Block Meet

A Decentralized Video Conferencing Solution

Synopsis Report submitted in partial fulfillment

of the requirement for the degree of

B. E. (Computer Engineering)

Submitted By

Deep Salunkhe

Omkar Patil

Pranav Redij

Sukant Thombare

Under the Guidance of

Dr. Sachin Bojewar

Department of Computer Engineering

(An Autonomous Institute Affiliated to University of Mumbai)

Vidyalankar Institute of Technology

Wadala(E), Mumbai 400 037

University of Mumbai

2024-25

CERTIFICATE OF APPROVAL

For

Project Synopsis

This is to Certify that

Deep Salunkhe

Omkar Patil

Pranav Redij

Sukant Thombare

Have successfully carried out Project Synopsis work entitled

Block Meet

A Decentralized Video Conferencing Solution

in partial fulfillment of degree course in

Computer Engineering

As laid down by University of Mumbai during the academic year

2024-25

Under the Guidance of

Dr. Sachin Bojewar

Signature of Guide Head of Department

Examiner 1 Examiner 2 Principal

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

	Name of student	Roll No.	Signature
1.	Deep Salunkhe	21102A0014	
2.	Omkar Patil	21102A0003	
3.	Pranav Redij	21102A0005	
4.	Sukant Thombare	21102A0037	

Date:

Acknowledgements

We would like to express our sincere gratitude to our mentor, Dr Sachin Bojewar for his continuous support and guidance. We also thank our department and the institute for providing us the opportunity and resources to work on this project.

Abstract

BlockMeet is a decentralized video conferencing application designed to overcome the limitations of traditional centralized platforms. Using a multi-layer architecture that includes a front-end React application, server, WebSocket signaling, and MongoDB database, BlockMeet enables secure, privacy-focused, real-time communication. The hydrocarbon-like architecture connects super peers to normal peers in a structured manner, ensuring efficient information flow. While the current prototype supports core video conferencing features, future versions will incorporate blockchain for interaction logging, IPFS for decentralized storage, and AI for meeting summarization.

Github Repo: https://github.com/Omkar-Patil-2003/BlockMeet

Table of Contents

1. Introduction

The increasing reliance on digital communication platforms has brought about the need for more secure and private solutions. Traditional video conferencing systems often rely on centralized servers, leading to data privacy concerns and the risk of single points of failure. BlockMeet addresses these challenges by implementing a decentralized video conferencing architecture using WebRTC and blockchain technologies, focusing on user privacy, data ownership, and censorship resistance.

2. Aim and Objectives

- ✓ To develop a decentralized video conferencing platform using a multi-layer architecture.
- ✓ To ensure high security and privacy through blockchain integration for logging interactions and decentralized storage using IPFS.
- ✓ To support scalable and reliable real-time communication with a hydrocarbon-like architecture.
- ✓ To enable future capabilities like AI-powered meeting summarization and storage of meeting recordings.

3. Literature Surveyed

- Existing video conferencing solutions like Zoom and Skype rely on centralized networks. These architectures suffer from limitations such as centralized data storage, potential data breaches, and censorship risks.
- Decentralized architectures offer advantages in terms of privacy and scalability but present challenges in consistency and network synchronization.
- WebRTC has emerged as a popular choice for peer-to-peer media transfer, while blockchain and IPFS provide secure and immutable data management solutions.

4. Problem Statement

The primary challenge with current video conferencing platforms is their centralized nature, which results in data privacy concerns, dependency on single points of failure, and potential content censorship. BlockMeet aims to build a decentralized solution to overcome these issues, providing a user-centric, secure, and censorship-resistant communication platform.

5. Scope

BlockMeet will initially focus on medium-sized meetings with basic conferencing features. Future versions will expand to include functionalities such as interaction logging on the blockchain, AI-generated meeting summaries, and decentralized storage for meeting recordings.

6. Proposed System

The BlockMeet system is built with a multi-layer architecture consisting of four main components: the front-end, server, WebSocket server, and MongoDB database. The proposed architecture follows a hydrocarbon-like structure, where super peers connect to two normal peers, facilitating decentralized data transfer. The system's architecture ensures fault tolerance, scalability, and privacy for medium-sized meetings.

- **Front-End**: A React-based application that provides a user-friendly interface for creating and joining meetings, enabling video/audio stream sharing.
- **Server**: Manages API requests, handles user authentication, and interacts with the database to store user information and meeting details.
- **WebSocket Server**: Acts as the signaling server for WebRTC, facilitating the exchange of Session Description Protocol (SDP) and Interactive Connectivity Establishment (ICE) candidates to establish peer-to-peer connections. It also maintains the mapping of super peers and normal peers in a meetings object.
- **Database (MongoDB)**: Stores user-related data, such as user profiles and meeting metadata. The database does not store super-peer mappings, as they are managed in-memory within the WebSocket server.

The future system enhancements include incorporating blockchain technology to log meeting interactions, integrating IPFS for decentralized storage, and using AI models to convert meeting recordings into text summaries for easy access.

7. Methodology

The development process follows an iterative approach to build the decentralized video conferencing solution:

- **Architecture Design**: Initial discussions led to the selection of a hydrocarbon-like architecture, where super peers manage connections between normal peers and relay data across the network.
- **Technology Selection**: WebRTC was chosen for real-time media communication, Ethereum blockchain for future interaction logging, and IPFS for decentralized storage solutions.
- **Implementation Phases**:
    1. **Basic Features Development**: Implemented core functionalities like meeting creation, joining, and decentralized audio/video streaming.
    2. **WebSocket Integration**: Set up a signaling mechanism using a WebSocket server to manage peer connections.
    3. **Testing and Validation**: Conducted extensive tests to ensure stable peer-to-peer communication and identify areas for improvement.
    4. **Future Development**: Planned integration of blockchain for logging and IPFS for storing recordings and summaries.

The methodology emphasizes continuous refinement and validation through user feedback and iterative enhancements.

8. Analysis

8.1 Process Model Used for the Project

The project utilizes an Agile methodology, enabling the team to adapt to changes quickly and continuously improve the system based on iterative feedback.

8.2 Feasibility Study

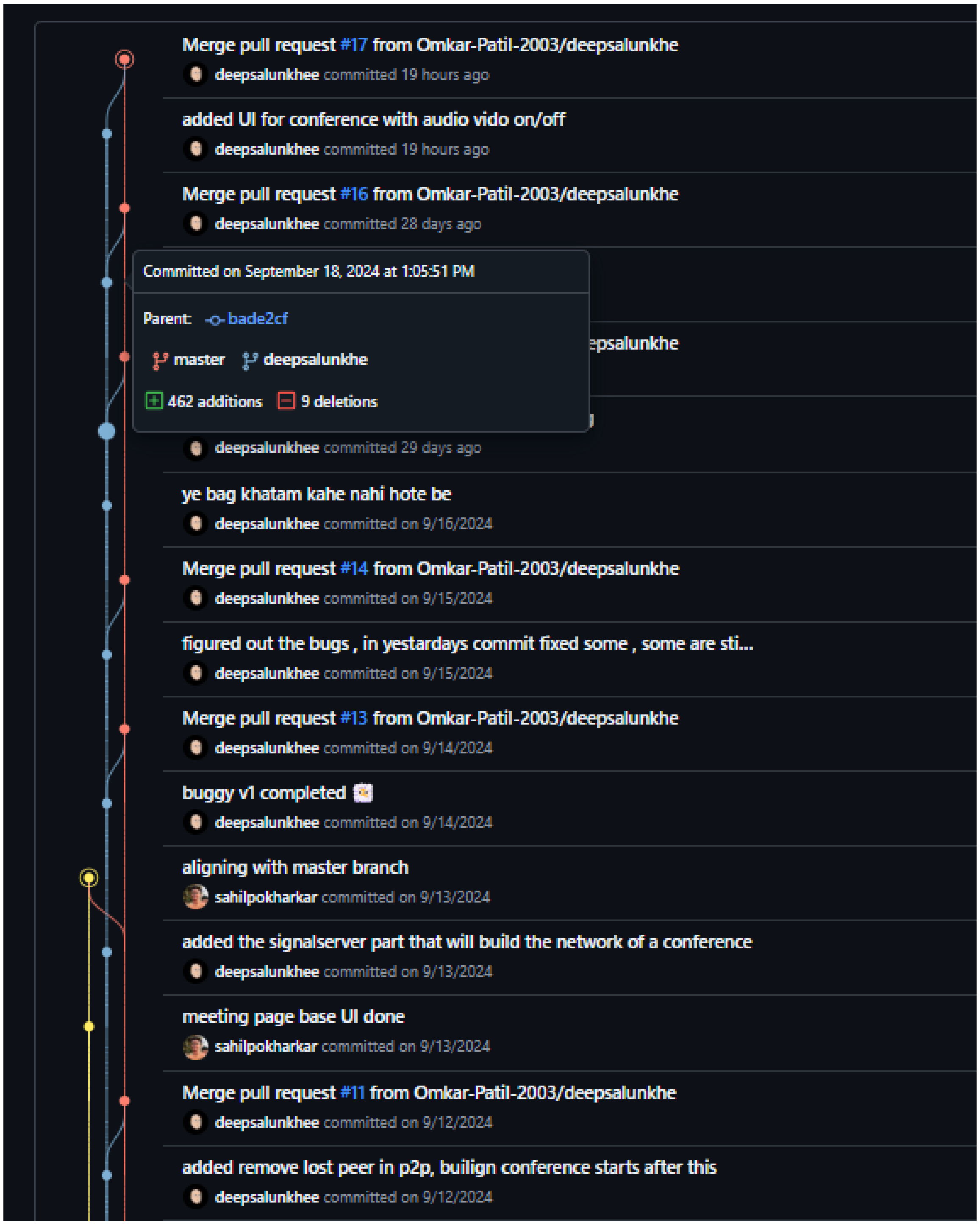
- Technical Feasibility: The use of React, WebRTC, and WebSocket technologies makes the implementation feasible, while Ethereum and IPFS are viable for future features.
- Economic Feasibility: The project leverages open-source technologies, reducing development costs. Integration with blockchain and IPFS may introduce some cost considerations in future versions.
- Operational Feasibility: The system aims to be user-friendly with a simple interface, making it easy for users to adopt.

8.3 Cost Analysis

The project primarily incurs costs related to hosting the WebSocket server and server backend. Future costs may involve gas fees for blockchain transactions and storage costs for IPFS.

8.4 Timeline Chart

The following are git commits charts



Merge pull request #17 from Omkar-Patil-2003/deepsalunkhe
deepsalunkhee committed 19 hours ago

added UI for conference with audio vido on/off
deepsalunkhee committed 19 hours ago

Merge pull request #16 from Omkar-Patil-2003/deepsalunkhe
deepsalunkhee committed 28 days ago

Committed on September 18, 2024 at 1:05:51 PM
Parent: bade2cf
master deepsalunkhe
462 additions 9 deletions

epsalunkhe

deepsalunkhee committed 29 days ago

ye bag khatam kahe nahi hote be
deepsalunkhee committed on 9/16/2024

Merge pull request #14 from Omkar-Patil-2003/deepsalunkhe
deepsalunkhee committed on 9/15/2024

figured out the bugs , in yestardays commit fixed some , some are sti...
deepsalunkhee committed on 9/15/2024

Merge pull request #13 from Omkar-Patil-2003/deepsalunkhe
deepsalunkhee committed on 9/14/2024

buggy v1 completed
deepsalunkhee committed on 9/14/2024

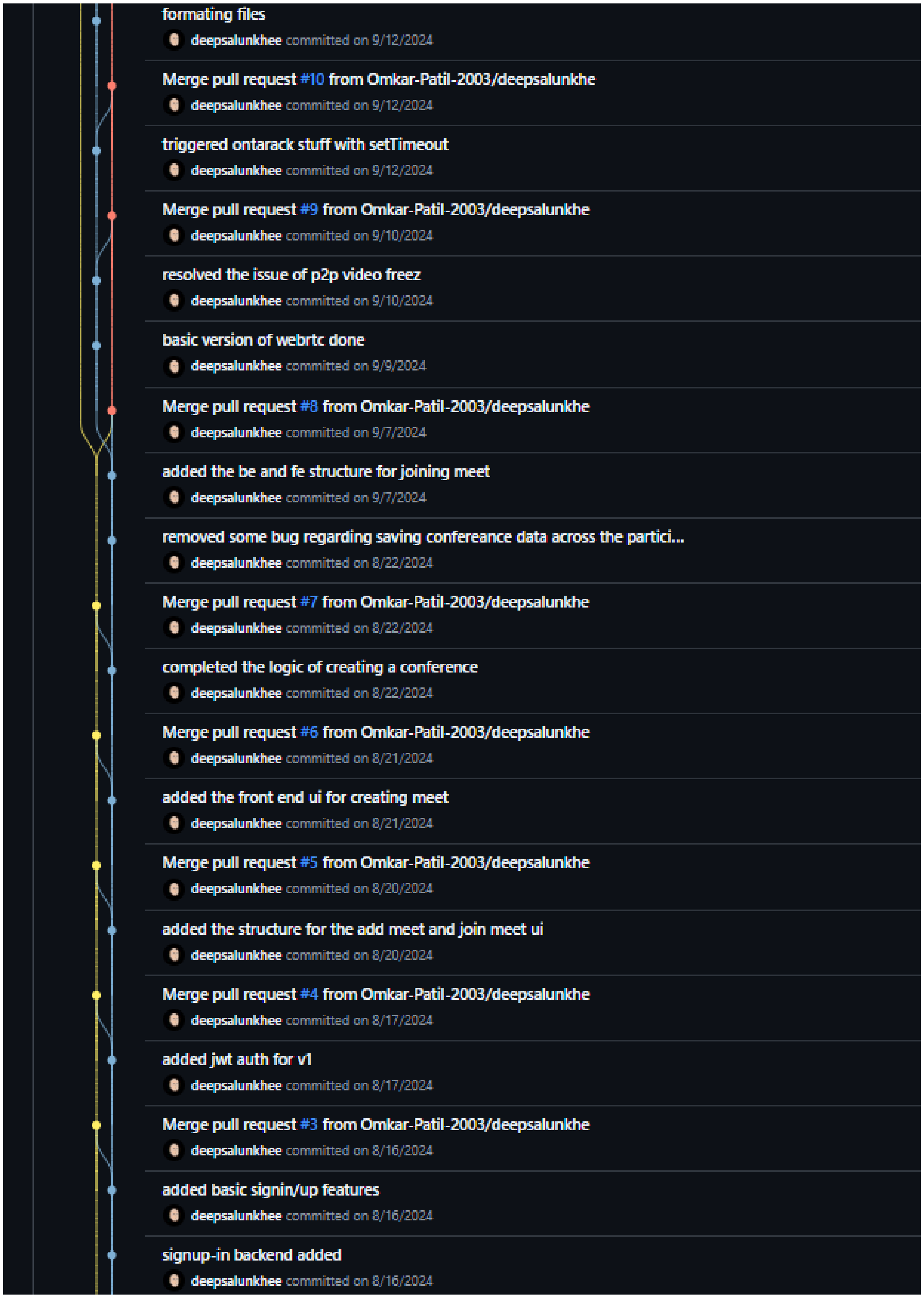
aligning with master branch
sahilpokharkar committed on 9/13/2024

added the signalserver part that will build the network of a conference
deepsalunkhee committed on 9/13/2024

meeting page base UI done
sahilpokharkar committed on 9/13/2024

Merge pull request #11 from Omkar-Patil-2003/deepsalunkhe
deepsalunkhee committed on 9/12/2024

added remove lost peer in p2p, builign conference starts after this
deepsalunkhee committed on 9/12/2024


formating files
deepsalunkhee committed on 9/12/2024

Merge pull request #10 from Omkar-Patil-2003/deepsalunkhe
deepsalunkhee committed on 9/12/2024

triggered ontarack stuff with setTimeout
deepsalunkhee committed on 9/12/2024

Merge pull request #9 from Omkar-Patil-2003/deepsalunkhe
deepsalunkhee committed on 9/10/2024

resolved the issue of p2p video freez
deepsalunkhee committed on 9/10/2024

basic version of webrtc done
deepsalunkhee committed on 9/9/2024

Merge pull request #8 from Omkar-Patil-2003/deepsalunkhe
deepsalunkhee committed on 9/7/2024

added the be and fe structure for joining meet
deepsalunkhee committed on 9/7/2024

removed some bug regarding saving confereance data across the partici...
deepsalunkhee committed on 8/22/2024

Merge pull request #7 from Omkar-Patil-2003/deepsalunkhe
deepsalunkhee committed on 8/22/2024

completed the logic of creating a conference
deepsalunkhee committed on 8/22/2024

Merge pull request #6 from Omkar-Patil-2003/deepsalunkhe
deepsalunkhee committed on 8/21/2024

added the front end ui for creating meet
deepsalunkhee committed on 8/21/2024

Merge pull request #5 from Omkar-Patil-2003/deepsalunkhe
deepsalunkhee committed on 8/20/2024

added the structure for the add meet and join meet ui
deepsalunkhee committed on 8/20/2024

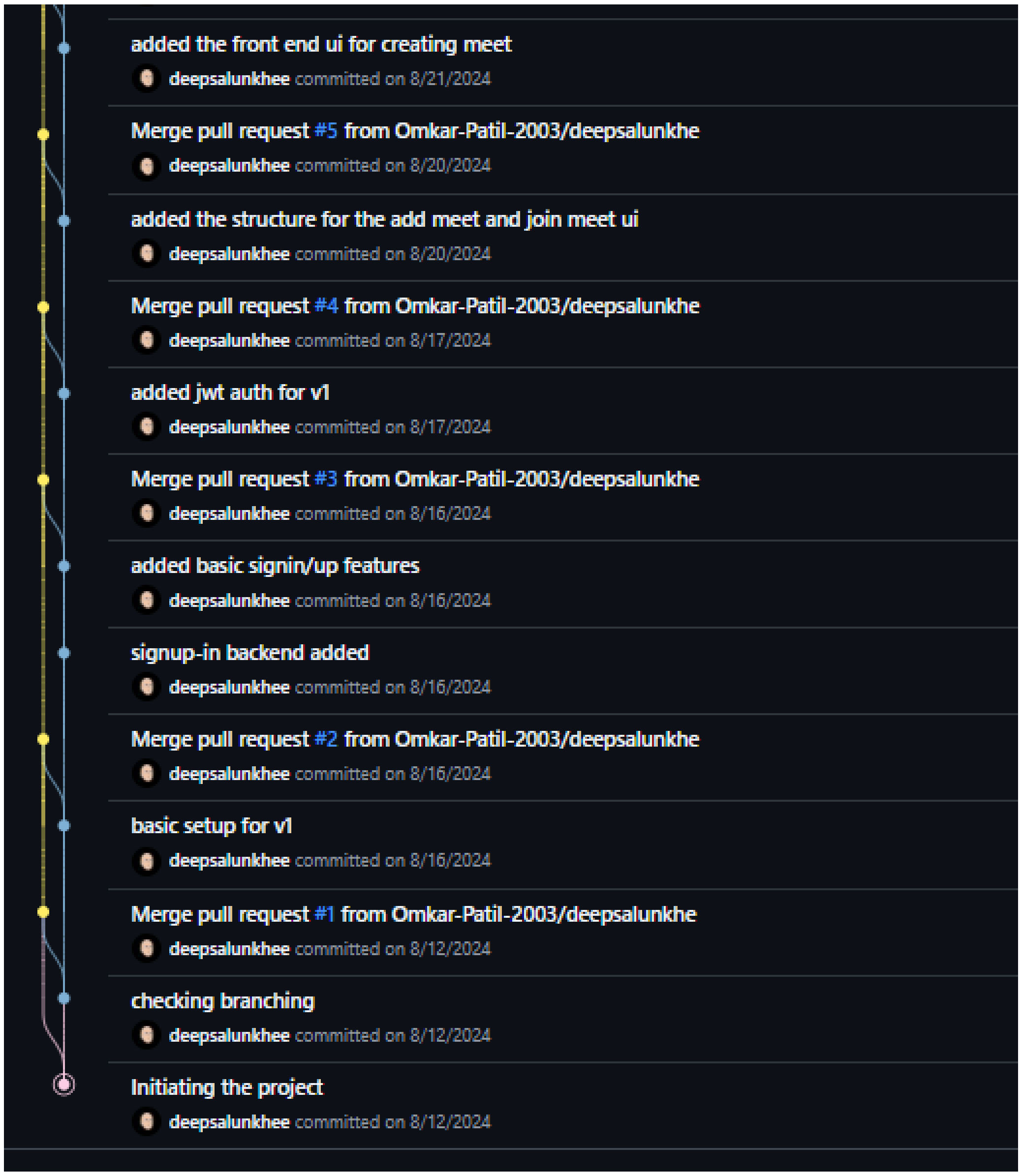
Merge pull request #4 from Omkar-Patil-2003/deepsalunkhe
deepsalunkhee committed on 8/17/2024

added jwt auth for v1
deepsalunkhee committed on 8/17/2024

Merge pull request #3 from Omkar-Patil-2003/deepsalunkhe
deepsalunkhee committed on 8/16/2024

added basic signin/up features
deepsalunkhee committed on 8/16/2024

signup-in backend added
deepsalunkhee committed on 8/16/2024



added the front end ui for creating meet
deepsalunkhee committed on 8/21/2024

Merge pull request #5 from Omkar-Patil-2003/deepsalunkhe
deepsalunkhee committed on 8/20/2024

added the structure for the add meet and join meet ui
deepsalunkhee committed on 8/20/2024

Merge pull request #4 from Omkar-Patil-2003/deepsalunkhe
deepsalunkhee committed on 8/17/2024

added jwt auth for v1
deepsalunkhee committed on 8/17/2024

Merge pull request #3 from Omkar-Patil-2003/deepsalunkhe
deepsalunkhee committed on 8/16/2024

added basic signin/up features
deepsalunkhee committed on 8/16/2024

signup-in backend added
deepsalunkhee committed on 8/16/2024

Merge pull request #2 from Omkar-Patil-2003/deepsalunkhe
deepsalunkhee committed on 8/16/2024

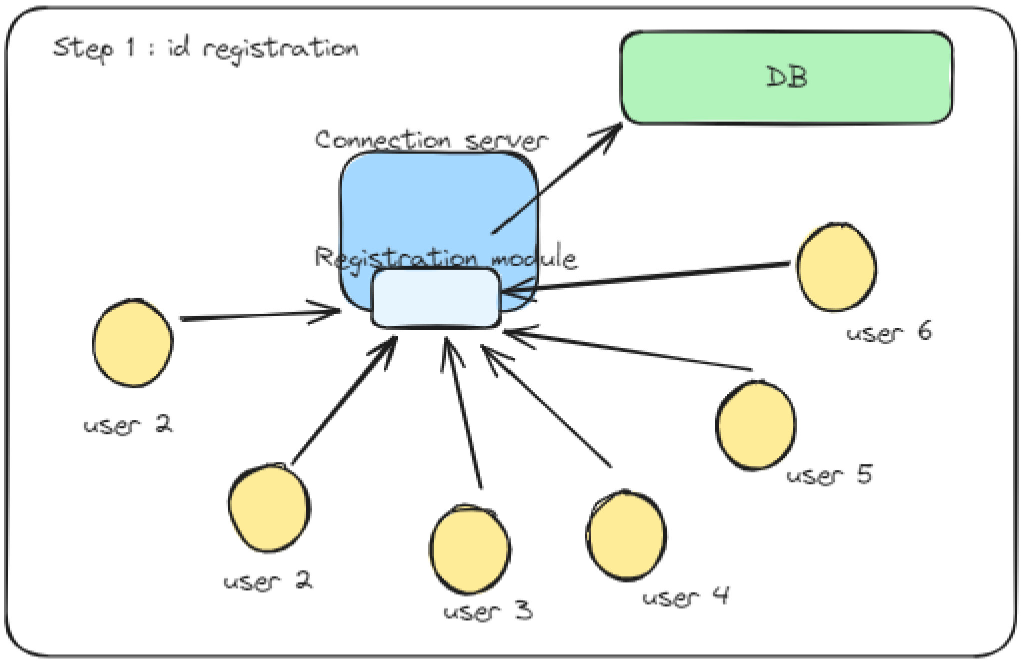
basic setup for v1
deepsalunkhee committed on 8/16/2024

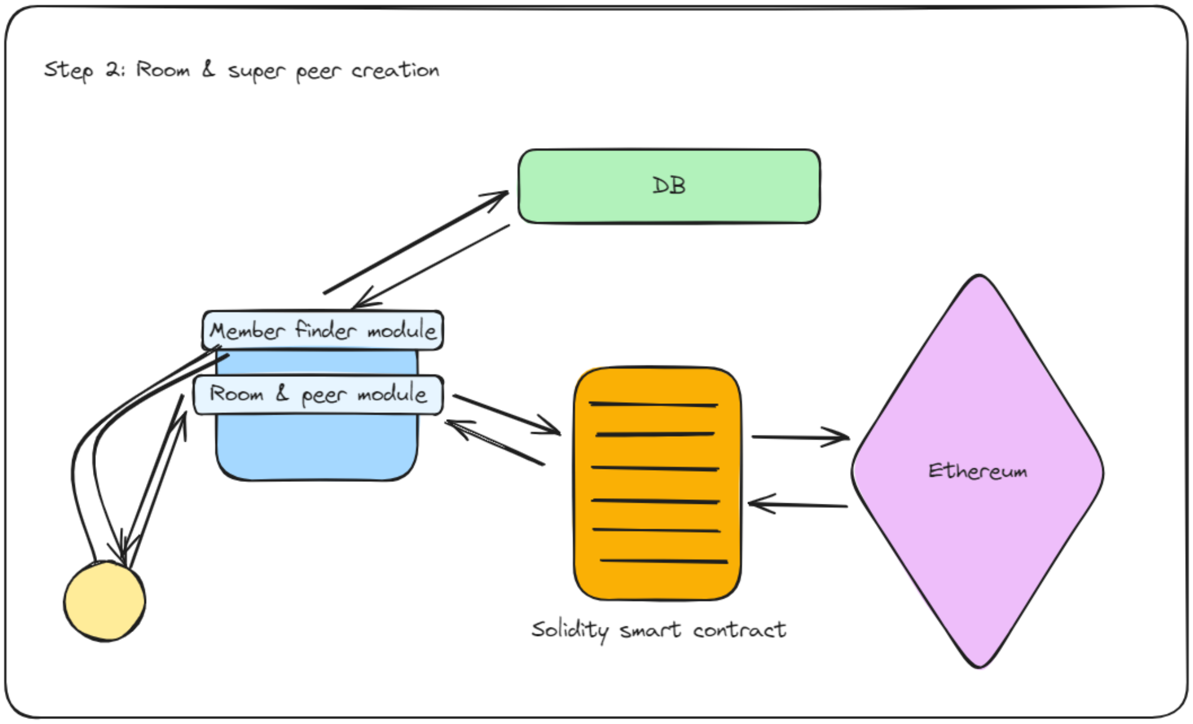
Merge pull request #1 from Omkar-Patil-2003/deepsalunkhe
deepsalunkhee committed on 8/12/2024

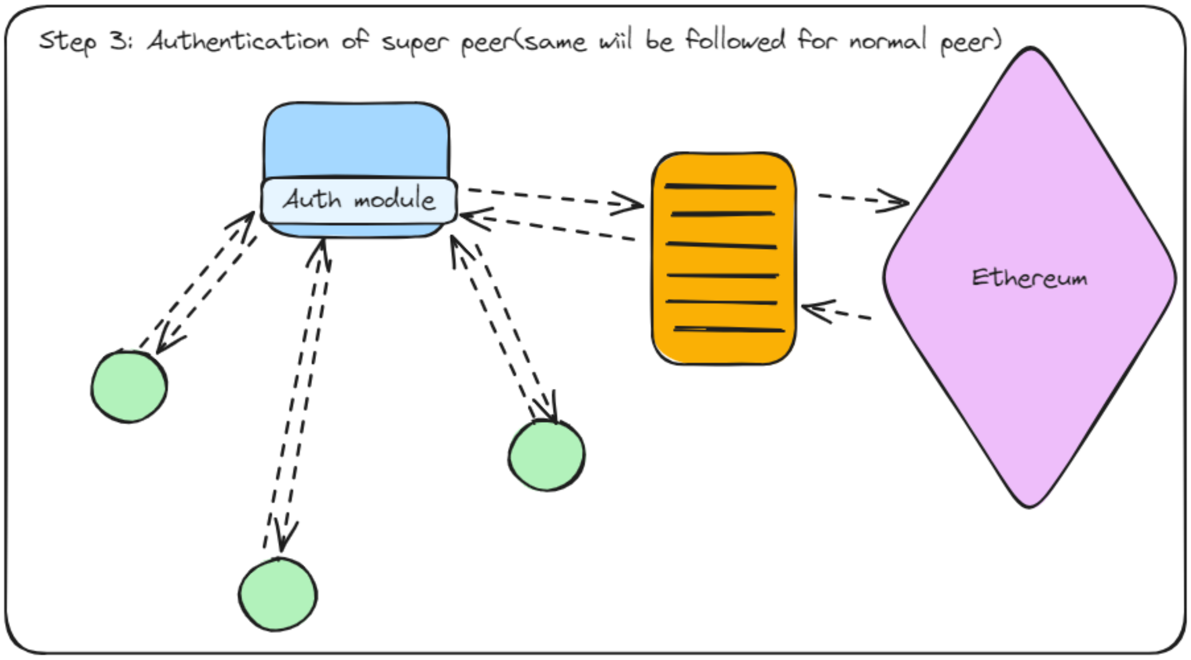
checking branching
deepsalunkhee committed on 8/12/2024

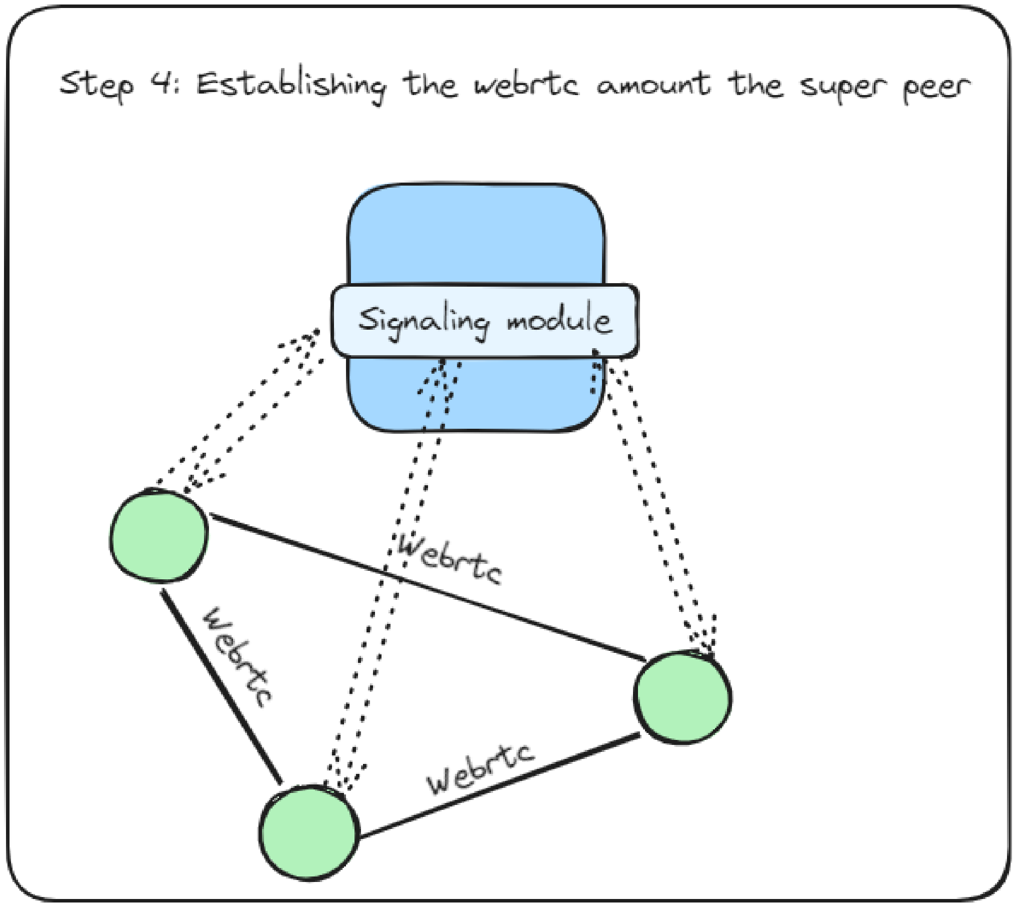
Initiating the project
deepsalunkhee committed on 8/12/2024

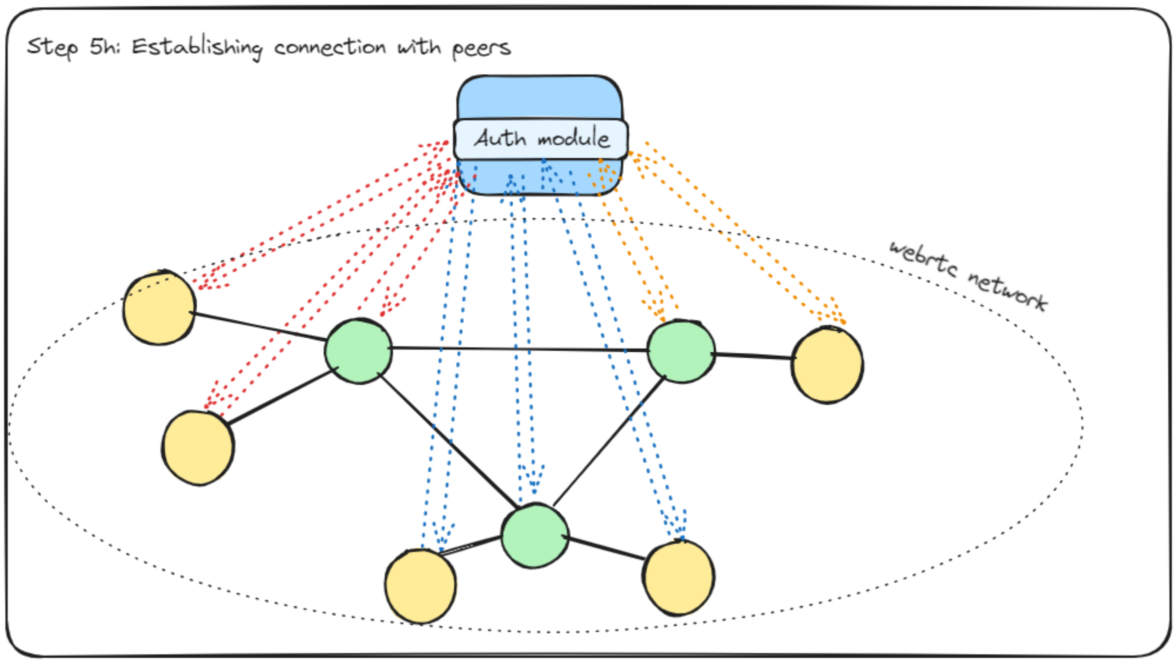
9. Design and Architecture

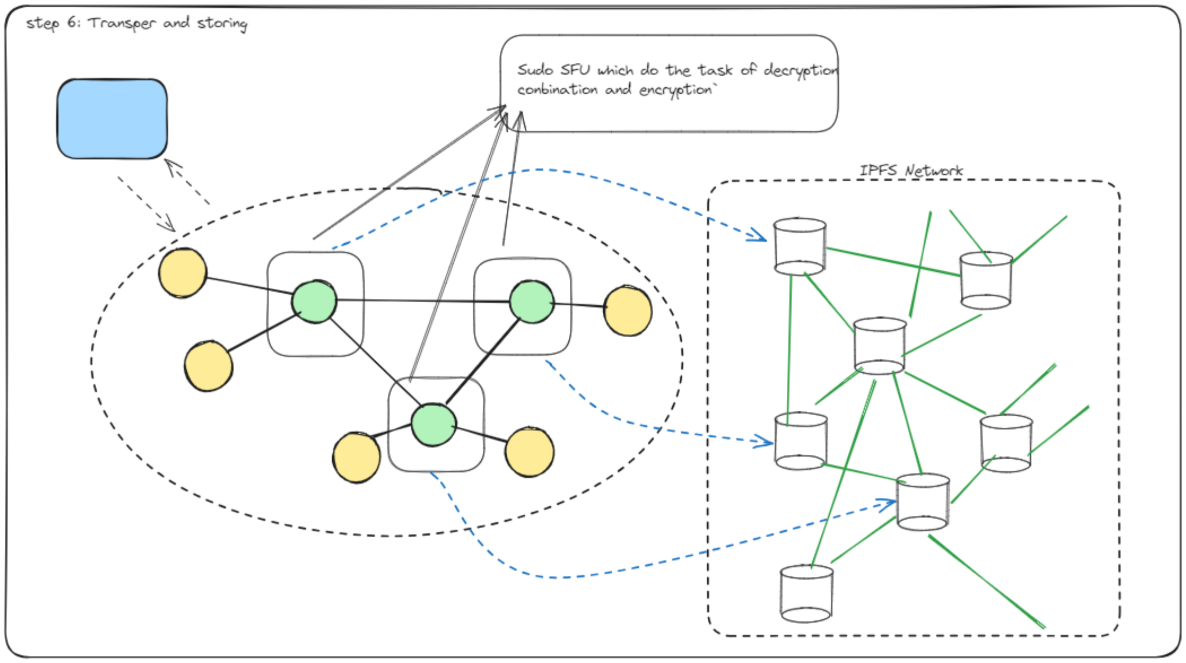


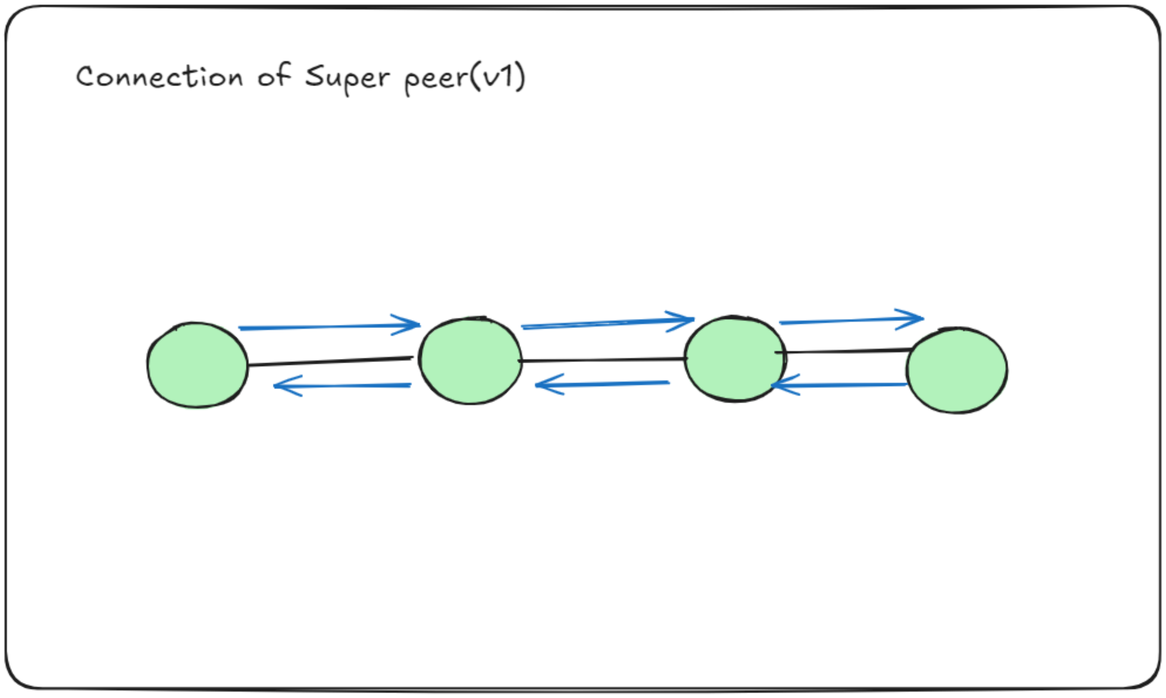


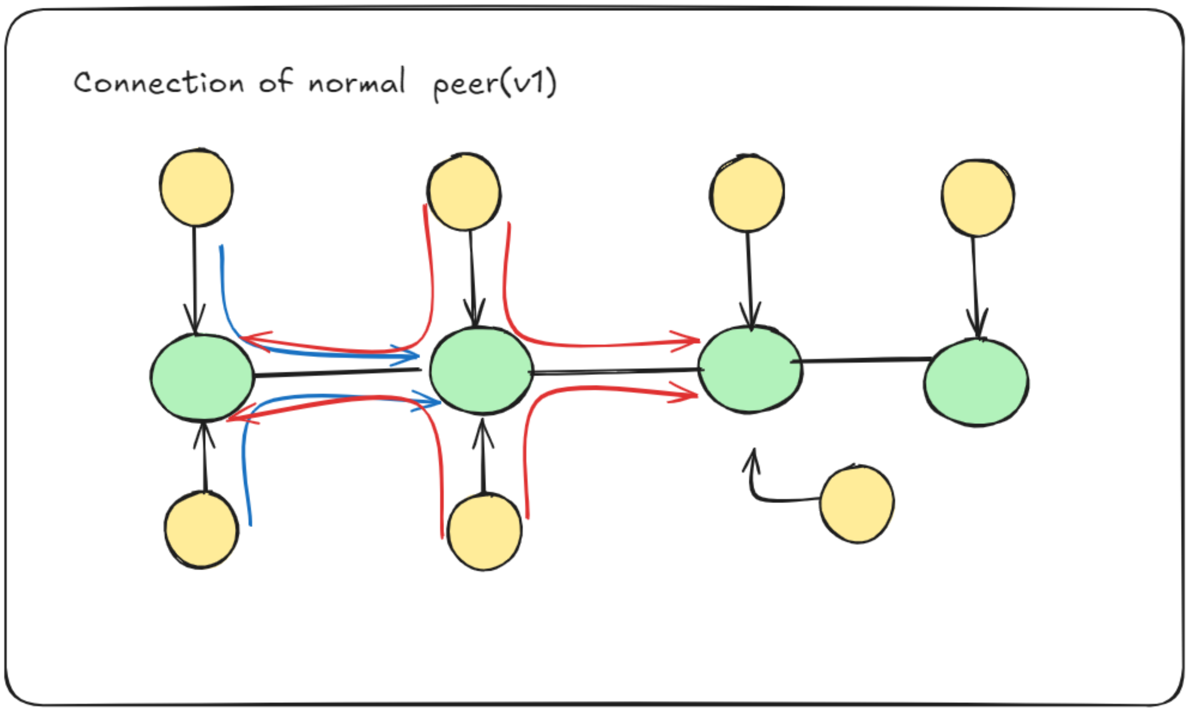

Step 3: Authentication of super peer(same wil be followed for normal peer)
Auth module
Ethereum


Step 4: Establishing the webrtc amount the super peer
Signaling module
Webrtc
Webrtc
Webrtc


Step 5h: Establishing connection with peers
Auth module
webrtc network


step 6: Transper and storing
Sudo SFU which do the task of decryption
conbination and encryption`
IPFS Network


Connection of Super peer(v1)


Connection of normal peer(v1)

10. Hardware and software Requirements

10.1 Hardware Requirements

Development Environment**:**

- Minimum: Intel i5 processor, 8 GB RAM, 256 GB SSD
- Recommended: Intel i7 or above, 16 GB RAM, 512 GB SSD

Deployment Environment**:**

- Server hosting with sufficient bandwidth to support signaling and data transfer for medium-sized meetings.

10.2 Software Requirements

- Front-End Development: React.js, Node.js, HTML, CSS
- Back-End Development: Express.js for the server, WebSocket library for signaling
- Database: MongoDB
- WebRTC: For peer-to-peer media communication
- Blockchain (Future Integration): Ethereum network, Solidity for smart contracts
- IPFS (Future Integration): For decentralized storage
- Development Tools: Visual Studio Code, Git, Postman (for API testing)

11. Reference

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.