Peer-to-Peer (P2P) File Sharing System Documentation

Package Diagram

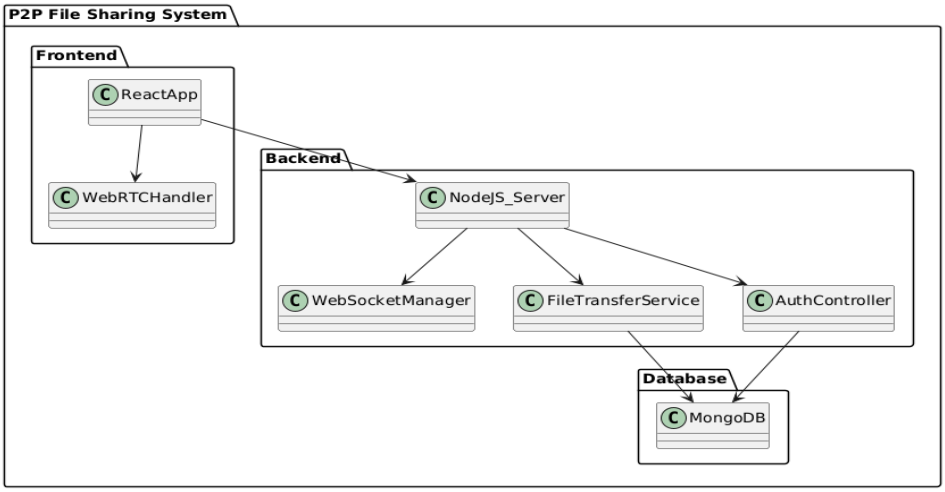
The package diagram illustrates the architecture of the Peer-to-Peer (P2P) File Sharing System, showcasing the relationships between different components.

Components:

1. **Frontend**
    - **ReactApp**: The main user interface that enables users to interact with the system.
    - **WebRTCHandler**: Manages WebRTC-based peer-to-peer communication.
2. **Backend**
    - **NodeJS_Server**: Acts as the central server for managing connections and authentication.
    - **WebSocketManager**: Handles real-time communication between clients and the backend.
    - **FileTransferService**: Manages file transfer requests.
    - **AuthController**: Ensures user authentication and authorization.
3. **Database**
    - **MongoDB**: Stores authentication data and transfer history.

The frontend communicates with the backend to establish connections, while the backend interacts with the database to store transfer records.

Package Diagram Image



Sequence Diagram

The sequence diagram describes the process flow of a file transfer between two users in the system.

Workflow:

1. **User A selects a file** to send.
2. **ReactApp (User A) sends a connection request** to the NodeJS server.
3. **NodeJS Server notifies ReactApp (User B)** about the incoming request.
4. **User B accepts the request**, and a WebRTC peer-to-peer connection is established.
5. **File is transferred in chunks** using WebRTC.
6. **ReactApp (User B) receives the file** and reconstructs it.
7. **NodeJS Server logs the transfer history** into MongoDB.
8. **A notification is sent** to both users confirming successful transfer.

This ensures an efficient peer-to-peer file sharing mechanism with authentication, logging, and real-time communication.

Sequence Diagram Image