

13/12/2023

# HLD - Messenger

Facebook Messenger ✓

Slack ✓

MS Teams ✓

WhatsApp ⊕

4 Steps

① MVP / Features

② Estimation of Scale

③ Design Tradeoffs

④ System design  
deep dive

## ① Features

→ send a msg / Receive a msg

→ 1:1 msg

→ Group msging

→ Chat history

→ Conversation where msgs are part of.

→ msg delivery near realtime

## Assumed set of features

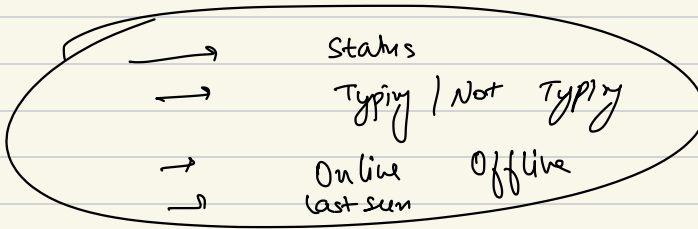
- ① user register
- ② authenticate / authorize

### Common Feature

Notification

### Popular Feature

Search



(2)

## Estimation of scale

$$\# \text{ User} = 2 \text{ B}$$

$$\# \text{ DAU} = 400 \text{ M}$$

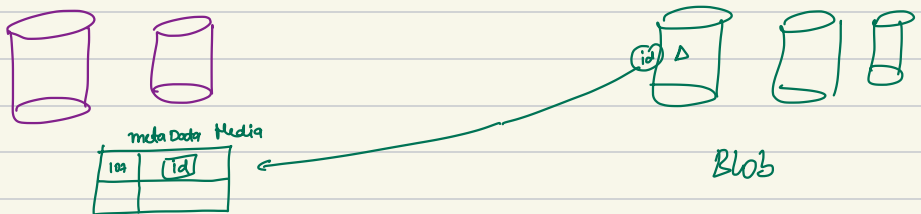
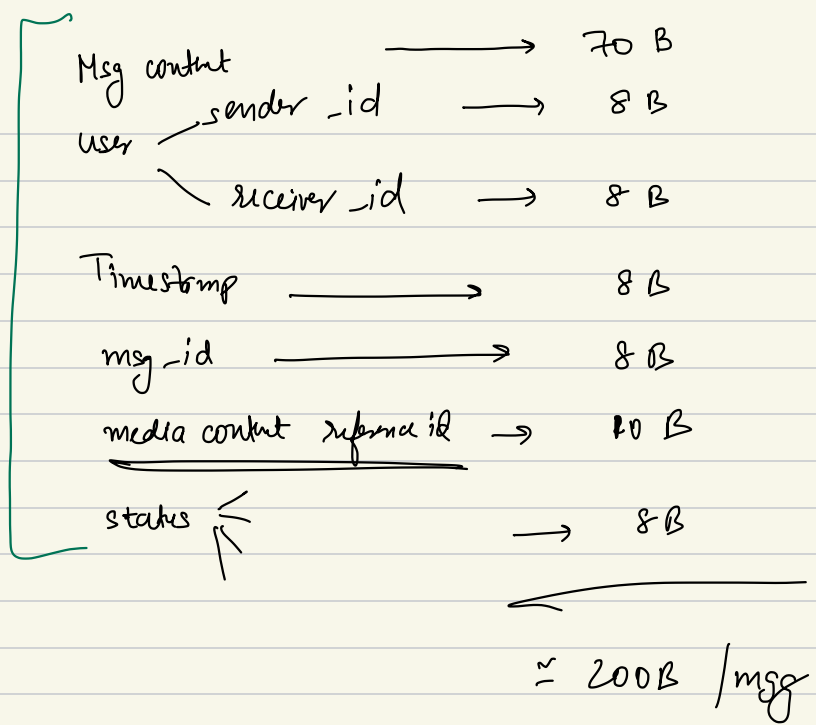
$$\text{Avg msgs / user / day} = 50$$

$$\begin{aligned} \# \text{ msg / day} &= 400 \text{ M} \times 50 \\ &= \underline{\underline{20 \text{ B} \text{ msgs / day}}} \end{aligned}$$

$$\boxed{\text{Write Ops} = \frac{20 \text{ B}}{86400}}$$

$$\text{Read Ops} \approx \underline{\underline{4}} \times \text{Write Ops}$$

Read + Write Heavy System 😊



---


$$20 \text{ B} \times 200 \text{ B}$$

$$4000 \text{ GB}$$

$$= 4 \text{ TB msg/day}$$

$$4 \text{ TB} \times \underline{\underline{365}} \times 10$$

$$= 12,000 \text{ TB} \quad \textcircled{\text{ö}}$$

$$= \text{PBs of data} \quad \textcircled{\text{ö}}$$

=

Sharding is a must...

### ③ Design Goals

① Latency: reasonably low latency

② Consistency: Highly Consistent

$C \gg A$

### ④ Design deepdive

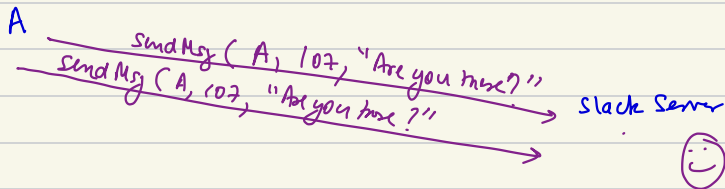
① API design

① sendMsg ( )

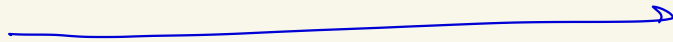
sender-id, conversation-id, text

decur

Idempoten

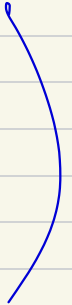


Client

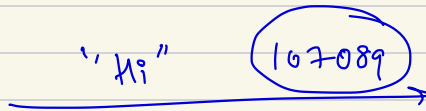


send Message (

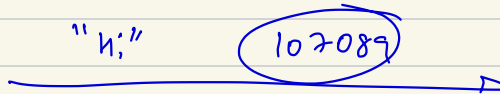
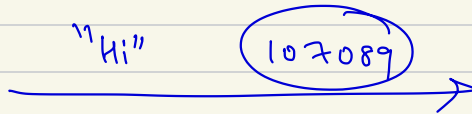
msg-id  
sender-id  
conv-id  
text  
Timestamp



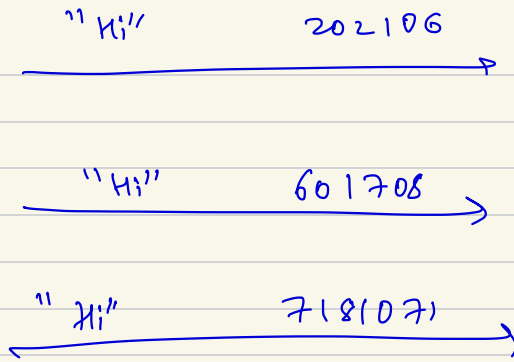
A



retry



A



To create idempotency, The client should generate a unique msg-id 😊

Combination inputs


- Timestamp
- sender-id
- conversation-id
- device-id



① sendMsg ( user-id - sender <sup>auth-token</sup> )  
                  conversation-id  
                  msg-id  
                  msg-text  
                  msg-metadata  
                  Timestamp


---

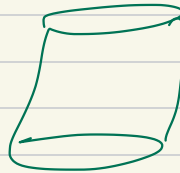
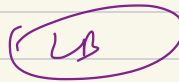
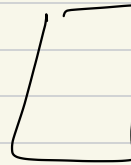
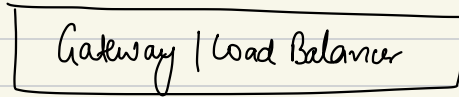
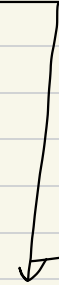
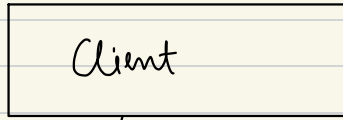
② getConversations <sup>auth-token</sup> → getConversationsList()  
( user-id,  
    offset,  
    limit )



- Nikhil →
- Deepak
- Mohsin
- Alfian
- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_

③ getMessages <sup>auth-token</sup> ( user-id  
                                  conversation-id  
                                  offset  
                                  limit )

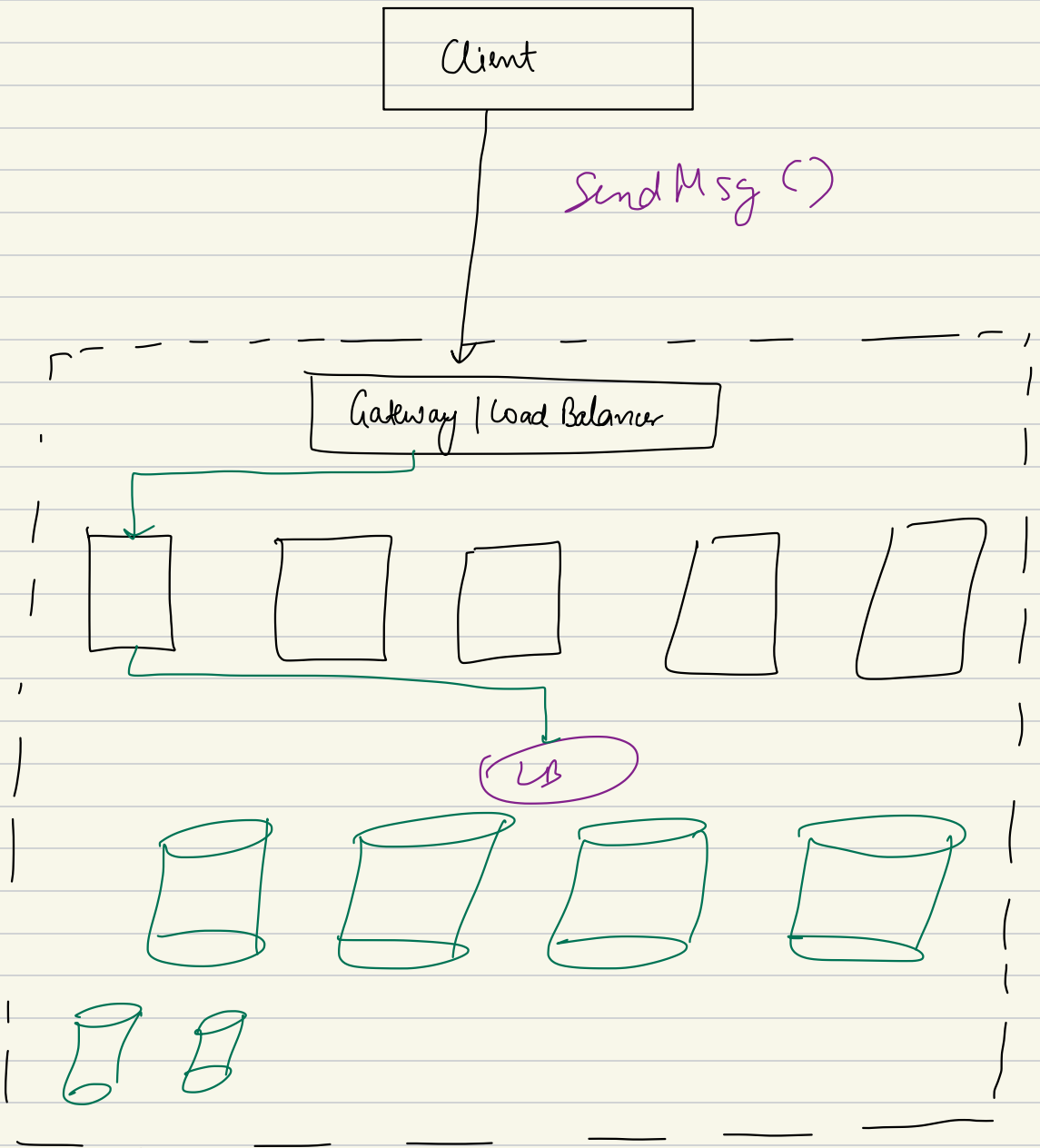




Mohsin



Deepak



# Sharding Key

2070

Mohsin

Deepak

3010

Aakash

RiHk

4079

Akshay

Ravi

9090

[  $M_1$   $M_2$   $M_3$

$M_{10}$  ]



## user collection

usr

107

name

Akshay

metadata of user

## Conn. colle

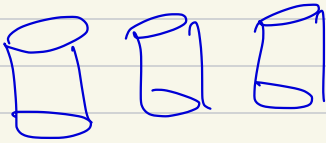
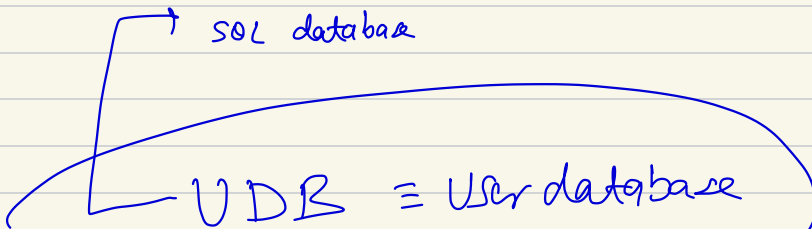
Connid.

1:1 / Group

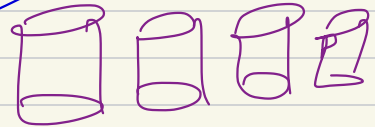
user ids of members

# messages collection

msg id	msg text	convid.	meta	Timestamp	sender id
A709		10709			



Sharding key  $\equiv$  user-id

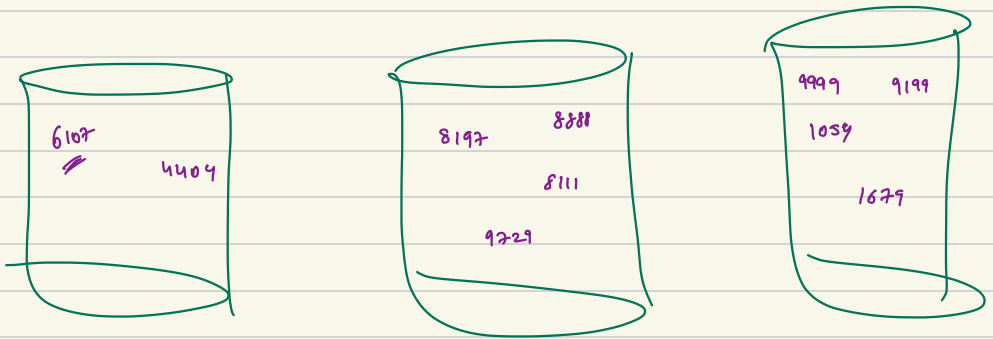


Msg Database

Msg DB

sharding key  $\rightarrow$  conversation -id

conv-id  $\rightarrow$  sharding key



conv-id  $\equiv$  6107, 4404, 5309, 9101,  $\rightarrow$

msg-id  $\ni$  A987, B101, 2426

userid  $\ni$  107, 111, 124, 600

Latest Conversations | :

$\text{inter shard} \equiv 2 \#$   
Group:  $\text{inter shard} \equiv N \#$   
②

Send Msg  
(intra)

Client

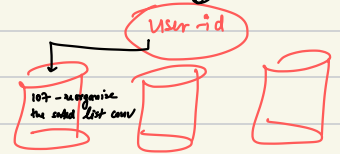
$107 \rightarrow \textcircled{5090}$ , "Text"  
 $\underline{107} \rightarrow \underline{9099}$ , —

Gateway

[members of  
conv-id]

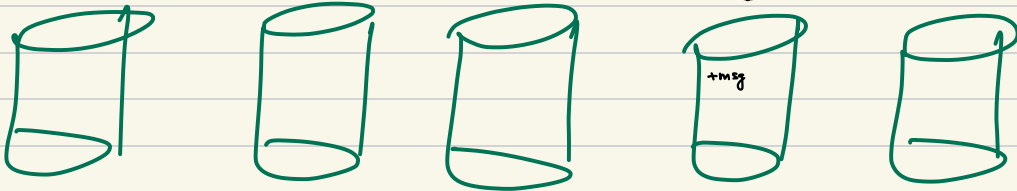


VDB



Latest Conversations

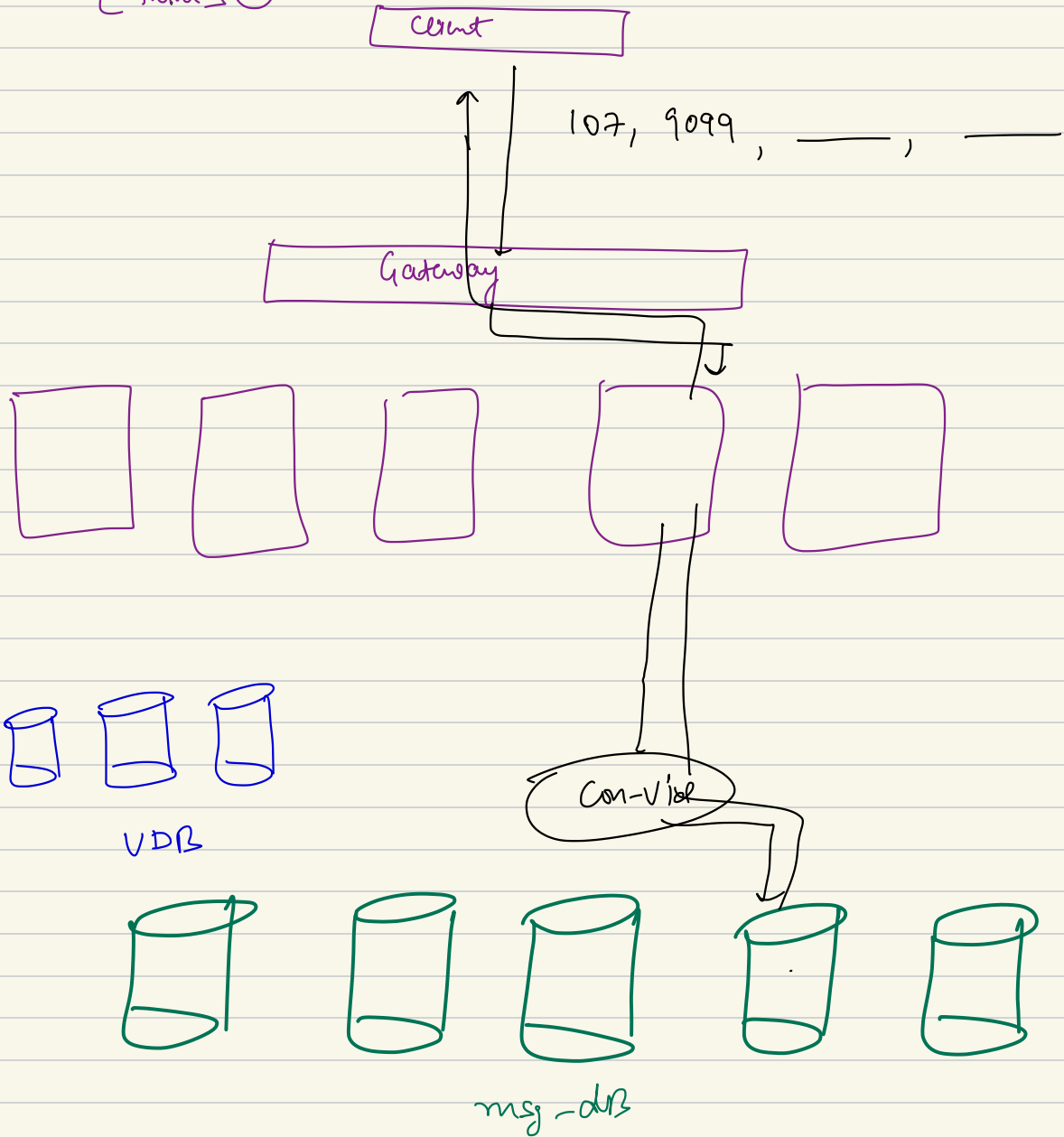
conv-id



msg-dbs

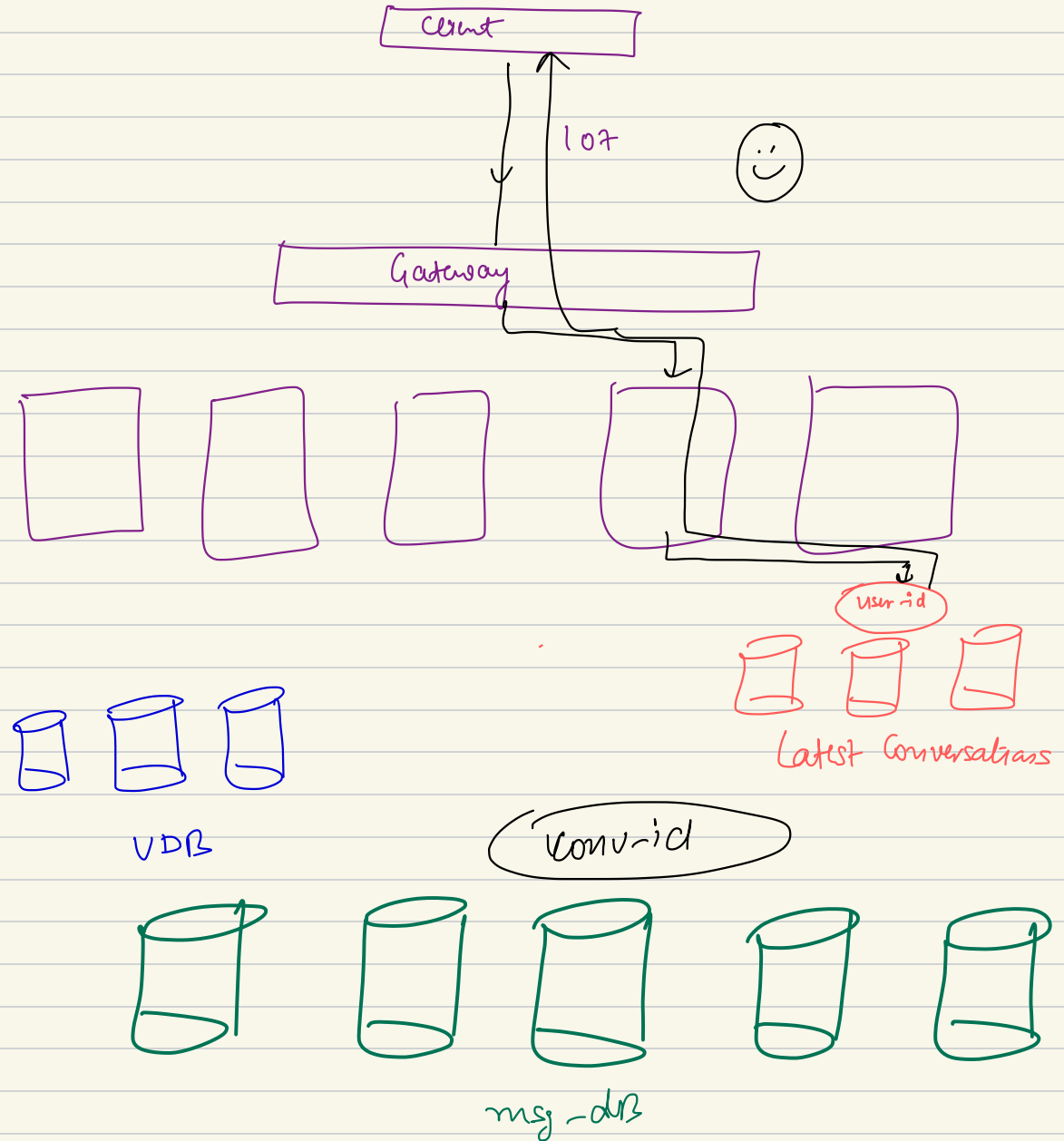
getMsg ( )

[ intra ] 😊





get ConversationList() → can't be handled in present solution as it will be a all shard query.



User is connected

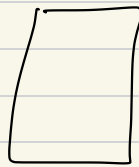
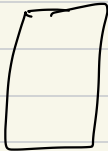
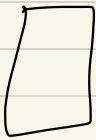
C1 → C2

Client 1

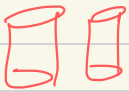
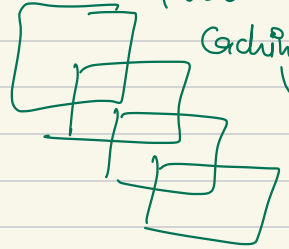
Client 2

Gateway / Load Balancer

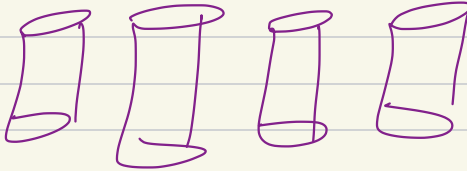
Stateful



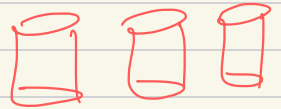
Global Caching



UDB

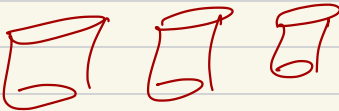
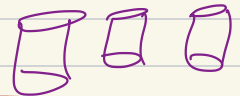


msgs db



Recent Conv DB

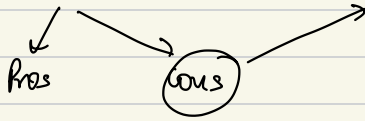
File Storage



Cold Storage / Archival Storage

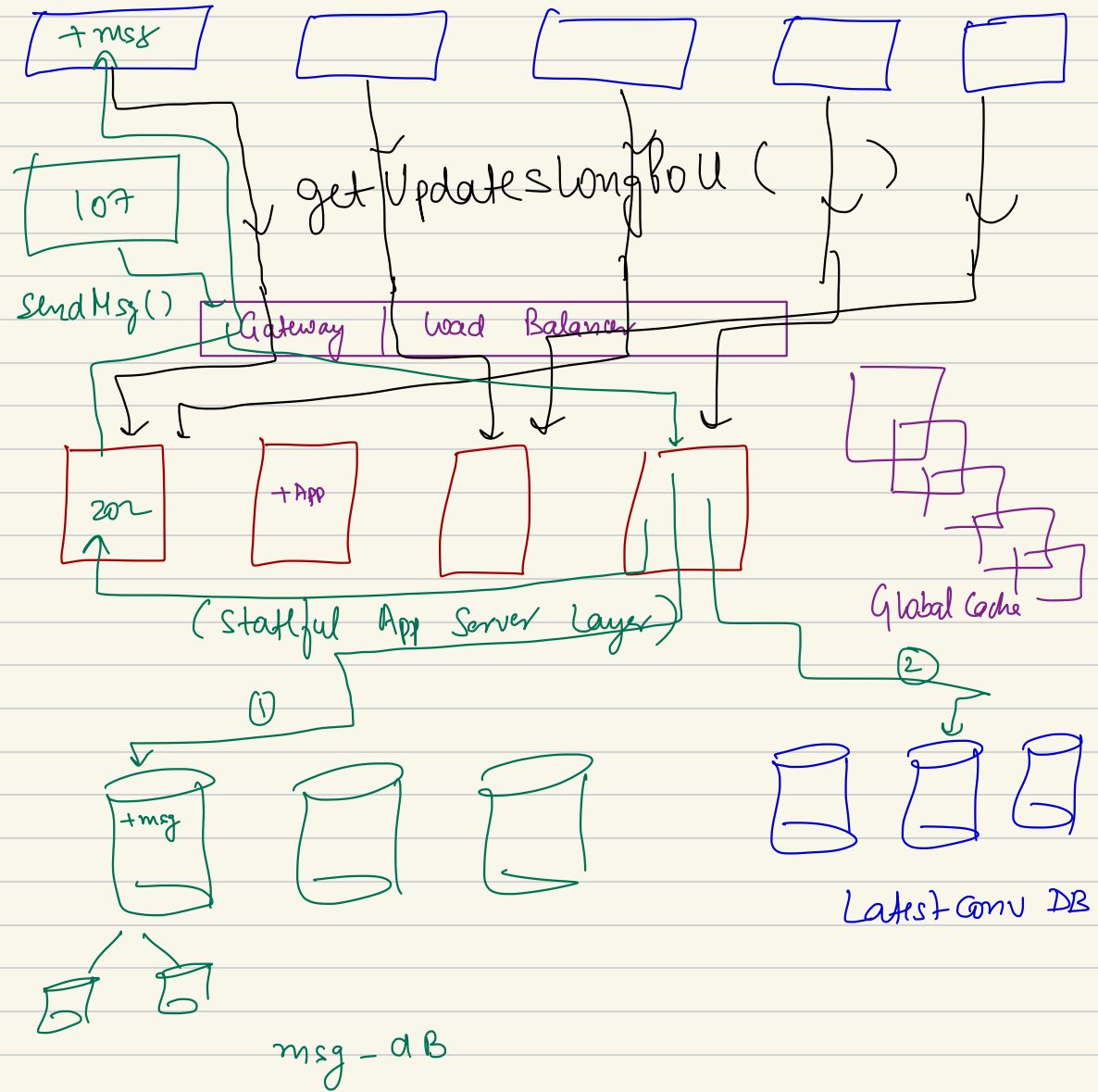
détour

① **web socket** → <sup>high</sup> resource consuming...



② **HTTP Long Polling** ☹️ ☹️ ☹️

1- 100's K



# Summary

