Solution Architecture – *Video Conferencing Web App*

📌 Overview

The solution architecture ensures that the web app is **secure**, **scalable**, and **user-friendly**, while providing **real-time communication** using modern technologies like React, Node.js, Socket.io, and Agora SDK.

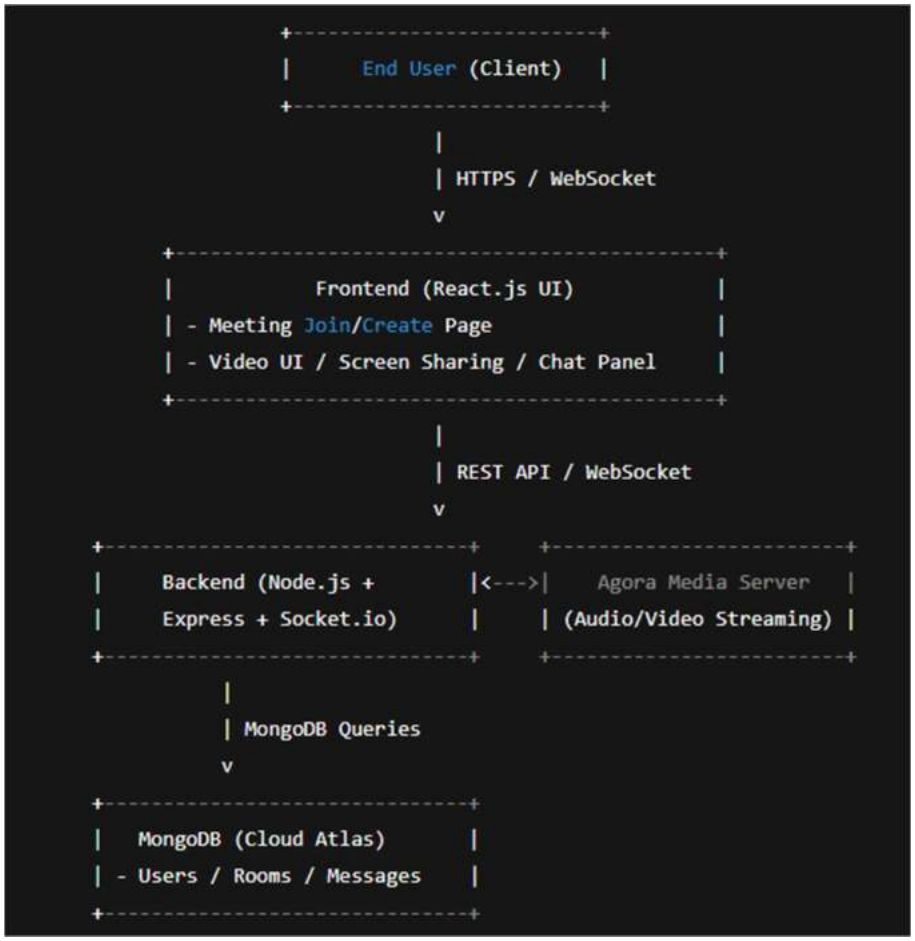
This modular architecture enables users to:

- Create/join video rooms instantly
- Communicate via video, audio, and chat
- Share their screen without extra tools
- Interact with a clean, mobile-responsive interface
- Be managed securely using token-based authentication

🧱 Key Architecture Features

- ✅ **Frontend (React.js + Tailwind CSS)**
  Handles the user interface, meeting screens, participant controls, and screen sharing options.
- ✅ **Backend (Node.js + Express.js)**
  Handles APIs for room creation, authentication, and token management.
- ✅ **Real-Time Communication (Socket.io + Agora SDK)**
  - Socket.io handles signaling and chat
  - Agora SDK manages real-time audio/video streaming and screen sharing
- ✅ **Authentication & Security**
  - JWT tokens for secure login and room access
  - HTTPS and role-based access for room host/guest actions
- ✅ **Database (MongoDB)**
  - Stores user data, meeting metadata, and chat history
- ✅ **Hosting/Deployment**
  - **Frontend:** Vercel
  - **Backend:** Render or Railway
  - **Database:** MongoDB Atlas

📐 Solution Architecture Diagram (Text Representation)



```
                +-------------------------+
                |     End User (Client)   |
                +-------------------------+
                             |
                             | HTTPS / WebSocket
                             v
      +-------------------------------------------+
      |             Frontend (React.js UI)        |
      | - Meeting Join/Create Page                |
      | - Video UI / Screen Sharing / Chat Panel  |
      +-------------------------------------------+
                             |
                             | REST API / WebSocket
                             v
+--------------------------------+    +--------------------------+
|      Backend (Node.js +        |<--->|   Agora Media Server    |
|      Express + Socket.io)      |    | (Audio/Video Streaming) |
+--------------------------------+    +--------------------------+
           |
           | MongoDB Queries
           v
+--------------------------------+
|   MongoDB (Cloud Atlas)        |
| - Users / Rooms / Messages     |
+--------------------------------+
```

🛡️ Key System Qualities

Quality	**Design Choice**
Scalability	Socket.io rooms + Agora channels allow high concurrency
Security	JWT tokens, HTTPS, room access roles
Availability	Cloud-hosted (Vercel/Render) with automatic scaling
Performance	Lightweight frontend with optimized components & lazy loading
Maintainability	Modular codebase and service separation
Extensibility	Future features (recording, analytics) can be plugged in easily