Design a chatting application like Whatsapp, Telegram etc.

Example of Chatting Application :—

① Whatsapp ② Telegram ③ Facebook Messenger
④ Slack ⑤ Teams ⑥ Slack ⑦ Web Ex

Functional Requirements :—

① User can do one-one communication or can do group messaging
② User can see the old messages
③ Messages should be send in proper order.
④ Send message in form of text, image, videos etc.
⑤ Notification about the message
⑥ Message should not be lost, if recipient is temporarily unavailable.

Additional features :— ① Message Encryption
② Delivery & Read status checks

Non-Functional Requirements

(1) Real-Time / Highly Performant, low latency
(2) Highly scalable (5) Secure (Encryption)
(3) Reliable

Scale of this System

(You can ask / learn the users / interviewer)

Total Active Users = 2 Billion
Daily Active Users = 500 million

Back of the Envelop Calculation / Infrastructure Estimation

Let, 1 person on an average send 100 messages per day,
Let average size of 1 message = 100 Bytes
Total data for storage of data for 1 day

$= 500 \text{ Million} \times 100 \times 100$
$= 500 \times 10^6 \times 10^4$
$= 5 \times 10^{12}$ Bytes
$= 5 \text{ TB}$

Storage for messages in on years $= 365 \times 5$ TB
$= 1825$ TB
$= 1.825$ PB

Bandwidth Estimation

Note:- one read ≈ One write
i.e No. of reads ≈ No. of writes

Data Generated in 24hrs = 5 TB

Data Generated in per second $= \frac{5}{24 \times 60 \times 60}$

$\approx 460 \text{ MB/s}$

Number of Servers / Machine Required

$$\text{No. of Machines} = \frac{\text{Total Number of connections}}{\text{Number of connections per server}}$$

Since, Daily active Users (DAU) = 5 Million
Let, To make a connection, Server take 100 Byte of data to make connection

Let, RAM of Server Machine = 16GB
Also, if 10 GB is available to make connection :-

$$\text{Number of Connection} = \frac{10 \times 10^9}{100}$$

$$= 10^8$$

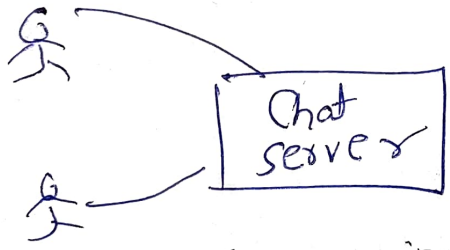
So, A single server machine is capable of making 10^8 connections.

Note:- Chatting is not a CPU intensive task

$$\text{No. of server Required} = \frac{50 \times 10^6}{10^8} = 0.5$$

So, server just accept the message and delays to another, hence we can handle 10^8 connections in single server machine itself it is capable to do that

But we do not do this → As it can lead to SPOF, single point of failure

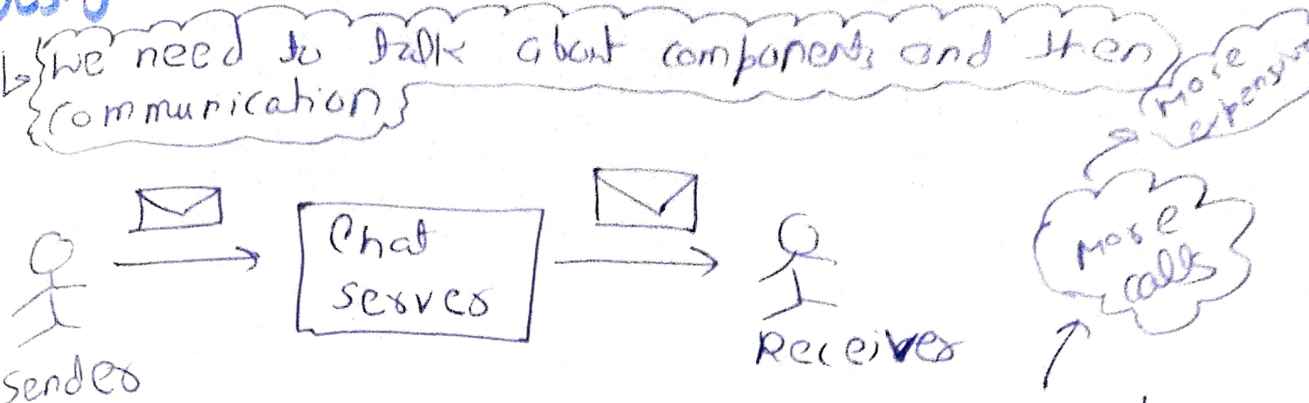


Why using multiple Server instead of Single Chat Server which can do work alone itself

① Easily Scale ② Higher Availability
③ Locality Advantage
↳ means if A person is talking in India to another person in India, their connection is maintained in a Chat server located in USA, so I will try that whenever two users communicates in different region their connection is made by that server so that they are closer to each other because, I have to design Low-latency application

Deep Dive into High-Level Design

↳ We need to talk about components and then communication



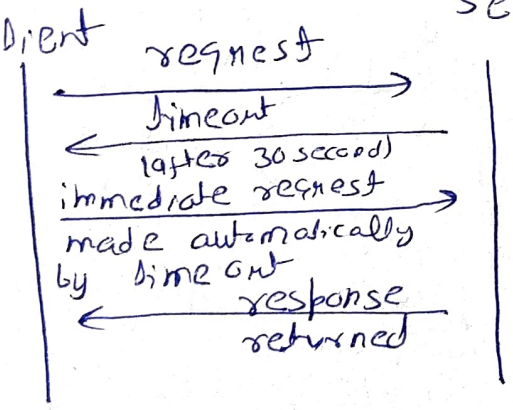
If http connection are made in messaging system to connect the users :-

B/w client & Server, there may be multiple http calls that result in blank response if receiver has not sent the response or not available to send the message.

↳ This strategy is called **Polling** in which you are calling the server multiple times, checking if server has the data you are looking for and then getting the data back from the user

To make it optimized :- → less number of calls → less expensive

DO Long Polling



Client connection made here will wait for a time which is decided by timeout, After 30 second let it be timeout it refuses timeout and again make the call immediately, if response is made by Server, it return the response

Long Polling helps in reducing the number of new http connections

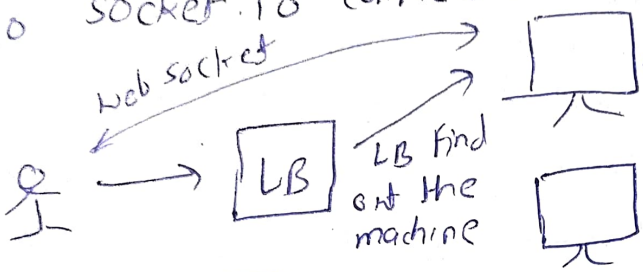
Disadvantage of Long Polling :-
Thread is blocked.

Note :- If you are using multitasking, try to avoid long polling

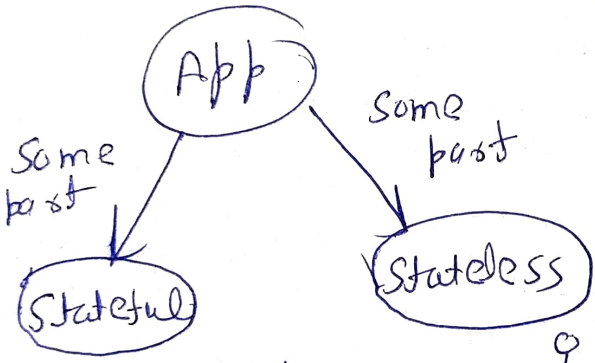
Important Point about Web Socket Connection
also called Stateful

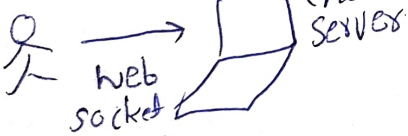
① 2 way communication.

② Why Stateful Communication :- Since you are making connection for a long period of time, we make consistent connection i.e machine is fixed for that connection so socket.io connection are involved



After finding machine, LB goes out of the picture and web-socket connection is done b/w user & machine/chat server directly





(connect with chat server)

text, img, video are also passed through web-socket connection

Several machines are here to provide different services like Authentication, service discovery, provide updates

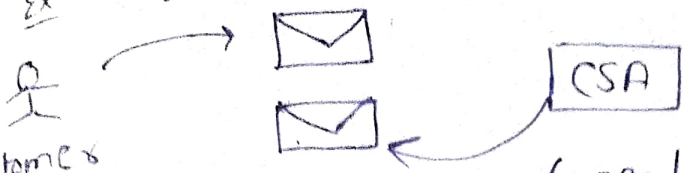
for these, thing you typically make use of http call

③ I make use of LB to manage the state of the system

Concept of Refresh Token / Access Token

① Mostly used in Banks

② Ex:- Customer service Agents

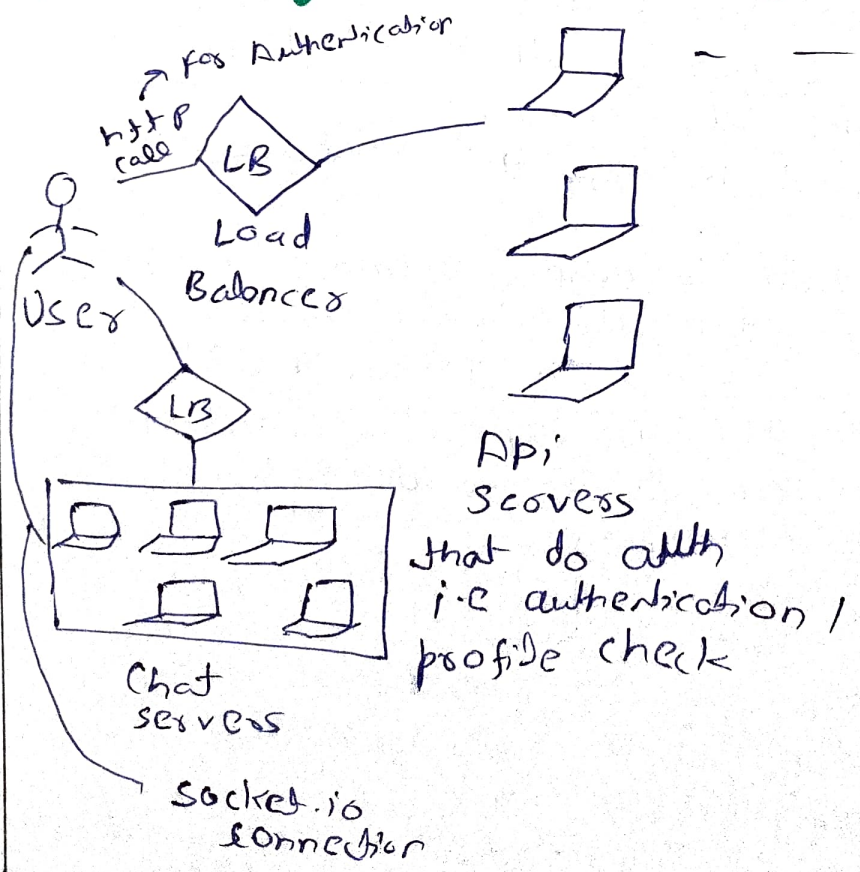


Let, customer do not go & send message for 2minute CSA become expired, all this happen from the concept of Refresh Token / Access Token

③ Online Banking Portal like HDFC, ICICI, if you do not use it for few minutes, you will realize that login is expired, you have to relogin again

Why it happens:- Everytime you do authentication, you will get a token, token has self life (Token has TTL i.e Time to live) so if everytime you send a message, there is a new token that gets generated which has TTL say of 60seconds, so within the 60seconds if you don't send the another message, this token does not get refreshed and if token expires your whole session i.e the web-socket connection which was created was disrupted so, basically you are able to manage because of the refresh token

How things look at high level design



Choose Appropriate Database to store message

1. Database should be scalable.
2. Message :-
   (a) All message must be stored
   (b) Historical messages → Load Later
   (c) People watch the latest message
3. Schema of Message :-
   1. Message id
   2. Time stamp
   3. From
   4. To
   5. Content

Make it as primary key i.e Message id as unique identifier.

To make a message unique in chronological order

Message ID :-
1. Should be made unique identifier
2. Sorted based on time i.e chronologically sort the messages

How to create GUID / UUID?

Problem statement of system Design

Used to create unique ID

Note:- RDBMS provide you feature of Sequence Generator that will generate Unique ID for your new data. but NoSQL world, this feature is not available.

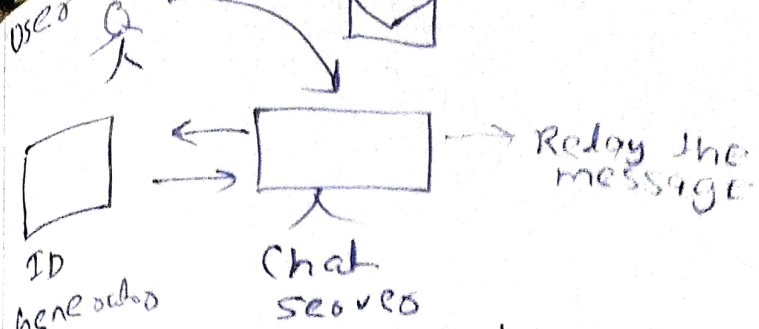
Since, data is huge we cannot rely over RDBMS, we have to dive into NoSQL world.

In NoSQL world, we can make use of ID Generator system that will do the same task as done by Sequence Generator in RDBMS.

How ID Generator System Works?

(a.) This system uses Process ID, MAC Address, timestamp, random number to generate unique ID of message

b.) Each thing is of 16 bits. so total 64 bits are used to make unique ID



Note:- Why not used counter in place of the ID generator system is because inconsistency can be created & spof lead to make our system less available & hence less reliable

Cluster Manager:-

In distributed world, we use cluster i.e group of machines to do a task, we make use of Load Balancer that know how to select a machine based on certain Algorithm like Round Robin, Sticky LB etc. but finding the right machine to perform task like finding right servers to do chatting- depends on multiple factors like how many connections are made by servers or maintained by server, health, state, locality etc. of server.

So, in distributed world, when you want to manage the cluster well Cluster manager comes into use.

What Cluster Manager can do:-

① Check if every machine is up or down i.e ~~health check~~ health

② It is able to do settings.

So, use Cluster manager to check how much load is taken by unique machine, how machines are connected to each other

③ You can do LB by Cluster Manager too.

④ You can also ensure that if one machine goes down it will bring up another machine in place of down machine.

Ex:- Zookeeper → One of the most common Cluster Manager to manage the cluster of servers.

See architecture of zookeeper & get information about it from Google or Chatgpt

Containers :-

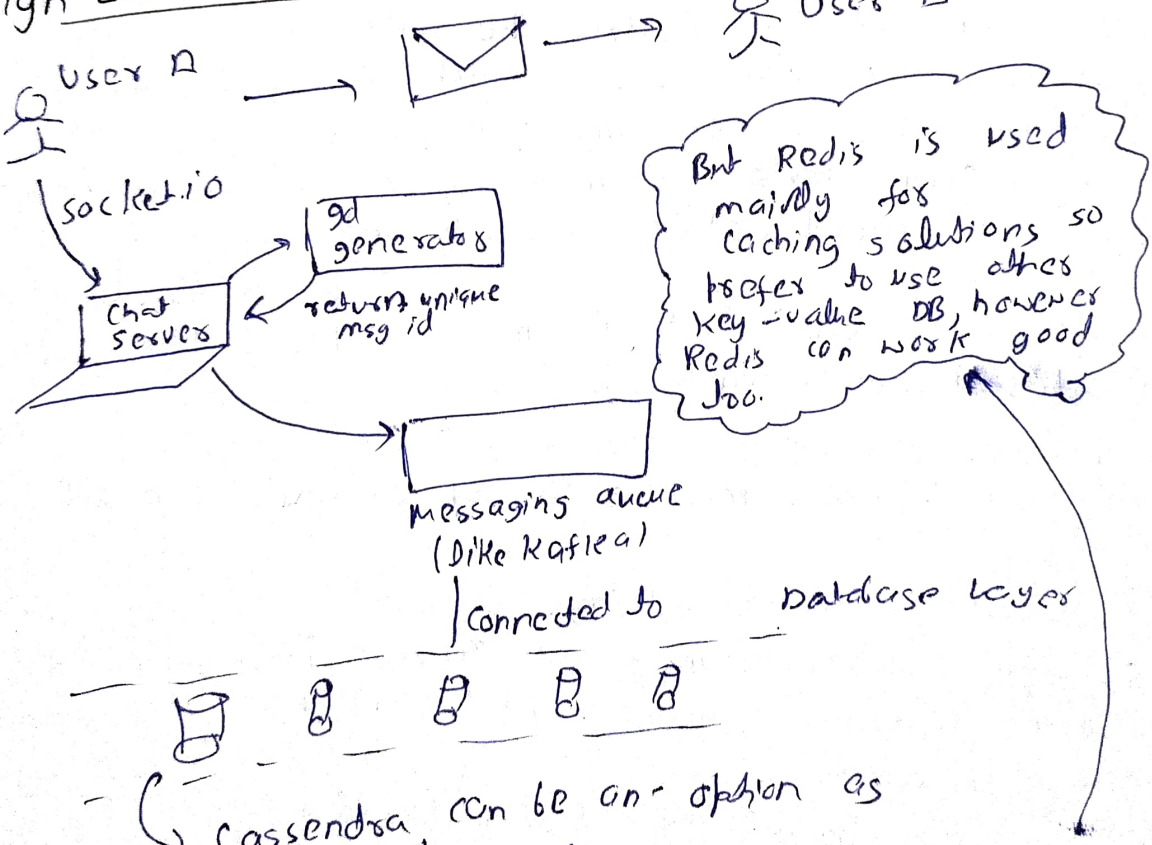
Example of Distributed system, Containers are self-sustained process which can itself kill or run.

KUbernative is used to manage the cluster of containers in Docker PC. Opposite to do task of IKEA

HTML CSS SQL JS → If all are in same, it is container.

Containers

High Level Design of Messaging System



Key-value DB can also be used to store data like Redis because for the basis of msg id it can retrieve the content in constant amount of time.

Note:- Facebook Messenger uses Cassandra to store data

Note:- Where do you store User Data

Since User data is small, so typically you make use of RDBMS. Note:- User data is relational & so that's why we are making of RDBMS i.e MySQL

Why Not used MongoDB instead of Cassendra in Messaging System

Since, Schema of message is fixed, it is not dynamic, flexible etc, we typically do not prefer MongoDB in that case.

MongoDB is used where dynamic schema is available like.

On facebook, posts has comments, comments can have comments or likes so here schema is not fixed; so in post we typically make of MongoDB to store post data.

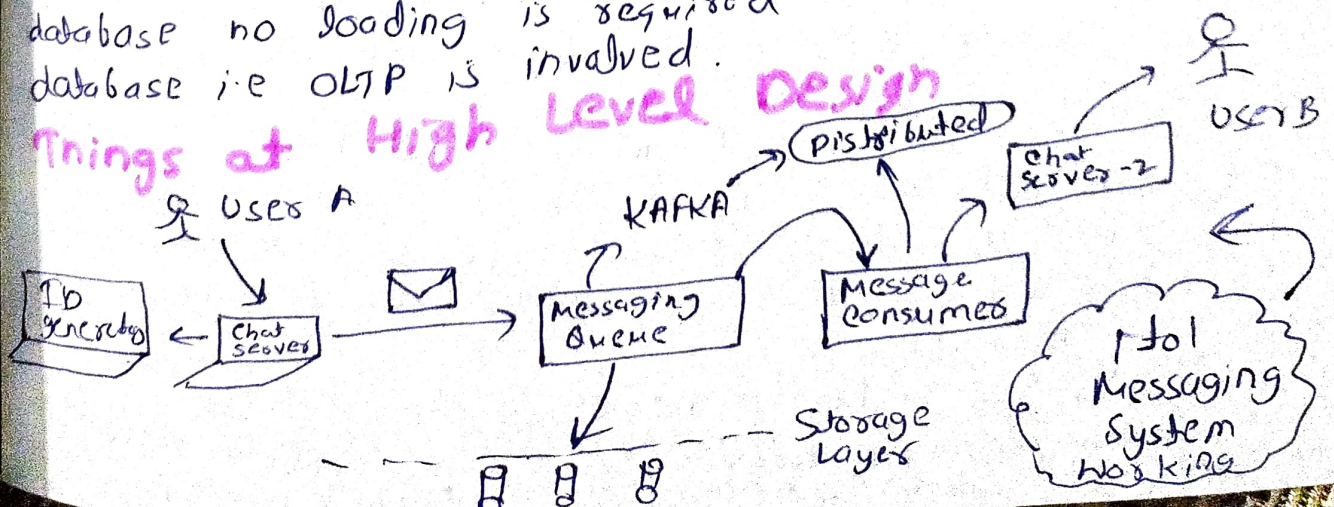
Posts Wills be stored in form of Documents/JSON and hence MongoDB Will le much better here.

Video/Audio take time to load in Messaging System while text take less time, Why?

Note:- You cannot store or it is not preferred to store audio/video files in Regular database, Distributed file system like cloud storage (HDFS, S3 etc) are preferred to store audio/video file and their links are stored in regular database. so in regular database, we store links of audio/video, hence you need to fetch content from link so it take time to load audio/video file.

Note:- On regular database no loading is required i.e transaction database i.e OLTP is involved.

Things at High Level Design



Note:- In one go, you cannot make a perfect system

Work of Message consumer:-

1. It gets the message from the queue and from message it knows which machine it is pointing now.
2. This message consumer will have the information that to which chat server our receiver is connected to.
3. It will make call to chat server-2 and send it the message.

How to ensure that our bandwidth of sending the message is not choke

To prevent choke of bandwidth of sending the message we can put some restriction that any message which is to be sent must be not greater than X MB. like 1 MB size you cannot send a message greater than it at a time.

Identifying weak points of HLD:-

1. Consuming from KAFKA → All message are sent to Kafka.
2. Find which chat server to send/pass Message

Improving KAFKA Topic

Since, you are system designer, so I can choose topic for each user.

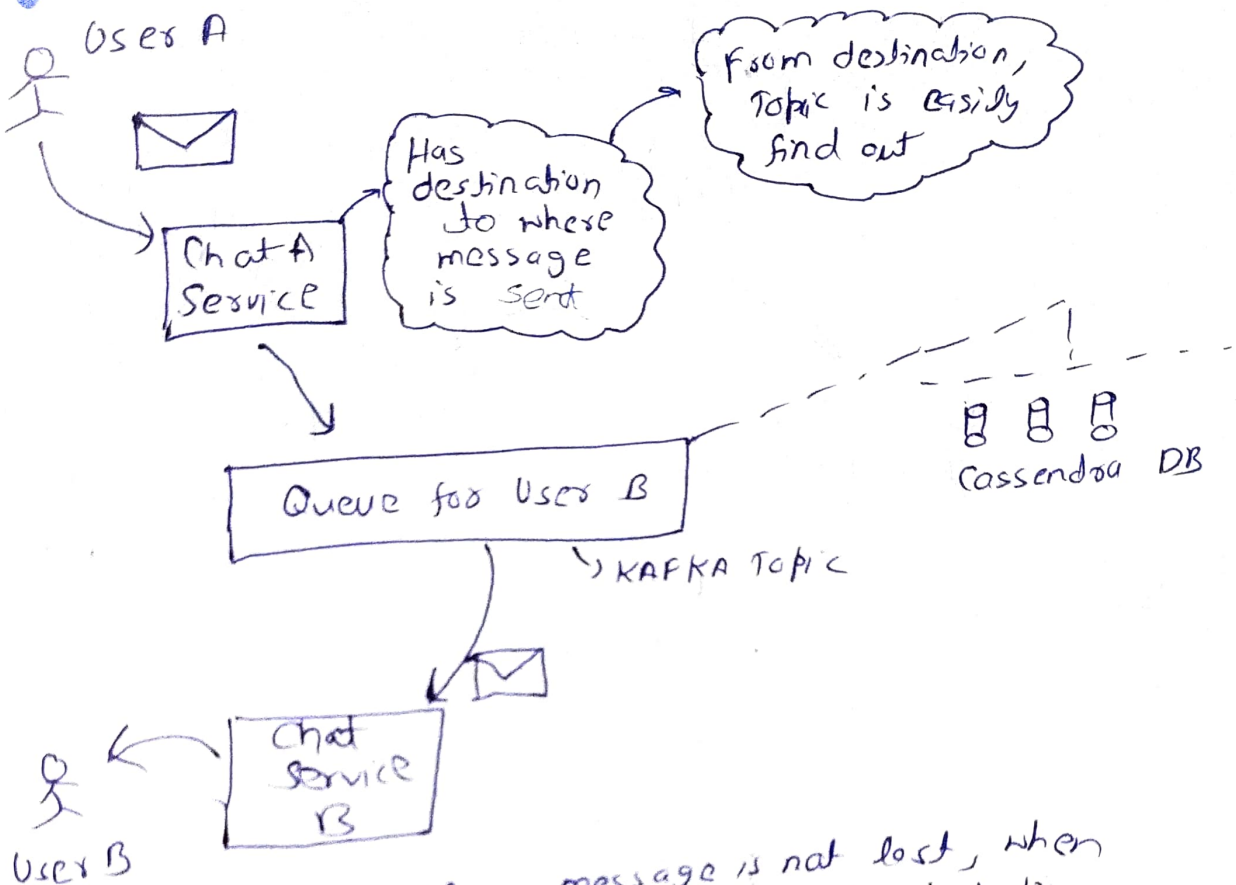
If I send all messages of different user to some topic, things get tough for message consumer to do task.

So, ensure each user is given a unique topic so Message consumer will not have search for user to show its message it directly call to that unique topic, get the message and display to the user.

Normally, I have to ensure that for each user when they register, they should be assigned Message Queue, to be decided for that user. Message Queue can be considered like a Mail Box.

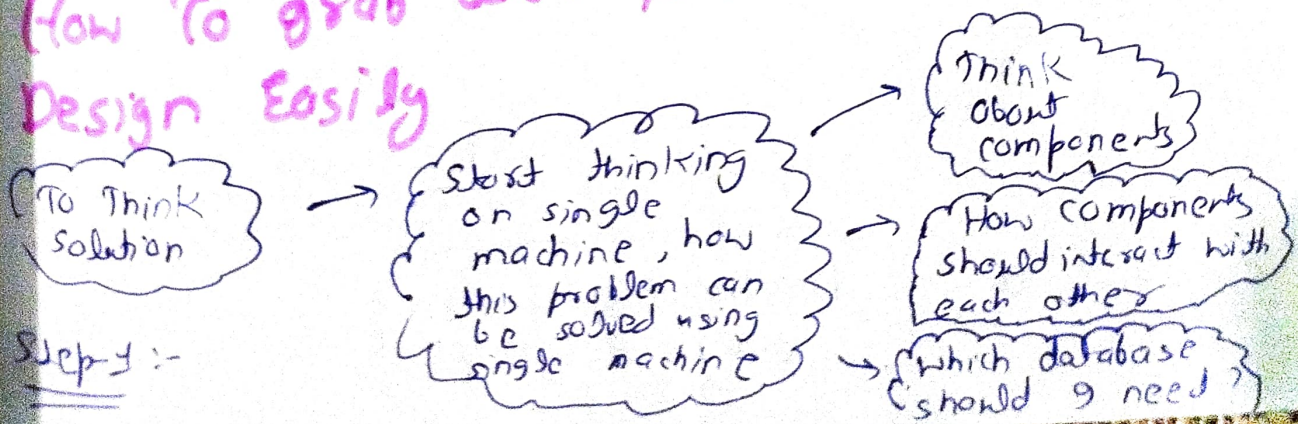
Now, this leads to scalable solution as no message consumer will be required, chat server can do work of message consumer.

Final Design or Improved Design of Messaging System



When User B is offline, message is not lost, when he/she becomes online, chat servers to which User B is connected check that it's Topic has some message if yes, takes message and display it to user, hence if userB is offline message is not lost also, no message consumer will be required as chat servers has replaced task of message consumers. This has led to more scalable design.

How To grab each question of System Design Easily



Note:-

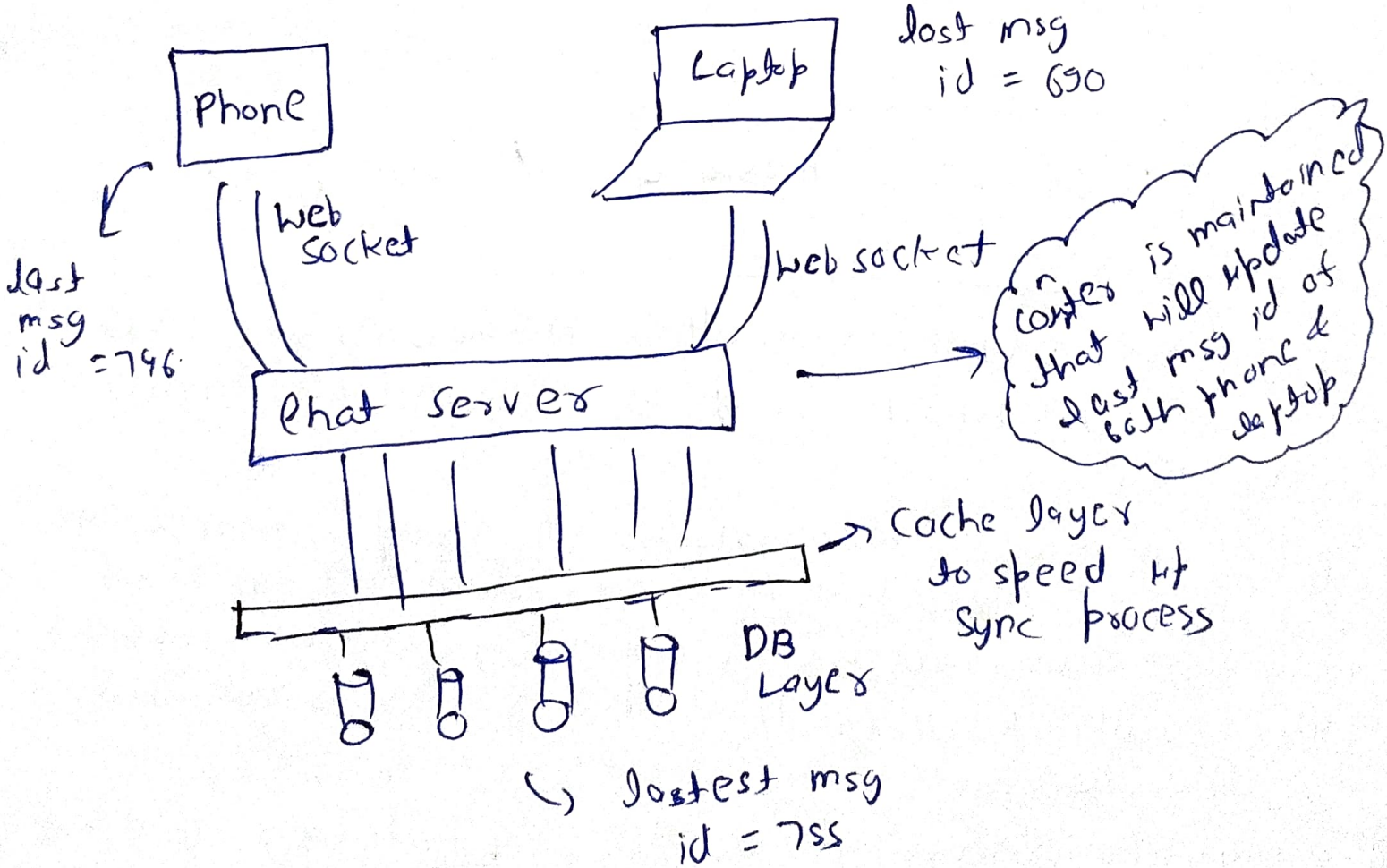
Polling and Long Polling connection are achieved with the help of http connection.

Additional features / Enhancement of messaging system

How to ensure that each device has the latest data / synced

Assume, user has laptop, mobile phone. Now, if he use Mobile and laptop remains off to use messaging system, we need to sync the activity of user to both devices so that this might not happen user get stale data at laptop.

More Modular Design



Back up feature in Messaging System

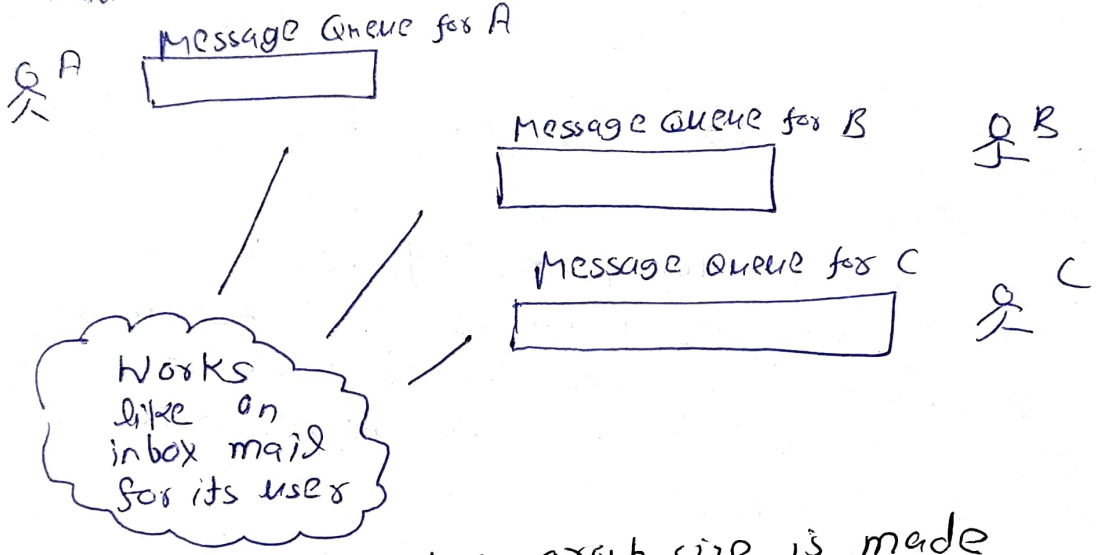
Backup is maintained in Local device and Cloud storage. These backups are for the seen data

Why Local Cloud Storage not Central server DB to do backup.

Reason is that Local Cloud Storage is faster than backup.

How to design Group Chat ?

① Every person is given it's own Message Queue



Why in messaging system, group size is made finite?

This is because.

> Number of events/ message ∝ Group Size published

For ex:- We Chat is another messaging application in which group size cannot be more than 500.

Note:- Message also consist of group information i.e Message consist of that it will of one-one communication or group communication. And on the basis of that chat server send msg to the required queue.

How my chat server decide that which msg in a group should be pushed to which msg queue of member ?

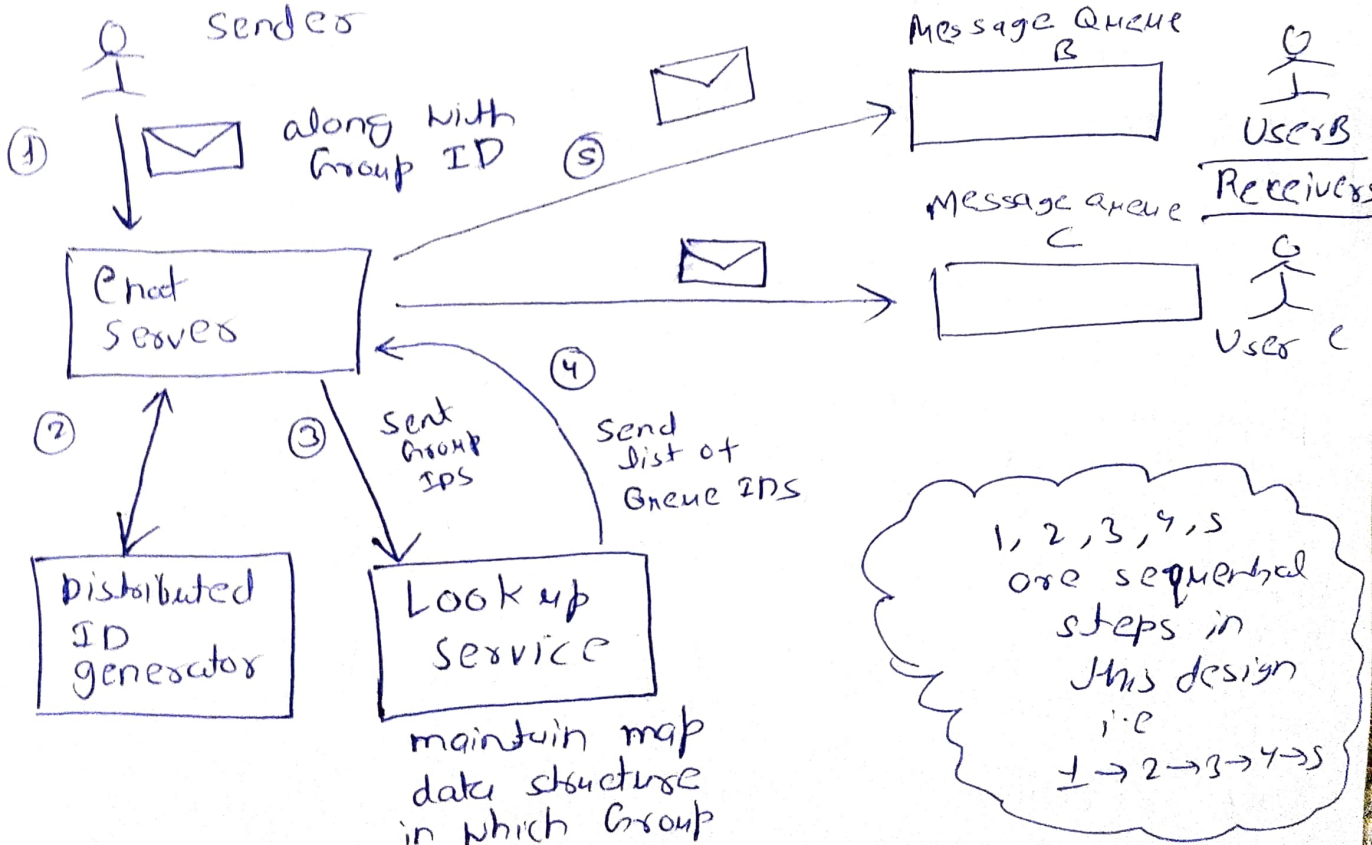
on system, there can be multiple group like $G_1, G_2 \dots G_n$.

But every group is assigned a unique id and we can create some kind of map which consist key as 'GroupId', value as 'List of UserIDs'

By maintaing a map data structure where in the **constant time** you can do **look-up** based on the groupid that what are the different users in the group.

Map
(key) → groupID
(value) → List of UserIds

More Modular Design



How to reflect the online/offline status of user to it's friends

A system called as Online status server is added which keep an eye or monitor the web-socket connection made b/w chat server and the user.

Work of the Online Status Server :-

It is going to check that what are the active web-socket connections, by the active socket connection it comes to know that what are the two sides of socket connection i.e it gets the User-ID.

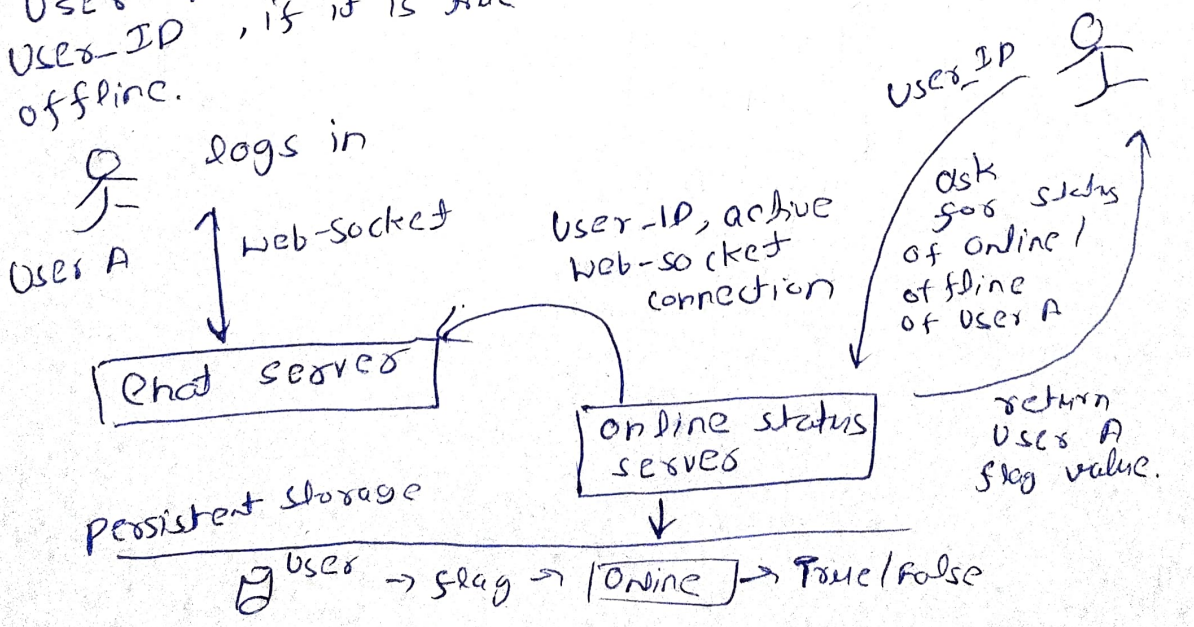
And for this User-ID in persisted storage, for each of the user a flag named status can be maintained that will tell that user is online/offline controlled by online status server.

So, whenever user logs in, status is set as online.

Where to use flag:- When there is boolean outcome, we typically make use of flag.

How do user check his/her friend are online or not

Make call to online status server by giving their User-Id, OSS send request to Persistent User DB to ask for the value of flag of User_ID, if it is true OSS return online otherwise offline.



Way to ensure if a system is alive

Let, we have a server and in it some application is running. Now if I have to go and check that my server is live or working or it has gone down..

Then we have to check that the health of our server/application is good or not

If it is healthy we say that our server is alive.

For this, we make use of something called as Heartbeat Test

Heartbeat Test

In this test, we make a server which is used to test the heartbeat.

By this Test, the machine/system after every x second has to tell about its health to the server that dude I am healthy. just like we have to tell our health status to our mother/father regularly.

After y second, if no heartbeat is sensed by Heartbeat server from system, it means that our server has gone down.

Why are we focusing on web-socket connection to check status of online/offline.

This is because when you close Whatsapp window (still it is running in background) your web-socket connection breaks up so keeping track of web-socket connection provide you real-time awareness of that user is online/offline.

How to see last active time of User:-

Everytime when web-socket connection is broken/terminated, we have something called log

So, online status server (OSS) has two tasks it gets a list of all web sockets which is active And it also sees that the web sockets which were terminated, what was the last time it has terminated. By OSS you check:-

① User is Online/offline → achieved in almost real-time

② Last seen feature

Note:- Sometime, it can happen that system cannot send message after x second. That's why you make checkmark that after y second if my system does not send a single response, i will consider that my system has gone down.

Ex:- x = 5second, y = 30 second.

Note:- Heart Beat server can contain state information but it is useful when you work in cluster environment like Big Data Cluster, Kubernative.