# **Real-time communication in a web application**

Real time communication in a web application, like a chat box, **WebSocket** is the best choice. WebSockets enable full-duplex communication between the client (React) and the server (Spring Boot) over a single, long-lived connection.

**Tech Stack for Chat Application:**

1. **Frontend:** React (with WebSocket API or Socket.io client)
2. **Backend:** Spring Boot (using Spring WebSocket)
3. **Database:** MongoDB (for storing chat messages)

**How It Works:**

1. **React (Client) initiates a WebSocket connection** to the Spring Boot backend.
2. **Spring Boot WebSocket server processes messages** and broadcasts them to other connected clients.
3. **Messages can be stored in MongoDB** for persistence and history retrieval.
4. **All connected clients receive real-time updates** when a new message arrives.

**Key Components:**

* **Spring Boot:** Uses @EnableWebSocket and @ServerEndpoint to handle WebSocket connections.
* **React:** Uses WebSocket API (new WebSocket("ws://localhost:8080/chat")) or socket.io-client for communication.
* **MongoDB:** Stores chat history in a messages collection.