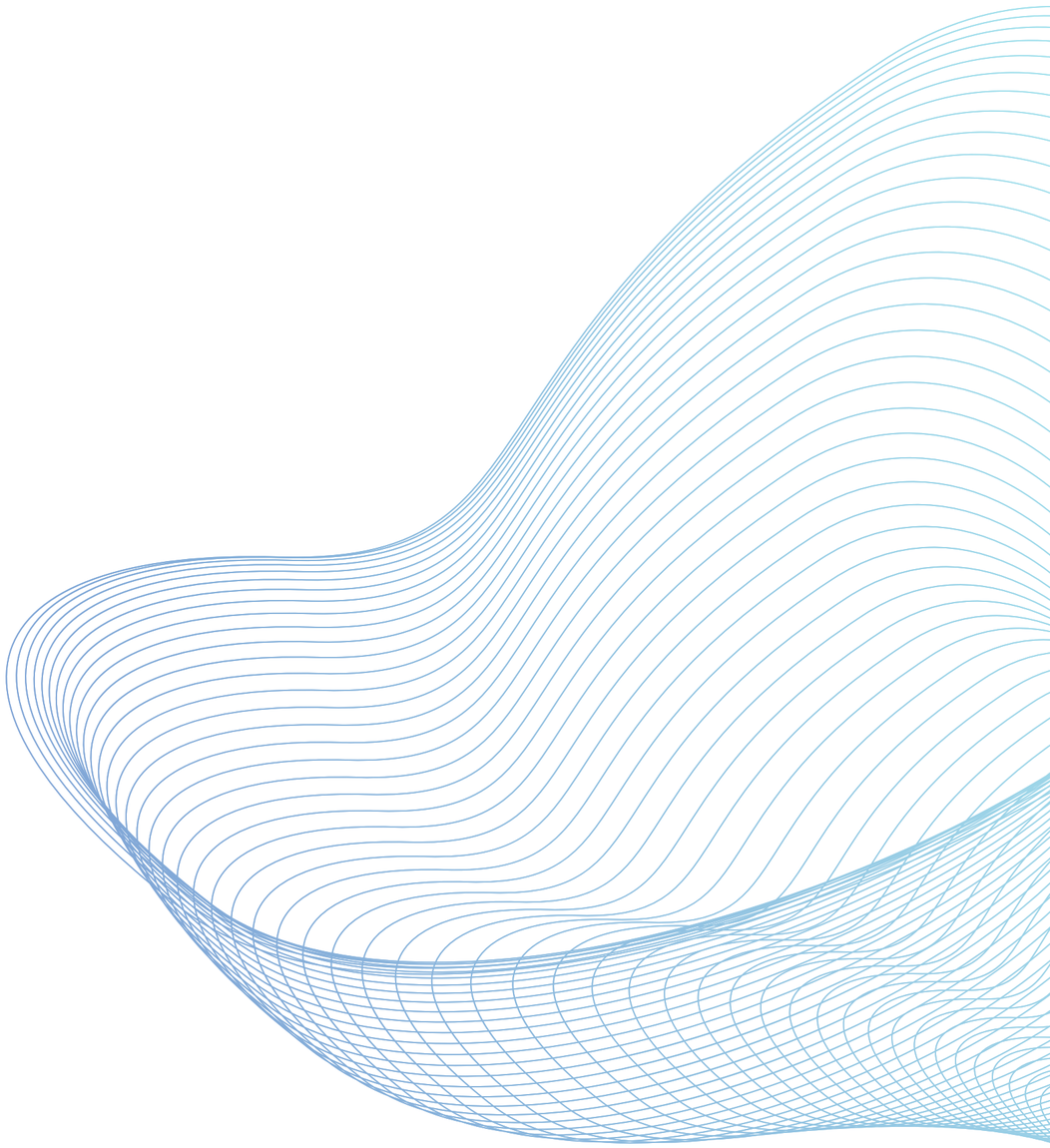
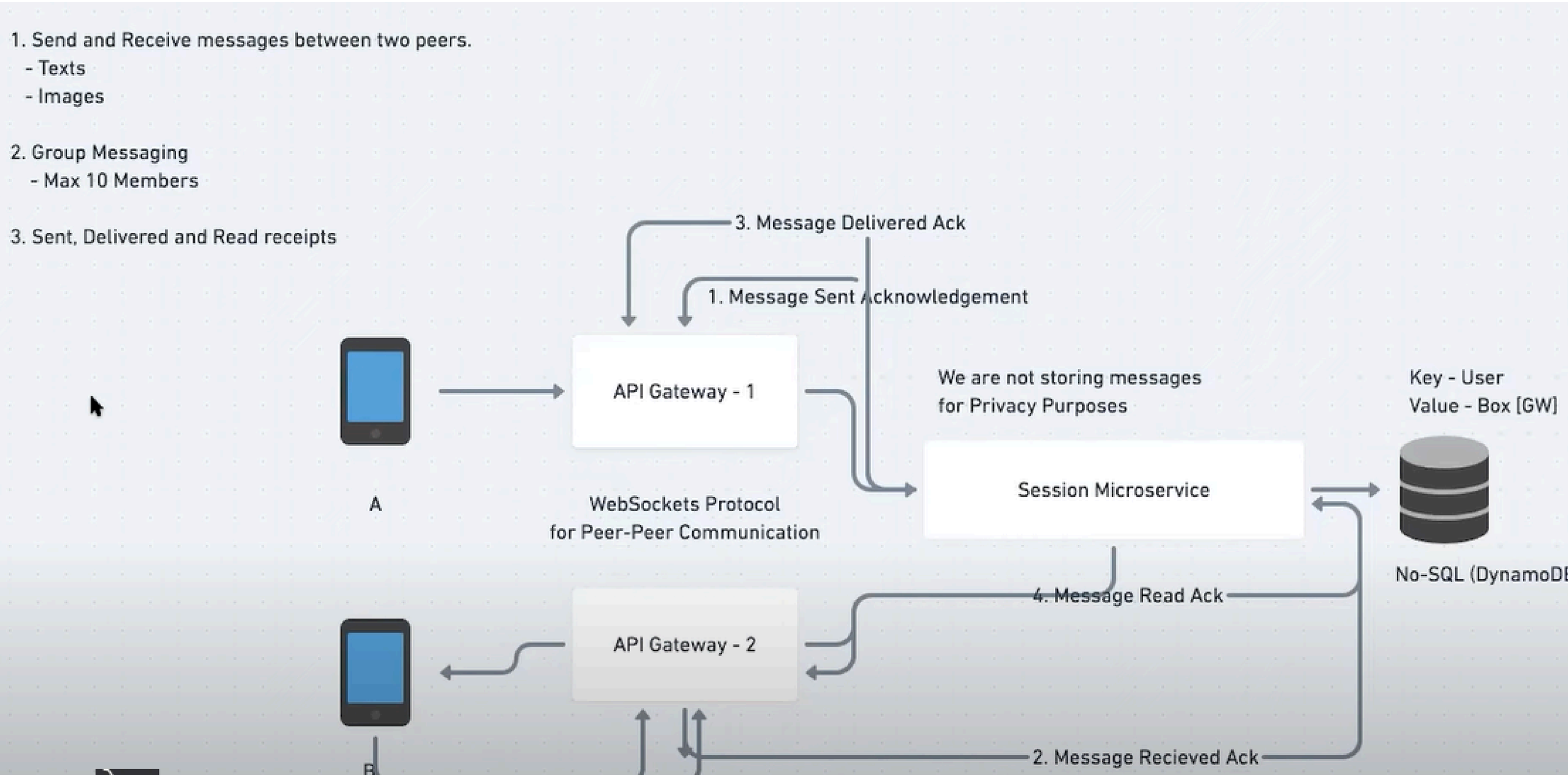
SYSTEM DESIGN



1. Send and Receive messages between two peers.
   - Texts
   - Images

2. Group Messaging
   - Max 10 Members

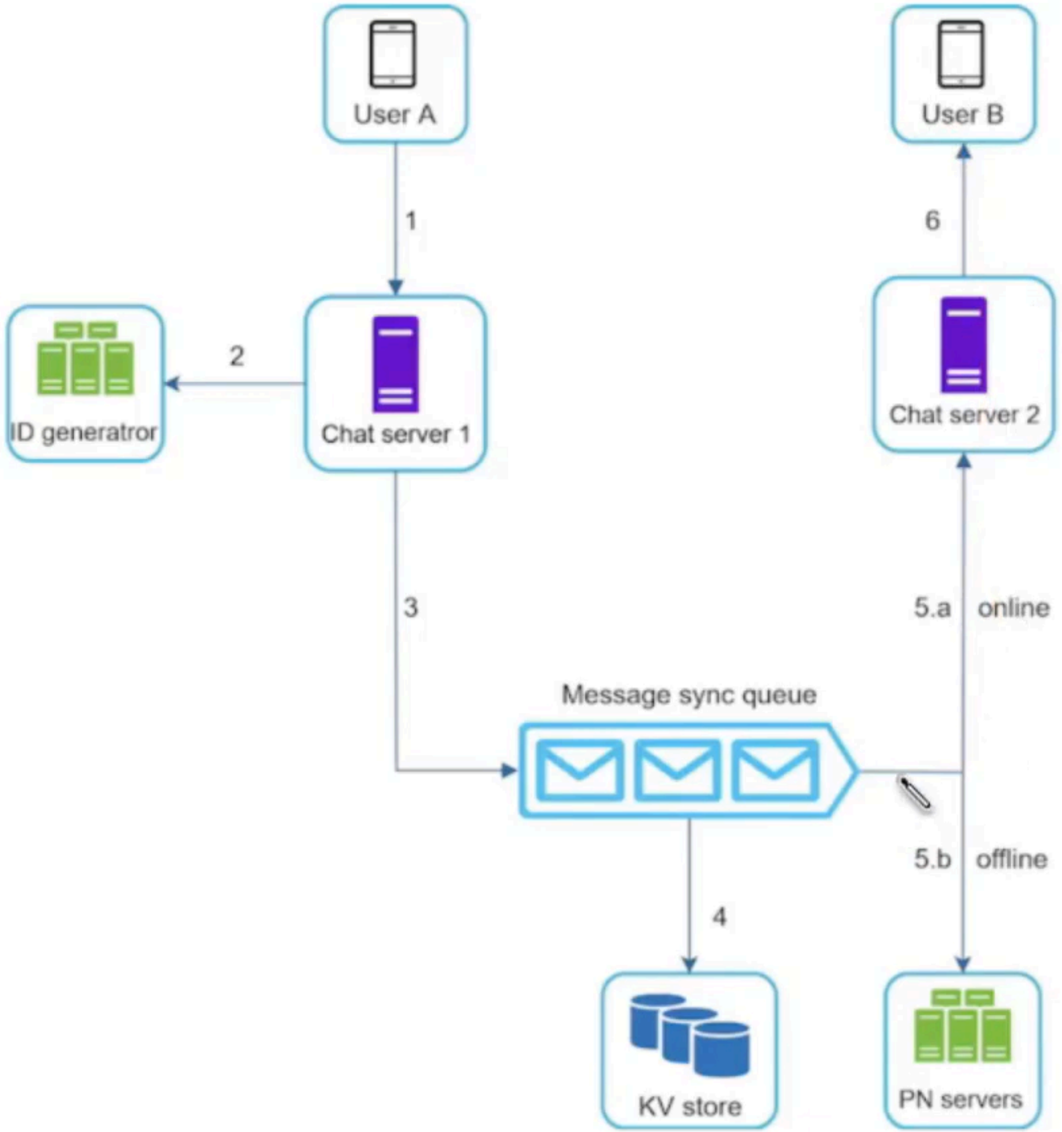
3. Sent, Delivered and Read receipts



Communication Flow

- Sent, delivered, read receipt
- Last seen
- Online status
- Typing status

1.00



SYSTEM OVERVIEW

The system allows users to send and receive messages in real time. It supports one-on-one chats and group chats, with features like message history, notifications, and user management.

COMPONENTS

FRONTEND

Built using Next JS . It includes chat windows and message input boxes,

WebSocket Client: For real-time communication. Libraries like Socket.IO or native WebSockets can be used.

COMPONENTS

BACKEND

WebSocket Server: Handles real-time messaging between clients.

Application Server: Manages user authentication, session management, and business logic.Built with Node.js

Database: NoSQL database like MongoDB is used.

FLOW OF OPERATIONS

User Authentication:

- Users sign in using NEXT-AUTH Authentication tokens are exchanged for WebSocket connections.

Real-Time Messaging:

- Users send messages via WebSocket connections.
- Messages are received by the WebSocket server and broadcast to the relevant users or groups.
- Messages are stored in the database for history.

FLOW OF OPERATIONS

Group Chats:

- Users create or join groups.
- Group metadata and participant information are stored in the database.
- WebSocket server handles real-time communication for all participants in the group.

THANK YOU