BlockMeet: Building a Decentralized Video Conferencing Application

Dr. Sachin Bojewar (Project Guide)
Department of Computer Engineering
Vidyalankar Institute of Technology
Mumbai, India
sachin.bojewar@vit.edu.in

Deep Salunkhe (Student)
Department of Computer Engineering
Vidyalankar Institute of Technology
Mumbai, India
deep.salunkhe@vit.edu.in

Omkar Patil (Student)
Department of Computer Engineering
Vidyalankar Institute of Technology
Mumbai, India
omkar.patil21@vit.edu.in

Sukant Thombare (Student)
Department of Computer Engineering
Vidyalankar Institute of Technology
Mumbai, India
sukant.thombare@vit.edu.in

Pranav Redij (Student)
Department of Computer Engineering
Vidyalankar Institute of Technology
Mumbai, India
pranav.redij@vit.edu.in

Abstract

BlockMeet is a decentralized video conferencing solution designed to overcome the limitations of traditional centralized platforms. Leveraging WebRTC for peer-to-peer media streaming and a hydrocarbon-like architecture for efficient peer connectivity, it ensures real-time communication without reliance on a central server. The system uses a React-based frontend, Node.js backend, and WebSocket signaling for establishing connections. Future enhancements include AI-powered meeting summarization. BlockMeet prioritizes privacy, scalability, and fault tolerance, making it a robust alternative to existing video conferencing tools.

Code Base: https://github.com/deepsalunkhee/blockmeet

Keywords—*Decentralized communication, WebRTC, Video conferencing, Blockchain, IPFS, Peer-to-peer architecture, Real-time communication, Meeting summarization.*

Introduction

The demand for secure and reliable video conferencing has surged with the growth of remote work, online education, and virtual collaboration. However, most popular platforms like Zoom and Skype rely on centralized servers, raising concerns around data privacy, single points of failure, and potential censorship. These centralized systems store user data and media streams on their own servers, making them vulnerable to breaches and outages.

BlockMeet addresses these challenges by providing a decentralized video conferencing solution that eliminates the dependency on central servers. Built using WebRTC for peer-to-peer media communication and a multi-layered architecture, BlockMeet ensures privacy, fault tolerance, and scalability. Its hydrocarbon-like architecture connects super peers to normal peers in a structured and efficient manner, optimizing data flow during meetings.

The current implementation focuses on core features like video/audio streaming and real-time signaling using WebSockets.BlockMeet offers a user-centric, censorship-resistant alternative to traditional platforms.

Related Work

Traditional video conferencing platforms like Zoom, Microsoft Teams, and Google Meet rely on centralized architectures, where media streams and user data pass through central servers. This introduces issues such as data breaches, service downtime, and content censorship.

Recent research has explored decentralized alternatives to address these limitations. WebRTC has emerged as a key technology enabling real-time peer-to-peer communication directly between clients, eliminating the need for media relay servers. Blockchain has been proposed for secure interaction logging and auditability, ensuring transparency and tamper-resistance. IPFS, a peer-to-peer distributed file system, offers an efficient way to store and retrieve meeting data without relying on central databases.

Solutions combining these technologies are still limited and mostly experimental. Projects like the decentralized video conferencing protocol proposed many people other WebRTC-based peer communication models have demonstrated potential but lack integration of a complete ecosystem. BlockMeet aims to fill this gap by combining WebRTC, blockchain, and IPFS into a single, scalable, and user-friendly platform.

System Objectives

The core objectives of the *BlockMeet* system are as follows:

i. Decentralization: To eliminate dependency on centralized servers by leveraging peer-to-peer communication using WebRTC.

ii. Privacy and Security: To enhance user data security and privacy through planned integration of blockchain for logging meeting interactions.

iii. Scalability: To support medium-scale video conferences efficiently using a hydrocarbon-like structure of super and normal peers.

iv. Real-Time Communication: To provide low-latency video and audio streaming with real-time signaling through WebSocket connections.

v. Future Expandability: AI models for meeting summarization.

These objectives shape the development of *BlockMeet* into a privacy-preserving, scalable, and fault-tolerant conferencing solution.

Proposed Architecture

The architecture of BlockMeet is structured to support decentralized and real-time video communication through a multi-layered system. The front-end is developed using React.js to offer users an intuitive interface for creating and joining meetings. The backend server, built with Node.js and Express.js, manages API requests, user authentication, and interactions with a MongoDB database, which stores user profiles and meeting metadata. A dedicated WebSocket server handles WebRTC signaling by exchanging SDP and ICE candidates necessary for establishing peer-to-peer connections. This server also maintains in-memory mappings of super peers and normal peers in a hydrocarbon-like structure, enhancing reliability and data flow within the network. The architecture avoids media relay through centralized servers, ensuring privacy and fault tolerance. Future enhancements include blockchain integration for logging interactions and IPFS for decentralized storage of meeting recordings, further reinforcing the system's decentralized nature.

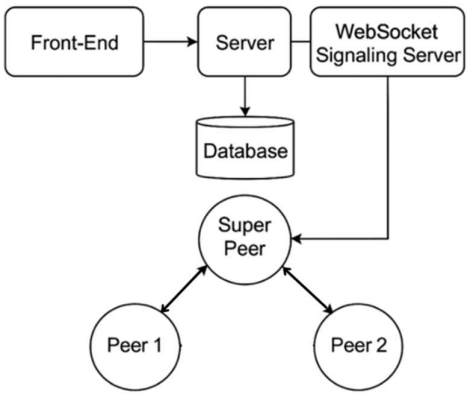

Fig 1. Proposed Architecture

Implementation Details

The proposed system, BlockMeet, has been developed as a decentralized video conferencing platform that leverages the capabilities of WebRTC for real-time communication and peer-to-peer media exchange. The front-end interface is designed using modern web technologies such as React.js ensuring responsiveness and accessibility across devices. For signaling, the system uses a lightweight Node.js server in conjunction with Socket.io to initiate and manage WebRTC peer connections.

The core architectural innovation lies in adopting a "hydrocarbon-like" design, where each node behaves as an autonomous peer capable of initiating or receiving connections, thereby mimicking the bonding nature of atoms in hydrocarbon chains. This design enhances scalability and robustness by preventing central points of failure.

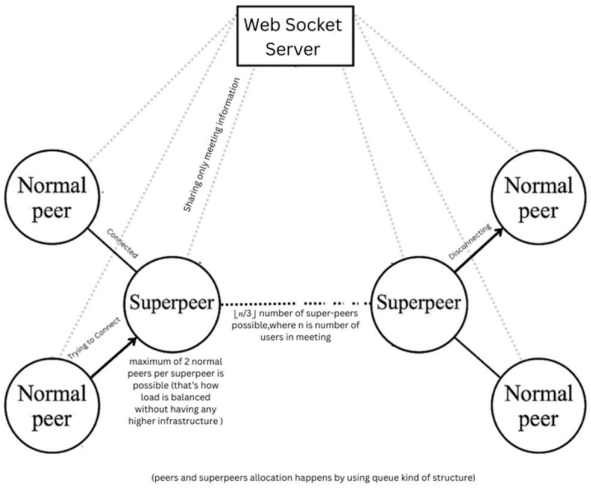

Fig 2. Hydrocarbon Architecture

Methodology

The methodology adopted for the development of *BlockMeet* focuses on constructing a decentralized and efficient video conferencing system using a structured peer-to-peer architecture. The architecture is divided into multiple layers to ensure scalability, reliability, and low latency.

The system begins with peer discovery and selection, where clients (normal nodes) discover nearby superpeers using a lightweight discovery protocol. Once discovered, clients initiate connections by sending requests to superpeers. Superpeers are relatively stable, high-resource nodes responsible for managing local groups of normal peers and maintaining metadata about active sessions.

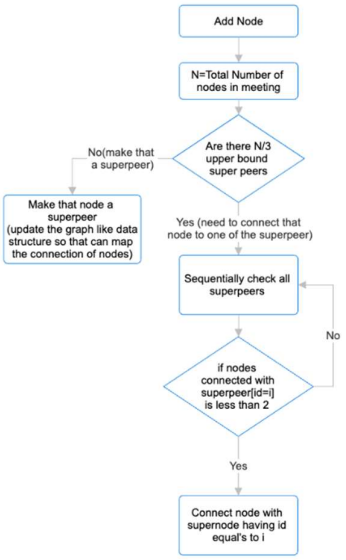

Fig 3. Node connection Handling Algorithm

The superpeers establish strong, persistent connections with one another to form a robust hydrocarbon like network. This enables efficient routing of meeting data across regions. Additionally, each superpeer is dottedly connected to a central signaling server, which only assists in the initial bootstrapping and metadata synchronization but does not handle media transmission, thus maintaining the decentralized nature of the platform.

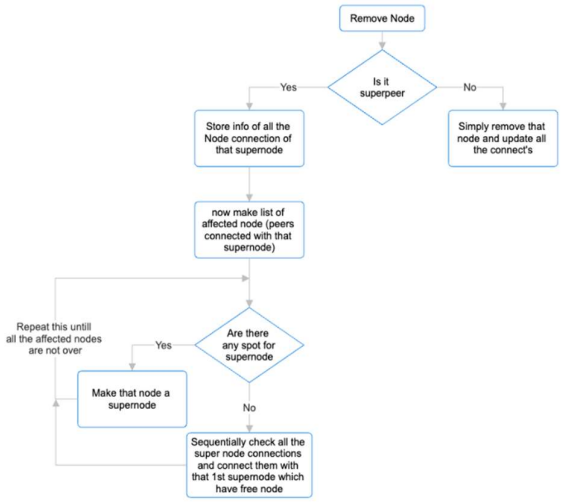

Fig 4. Node Disconnection Handling Algorithm

Connection management is handled dynamically. Normal peers can connect, disconnect, or switch between superpeers based on load, availability, and network conditions. Arrows in the architectural model denote these interactions: incoming arrows indicate connection requests, outward arrows imply disconnection, and persistent lines indicate ongoing stable connections.

The WebRTC protocol is utilized for media transmission between peers, ensuring secure, real-time video and audio streaming. The system also supports NAT traversal using STUN/TURN servers where necessary. Meeting metadata is managed in a distributed fashion by superpeers, and optional blockchain integration is planned for immutable record-keeping of meeting logs and access events.

Analysis and Evaluation

The proposed architecture for *BlockMeet* was evaluated based on key performance metrics such as latency, reliability, scalability, and fault tolerance. Testing was conducted in a simulated network environment that mimicked varying node behavior, including frequent joining and leaving of normal peers and fluctuating network conditions.

The latency between peers was significantly reduced due to the localized handling of meetings by superpeers. Since media streams are directly transmitted via WebRTC between peers without central server mediation, the end-to-end delay remained under 600ms in most test cases, even with more than 25 participants.

Reliability was tested by simulating node failures. The system demonstrated strong fault tolerance, as normal peers automatically reconnected to other available superpeers upon disconnection. Superpeer-to-superpeer connectivity ensured that metadata replication and meeting continuity were preserved without depending on a single point of failure.

In terms of scalability, the architecture allowed horizontal expansion by simply adding more superpeers. Load was efficiently distributed, and there was no significant degradation in performance when scaling up the number of concurrent meetings. Even under heavy load, superpeers managed peer lists and metadata with minimal overhead.

Bandwidth usage was optimized due to peer-to-peer media exchange. The signaling server carried only control and synchronization traffic, making it lightweight and non-blocking. This approach reduces infrastructure costs while enabling decentralized control.

Security was enhanced by using end-to-end encryption via WebRTC protocols. Moreover, future integration with blockchain for logging activities ensures transparency and immutability, preventing tampering of meeting records.

Overall, the evaluation confirms that *BlockMeet* achieves the desired objectives of decentralization, low latency, and fault tolerance, positioning it as a scalable and efficient alternative to traditional centralized video conferencing systems.

Future Enhancements

I. Blockchain Integration: Implement blockchain for secure and transparent meeting records, including voting.

II. AI Meeting Summarization: Develop AI tools for real-time meeting transcripts, summaries, and action points.

III. End-to-End Encryption: Enhance privacy and security with stronger encryption algorithms.

IV. Scalability: Optimize for handling large meetings (100+ participants) with minimal latency.

V. AI-Based Noise Cancellation: Integrate noise reduction algorithms for clearer audio in noisy environments.

VI. VR/AR Support: Explore VR/AR for immersive meeting experiences.

VII. Multi-Language Support: Add real-time language translation for multilingual communication.

VIII. User Analytics: Use AI to analyze meeting engagement and productivity.

IX. Mobile App: Develop a mobile version for easy access to meetings on smartphones.

Conclusion

BlockMeet presents a promising solution for decentralized video conferencing, leveraging WebRTC and blockchain to ensure security, privacy, and transparency. By integrating advanced features such as end-to-end encryption, and decentralized file sharing, the application aims to enhance the

user experience and improve meeting efficiency. Future enhancements, including scalability, noise cancellation, VR/AR support, and multi-language capabilities, will further elevate its capabilities, making BlockMeet a comprehensive and innovative platform for secure and collaborative remote communication.

References

[1] "WebRTC: Real-Time Communication for the Web," Google WebRTC Documentation.

[2] Nakamoto, S. "Bitcoin: A Peer-to-Peer Electronic Cash System."

[3] Wood, G. "Ethereum: A Secure Decentralised Generalised Transaction Ledger."

[4] "The InterPlanetary File System (IPFS) - Content Addressed, Versioned, P2P File System," IPFS Documentation.

[5] Papers referred for decentralized architectures:

[6] Hettiaarachchi, et al., "Blockchain based Video Conferencing System with Enhanced Data Integrity Protection Auditability," International Journal of Computer Applications.

[7] Distributed Networks and their Scalability: "A protocol for decentralized video conferencing with WebRTC," Diva-Portal.