SYSTEM DESIGN EXAMPLE

1. DEFINE MVP
2. ESTIMATE SCALE
3. DESIGN GOALS
4. API DEFN
5. DEEP DIVE

1. Send a message ✓
   1:1 conversation
2. User sign up ✓
3. Group conversation
4. Broadcast
5. Fetch conversations & ✓
   messages in my mailbox
6. Read ✓ / unread, $\boxed{\text{sent / delivered}}$ ✓
7. Status of user: Online / offline.
8. Realtime ← latency should be low.
9. Notification about new message.

Scale:

50M
$\boxed{5B}$ messages
↓
5B * 500
$\boxed{= 2.5 \text{ TB}}$

NO. 1000s of machine to store.

1. Do all msgs. fit in one machine

2. Can all queries be handled by one machine.

NO

2.5 * 365 * 10

9000TB = $\boxed{9PB}$

QPS: 5B *

$$\frac{5 * 10^9 \times 3}{86400}$$

$$\frac{1}{6} * 10^6 \qquad \frac{15000 * 10^6}{86400}$$

---

1. CP ← $\boxed{\text{Consistent}}$
2. Latency low ← realtime

Send button → delivered to

Send button; recipient

---

① User signup

② send Message

— sendMessage (userId, recipientId, text
timestamp, messageId)

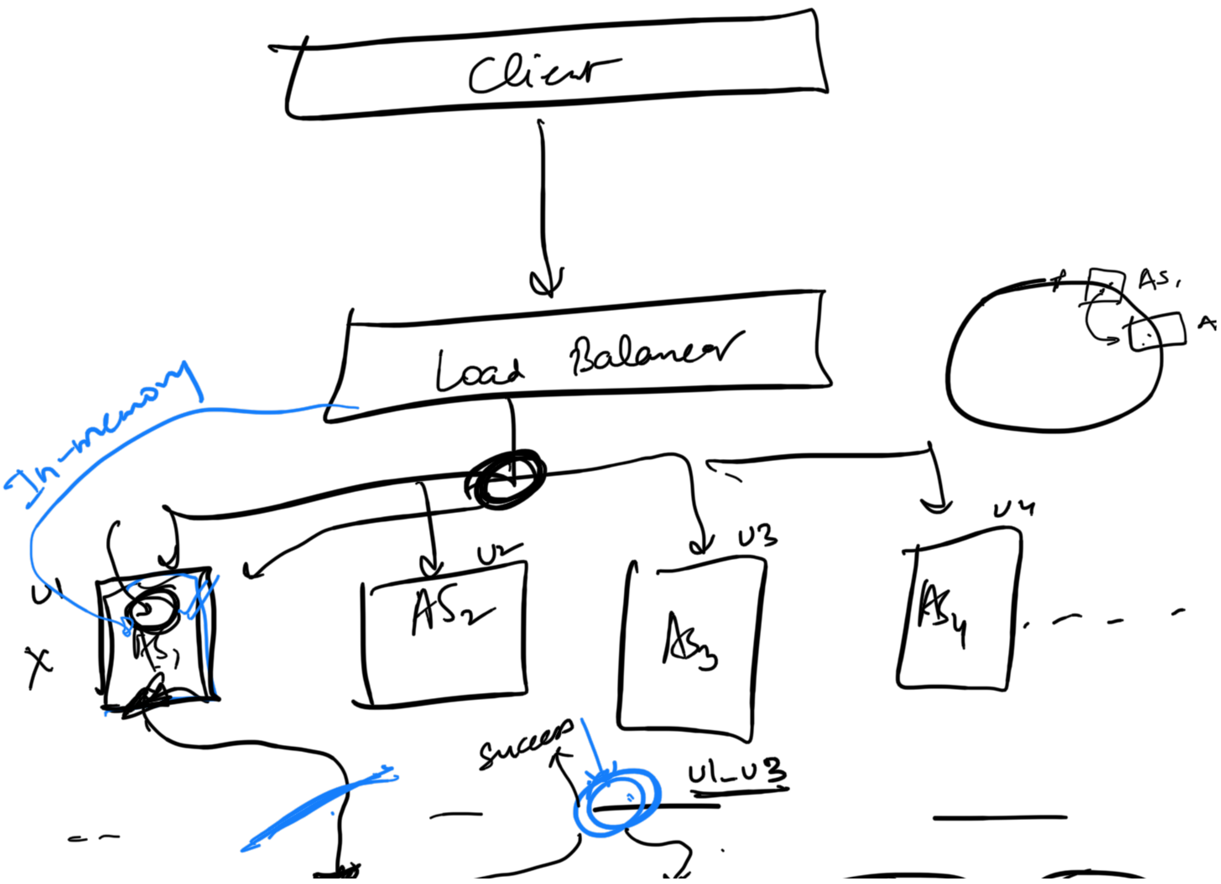
③ Fetching conversation

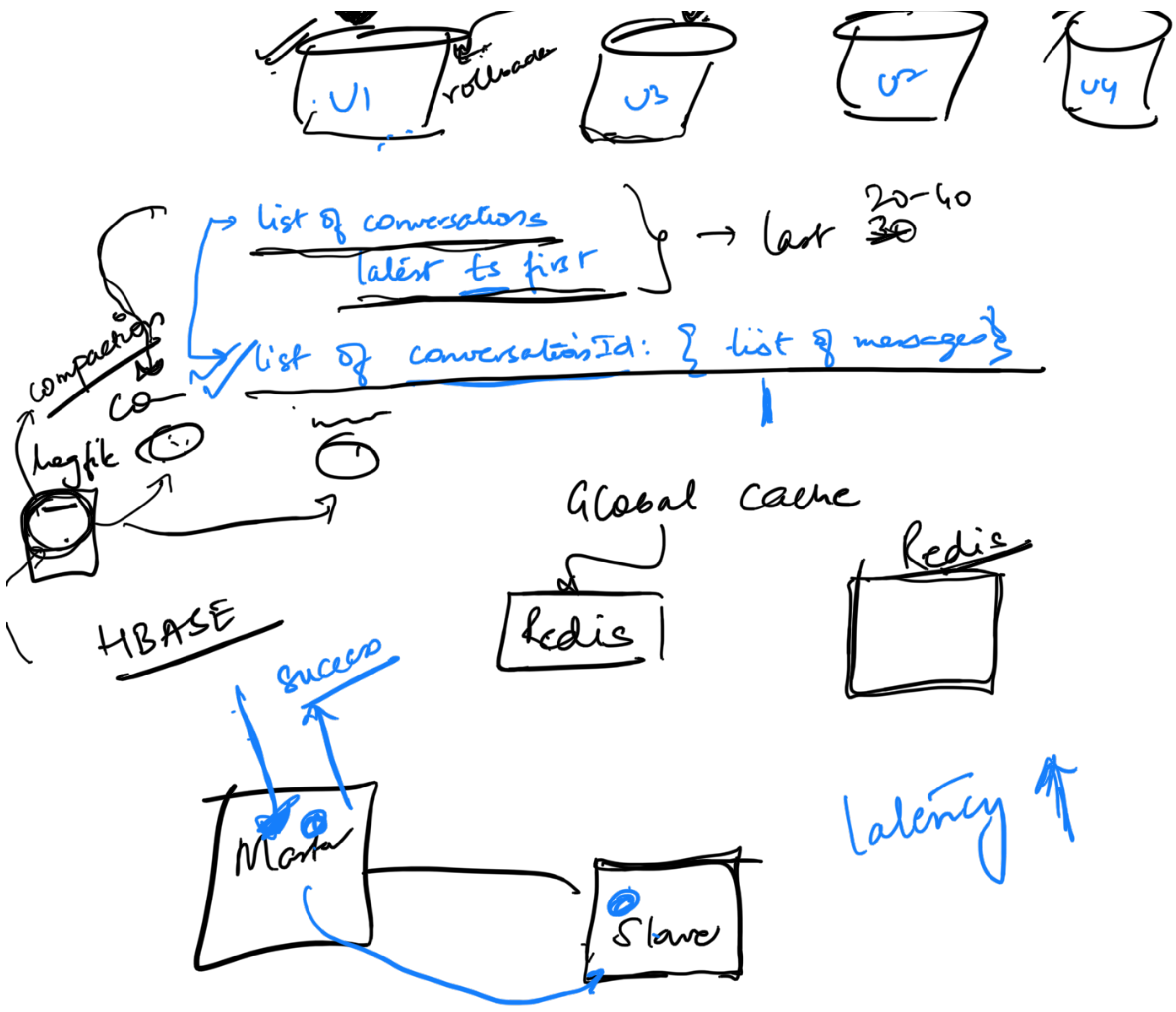
— fetchConversations (userId , numConversat
offset , delta , timestamp)

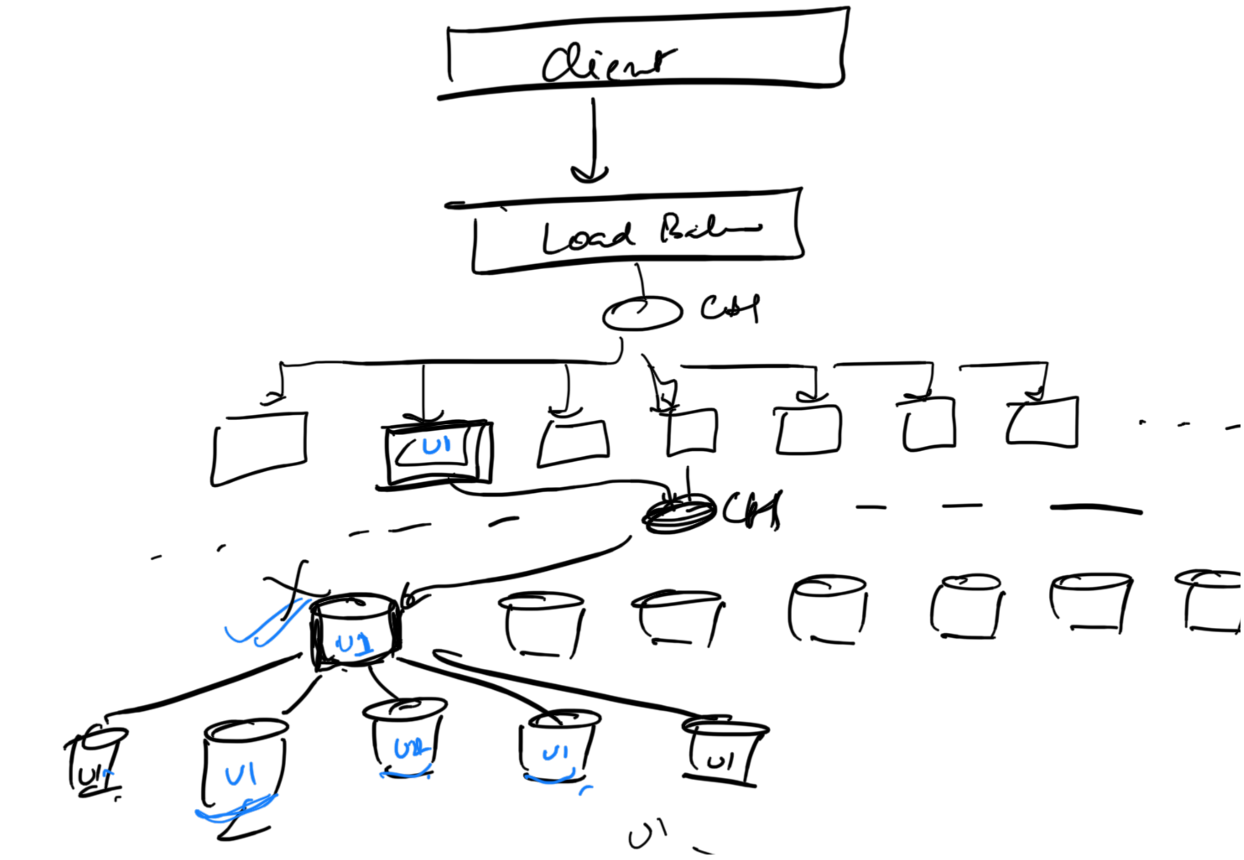
— fetchMessages (userId , conversationId ,
numMessages , offset , delta , time

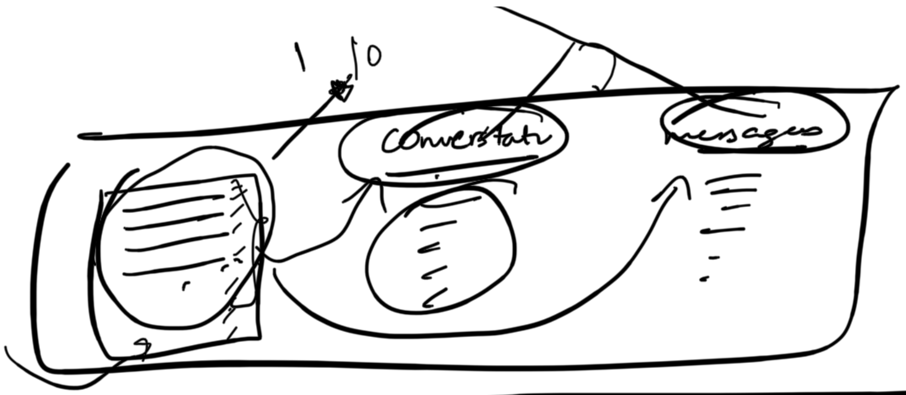
④ Read / Unread , sent / delive

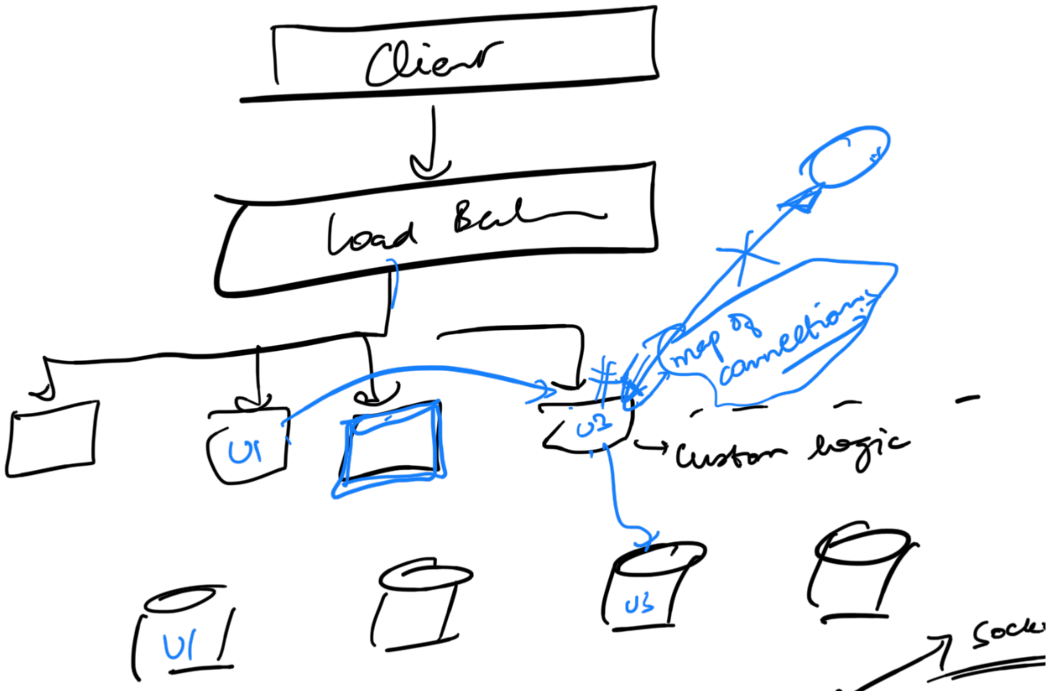
⑤ notify User




U1
rollback
U3
U2
U4
list of conversations
latest ts first
→ last ~~30~~ 20-40
list of conversationId: { list of messages }
compaction
CO
logfile
Global cache
Redis
Redis
HBASE
success
Master
Slave
latency ↑

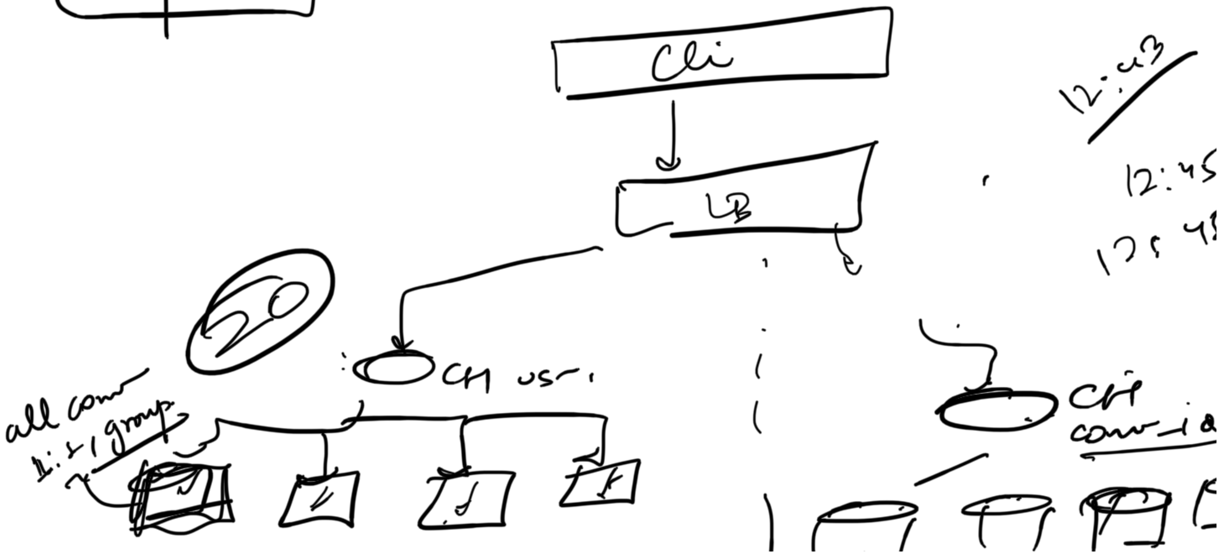

Client
Load Bal
CH
U1
CH
U1
U1
U1
U2
U1
U1
U1

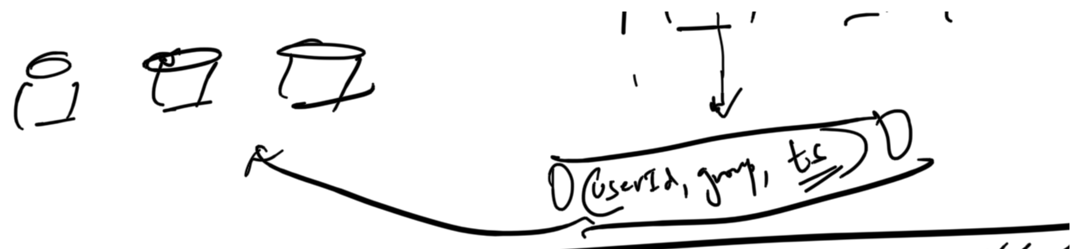


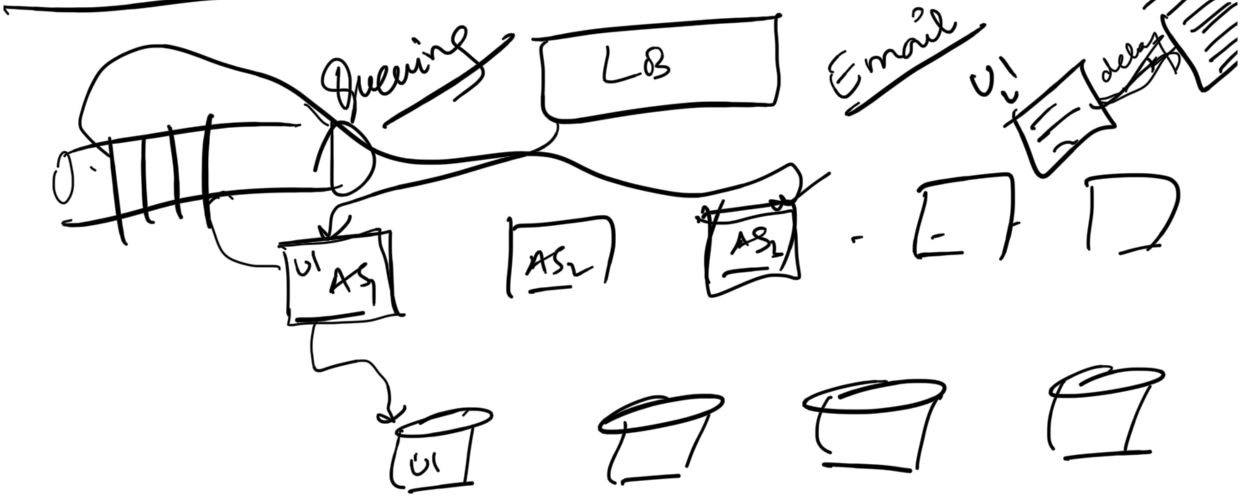


User_device

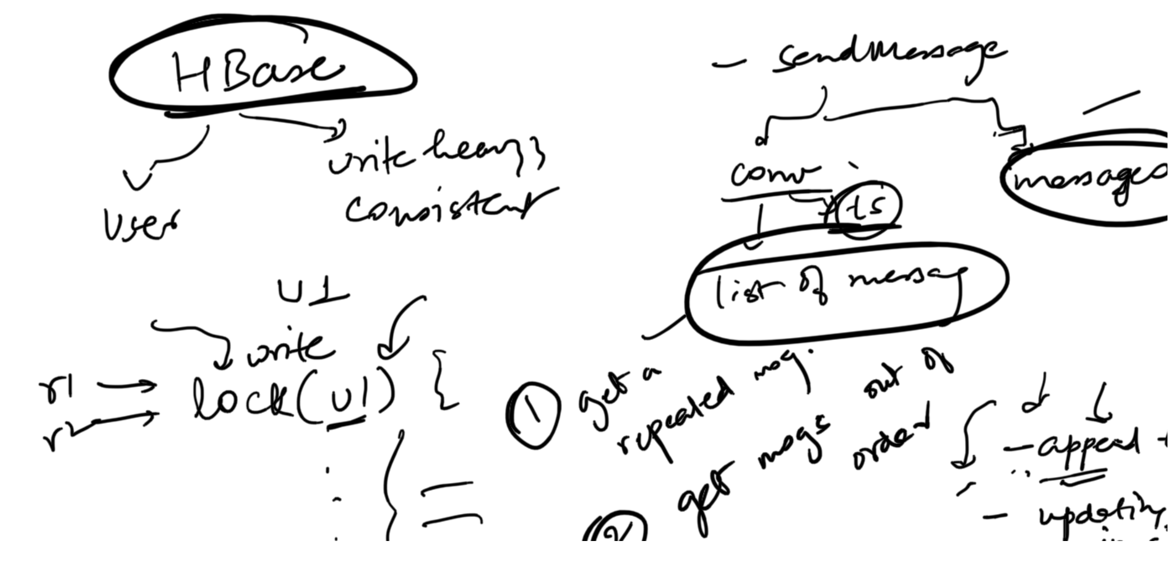
User	ipad
	iphone
	web
	⋮

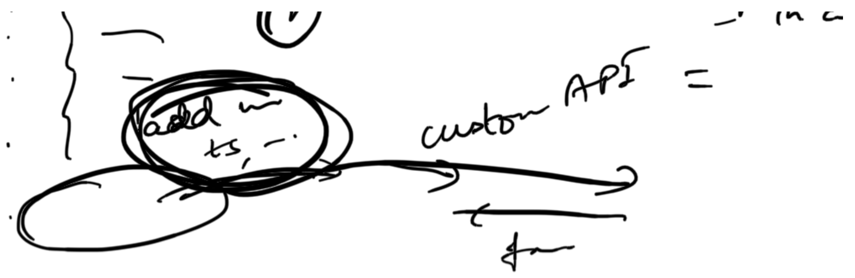






- Caching in consistent fashion for a lot of data: AS stateful
- Write heavy system: Write to a log file.
  ↳ Quorum
- Sharding by $\boxed{\text{user}}$ getConversations.
- ~~Not~~ Notif system → need a map of all user devices + policy engine




in a
add
ts,
custom API =
fun

}