Project Design Phase

# Solution Architecture

Date: 28-05-2025

Project Name: Video Conferencing App

Maximum Marks: 4 Marks

## Solution Architecture:

The Video Conferencing App follows a modular, scalable architecture optimized for real-time communication using WebRTC. It enables peer-to-peer (P2P) media streaming while using a minimal signaling server for user coordination and room management. The architecture is designed to be open, extensible, and easily deployable in cloud or on-premise environments.

- Client-side UI developed using HTML, CSS, JavaScript (or optionally React)

- Node.js and Express power the backend, handling routing and socket events

- WebRTC is used for real-time video/audio streaming, and PeerJS abstracts peer connections

- Socket.IO handles signaling between peers and room lifecycle events

- Room IDs are generated dynamically and shared as unique links

- Screen sharing is supported via the browser’s media devices API

- Recording is enabled using MediaRecorder and stored locally or optionally sent to backend

- Frontend and backend are separated for easier scaling and testing

- Optional MongoDB database can log session metadata, diagnostics, or analytics

- Support for future enhancements: password-protected rooms, chat history persistence, TURN server integration

## Non-functional Characteristics:

- \*\*Low Latency:\*\* Peer-to-peer streams minimize delay.

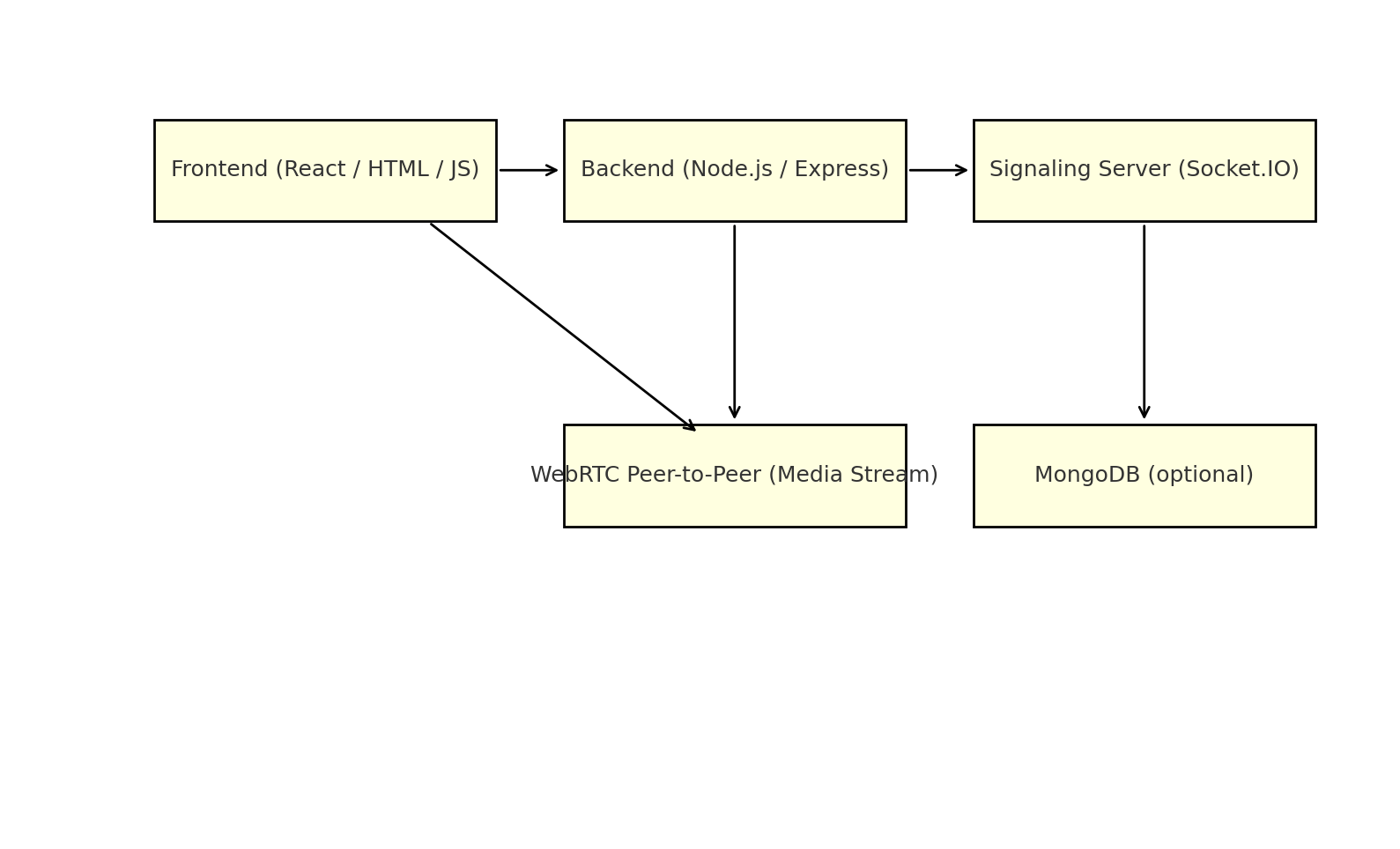
- \*\*Scalability:\*\* Signaling server can handle thousands of concurrent users with clustering or horizontal scaling.

- \*\*Security:\*\* No media stored on server; streams are encrypted by design via WebRTC.

- \*\*Extensibility:\*\* Easily integrate with calendar APIs, user auth systems, or analytics tools.

- \*\*Portability:\*\* The app can run on any modern browser and server platform (Linux/Windows).

## Example - Solution Architecture Diagram:



## Final Diagram - Complete Peer Connection Workflow:

This final diagram presents a complete overview of the peer connection process between two users. It covers all key steps including SDP offer/answer exchange via the signaling server, ICE candidate sharing, and the establishment of direct peer-to-peer media streams using WebRTC.

