

WhatsApp System design :-

Requirement gathering :

- Functional :
- ① 1-1 send and receive Text
 - ② Group msg. support
 - ③ Last seen/online
 - ④ User Login

- Non-functional
- ① Scalability
↳ High traffic
 - ② Low-Latency
 - ③ Availability

Back of envelope :

$$\text{Total users} = 2B$$

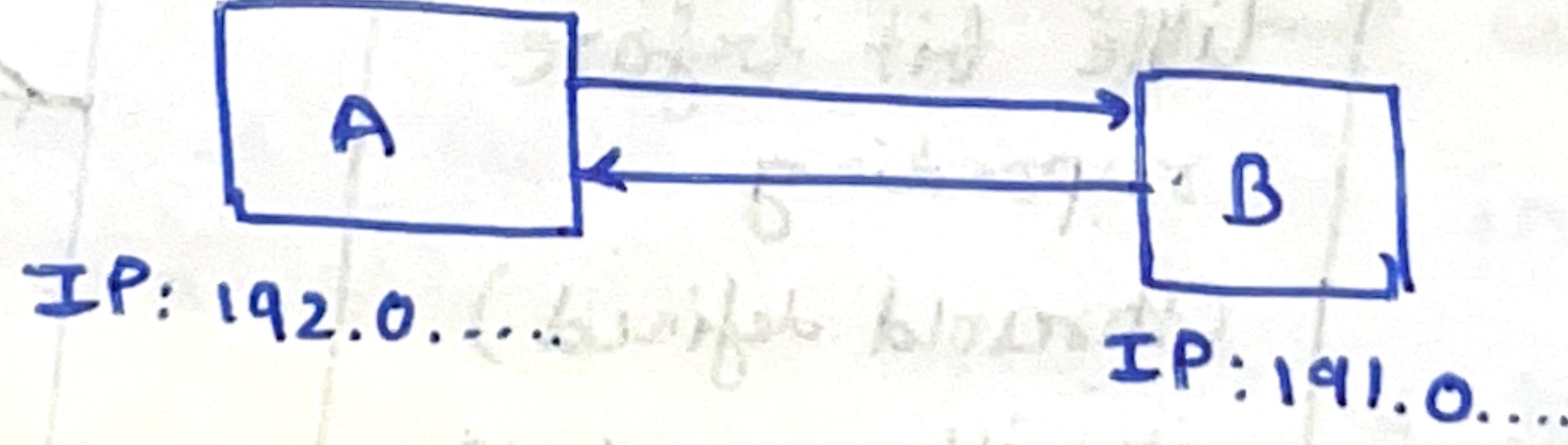
$$\text{DAU} = 50 \text{ M users}$$

$$\text{avg. user} \rightarrow 40 \text{ msg/day} = 2B \text{ msg/day}$$

$$\text{avg. } 1000 \text{ byte per msg} = 2 \times 1000 \text{ B.} - 2 \text{ TB/day}$$

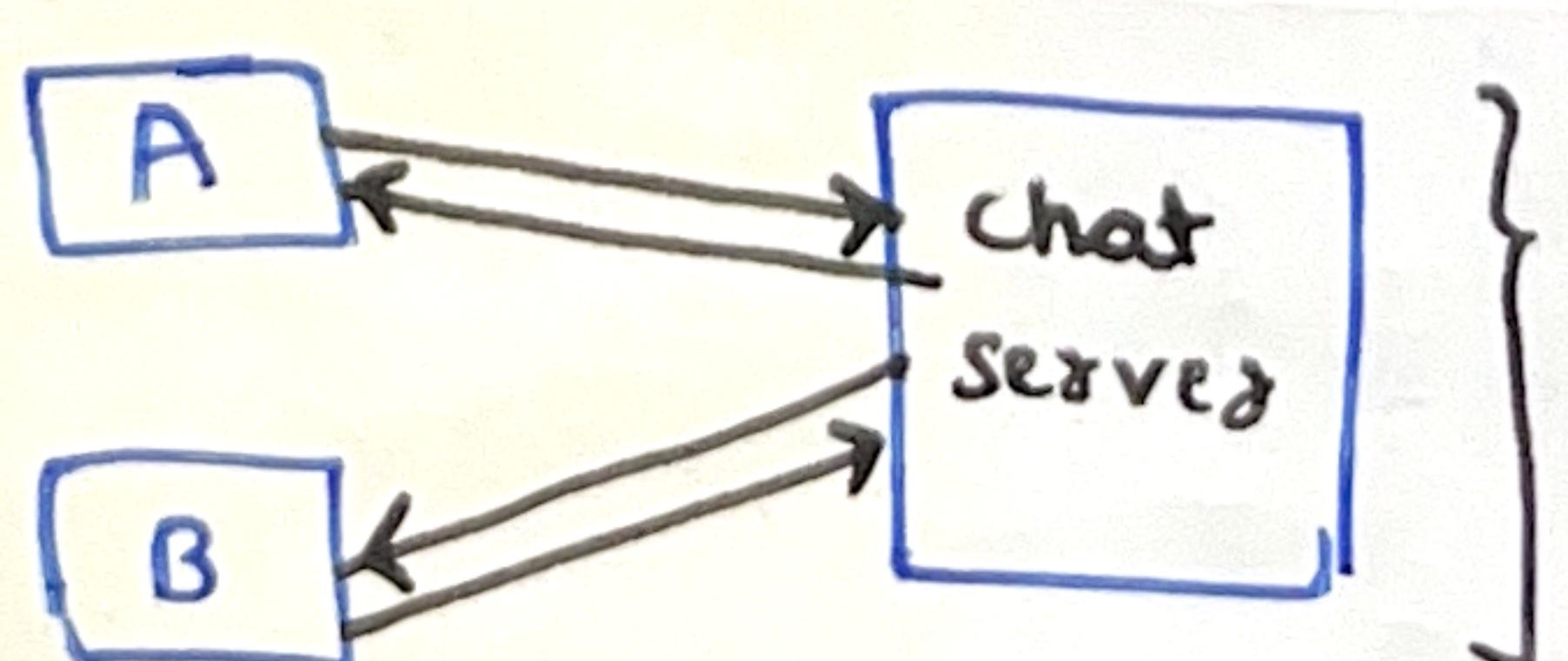
$$10 \text{ years} = 20 \text{ TB} \times 365 \times 10 =$$

design :



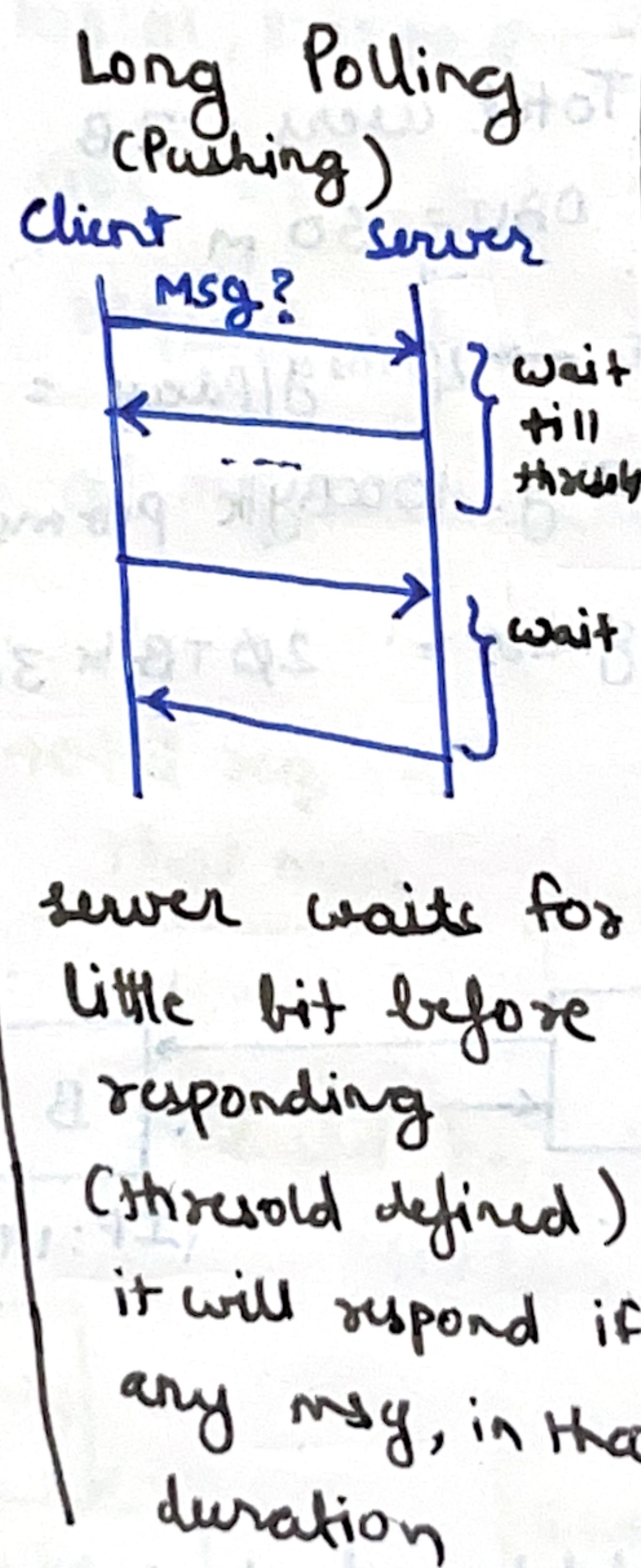
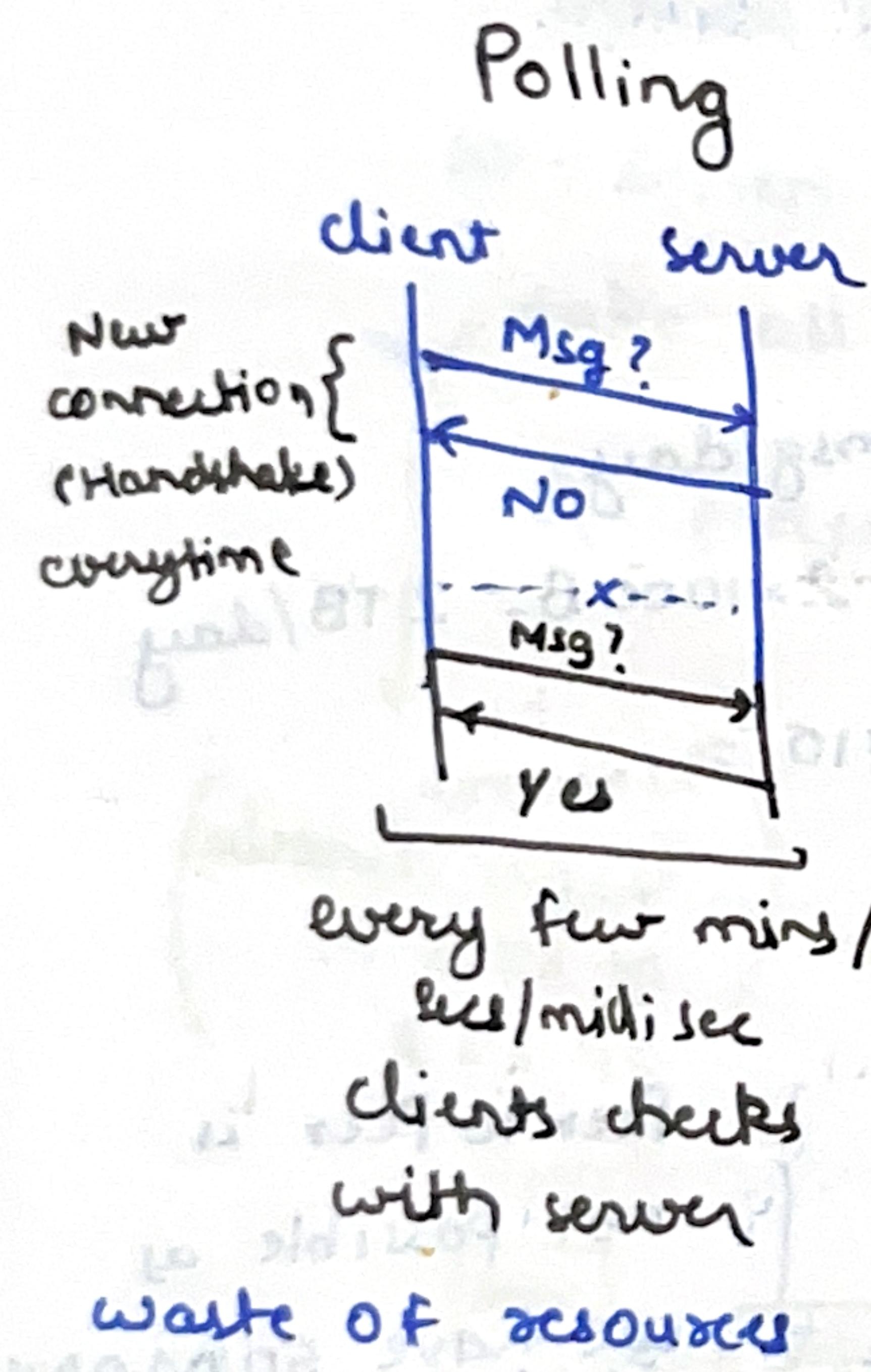
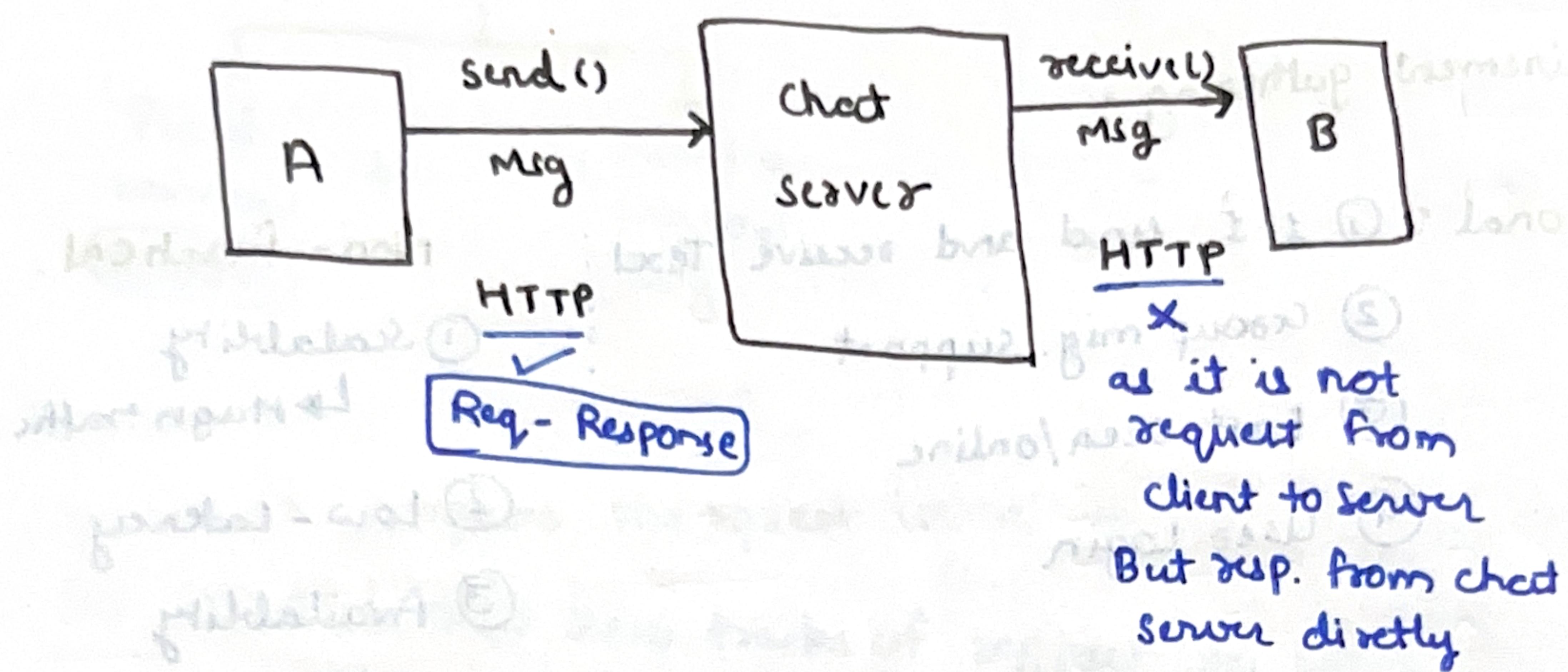
} Peer to Peer is not possible as we have 50M DAU and we need scalability.

SO there would be chat server.

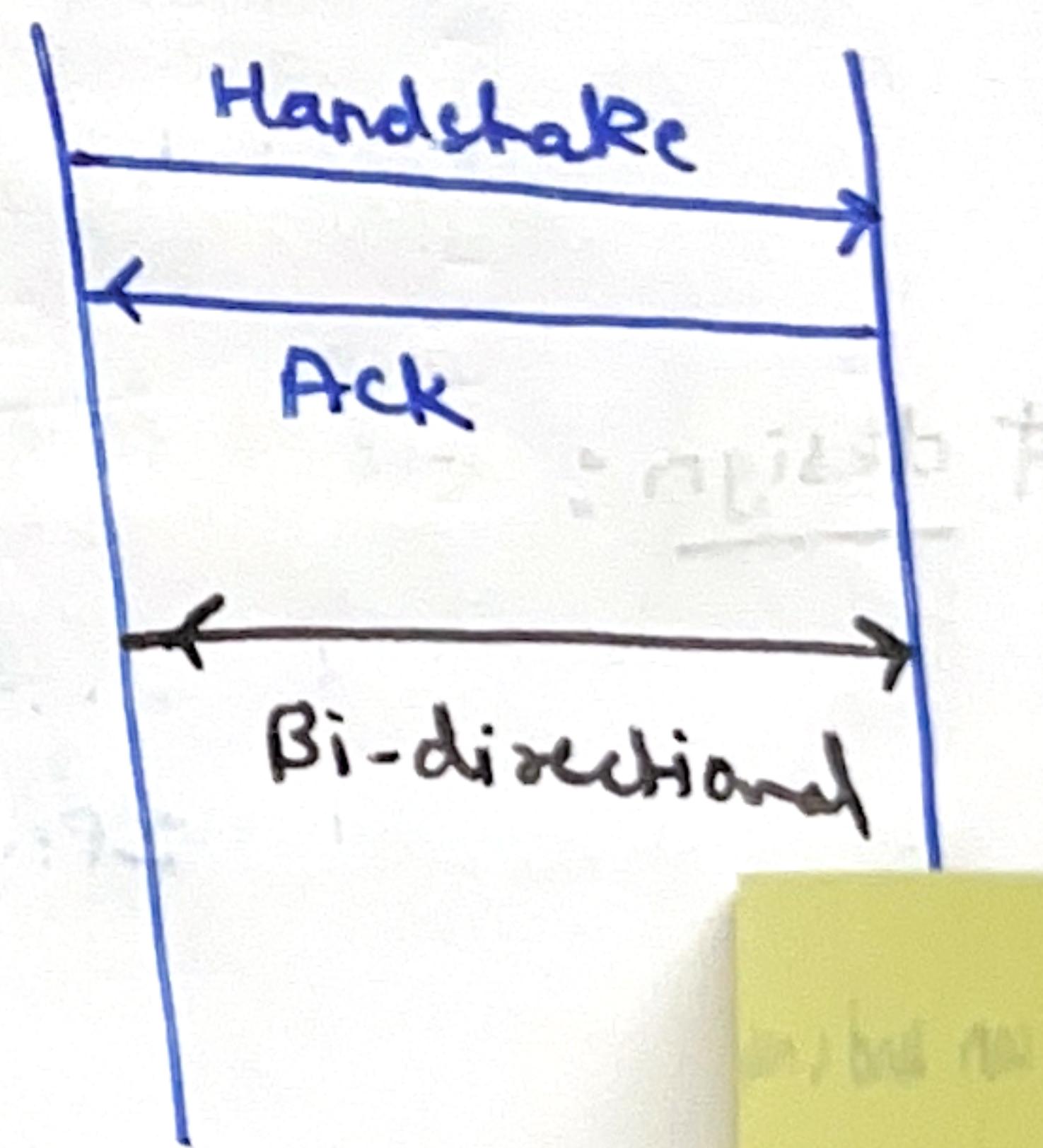


client - server architecture

Protocols to use:



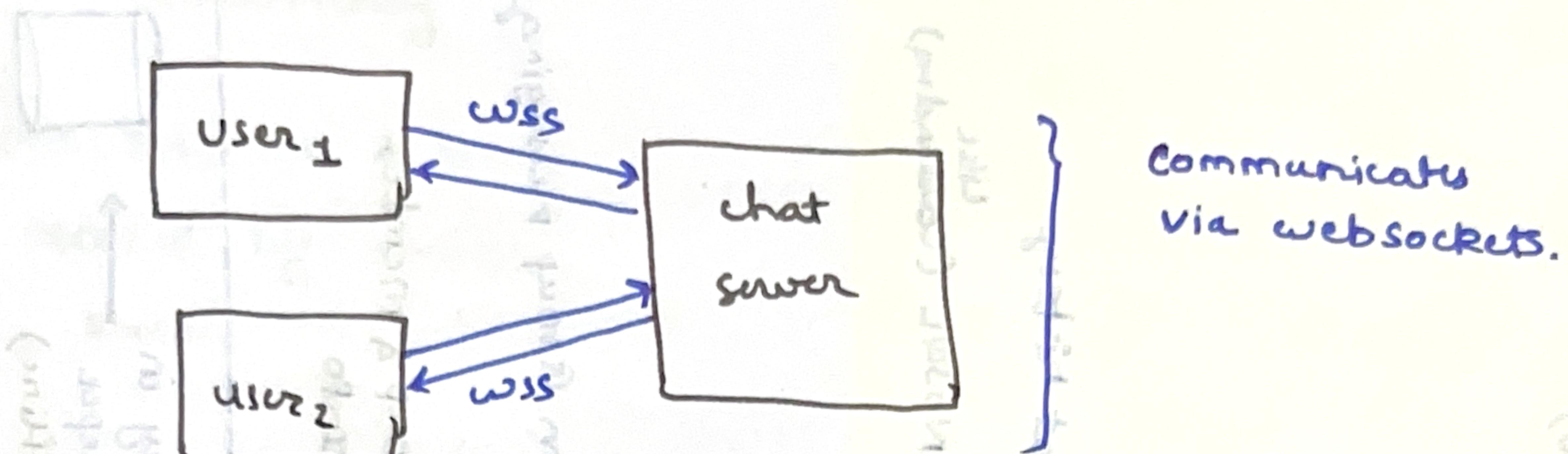
: web socket
Bidirectional
Persistent connection



client can send a message
and server can also send
message, Bi-directional
comm.,

⇒ connection will remain
open till one of them
shuts down the connection
or Internet Breaks.

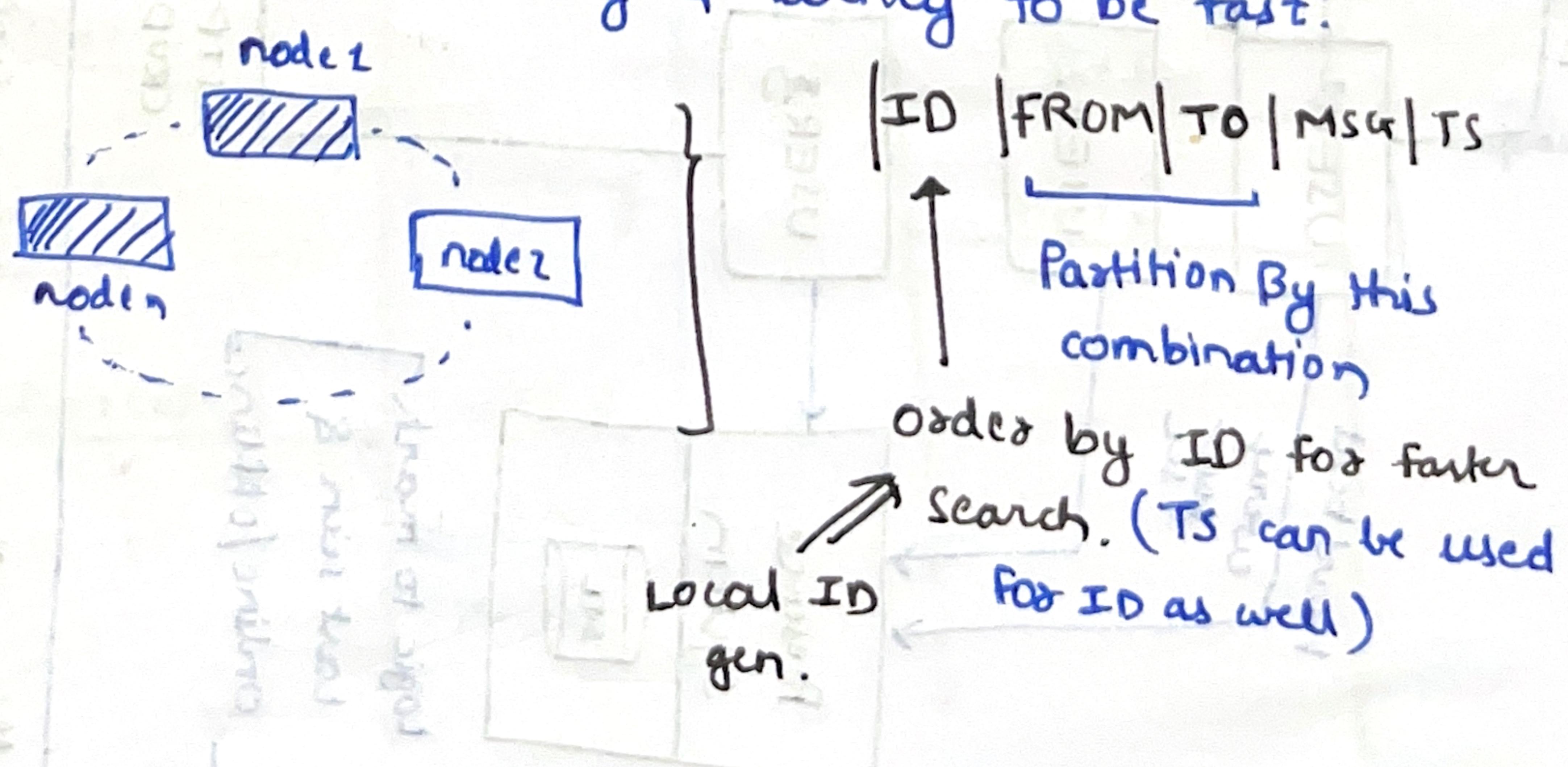
communication using websockets:

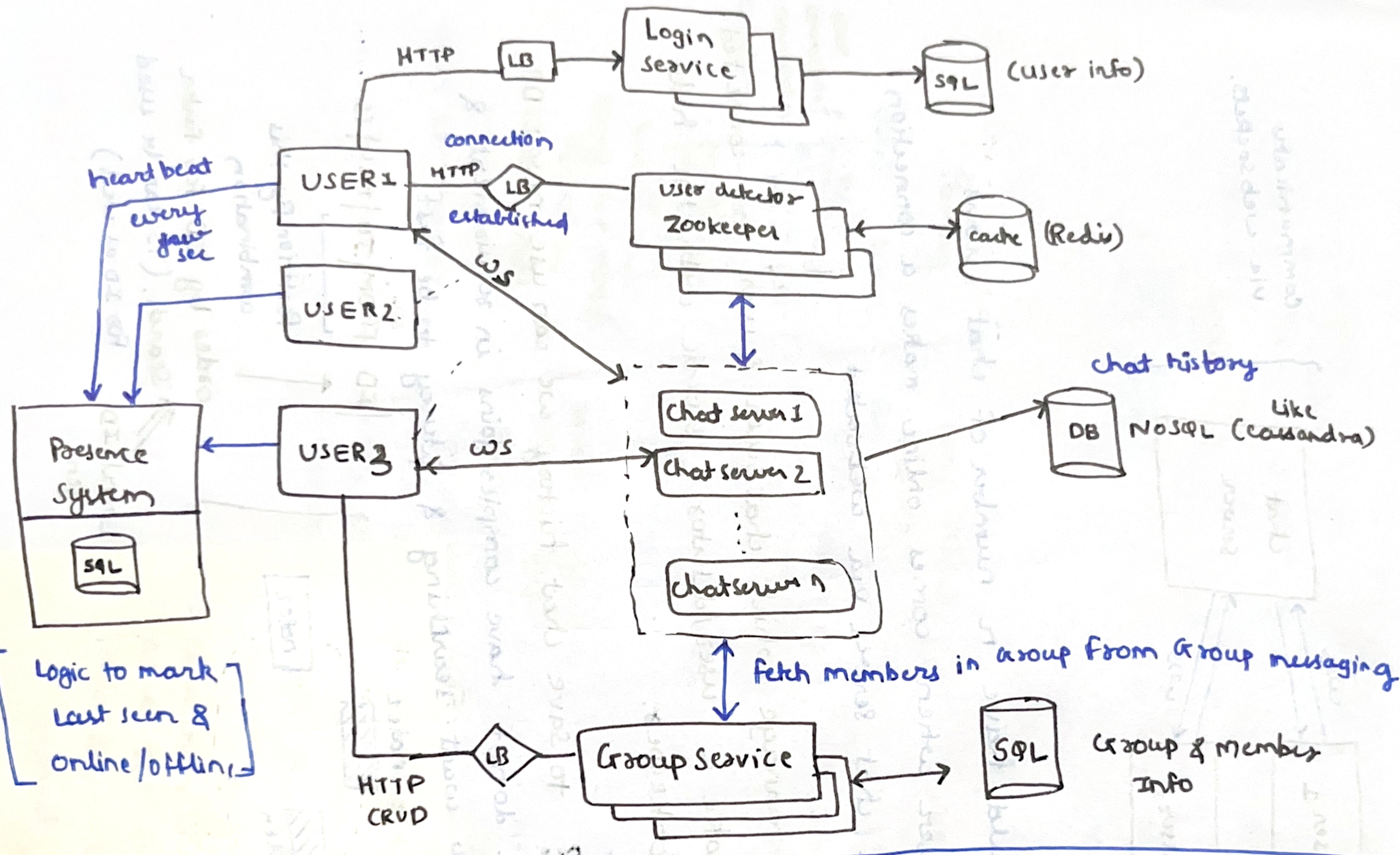


- we would have n number of chat servers
- each user when comes online makes a connection to one chat server via websocket

How do we manage which chat server & user are connected?
 ⇒ we make a user/detector service with the help of Zookeeper.

Chat History? to save chat history we can use nosql DB as we do not have complex joins in requirement & we want searching & latency to be fast.





Handling when one server is offline
& can't receive msg:

USER1 → **send()**
user2

No entry in
zookeeper
(User offline)

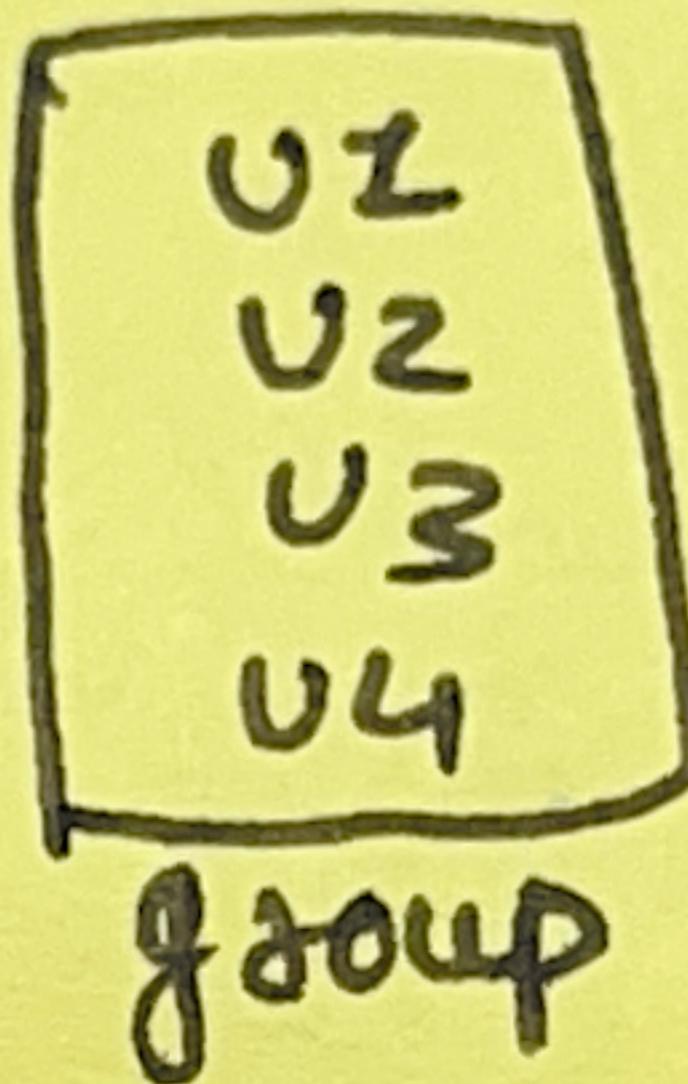
entry in
DB

USER2 → **connection**
ctd.

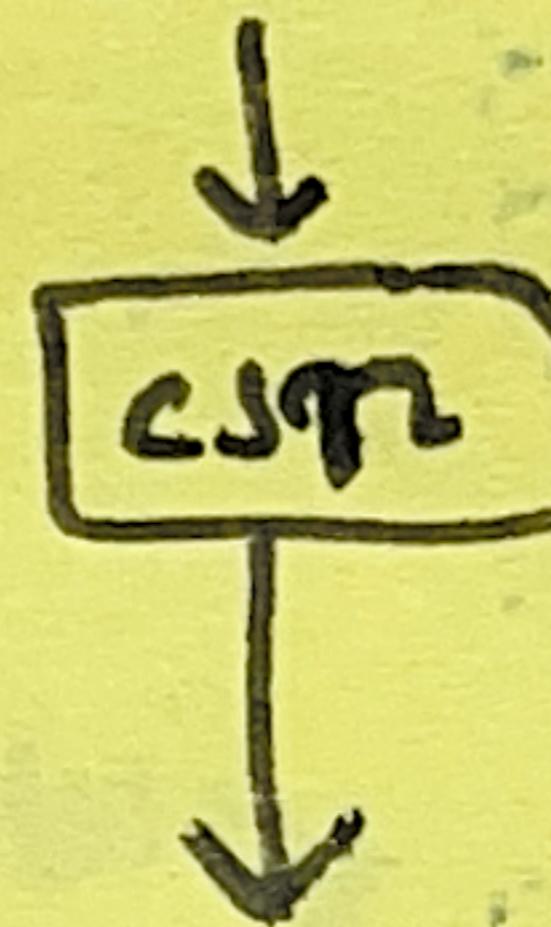
→ **CSi** → **read DB** → **send pending**
msg.

in case of group msg.

U1 →



groupId = 501



Group serve
to fetch
group user
Id's

Zookeeper
to find U of each user

send
msg to
those