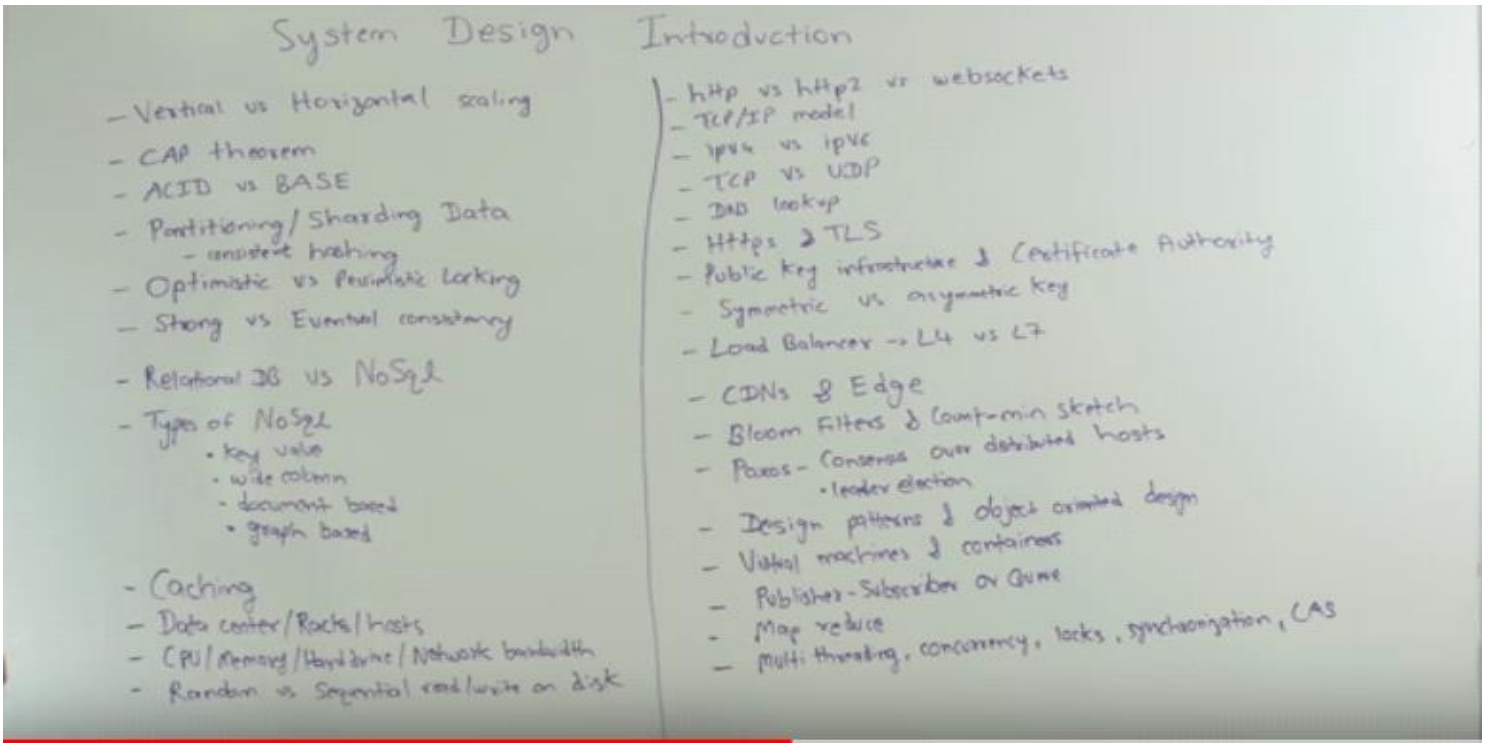
0. System Design Strategy

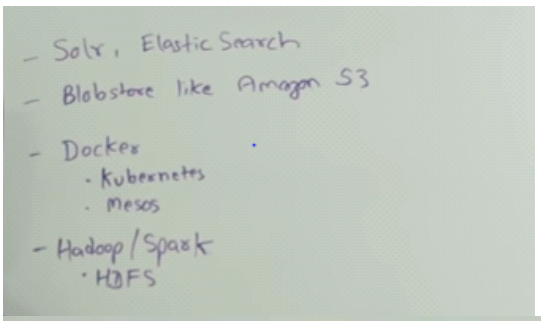
Wednesday, July 28, 2021 9:57 AM

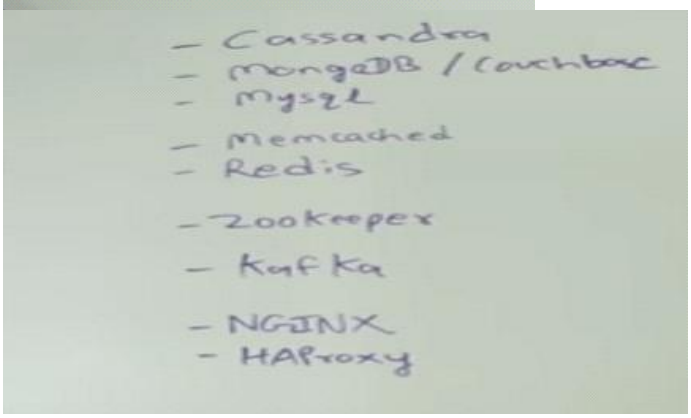
1. Requirement -> Functional
2. Constraints, Assumption, Risk, Scope
3. API
4. Data - Estimation - SLO, SLA, RTO, RPO
5. NFR
6. Data Modeling
    1. CA - SQL
    2. CP -> Mongo
    3. AP -> Dynamo, Casandra
    4. RPU / WPU
    5. Read vs Write
    6. PK, Sort, Secondary Index
    7. Shading, replication
    8. TTL
7. Tradeoff - Storage vs Compute
8. Scalability
9. Resiliency

System Design

Thursday, December 12, 2019 3:48 PM







Capacity Planning

Monday, July 26, 2021 10:26 PM

1. Storage
2. Network bandwidth

Scale:

How much per document -
Bandwidth - Concurrent User
Read or Write Intensive -

```
Scale
 - 500M MAU
 - Distributed across globe
 - Equally load across the day
 - 24M daily active users
   - 1M users per hour
   - .1M users creating board
   - .9M users would see the boards sharing
   - Read heavy service

 - .1 M users per hour
   - .1M
   - 100K/60 new board per min
     - 1.6k boards per min
   - per board 5 links
     - 1.6k * 5 links save per min
   - 900K/60 view boards per min
     - 15k boards per min
```

Database:

CP: Document based
MongoDB - > Limitation - 16 MB

AP: Columnar
Casandra, dynamoDB

Partitioning, Sharding?
Cross Partition Query

Tradeoff:

Redundant Data (Storage) / Scanning Query (Compute)

Capacity Planning

10^3 = KB
10^6 = 1 MB
10^9 = 1 GB

.4 GB = 400 MB

.4 * 10^3 = 400 MB

500 * 10^6 = 500 Million

1 B = 1 * 10^9 = 1 Billion

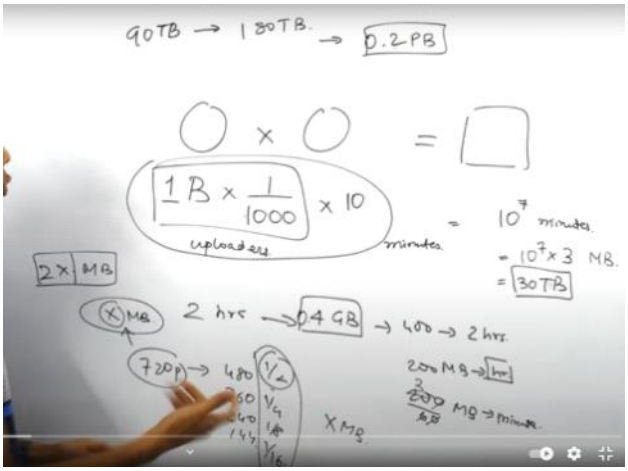
10^9 users * 1/1000 Uploads *10 minutes = 1o^7 minutes

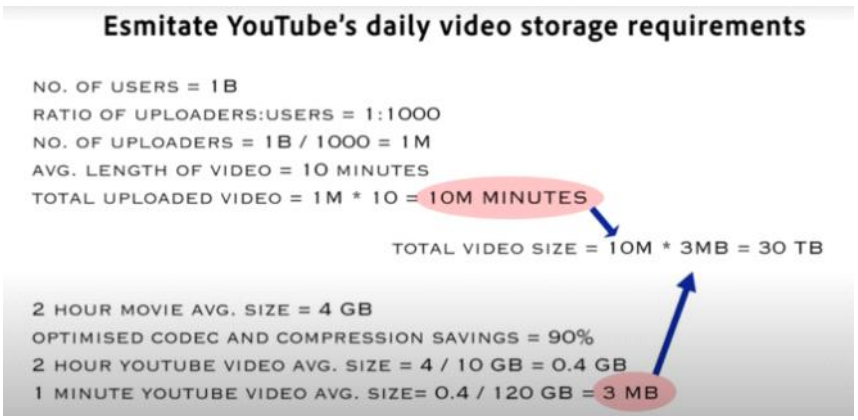
Lets Assume:

2 hours video = 4GB => Filter => .4 GB = 400 MB per 2 hours
= 400 MB / 2*60 =3 MB per minutes

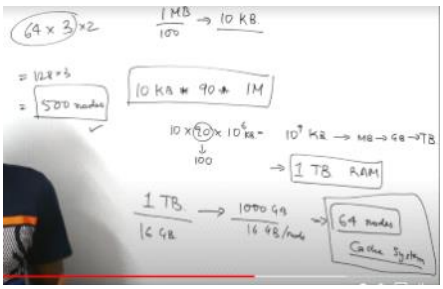
Total Size required = 10^7 * 3^ 6 => 3^13 => 30^12 => 30 TB per minutes

Fault + Reduandancy = > 3* 30 TB => 90 TB
Different resolution => 90 * 2 => 180 TB => .2 PB





Caching



Per original image = 1 MB filter image 10 KB

10 KB * 90 days most recent * 1 M image
= > 10^3 & 100 & 10^6 = 10^11 ~ 1 TB

Use case-3

- **Database Capacity planning**

Let's assume you have a table with columns as ID, Name, Date of Birth

Field Storage : ID =3, Name =21, Date of Birth=6 Total = 30

Table size: 500000 * 30 * 1.5 = 22500000 = 22 MB

Index size = number of records in the table being indexed x (7 + number of fields indexed +field storage) x 2

= 500000*(7+1+30) * 2 = = 38000000 =38 MB

Database size = *Sum of all table sizes + Schema size (100 MB) + index size*

Reach goals

Example used Twitter

Total users : 500 million

Active users : 2 million

Average photo size : 200KB

1 million photos every day

Total space for one day : 1 M X 200KB = 200GB

Total space required for one year = 365 X 200 * 1 = 73 TB

1. Social

Friday, September 24, 2021 10:19 AM

1.1 Twitter

Tuesday, July 27, 2021 1:35 PM

https://medium.com/@narengowda/system-design-for-twitter-e737284afc95

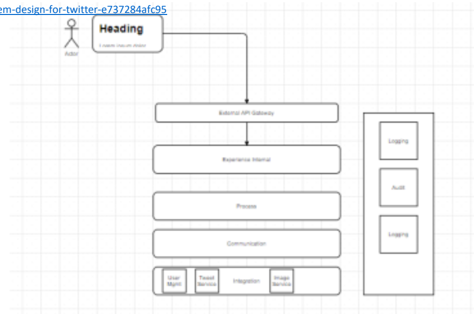
Core functionalities:

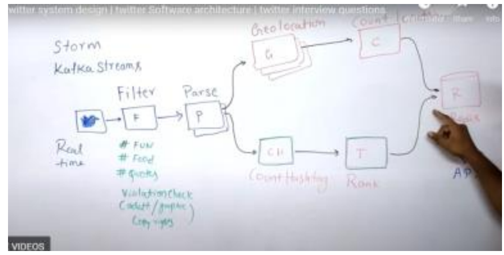
- User is share content (Text and Images) - Tweet
- User Profile Management
  - Profile details
  - User will be following other users
- Feed : According to each user.
  - Based on users is being followed
  - Latest on top
- HashTag Search
  - All public tweets

1. Tweet
2. Retweet
3. TimeLine
4. Hashtag

Design:

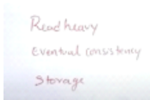
1. Pre compute TimeLine
   a. Active Users
   b. Threshold Users
2. Hybrid

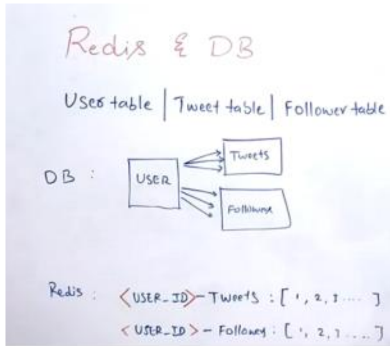


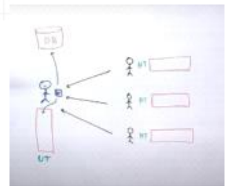


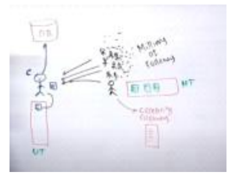
Persistence Layer:

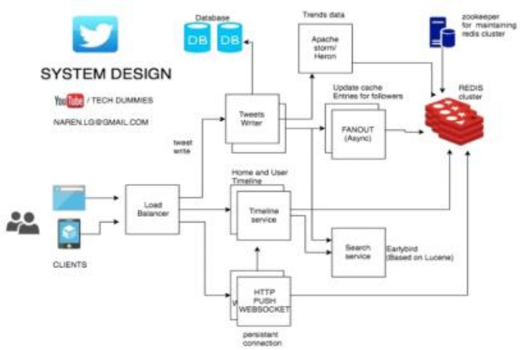
- User Profile Data
  - Partitioning based on user_id (Mongo-DB)
  -
- Tweets of Users
  - Text
  - Images
  - Properties
    - User ID
    - Content
    - Hashtags
    - Time Of Posting
  - Sharding By UserID
  - Cassendra
- Save users followers
  - User Profile DB separate schema (Partitioned Based on user ID)
    - UserID -> Followed Users
    - Followers -> UserID (friends ID)
- HashTag based indexes of Tweets
- Image Server
  - AWS S3

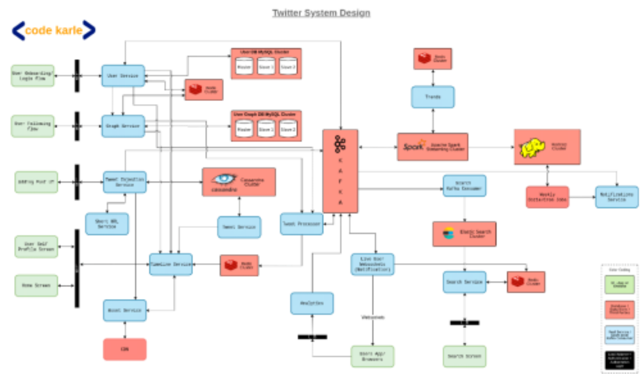












Capacity Planning :

Capacity Estimation: Back-Of-The-Envelope Calculation for System Design | Twitter Example

Total number of users = 600 million
Daily active users = 200 million
25% of the users tweet, each 2 tweets per day
200 * 25% * 2 = 100 million new tweets per day
200 million * 100 = 20B views per day

600 × 10^6 Users
200 × 10^6 Active users
25% Active users - twist
200 × 10^6 × 25/100 × 2 = 100m write
200 m AU reads 100 twit
200 × 10^6 × 100 = 20 × 10^9 read

100 characters per tweet on an average
2 bytes per character
200 bytes + 50 bytes = 250 bytes
Tweet Text Data = 25GB/day

100 * 10^6 * 250

1000 * 1000 * 1000

100 × 10^6 × 250 / 10^9 GB
= 25 GB

100 characters per tweet on an average
2 bytes per character
200 bytes + 50 bytes = 250 bytes
Tweet Text Data = 25GB/day
1/20 will have an image associated with it: 200KB
1/100 will have a video associated with it: 2MB

100 * 10^6 * 200 / (20 * 1000 * 1000 * 1000) + 100 * 10^6 * 2 / (100 * 1000 * 1000)

Image + Video = (1 + 2) = 3 TB /day
Total final storage: 25 GB + 3 TB ~= 3 TB/day

Incoming to server = 3 TB/day ~= 23 MB/sec
Outgoing text tweet data = 20B * 250bytes ~= 60 MB/sec

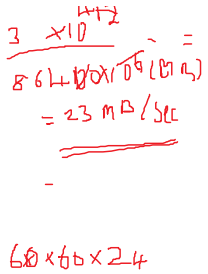
65
4+2
3 × 10 =
864 × 10^6 (GB)

Total final storage: 25 GB + 3 TB ~= 3 TB/day

Incoming to server = 3 TB/day ~= 23 MB/sec
Outgoing text tweet data = 20B * 250bytes ~= 60 MB/sec

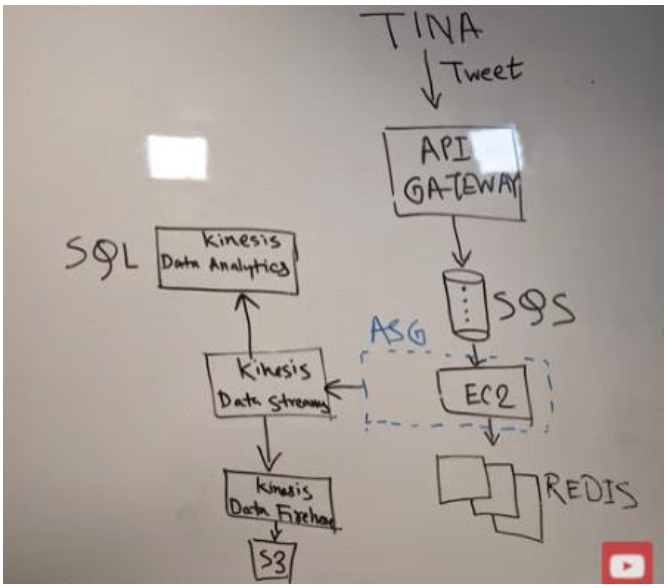
$$\frac{20 * 10^3 * 10^6 * 250}{86400 * 1000 * 1000}$$

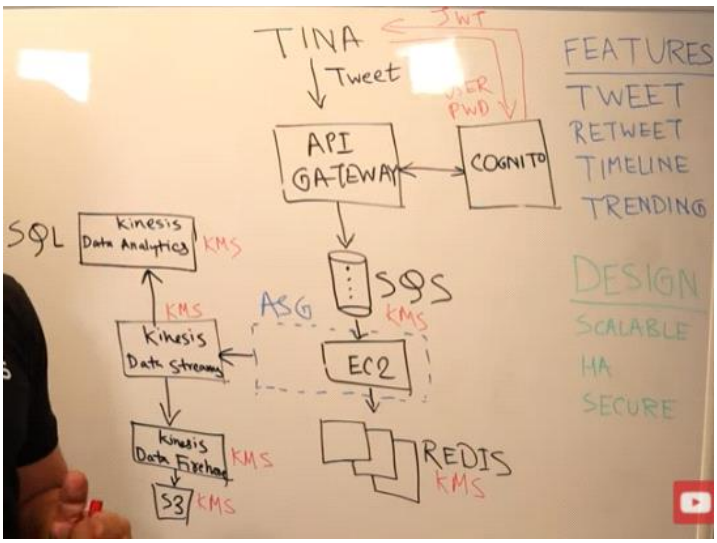
Incoming to server = 3 TB/day ~= 23 MB/sec
Outgoing text tweet data = 20B * 250bytes ~= 60 MB/sec
1/20 will have an image associated with it: 200KB
Outgoing image data ~= 2.5 GB/sec
1/100 will have a video associated with it: 2MB
When we show 5 videos, users watch 1 (1/5)
Outgoing video data ~= 1 GB/sec
Total Outgoing = 60 MB/sec + 2.5 GB/sec + 1 GB/sec ~= 3.5 GB/sec



raj

Tuesday, September 28, 2021 5:25 PM





Twitter

Thursday, September 23, 2021 9:23 AM

Functional

- Tweet
- Retweet
- Timeline
- Trending

Non Functional

- Business Continuity and Disaster Recovery
    - Fault Tolerance

- Cost
- Security
- Monitoring
- Scalability and Reliability
- Deployment
- Hybrid

Architecture Patterns:

- Communication

- Asynchronous
- Message Broker
- Event Streaming
- SAGGA Orchestrator

- Security

- Authentication
- Authorization
- Federator
- Reverse Proxy

- Data Pattern

- Access Trends
- CQRS
- Relevance Tags
- Repository
- Unit of work

Data Storage and Network bandwidth:

Total Users: 500 * 10^6
Active Users: 200 * 10^6
Write Message : 200 * 10^6 * 25/100 * 2 = 100 * 10^6
Per Day = 200 * 100 * 10^6 = 20 * 10^9

Storage: (bits)

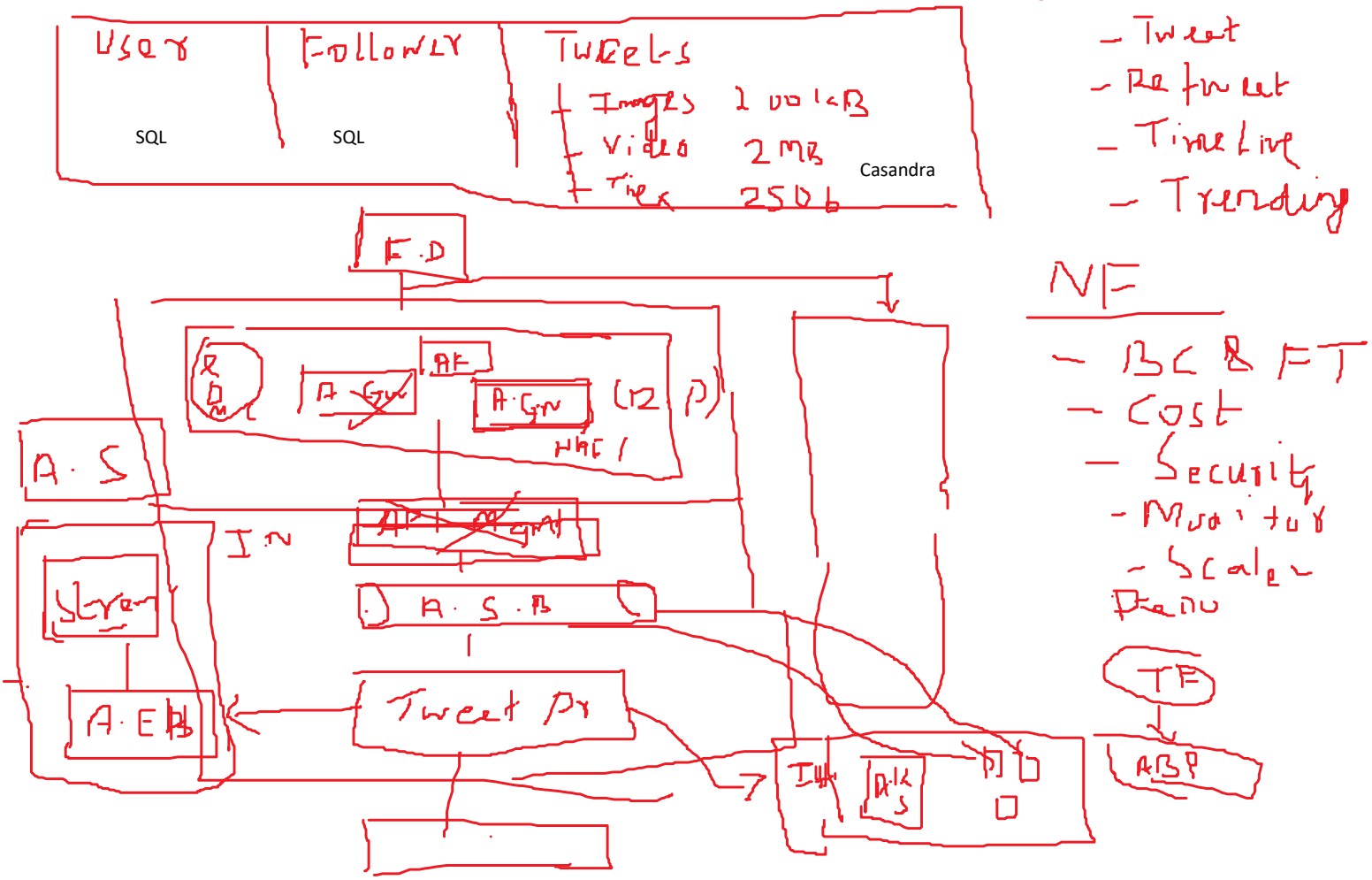
Text = 250 b => 100 * 10^6* 250 => 25 * 10^9 = 25 GB
Image 200 KB => 100 * 10^6 * 200 * 10^3/ 20 = 1 * 10^12 = 1TB
Video 2 MB

Network

Incoming: 20 * 10^12 / 86400 = 34 MB /s
Outgoing : 20 * 10^9 * 250/ 86400 = 60 Mb/s

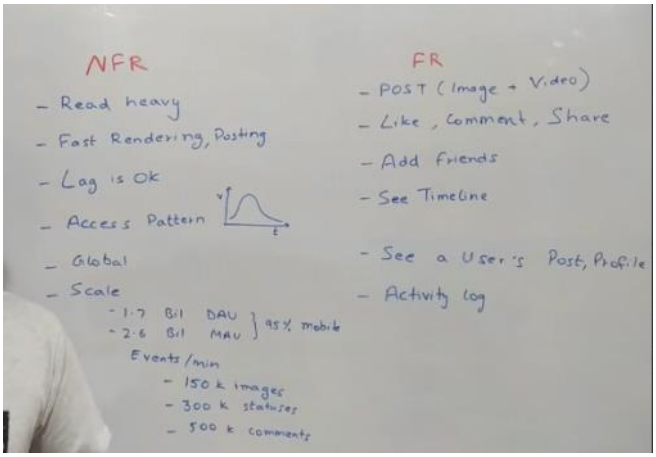
White board

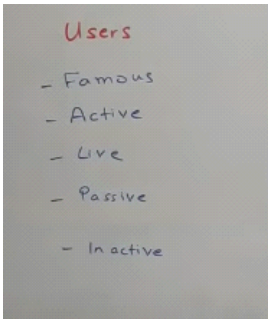
Wednesday, September 22, 2021 10:36 PM

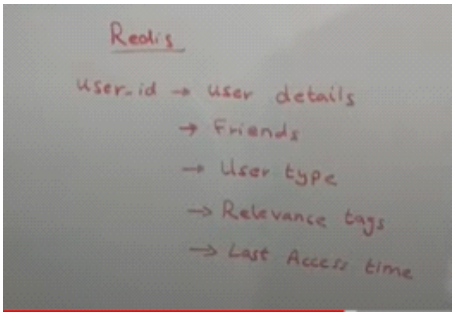


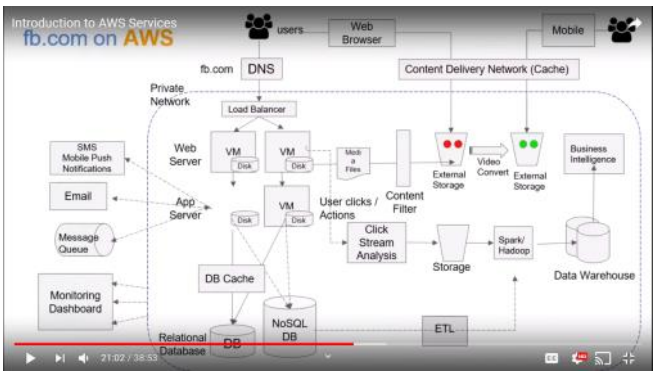
1. 2. Facebook

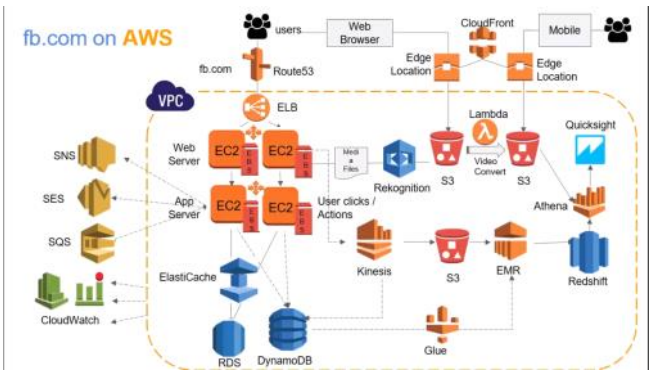
Friday, February 7, 2020 9:25 PM

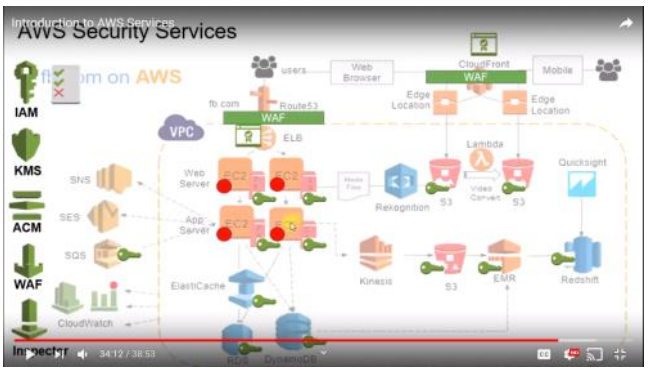


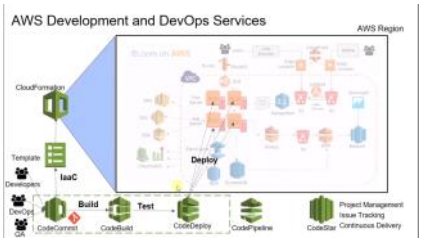











AWS Development and DevOps Services
AWS Region
CloudFormation
Template
IaaC
Developers
DevOps
QA
Build
Test
Deploy
CodeCommit
CodeBuild
CodeDeploy
CodePipeline
CodeStar
Project Management
Issue Tracking
Continuous Delivery


Templates
A template in AWS CloudFormation is a formatted text file in JSON or YAML language that describes your AWS infrastructure
To create, view and modify templates you can use AWS CloudFormation Designer or any text editor tool
Template in Amazon CloudFormation consists of nine main objects:
Format version
Description
Metadata
Parameters
Mappings
Conditions
Transform
Resources
Outputs
simplilearn

2. E-commerce

Friday, January 24, 2020 10:45 AM

1. Customer
2. Place Order
3. Fulfillment Order
4. Validate Order
5. Shipping Order

Customer orders,

billing,

payment,

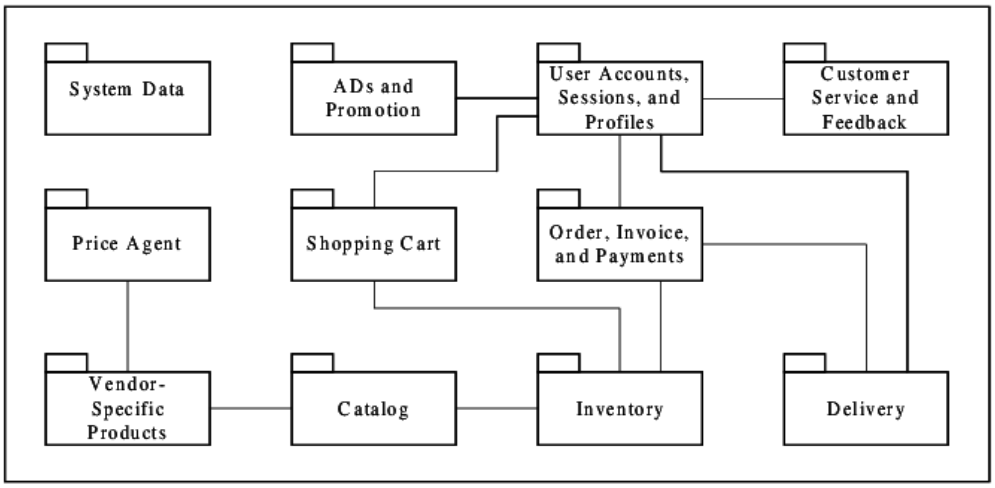
inventory,

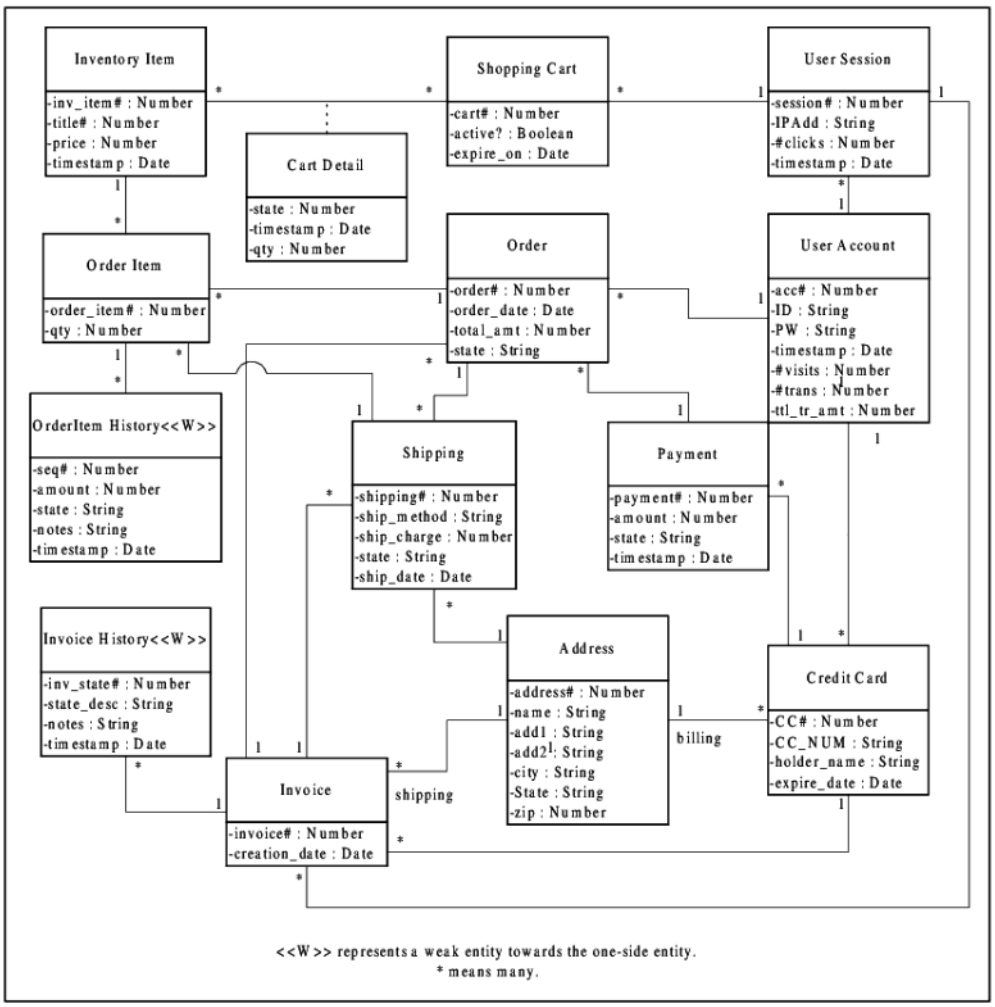
shipping.

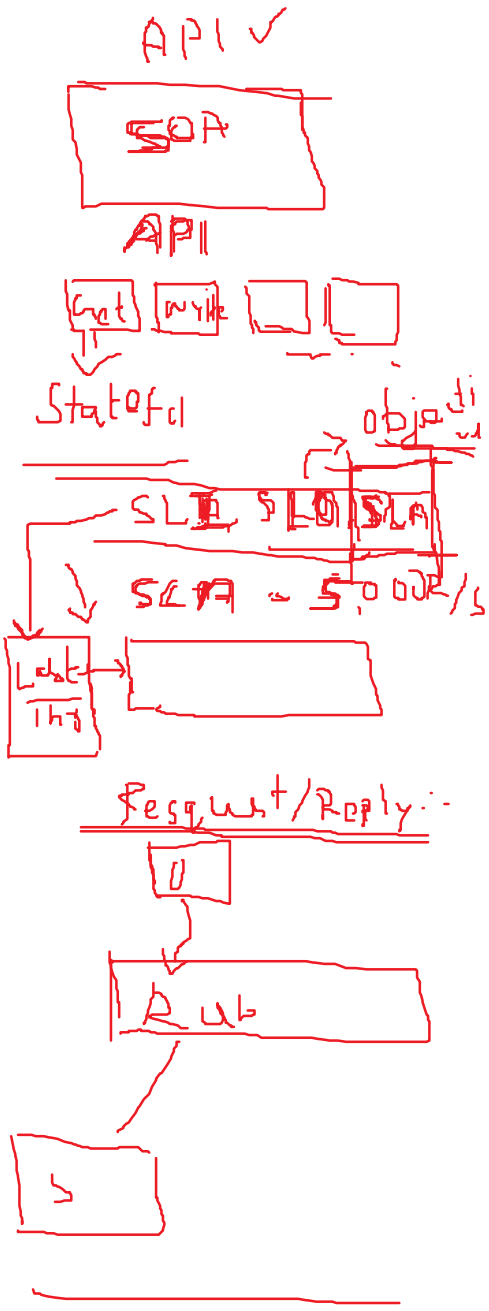
Insurance Business Domain
Claim
Billing
Invoice
Accounting
Corporate HR

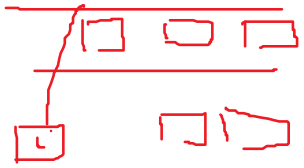
https://www.researchgate.net/figure/A-simplified-database-schema-for-e-commerce-transaction-processing_fig2_2359510

https://www.researchgate.net/figure/A-simplified-database-schema-for-e-commerce-transaction-processing_fig2_2359510\\



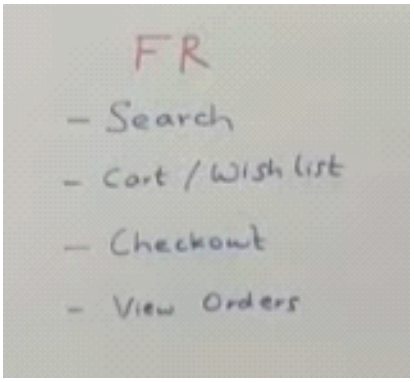


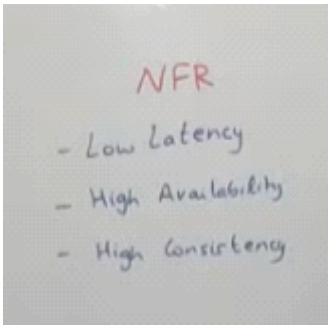




Functional

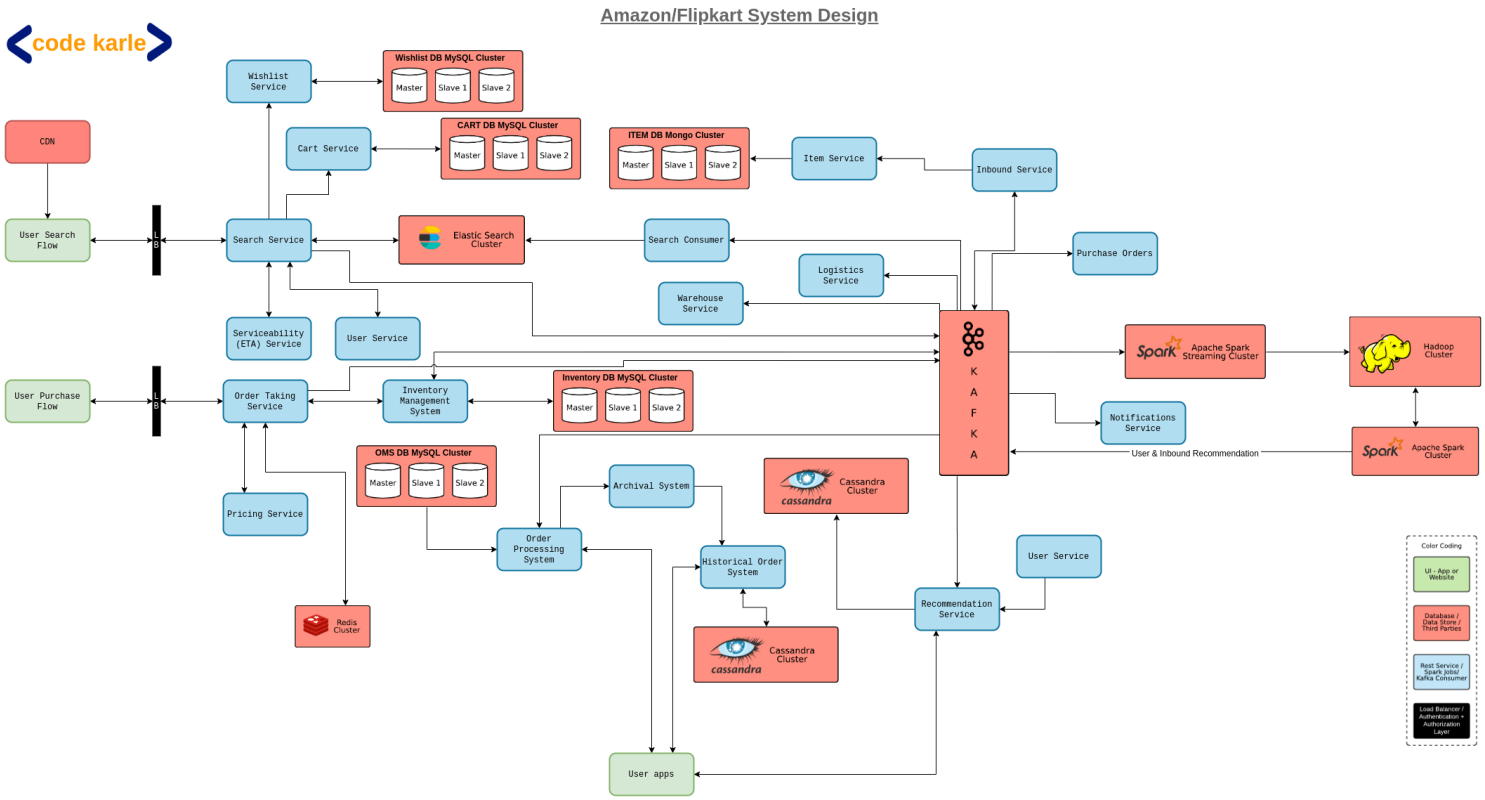
Friday, September 24, 2021 12:15 PM





System Design

Thursday, September 23, 2021 2:03 PM

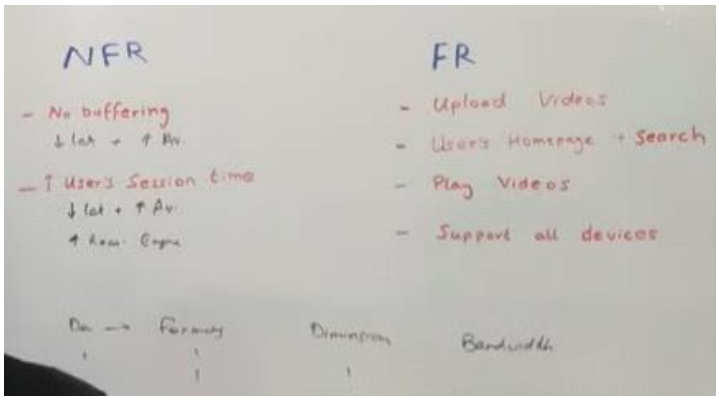


3. On-demand Video

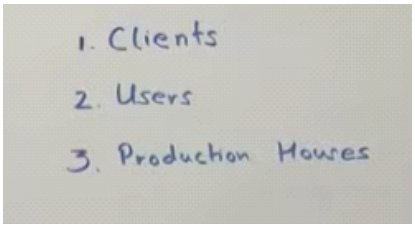
Friday, September 24, 2021 10:21 AM

3.3 Netflix 1

Friday, September 24, 2021 11:11 AM



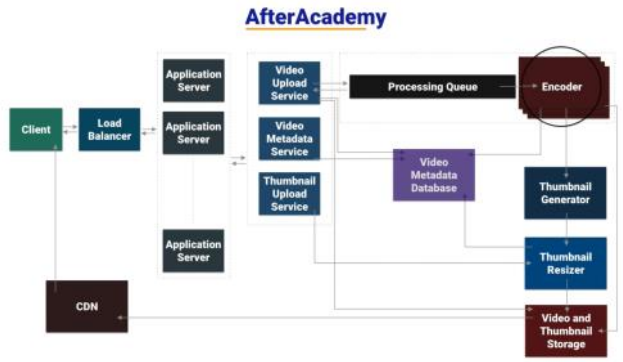
1. Devices
2. Format
3. Dimension
4. Bandwidth



3.1 YouTube

https://www.youtube.com/watch?v=XkFL5sAmp-g&t=1s

Friday, July 30, 2021 7:01 PM



Storage System

- Video Storage: HDFS
- Thumbnail Storage: S3 bucket

What More?

- Video duplication: Block-matching algorithm
- API for deleting a video
- Notification to the client

Database Design

- Video Metadata
- MySql

VideoMetadata

videoID
title
description
thumbnailPath
videoRawPath
videoPath
userId

API Design

uploadVideo(accessToken, title, desc, video)

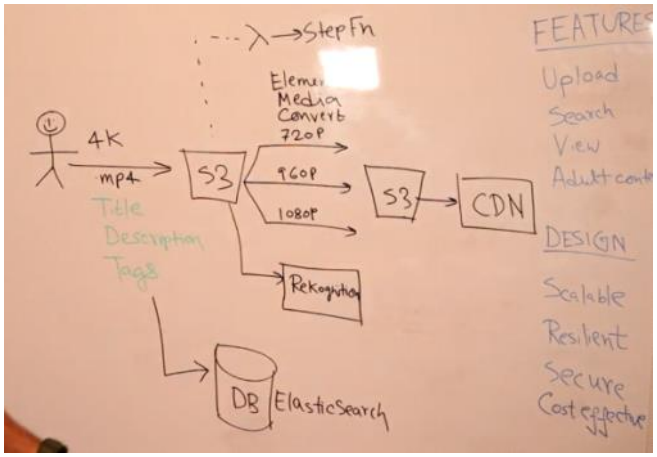
videoMetadata(accessToken, videoId, title, desc)

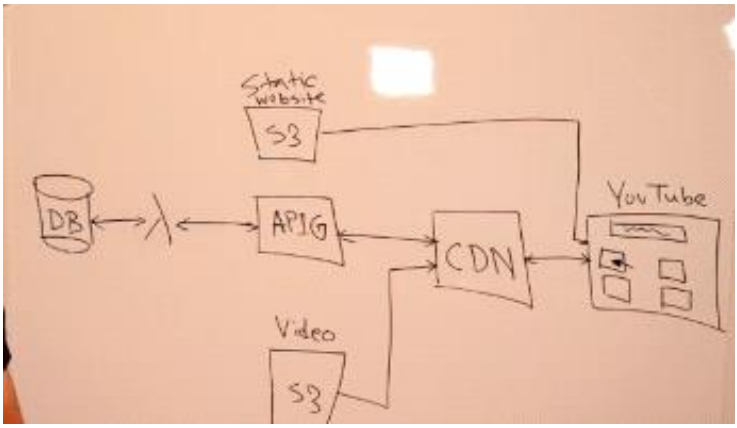
uploadThumbnail(accessToken, videoId, thumbnail)

3.2 Netflix - Raj

Friday, September 24, 2021 10:22 AM

Azure media service - Element media convert
Face deduction -> Rekognition





HTTP LIVE Streaming

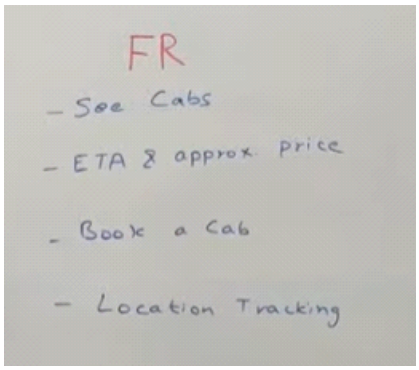
Ffmpeg

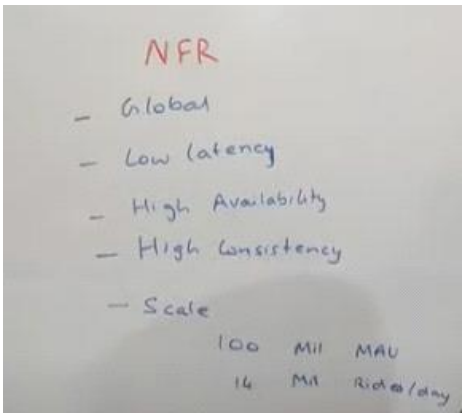
Tuesday, September 28, 2021 3:19 PM

```
ffmpeg -i test.mp4 -hls_time 10  -hls_playlist_type vod -hls_segment_filename "video_segments_%0d.ts" hls_master_for_test.m3u8
```

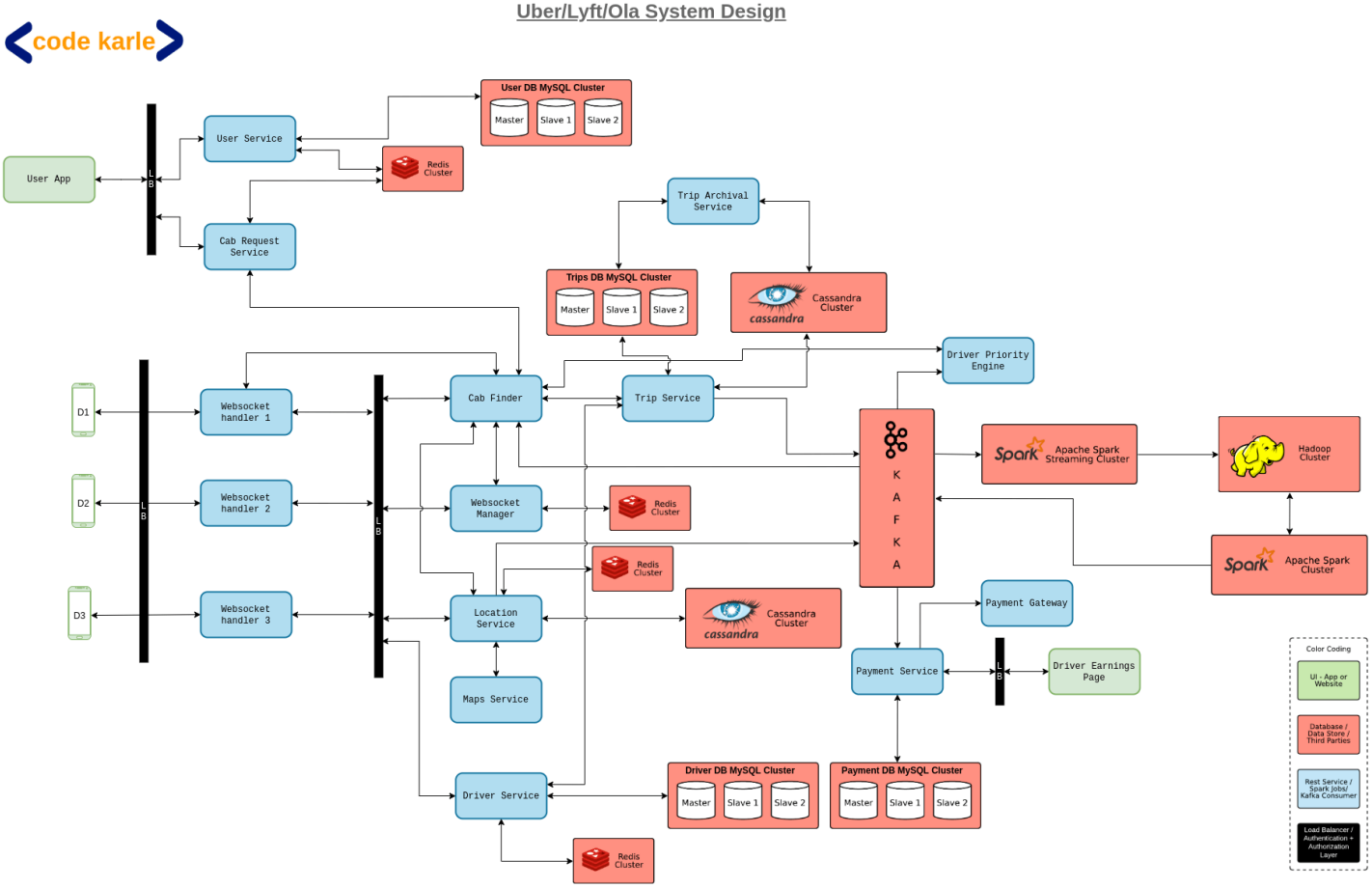
4. Car Tracking

Friday, September 24, 2021 11:46 AM





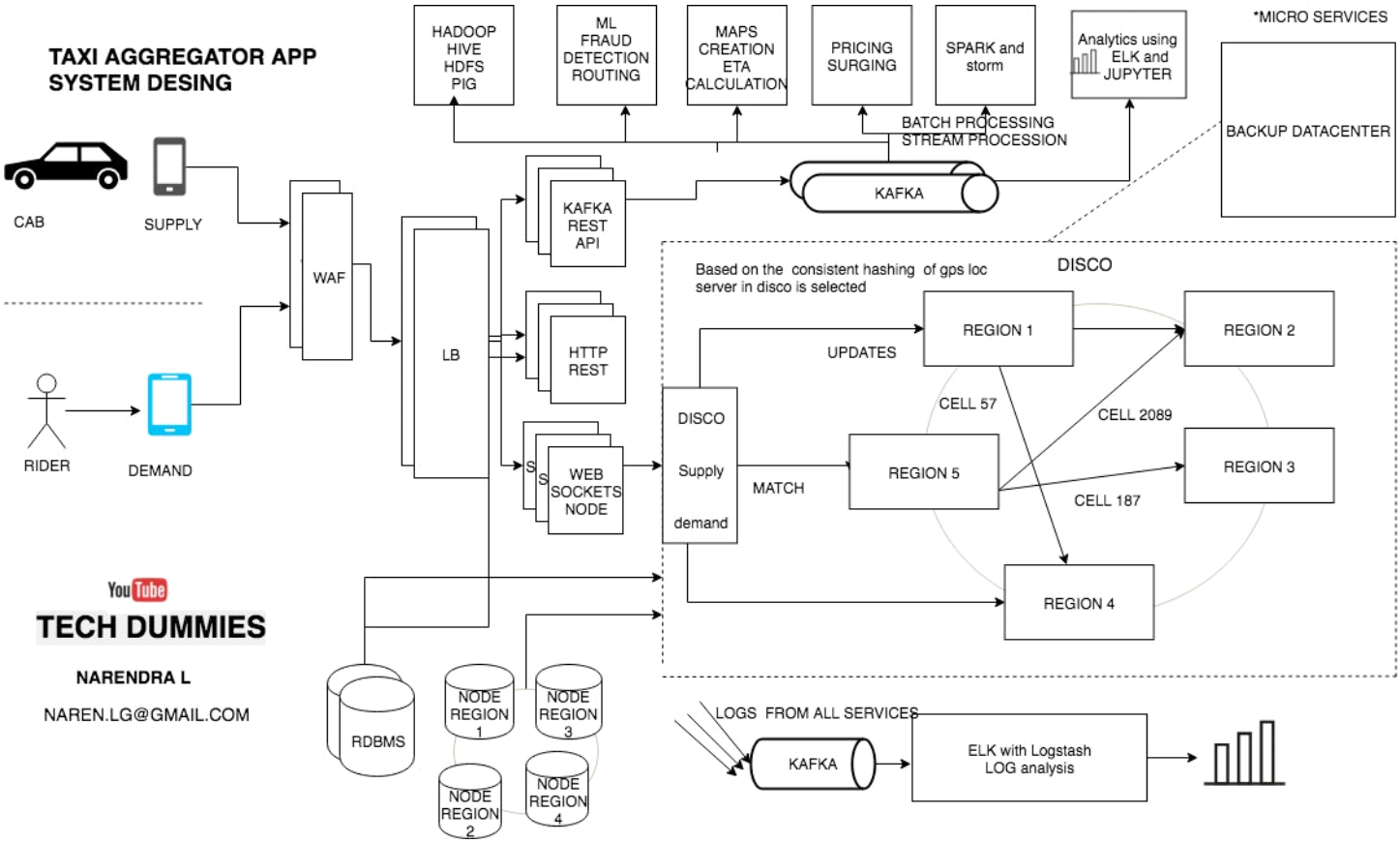
Uber/Lyft/Ola System Design



Narendra

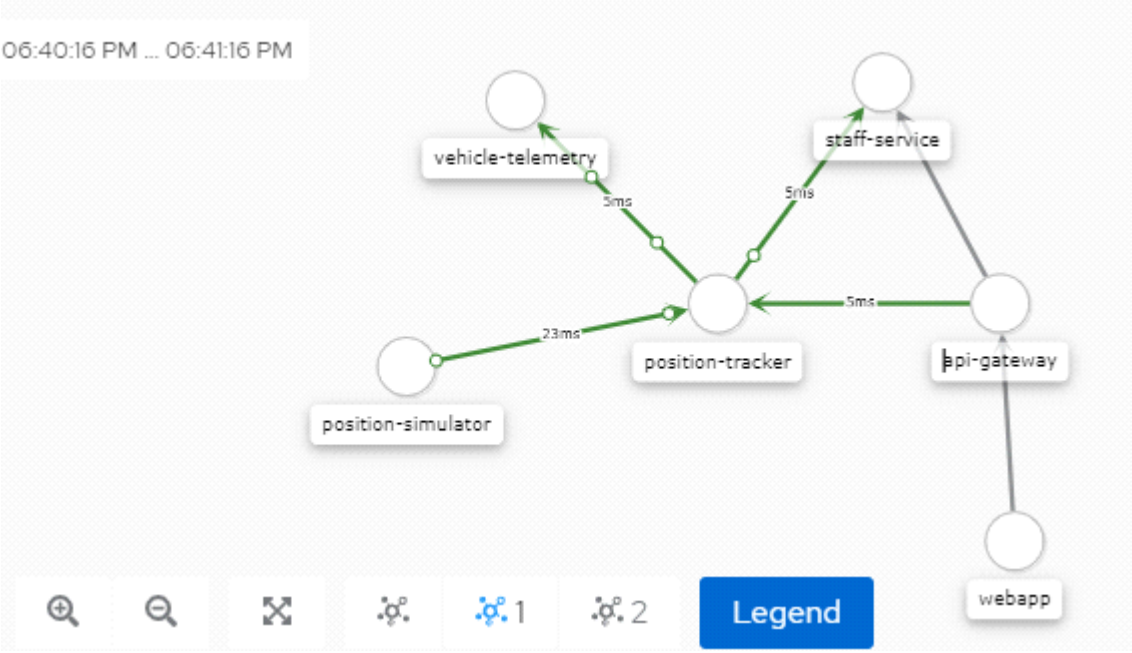
Friday, September 24, 2021 12:22 PM

Google s2 library



4.1 Vehicle Tracker

Thursday, August 26, 2021 6:41 PM



5. Online Chat System

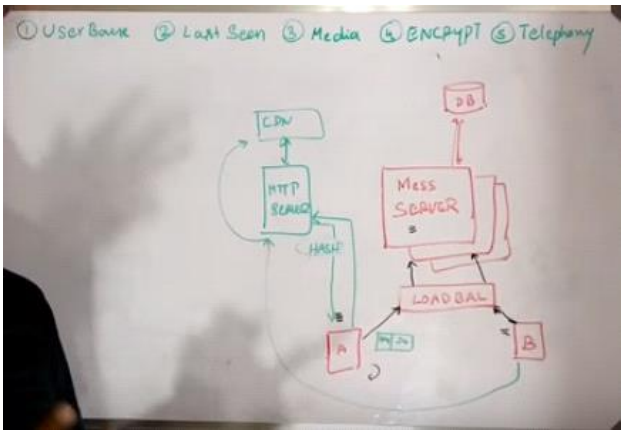
Thursday, December 12, 2019 11:06 AM

Protocol:

- TCP
- UDP
- WebScoket
- BOSH
- Long Polling
- XMPP

Load Balancer Configuration

- Load
- Session

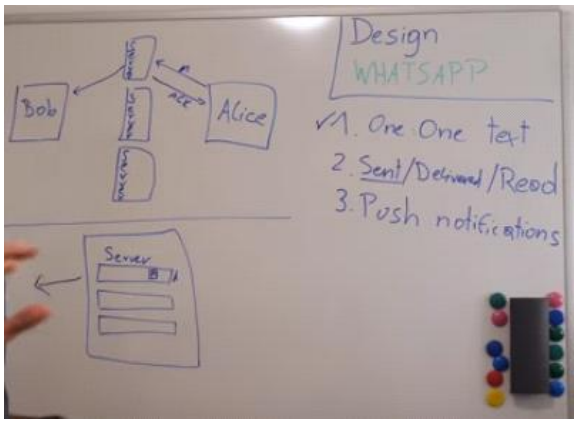


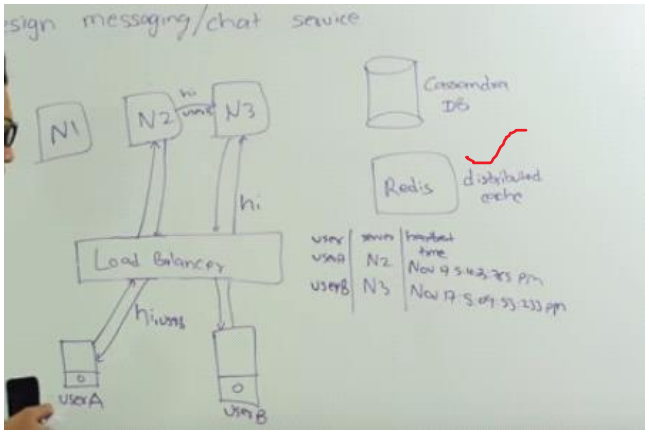
Queue Free when no message

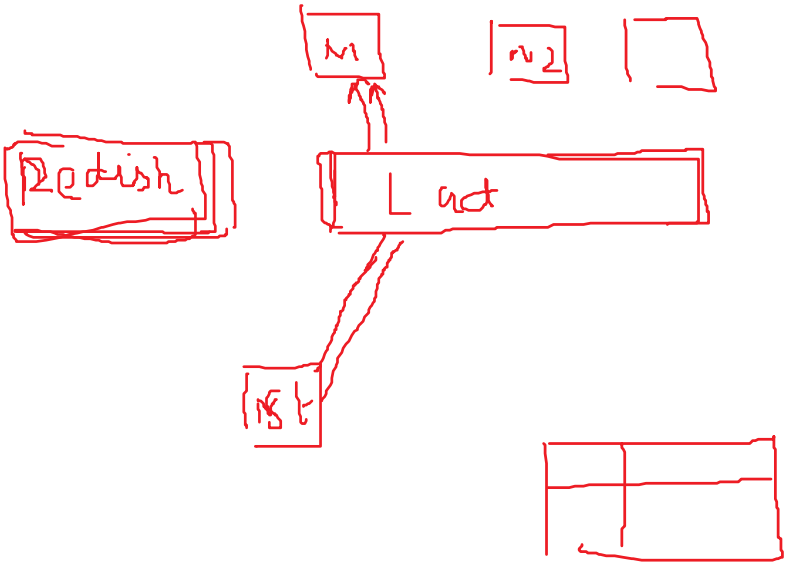
Ephemeral
Queue health

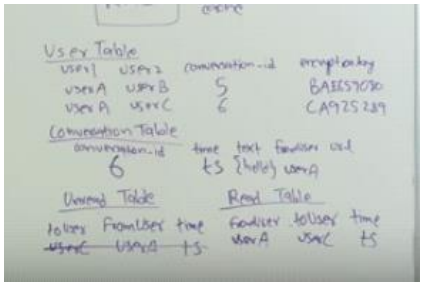
High Availability

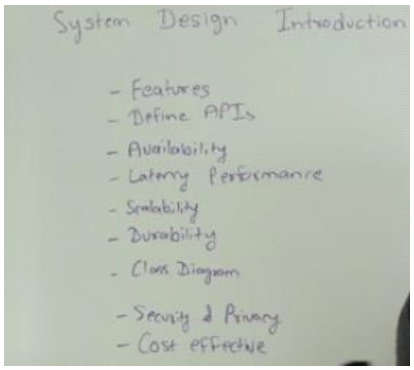
1. Ordering message [Session Affinity / Sticky Session]




N1
N2
N3
hi
hi
Load Balancer
Redis
distributed cache
hi, user1
UserA
UserB

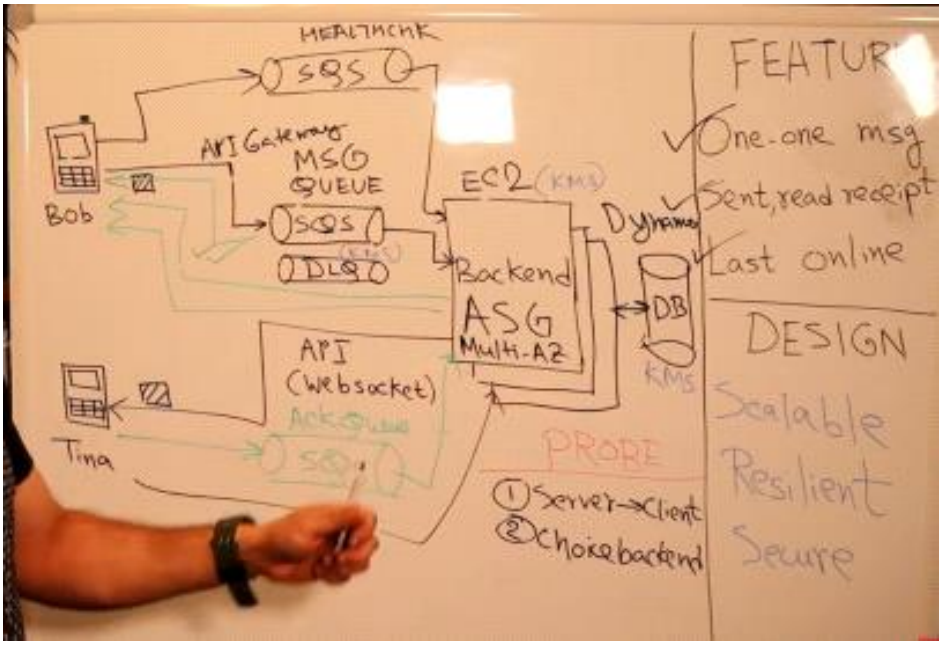

N1
N2
Redish
L ect

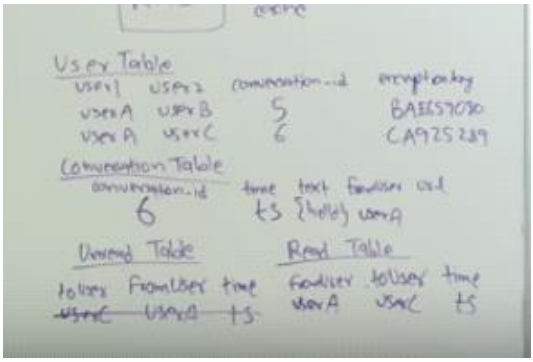

User Table
Conversation Table
Unread Table
Read Table


System Design Introduction
- Features
- Define APIs
- Availability
- Latency Performance
- Scalability
- Durability
- Class Diagram
- Security & Privacy
- Cost effective

Raj

Friday, September 24, 2021 12:04 PM



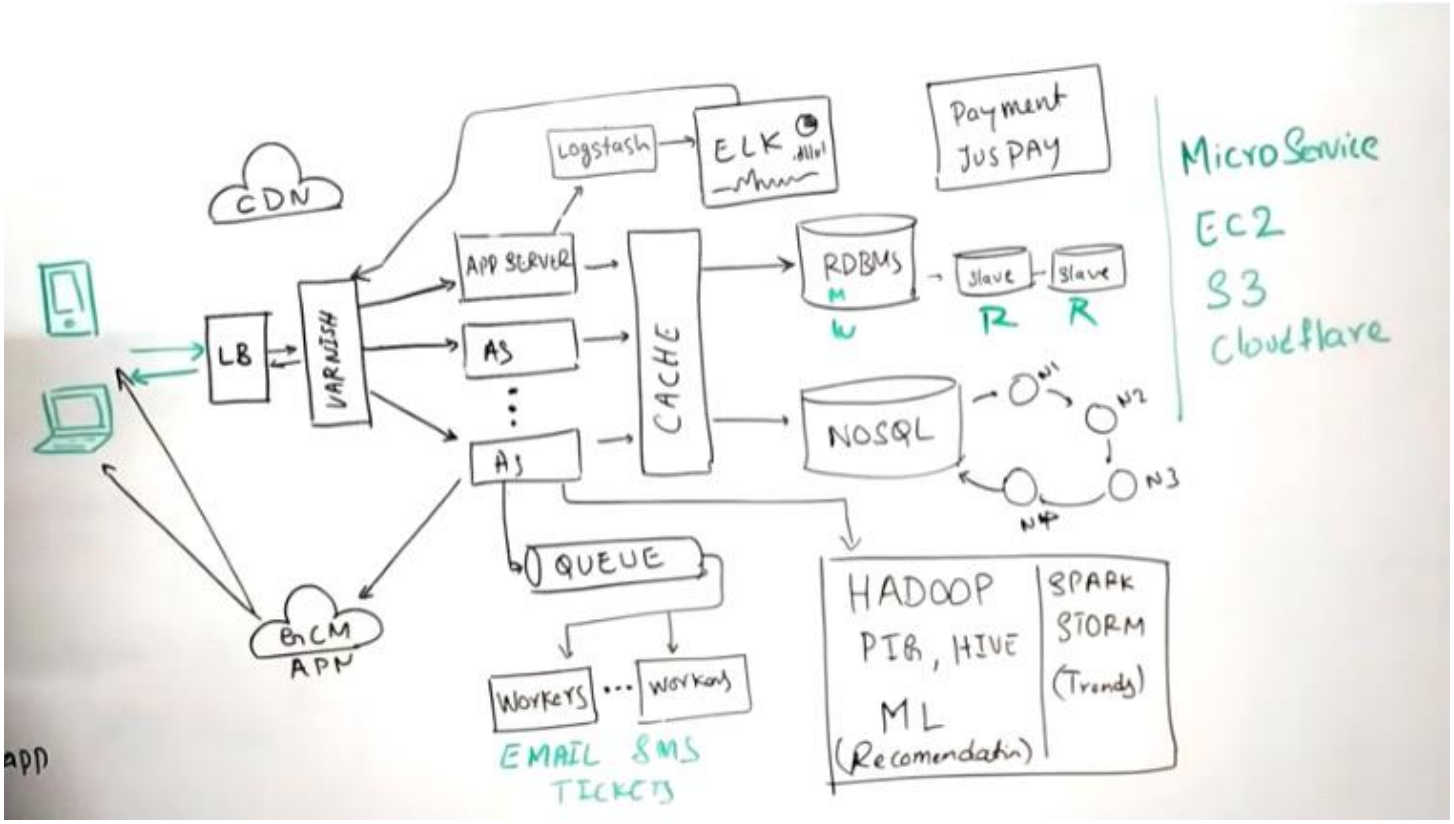


6. Online booking

Friday, September 24, 2021 2:18 PM

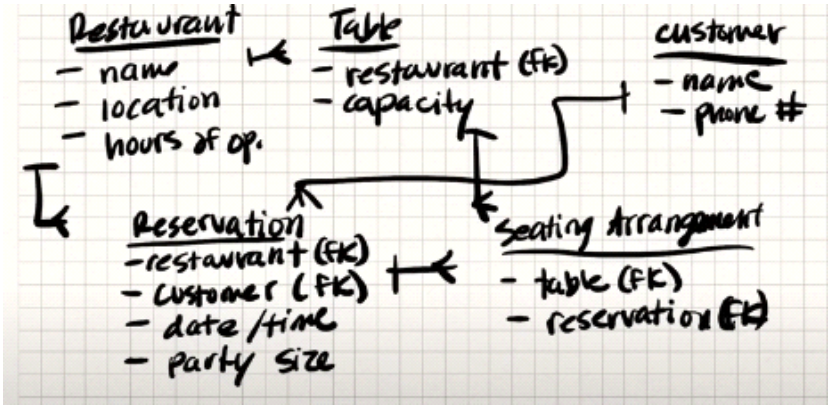
Movie

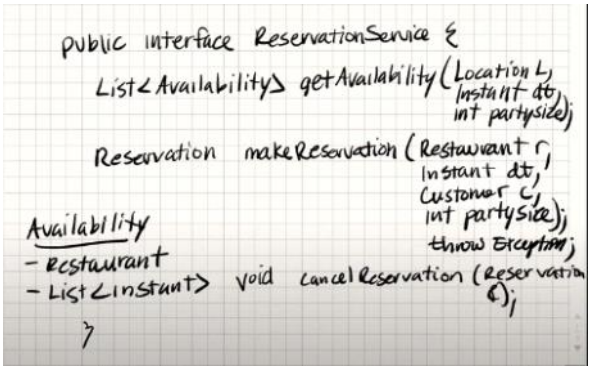
Friday, September 24, 2021 5:04 PM



Restaurant

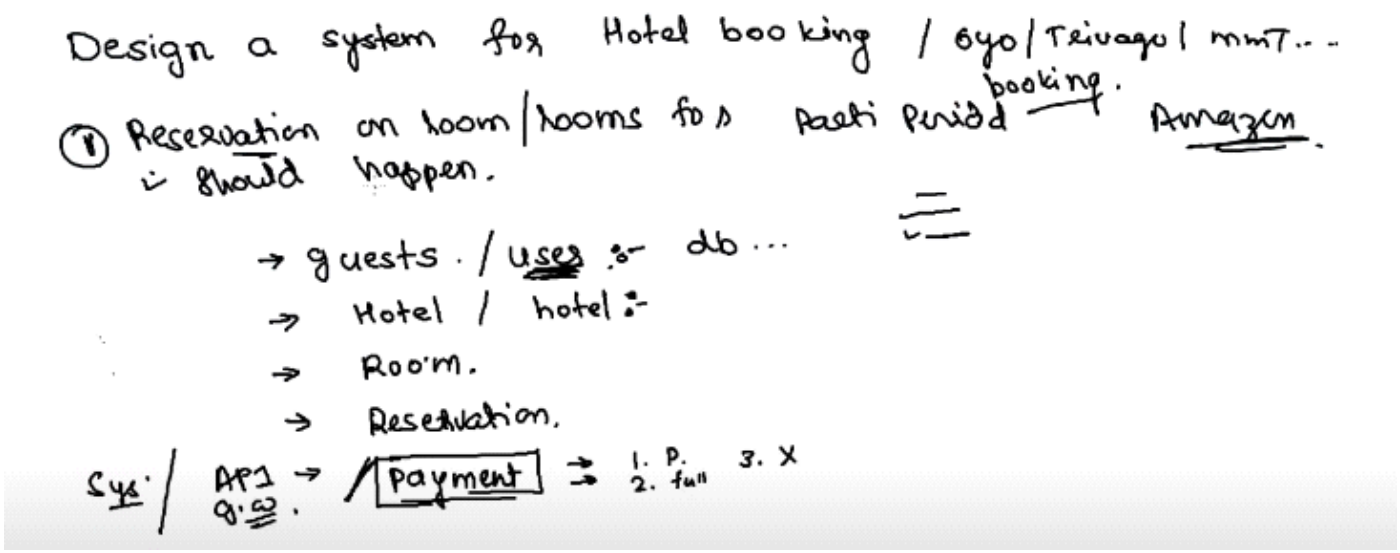
Friday, September 24, 2021 4:51 PM

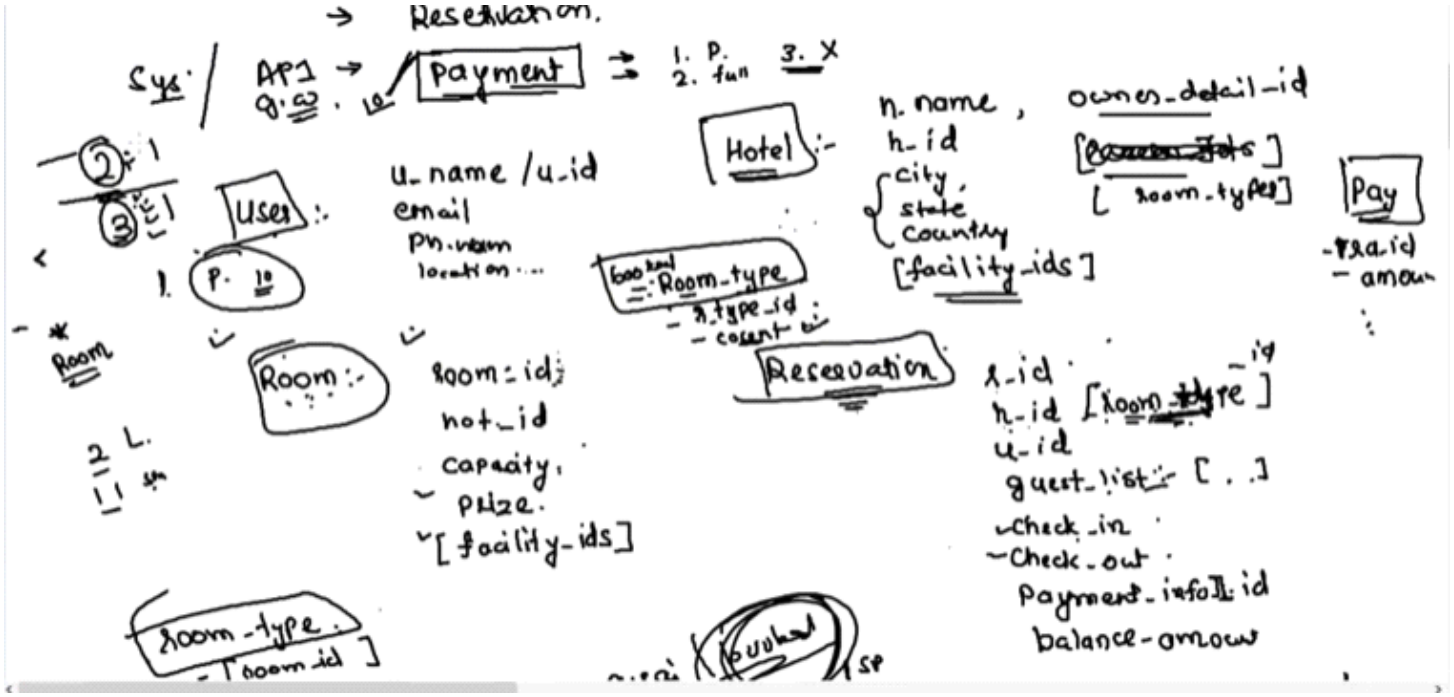


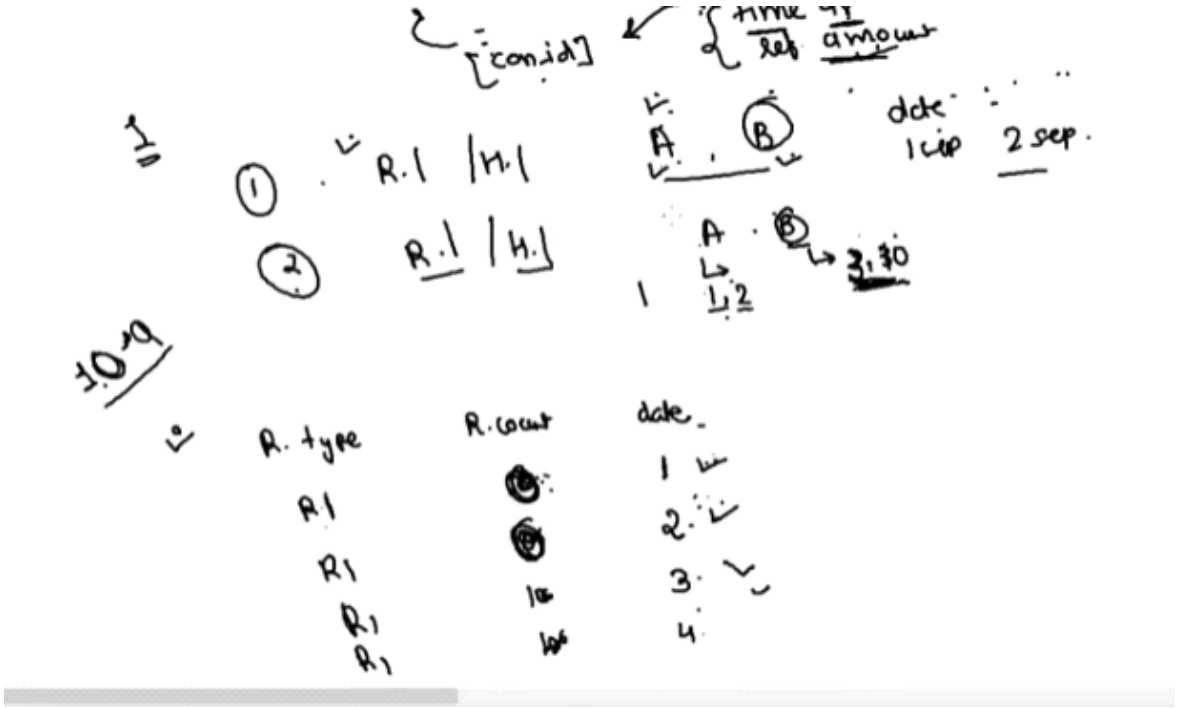


Hotel Booking

Friday, September 24, 2021 3:36 PM

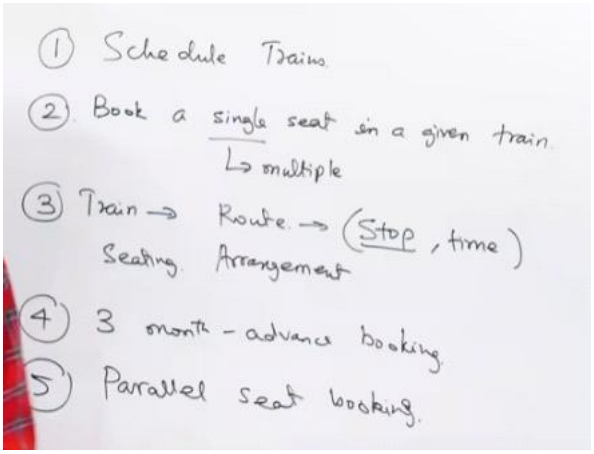


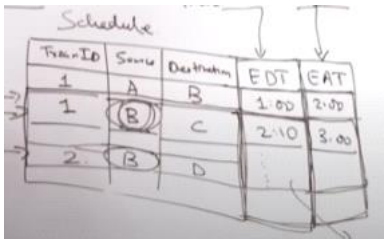


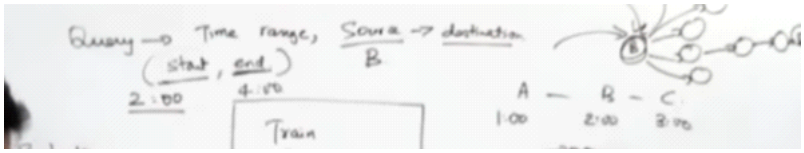

[conid]
amount
A
B
date
1 sep
2 sep
R.1
R.1
3.30
10-9
R. type
R. count
date
R1
R1
R1
R1

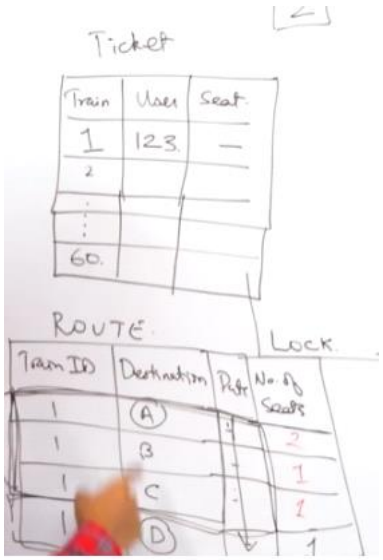
IRCTS

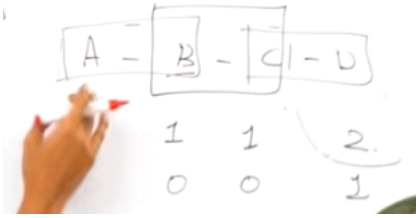
Friday, September 24, 2021 2:42 PM











7. Google Map

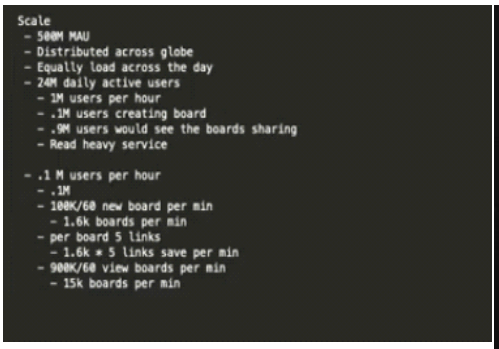
Friday, September 24, 2021 2:20 PM

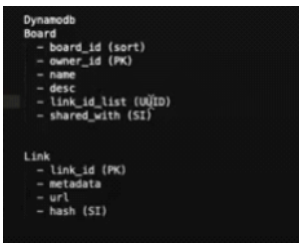
8. Notification Service

Friday, September 24, 2021 2:20 PM

8. Pinterest

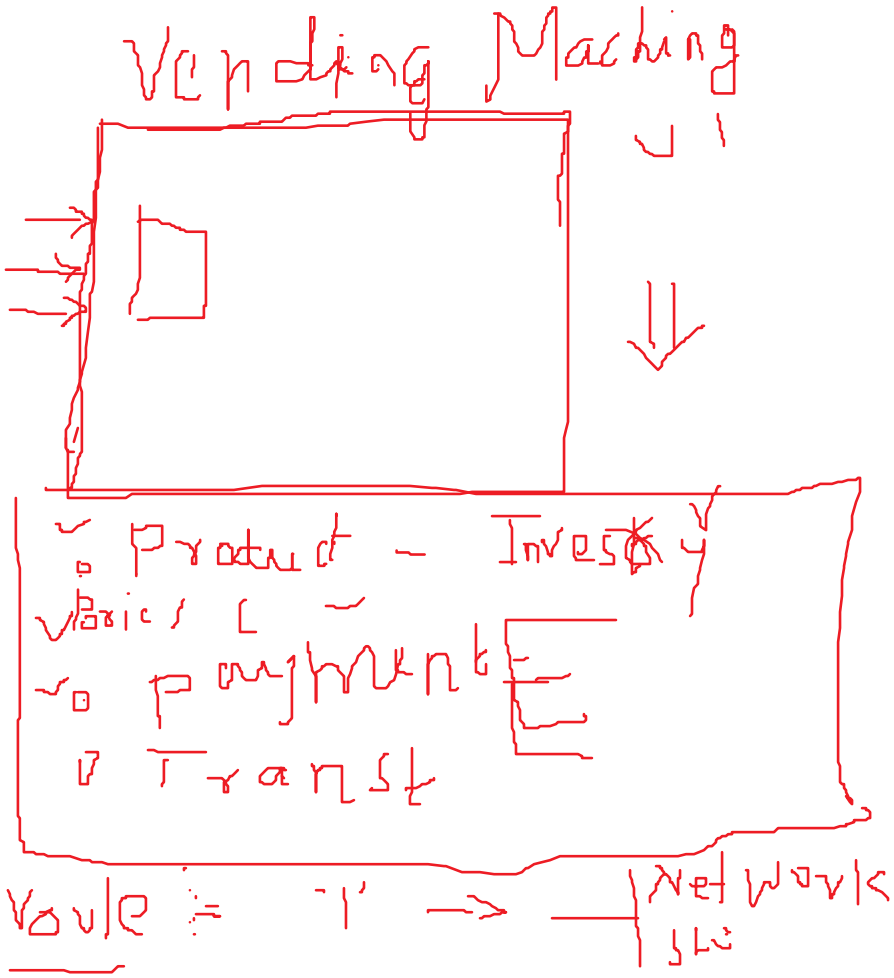
Tuesday, July 27, 2021 9:49 AM





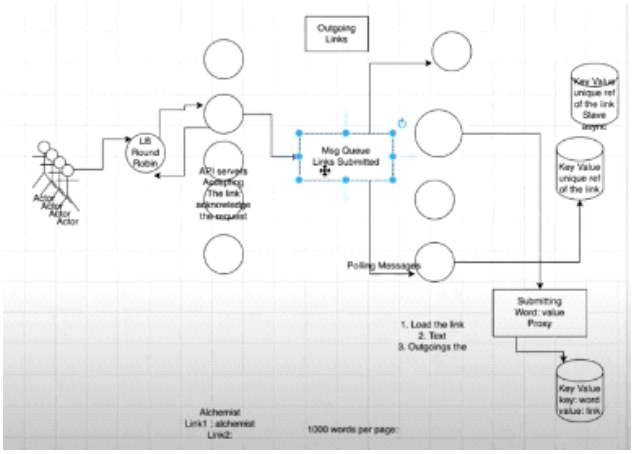
9. Vending machine

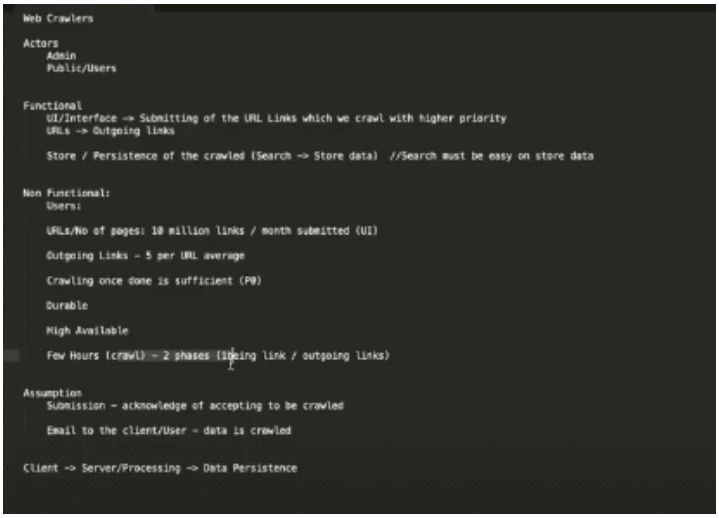
Wednesday, August 18, 2021 11:32 AM



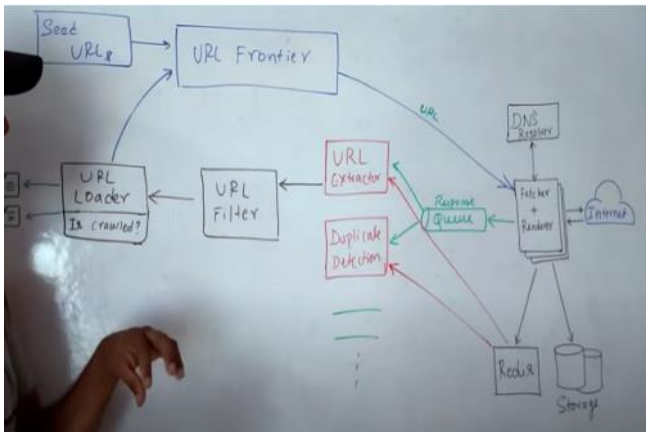
10. Web Crawler

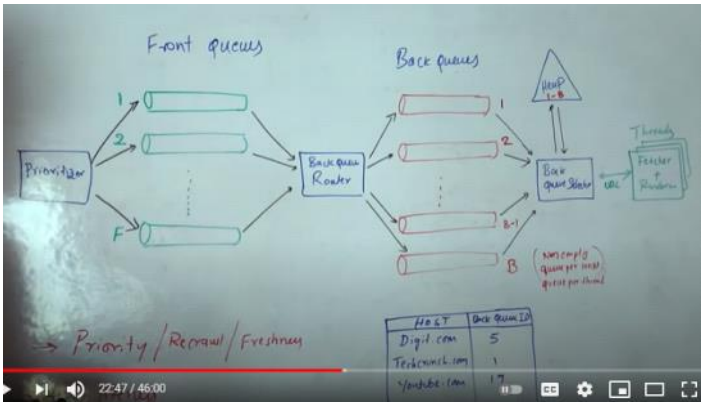
Monday, July 26, 2021 10:26 PM





```
Web Crawlers

Actors
    Admin
    Public/Users


Functional
    UI/Interface -> Submitting of the URL Links which we crawl with higher priority
    URLs -> Outgoing links

    Store / Persistence of the crawled (Search -> Store data)  //Search must be easy on store data

Non Functional:
    Users:

    URLs/No of pages: 10 million links / month submitted (UI)

    Outgoing Links - 5 per URL average

    Crawling once done is sufficient (P0)

    Durable

    High Available

    Few Hours (crawl) - 2 phases [Ingoing link / outgoing links)


Assumption
    Submission - acknowledge of accepting to be crawled

    Email to the client/User - data is crawled

Client -> Server/Processing -> Data Persistence
```



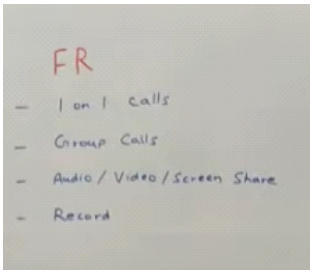


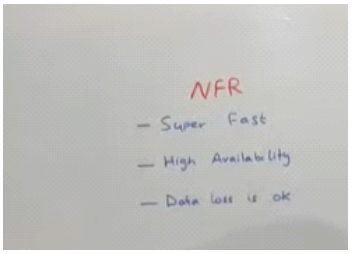
11. Video Communication [ZOOM]

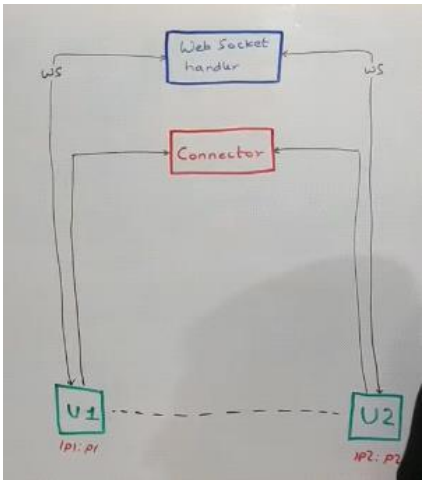
Friday, September 24, 2021 9:59 PM

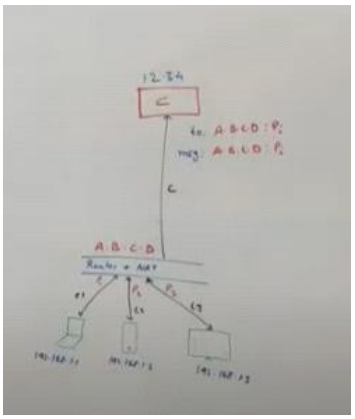
Video Call

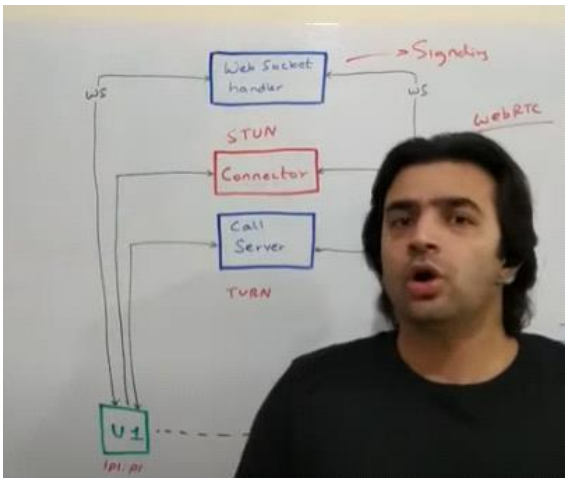
Tuesday, September 28, 2021 12:14 PM

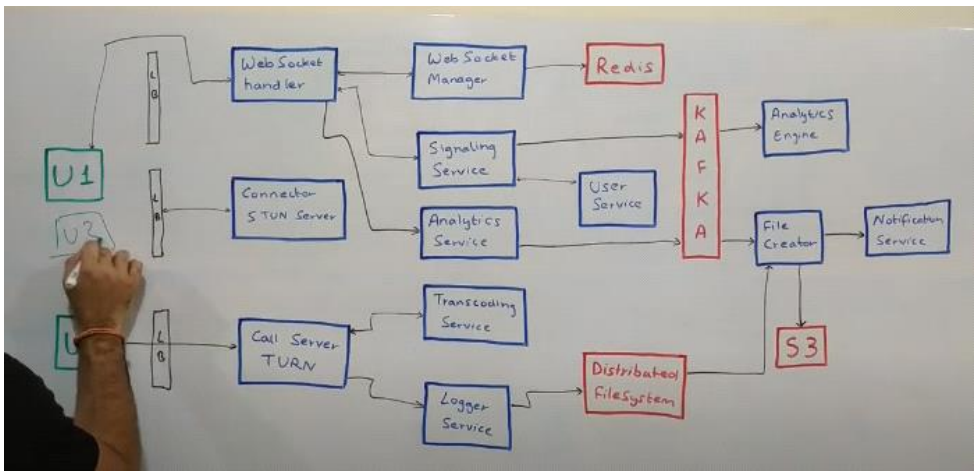








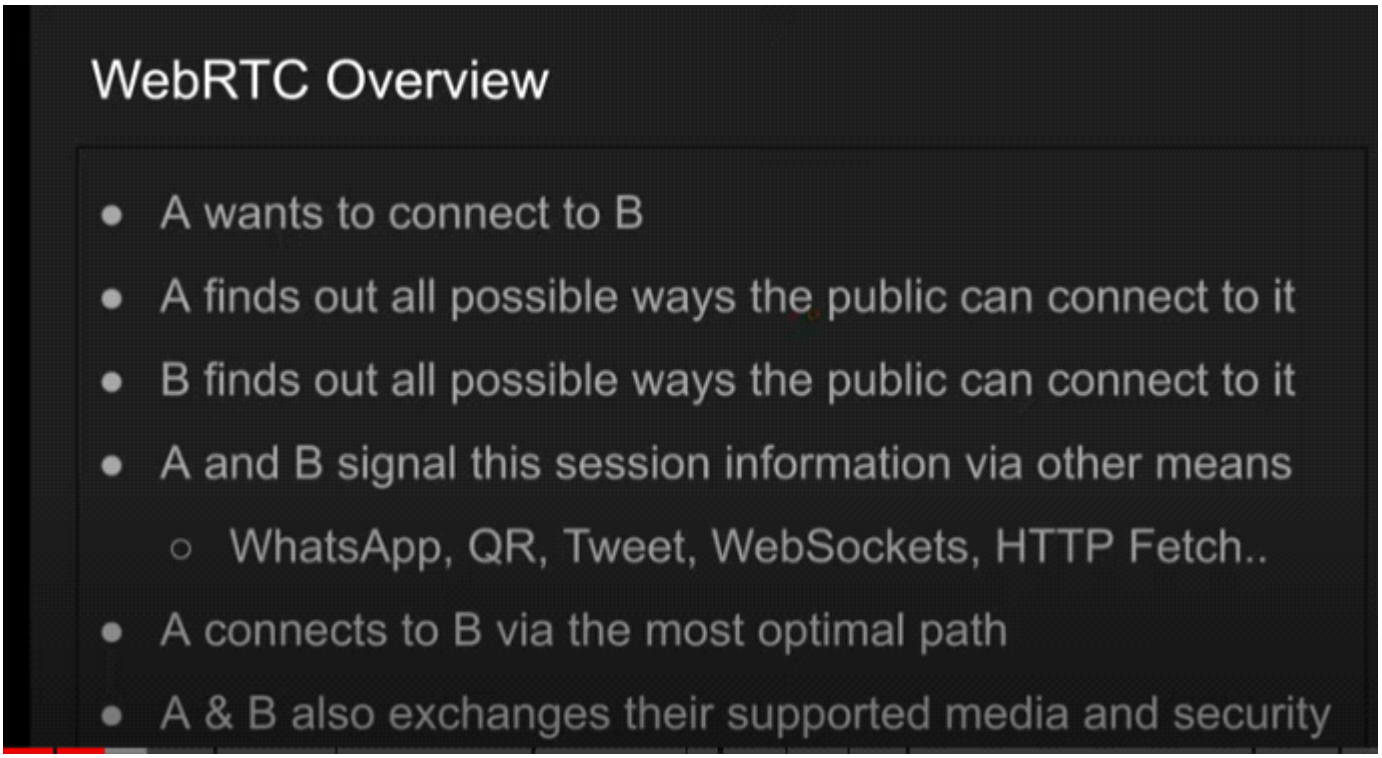


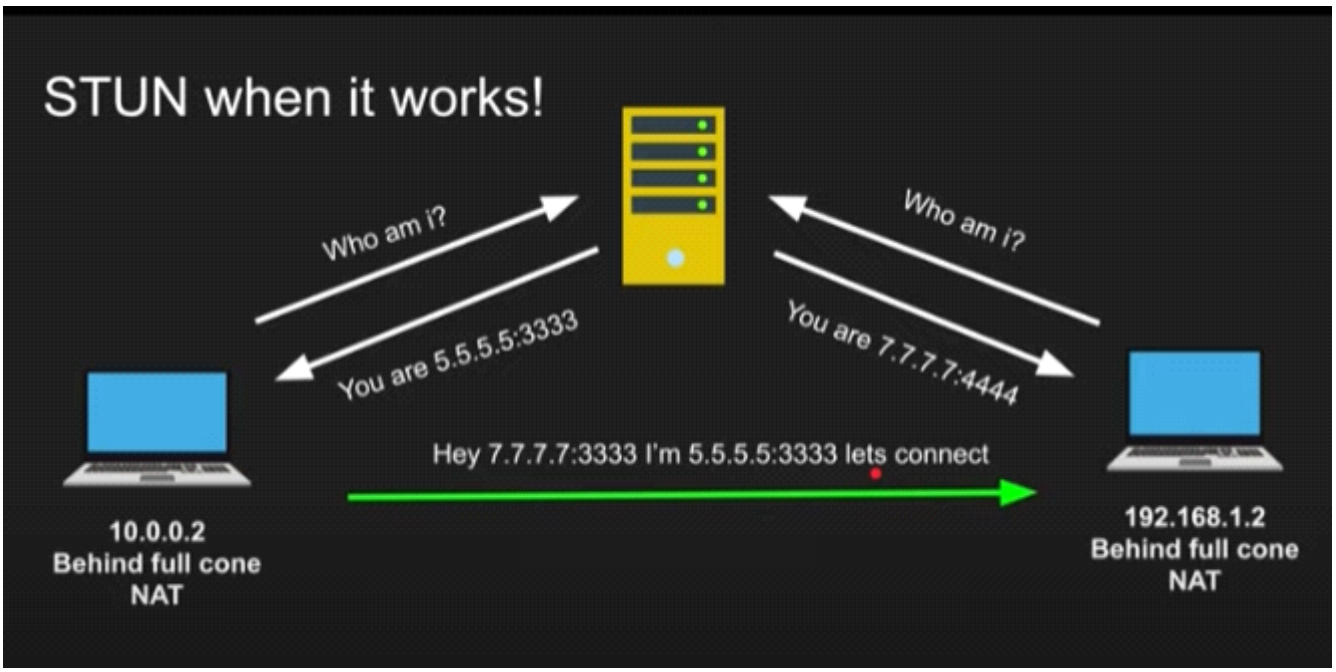


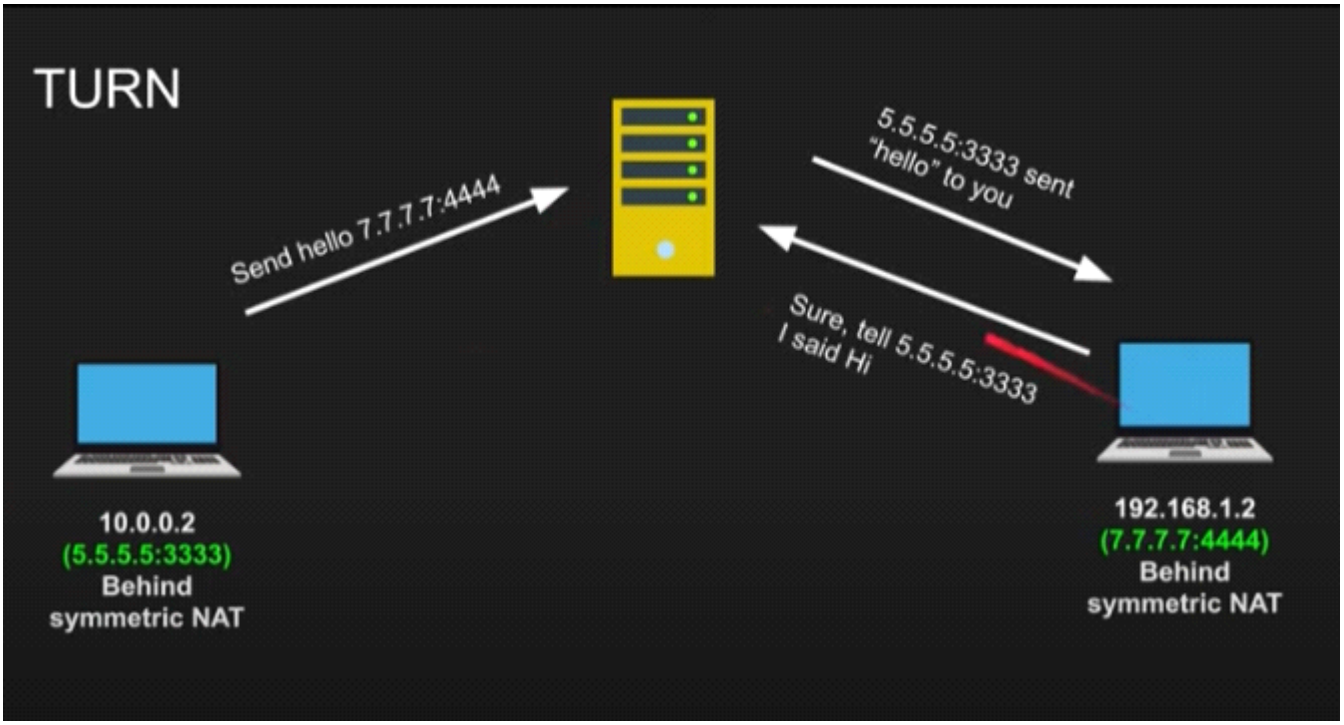
webrtc

Tuesday, September 28, 2021 2:27 PM

[WebRTC Crash Course]







ICE

- Interactive Connectivity Establishment
- ICE collects all available candidates (local IP addresses, reflexive addresses – STUN ones and relayed addresses – TURN ones)
- Called ice candidates
- All the collected addresses are then sent to the remote peer via SDP

SDP

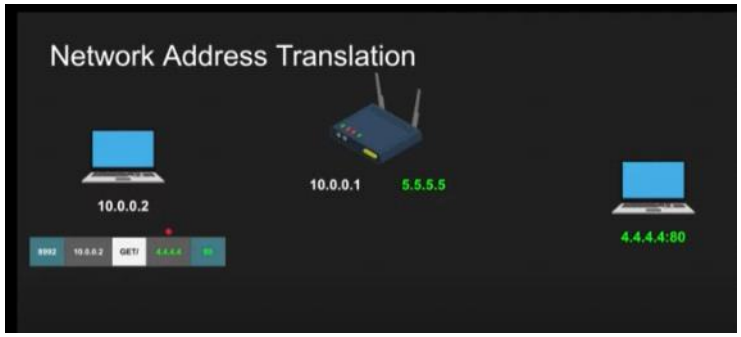
- Session Description Protocol
- A format that describes ice candidates, networking options, media options, security options and other stuff
- Not really a protocol its a format
- Most important concept in WebRTC
- The goal is to take the SDP generated by a user and send it “somehow” to the other party.

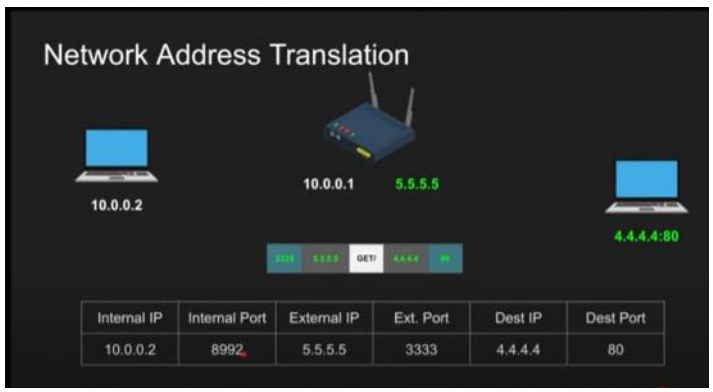
WebRTC Demystified

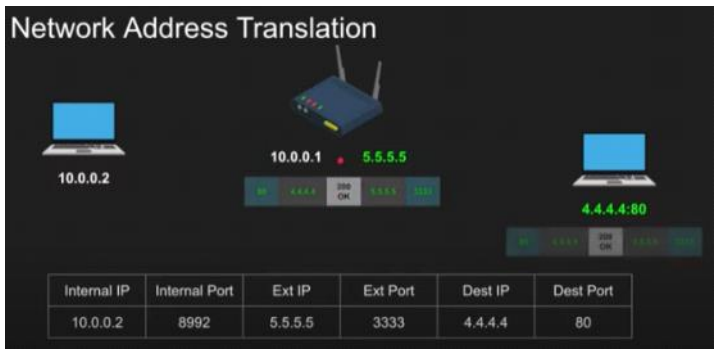
1. A wants to connect to B
2. A creates an “offer”, it finds all ICE candidates, security options, audio/video options and generates SDP, the offer is basically the SDP
3. A signals the offer somehow to B (whatsapp)
4. B creates the “answer” after setting A’s offer
5. B signals the “answer” to A
6. Connection is created

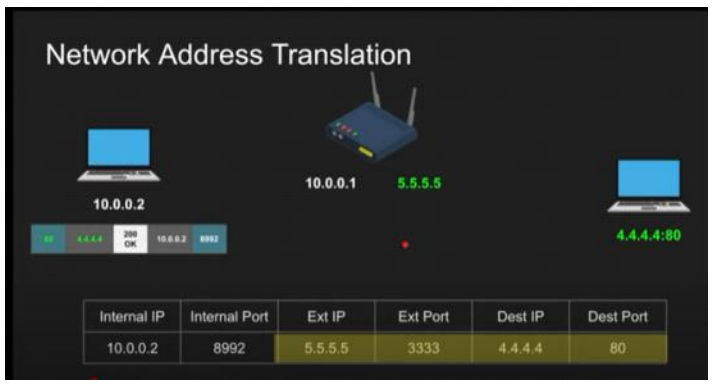
NAT

Tuesday, September 28, 2021 2:33 PM









ARP -> Address resolution Protocol