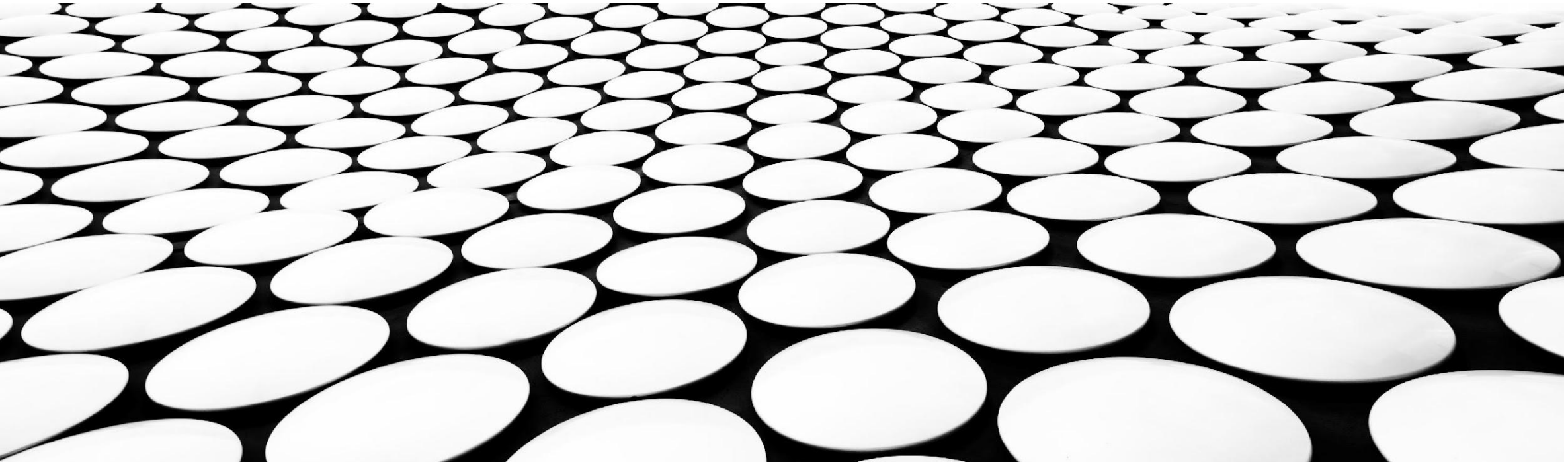
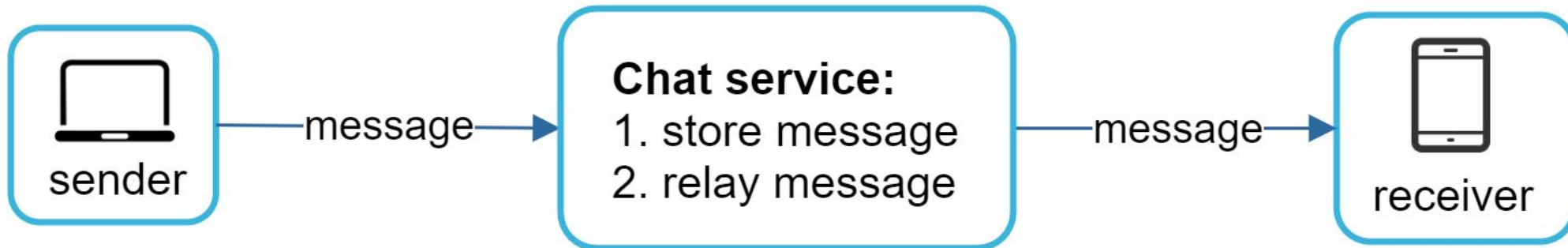

BC05 - REAL PROBLEMS ANALYSIS

MVP



DISEÑO DEL SISTEMA DE LA APLICACIÓN DE CHAT





REQUERIMIENTOS FUNCIONALES

- One-one chat
- Group chat
- Read receipt
- Online status
- Push notifications
- share multimedia
- Multi device support



REQUERIMIENTOS NO FUNCIONALES

- Low latency
- Highly available
- Highly scalable

ESTIMACIONES DE USO:

- Total active users: 500M
- On average a user sends 30 messages per day.
- Total messages per day= $500M \times 30 = 1500M = 1.5B$
- msgs per day= $1.5B / 3600 \times 24 = 18k$ msgs per sec

ESTIMACIONES DE ALMACENAMIENTO

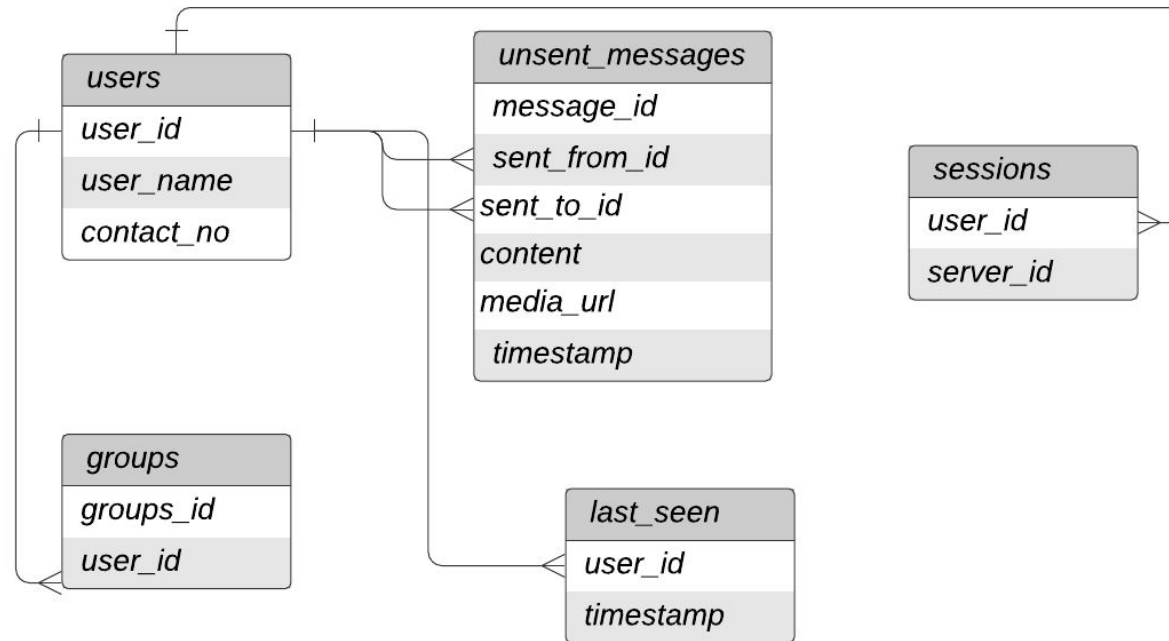
- Total messages per day= 1.5B
- considering each message is on an average of 50 KB. Total storage required to store all messages= $1.5B * 50kb = 75pb$
- Messages are not going to be stored. Let's say 1: 10 of the above data is for undelivered messages. And we are going to store undelivered messages only for 30 days.
- Storage required for undelivered messages for one day= $75pb/10 = 7.5pb$
- Messages in 30 days= $7.5 PB * 30 = 225PB$



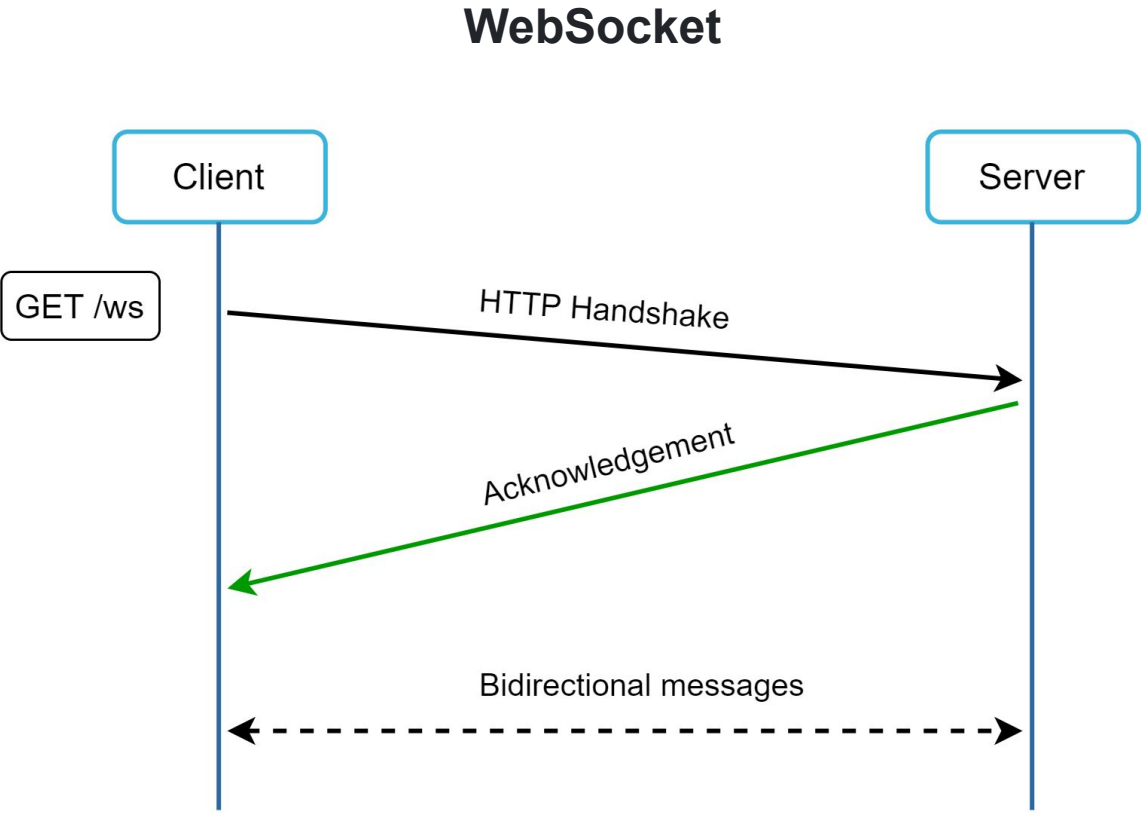
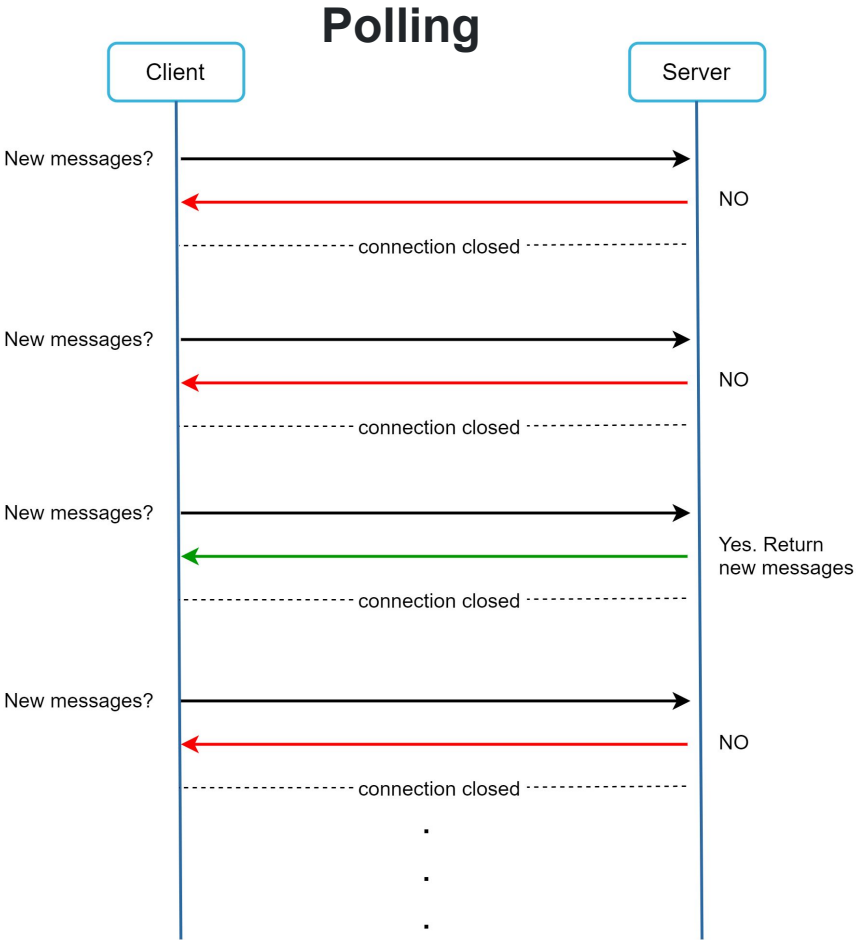
API BÁSICOS

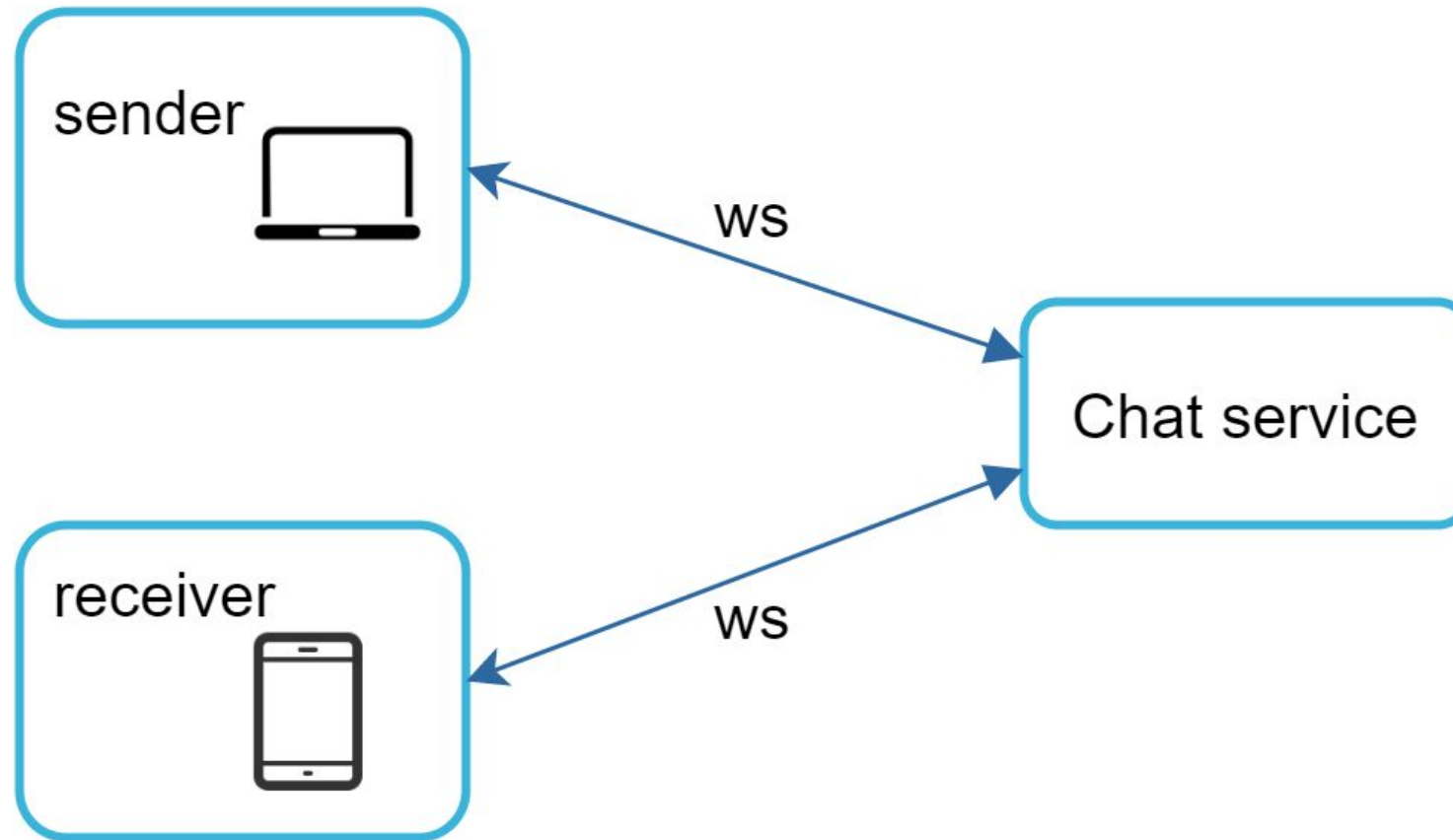
- `Send_Message(sender_userID, reciever_userID, text)`
- `Get_Messages(user_Id, screen_size, before_timestamp)`

ESQUEMA DE BASE DE DATOS RELACIONAL



CLIENTE / SERVIDOR





DISEÑO DE ALTO NIVEL

