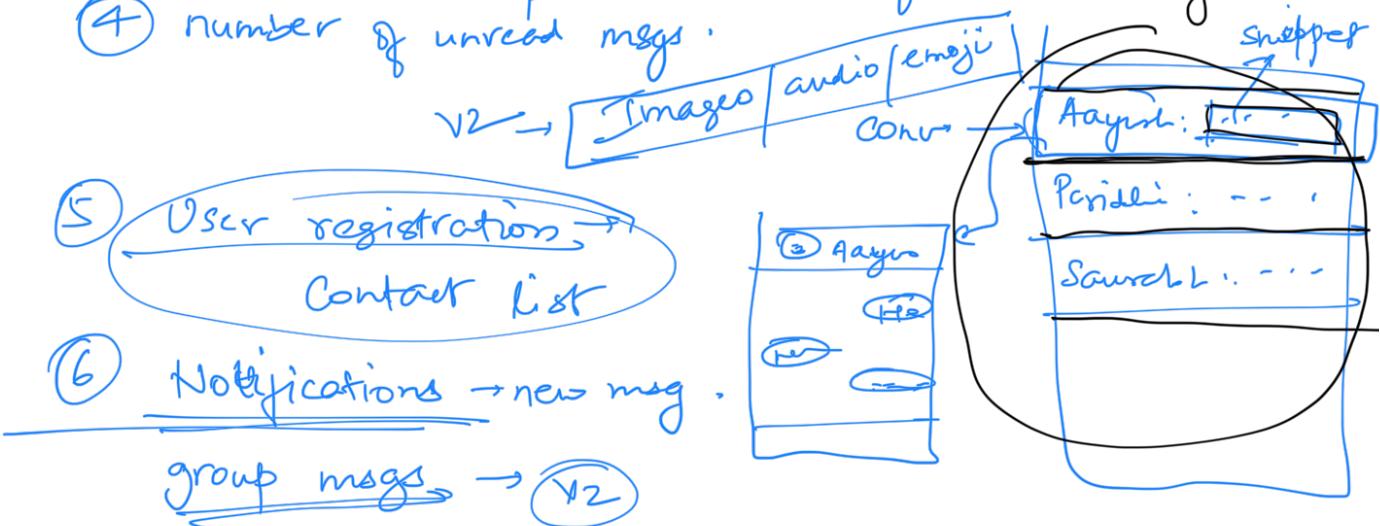


DESIGN MESSENGER

MVP:

- ① 1:1 chats
 - ↳ send messages
 - receive msgs.
- ② List of conversations - most recent
 - ↳ snippet last msg first
- ③ List of messages given conversation
 - ↳ timestamp - most recent first
- ④ number of unread msgs.



Non-functional req.:-

- ① Low latency
- ② HC

Not consistent

- messages out of order
- later msgs via older ones not visible yet

Estimate scale

- (20 Billion msgs sent / day)
 * (~ 200 bytes)

- YES! e
- BOTH e
- ① Sharding }
- ② R or w heavy }
- ③ Capacity →

20 Billion ~ 2.15

msg

$\sim 200 * 200 \text{ bytes}$

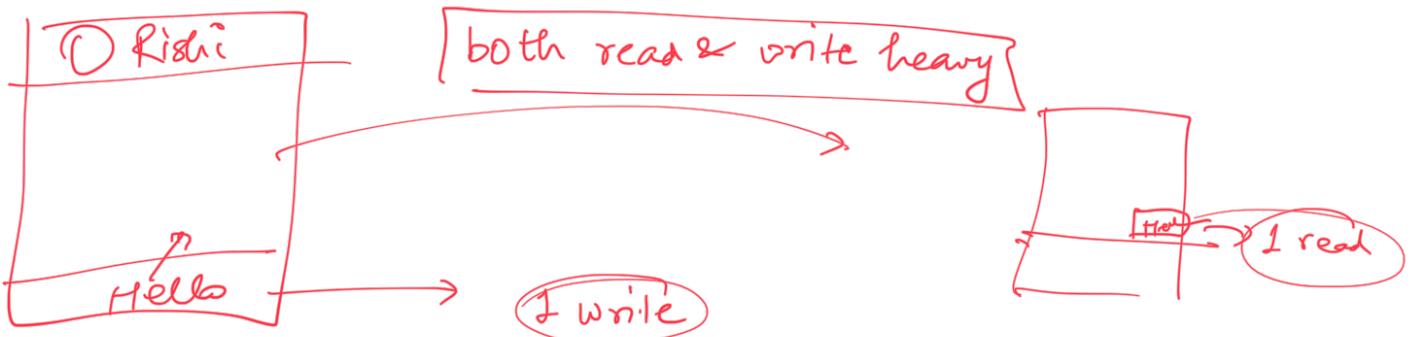
$$= 20 * \frac{4000}{200} \text{ GB} = \boxed{4 \text{ TB}}$$

$$4 \text{ TB} * 365 * 10 \approx 15000 \text{ TB}$$
$$\approx \underline{\underline{15 \text{ PB}}}$$

100 B ← [-text]
 " 8 B ← [-sendgrid]
 8 B ← [-recipient_id]
 8 B ← [-timestamp]
 8 B ← [-msg_id]

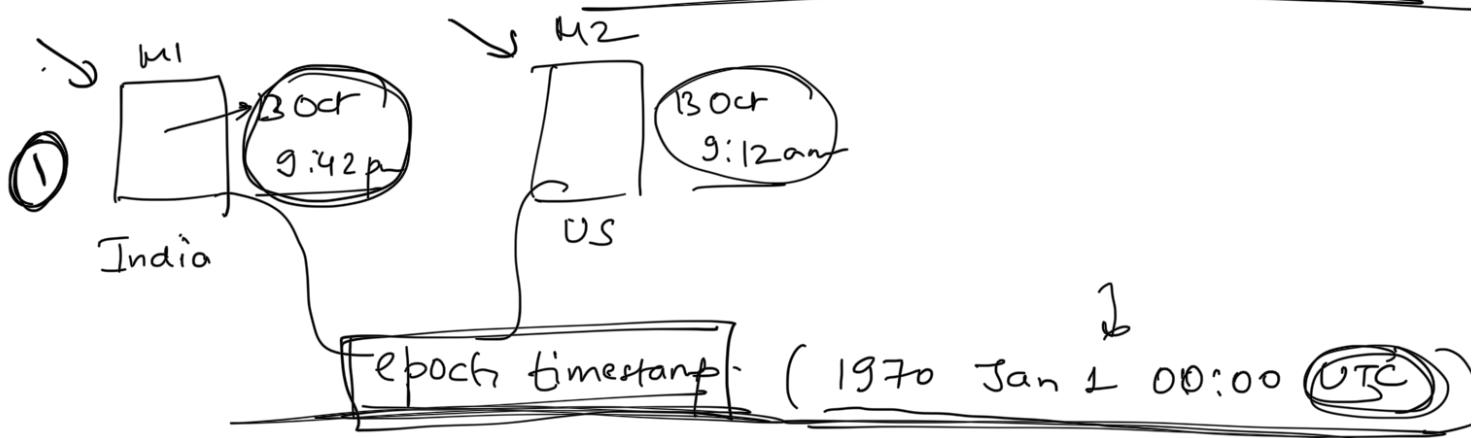
[-
 =
] metadata

$\sim 200 \text{ Bytes}$



API

Send Msg (~~text~~, sender, recipient,



②

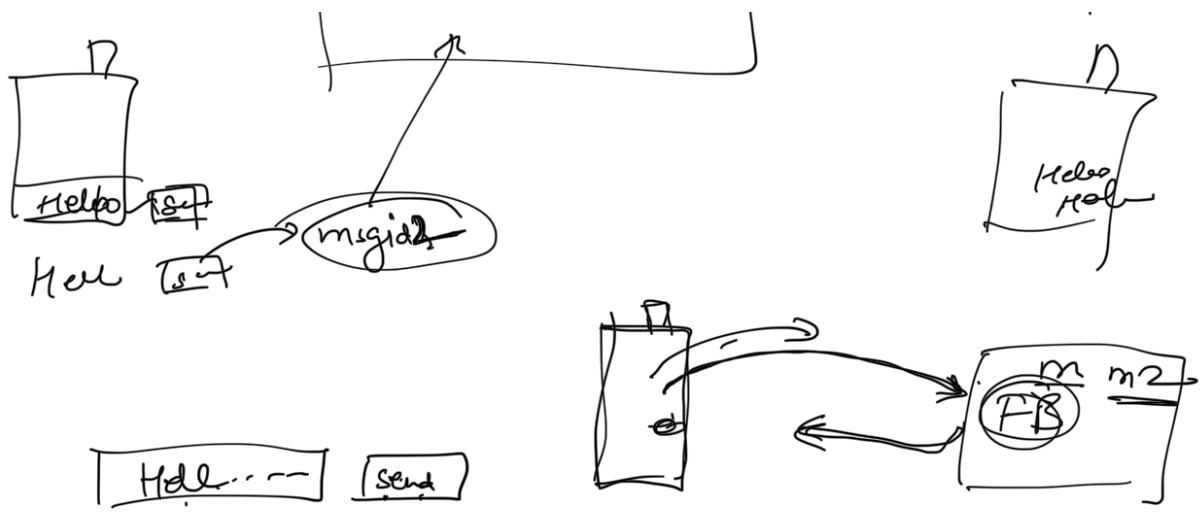
Idempotent

$$X = 10$$

X++;

$$\frac{X}{10}$$





msg:

↗ text → "-----"
 ↗ send → -----
 ↗ rec →
 ↗ msg id →

Message {

get -

"Hello Sahil" ↘
send ↗

Set - —

"Hello Sahil" ↗
Send ↘

Message Obj 1;

Obj. setText("Hello Sahil");

Set

Set msgId ("---")

obj1.setText("Hello Sahil")

Set

Set

Set msgId ("---")

Timestamp

- Timestamp
 - User Id
 - Device Id

D1
 (uuid)

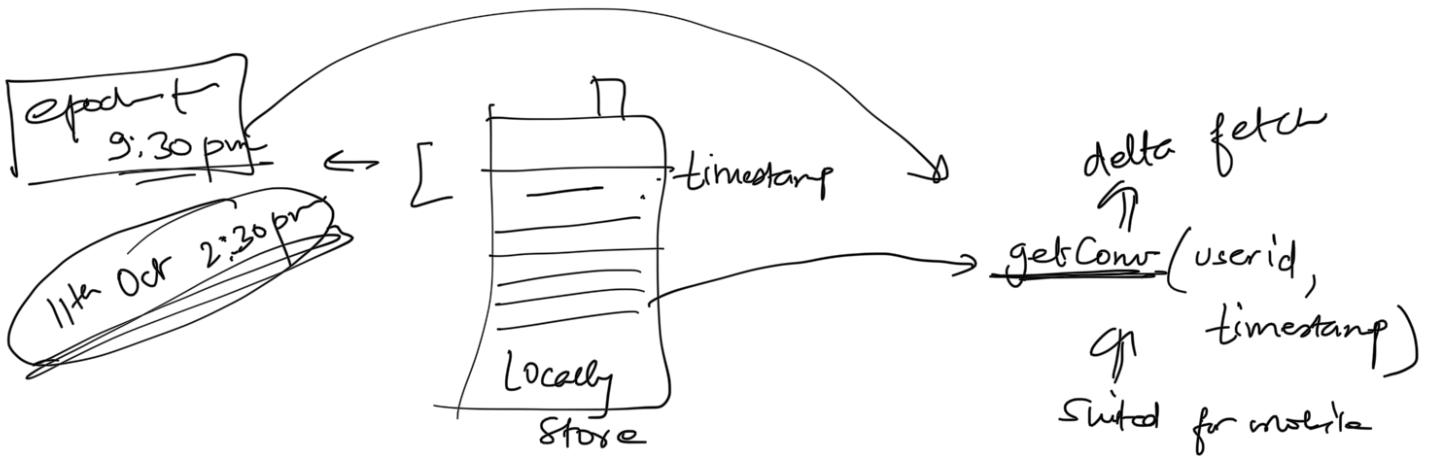
D2



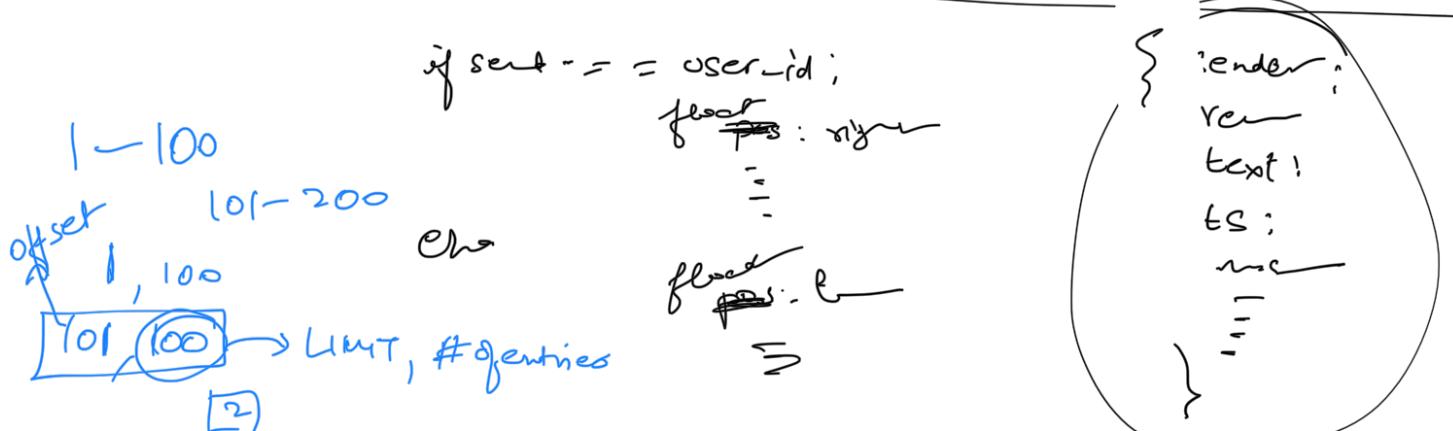
- Send Message(sender, recipient, text, msgId)



- get Conversations(userId, num-conv, offset)

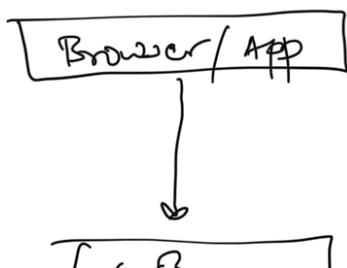


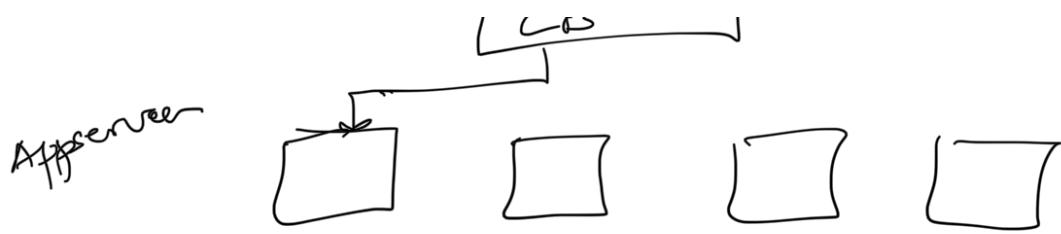
- get Messages(userId, conv-id, num-msgs, offset)



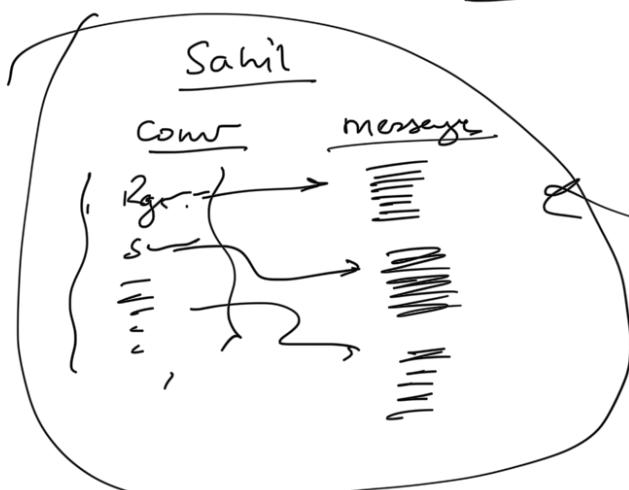
Userid1 Userid2

"Userid1#Userid2"





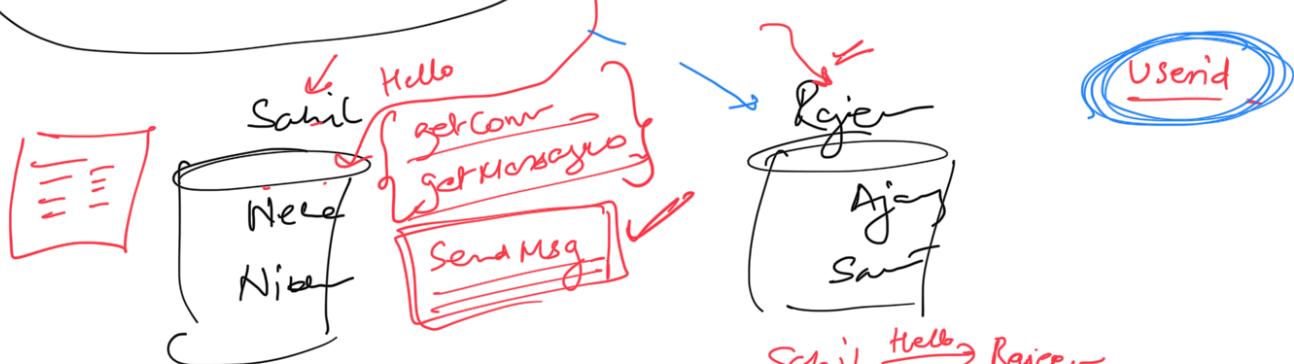
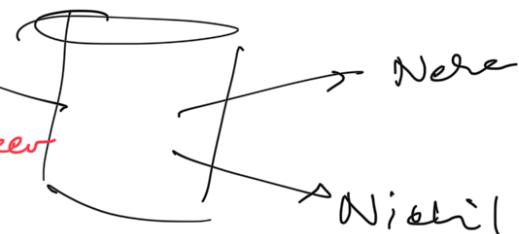
User id



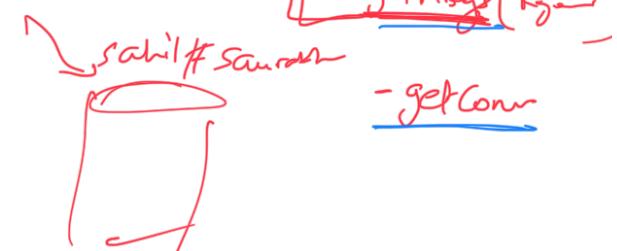
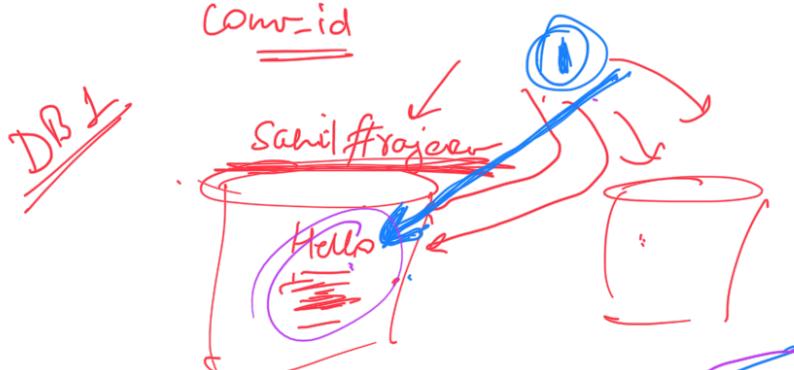
Rajeev

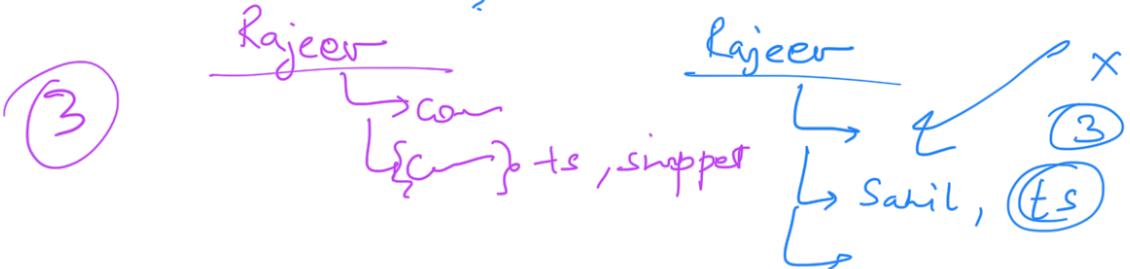
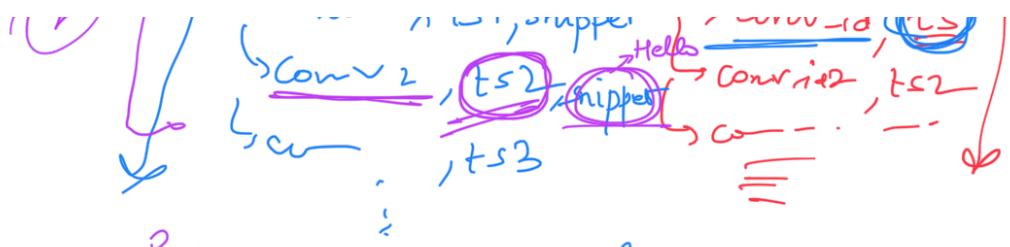
- ① SendMsg
- ② GetConv
- ③ GetMessage

Sahil → Rajeev
Hello



Conv-id





Userid



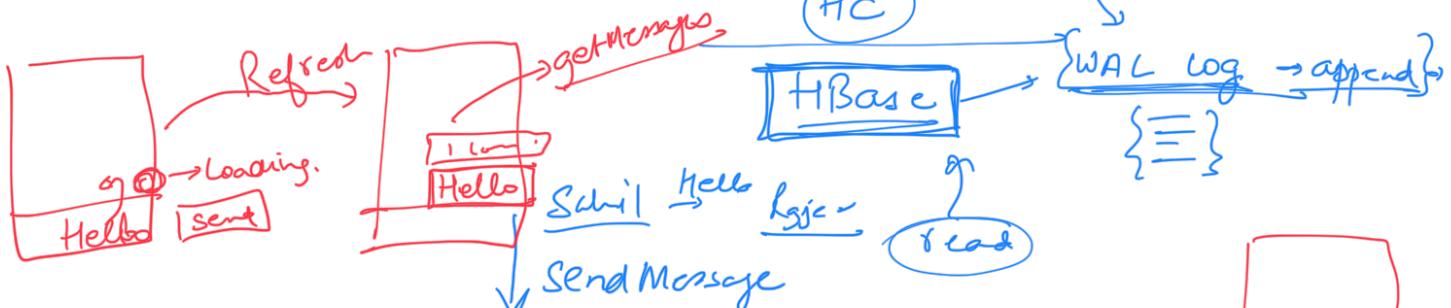
VS CF

VS

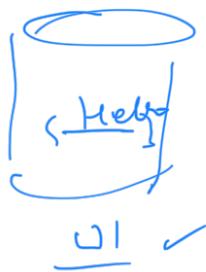
Document

Write heavy

Hc



2
Sahil ✓



1



Hello

