Chat Application System Design.

Functional requirements

① 1-1 chat
② Group chat
} text/ images/ videos

③ Read receipt
④ last seen
⑤ whatsApp status / stories → watch already uploaded video
⑥ Chat are stored at server / user app level ?

NFR

① Availability ↑
② Low latency
③ Scalability
↳ 2.24 Billion monthly users.

Envelope Estimations

① Number of users - 2.24B monthly users. (whatsApp)
② Storage - messages are stored on server & Telegram, FB messenger ?
③ ~ 100B messages / day (whatsApp)

Text : Images : Videos
(70 : 20 : 10) → assumption

128 bytes 1KB/image 10KB/video → Compression of media is priority requirement to save space.

∴ Total storage / day

= 70B * 128 Bytes + 20B * 1KB + 10B * 10KB

= 8.2TB + 18.6TB + 92TB

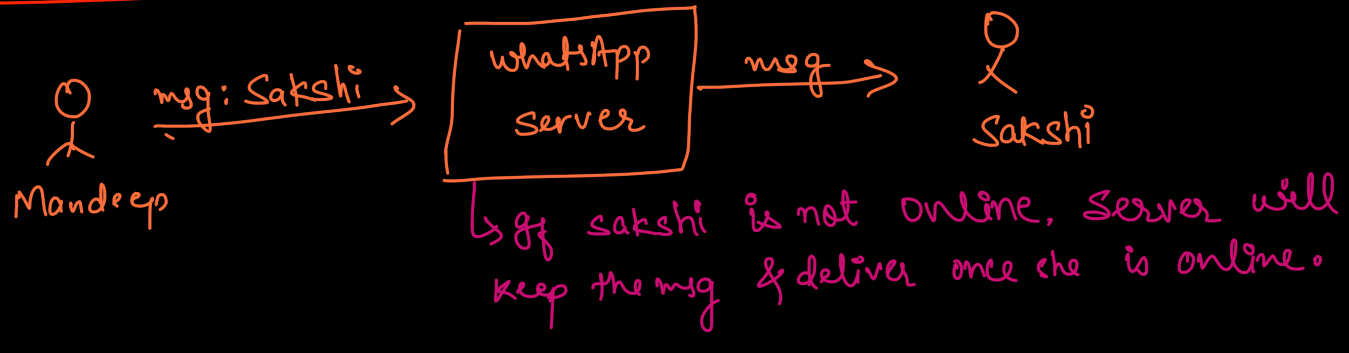
= ≈ 118 TB / day

① This is huge storage requirement basis our assumptions. The point of consideration would be if we really need to keep msgs on server all the time?

② The compression algorithm should be really good to optimize storage space.

③ There are lot of images/videos that are forwarded/shared again and again. Can we somehow figure out while storing and only store it once?

System High Level View



↳ If sakshi is not online, server will keep the msg & deliver once she is online.

How will users establish connection with servers?

① A bi directional connection is preferred to avoid TCP

handshakes again & again over HTTP.

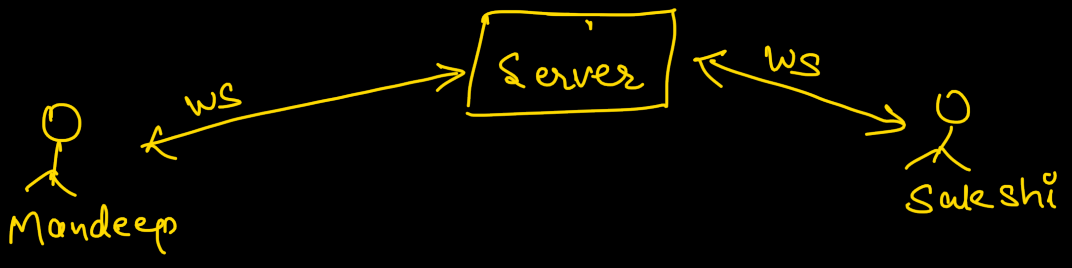
Websocket → operate over 80/443 port

② whatsApp uses XMPP protocol.

Application need to implement "presence" functionality.

Extensible Messaging & Presence Protocol.

↳ especially built for instant msging/presence/contacts list.



Storage ① Maintain user information → userName, phone, photo
↳ when was last online?

② User Chats → texts
→ images
→ videos

① User Information

for user "Mandeep"

① Name
② phone Number
③ Status
④ last active → As this is frequently updated info, can be handled by a separate service + database to avoid user DB updates.
⑤ Account created on
⑥ Account status
⑦ profile picture → link to image dumped in S3.
⑧ user preferences { }
⑨ Contact List

ⓐ any relation between different users?
↳ if a user account is deleted, can it be still in contacts of another user? → for simplicity, assume phone contact list = whatsApp contact list

ⓑ Transactions support? → NO

© Scale of data? → total Number of users.

Both SQL and NoSQL datastores can be explored for keeping user information.
Traditionally, looking at different systems such as Instagram, user info is kept in relational datastores.
with partitioning to scale as users grow.

2) User chats

- @ message Id → unique Id can be local to 1-1 / group
- (b) message from
- (c) message To
- (d) Timestamp
- (e) Content → chat / image URL / video URL

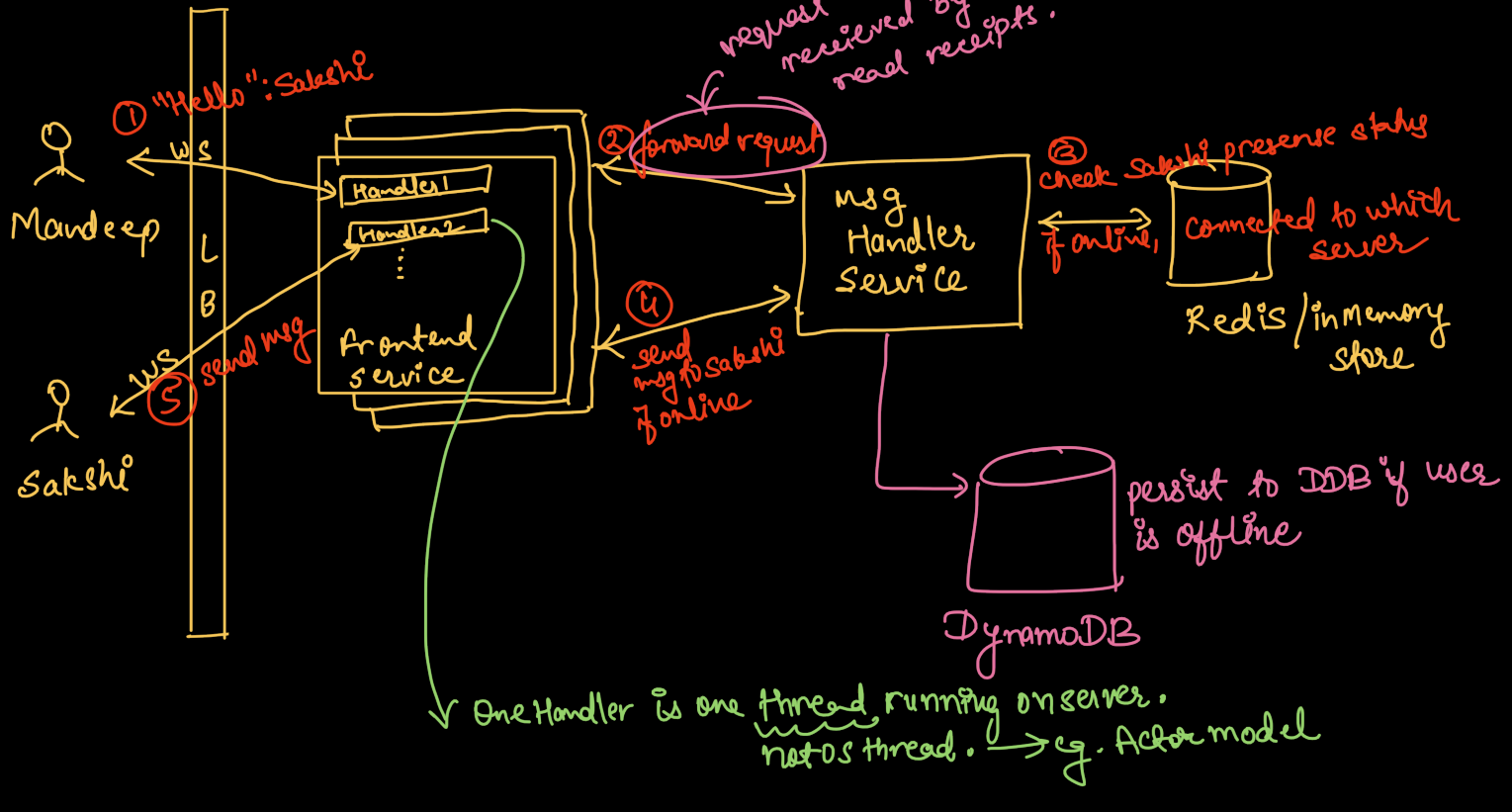
Group msg

- @ GroupId
- (b) message from
- (c) Timestamp
- (d) content
- (e) message Id

@ Scale of Data? → Discord uses ScyllaDB

(b) No relations

Non relational datastore can be preferred choice here.
Such as KV store - dynamoDB.



A Key Consideration ⇒ Should msg Handler Service persist msg to DDB before forwarding to receiver?

↳ Not required ⇒ ⓐ It will save extra hop for storage, for WhatsApp like systems. For FB messenger, it can update storage asynchronously.

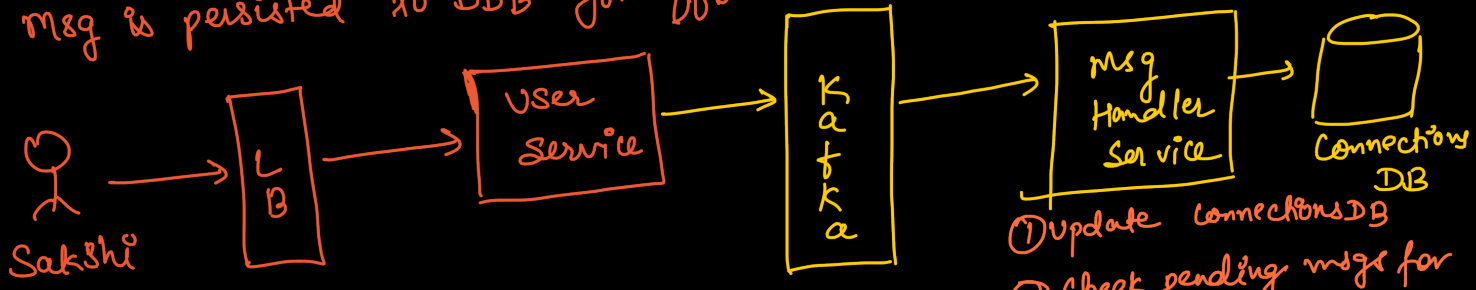
ⓑ In case ack is not received by client in some time, It can resend the msg. ↔ Duplicate msgs at receiver end can be handled in client application basis messageId.

B Frontend service ⇒ The servers will maintain persistent connection with online users.

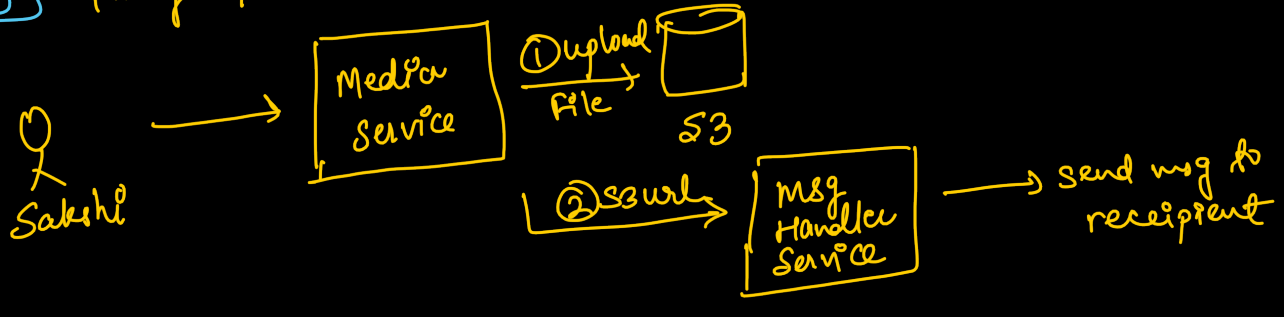
Information of which server is connected to which server is maintained in in-memory connections DB.

C Workflow → Sakshi is offline.

msg is persisted to DDB for offline users.



D Image | Video



E Group messages

Msg Handler Service forwards same msg to multiple clients.

What if few users are online & few offline?

↳ msgs are stored in DDB for offline users similar to 1-1 chats.

Subscribe for more such content

Happy Learning ☺