Decentralized Video Conferencing Web-Application

PROJECT PHASE I REPORT

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering with specialization in Blockchain Technology

By

BALA R (Reg. No – 41613002)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)

CATEGORY - 1 UNIVERSITY BY UGC

Accredited “A++” by NAAC I Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600119

AUGUST - 2024

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Category - I University by UGC
Accredited "A++" by NAAC | Approved by AICTE
www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Phase1 report is the bonafide work of **Bala R (41613002)** whocarried out the Project entitled "**DECENTRALIZED VIDEO CONFERENCING WEB-APPLICATION"** under my supervision from June 2024 to December 2024.

Internal Guide

Dr.D.MENAKA, M.E., Ph.D.,

Head of the Department
Dr. A. MARY POSONIA, M.E., Ph.D.,

Submitted for Interdisciplinary Viva Voce Examination held on ____________________

Internal Examiner **External Examiner**

DECLARATION

I, **Bala R (Reg. No- 41613002),** hereby declare that the Project Phase1repoentitled "**DECENTRALIZED VIDEO CONFERENCING WEB-APPLICATION**" done by me under the guidance of **Dr.D.MENAKA, M.E., Ph.D.,** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering with Specialization in Blockchain Technology.**

DATE:

PLACE: Chennai **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **Sathyabama Institute of Science and Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph. D.**, **Dean**, School of Computing, and**, Dr. A. MARY POSONIA, M.E., Ph.D., Head of the Department** of Computer Science andEngineering with Specialization in Artificial intelligence for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. D.Menaka, M.E., Ph.D.,** for her valuable guidance, suggestions, and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

In today's interconnected world, video conferencing is essential for personal and professional communication, yet traditional systems relying on centralized servers face privacy, security, and single-point failure issues. This project introduces a decentralized video conferencing web application that leverages advanced technologies for secure, private, and efficient communication. By utilizing peer-to-peer (P2P) communication and Web Real-Time Communication (WebRTC), the application enables real-time audio, video, and data sharing directly between browsers, significantly reducing latency and enhancing user experience. A key feature is the integration of smart contracts, which automate and safeguard processes, eliminating third-party intermediaries and ensuring secure, transparent interactions. Developed using React.js and Next.js, the application boasts a robust framework for interactive user interfaces and enhanced performance through server- side rendering and static site generation. By embracing decentralization, this application provides a secure and private communication platform, addressing traditional systems' limitations and protecting user data. This innovative approach enhances privacy and security, setting a new benchmark for decentralized applications and paving the way for secure, efficient digital communication solutions.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In today's interconnected world, video conferencing has become essential for communication across various domains, from personal interactions to professional meetings. However, traditional video conferencing systems, which rely on centralized servers, pose significant challenges such as privacy concerns, security vulnerabilities, and potential single points of failure. To overcome these issues, a decentralized video conferencing web application offers a promising solution, leveraging advanced technologies to ensure secure, private, and efficient communication.

This project centers on the concept of peer-to-peer (P2P) communication, eliminating the need for central servers by establishing direct connections between users. By utilizing Web Real-Time Communication (WebRTC), the application enables real-time audio, video, and data sharing directly between browsers. WebRTC's capabilities significantly reduce latency and enhance the overall user experience, making it an ideal choice for this decentralized approach.

A key feature of the application is the use of smart contracts, which are self-executing contracts with the terms directly written into code. Smart contracts automate and safeguard various processes within the application, eliminating the reliance on third-party intermediaries and enhancing security by ensuring that all interactions are securely and transparently recorded.

The application is developed using cutting-edge web technologies, including React.js and Next.js. React.js offers a robust framework for building interactive and dynamic user interfaces, while Next.js enhances the performance and scalability of the application through server-side rendering and static site generation. These technologies together ensure a seamless and responsive user experience.

By embracing decentralization, this video conferencing web application provides a secure and private platform for users to communicate. It addresses the limitations of traditionalsystems, ensuring that user data remains protected and interactions are free from centralized control. This innovative approach not only enhances privacy and security but also sets a new benchmark for future decentralized applications, paving the way for more secure and efficient digital communication solutions.

CHAPTER 2

LITERATURE SURVEY

[1] Title : DecVi: Adaptive Video Conferencing on Open Peer-to-Peer Networks (2023)

Authors : Jingren Wei, Shaileshh Bojja Venkatakrishnan.

Video conferencing's shift to decentralized, blockchain-incentivized peer-to-peer networks counters centralization issues like data center proximity bias and privacy concerns. DecVi proposes a decentralized multicast tree protocol, adapting through an exploration-exploitation framework akin to multi-armed bandit problems. It achieves high reliability and flexibility without requiring global server coordination.

[2] Title : A REVIEW OF MYFRAMES –VIDEO CONFERENCING WEB APPLICATION USING WEBRTC [2021]

Authors : Ganesh Vishnu Parbat, Altaz Altaf Daruwala, Omkar Vinay Joshi, Aman Sanjay Singh, Prof. Dr. K. C. Nalavade.

This paper reviews video conferencing software facilitating visual communication across industries. Existing platforms suffer from connectivity issues affecting audio-visual quality and compatibility issues with browsers and operating systems. Security enhancements via WebRTC address these concerns using protocols like SRTP for encryption and authentication, and Opus codec for high-quality voice transmission.

[3] Title: Comparative Study of WebRTC Open Source SFUs for Video Conferencing [2018]

Authors :Emmanuel Andre Nicolas Le Breton∗§, Augustin Lemesle∗§, Ludovic Roux

and Alexandre Gouaillard .

WebRTC SFUs, crucial for Simulcast in WebRTC 1.0, are widely adopted for video conferencing and broadcasting. This paper introduces a novel approach using the KITE testing engine to compare five open-source WebRTC SFUs under load. The study reveals valuable insights into SFU scalability, previously unexplored in scientific research.

[4] Title: DeFake: Decentralized ENF-Consensus Based DeepFake Detection in Video Conferencing(2024).

Authors: Deeraj Nagothua, Ronghua Xua, Yu Chena, Erik Blaschb, Alexander Avedb

Current video conferencing lacks real-time media authenticity verification, leaving it vulnerable to deepfake and replay attacks. This paper proposes DeFake, a decentralized audio authentication using Electrical Network Frequency (ENF) signals. The Proof-of-ENF (PoENF) algorithm ensures robust deepfake detection in audio streams with minimal computational overhead, enhancing real-time security for video recordings.

[5] Title: DHT- and blockchain-based smart identification for video conferencing[2022].

Authors:Morteza Alizadeh, Karl Andersson, Olov Schelen

Video conferencing faces universal challenges in integrity, security, and authentication. This paper proposes a decentralized smart identification scheme using biometrics, machine learning, blockchain, and distributed hash tables. It enhances system storage and immutability, comparing architecture efficiencies with experimental results favoring blockchain and distributed hash tables despite longer execution times.

[6] Title: DIGIPARTY - A DECENTRALIZED MULTI-PARTY VIDEO CONFERENCING SYSTEM[2018]

Authors: Ling Chen, Chong Luo, hang Li, and Shipeng Li

Despite advancements since Microsoft NetMeeting, Internet video telephony remains primarily point-to-point. This paper introduces DigiParty, a fully distributed multi-party video conferencing system using a full mesh architecture and a novel conference control protocol. DigiParty aims to provide easy-to-use, global connectivity for families and friends via the Internet, compatible with various internet connections and instant messaging services.

[7] Title: End-to-End PQC Encryption Protocol for GPKI-based Video Conferencing System(2023).

Authors: Yeongjae Park 1, Hyeondo Yoo 1, Jieun Ryu 1, Young-Rak Choi 1, Ju-Sung Kang 2 and Yongjin Yeom.

This paper examines security issues in video conferencing, focusing on end-to-end encryption (E2EE) protocols. It compares Zoom and Secure Frame (SFrame) systems, then proposes an E2EE mechanism using post-quantum cryptography (PQC) Key Encapsulation Mechanism (KEM) for Government Public Key Infrastructure (GPKI)-based systems, enhancing security against future quantum threats.

[8] Title: Video-conference Communication Platform Based on WebRTC Online meetings [2020].

Authors: Jelena Caiko, Antons Patlins, Arapov Nurlan, Vladimir Protsenko

The shift to remote work during the pandemic has highlighted challenges in video conferencing, particularly for small to medium-sized businesses lacking adequate facilities.

This article aims to address these issues by designing and implementing Jitsi, an open-source video conferencing prototype, offering accessible meeting solutions without high costs or space constraints.

[9] Title: Analysis and Design of Decentralized Conferencing using Wi-Fi based on P2P Architecture[2014].

Authors: Aditya Trivedi, Niseant Chaubey

This paper explores audio/video calls and conferencing over Wi-Fi with minimal cost, proposing a decentralized approach to avoid single points of failure in client-server architectures. It introduces a peer-to-peer (P2P) model using a mobile number to IP address conversion algorithm, enabling Android-based mobile phones to facilitate WLAN-based audio/video calls and conferencing.

[10] Title: A Scalable based Multicast Model for P2P Conferencing Applications.MouradAMAD[2019].

Authors: Zahir HADDAD, Lachemi KHENOUS, KamalKABYL

This paper explores multicast conferencing efficiency and the benefits of Peer-to-Peer (P2P) models for scalability and fault tolerance. It proposes a novel approach combining SIP for call control, Chord for dynamic network organization, and application layer multicast. Performance evaluation shows improved traffic flow and transmission efficiency in conferencing applications.

2.1 Review on Existing System

Current video conferencing systems often rely on centralized architectures, leading to issues such as single points of failure, limited scalability, and privacy concerns. These centralized systems can be impractical and costly for small and medium-sized businesses, requiring large conference rooms and high rents. Traditional client-server models for video conferencing suffer from inefficiencies and centralized points of failure, while Wi-Fi-based audio/video calls using centralized databases necessitate user registration and face similar centralization issues. Multicast conferencing using separate links between participants results in high traffic and inefficiency. Although Peer-to-Peer (P2P) models offer scalability, robustness, and fault tolerance, they are often combined with centralized components for call control, limiting their effectiveness. This highlights the need for fully decentralized solutions that leverage P2P architectures and efficient multimedia distribution to overcome the limitations of current systems.

2.2 Inferences and Challenges in Existing System

Existing video conferencing systems often rely on centralized servers, which introduce privacy concerns, security vulnerabilities, and potential single points of failure. These systems can lead to data breaches and unreliable performance due to server overloads or outages. Despite advancements in technology, traditional systems still face significant challenges in ensuring secure and private communication. Users are often dependent on third-party intermediaries, which can compromise data integrity and transparency. Addressing these issues requires a shift to decentralized models, where peer-to-peer (P2P) communication and advanced technologies like WebRTC can mitigate these challenges by offering direct user connections, reduced latency, and enhanced security.

CHAPTER 3

REQUIREMENT ANALYSIS

The project requires a robust computing environment capable of handling real-time audio and video processing efficiently. It will utilize advanced web frameworks like React.js and Next.js for building the user interface and ensuring high performance. The WebRTC protocol is essential for enabling seamless real-time communication, while blockchain technology will be implemented for secure smart contracts. Adequate setup for a blockchain environment is needed to manage and execute these contracts effectively. Additionally, the system must be designed to be scalable and adaptable to various devices and network conditions to ensure consistent performance.

3.1 Necessity and Feasibility Analysis of Proposed System

1. **Privacy and Security**: Decentralization enhances user privacy by enabling direct peer-to-peer connections, minimizing reliance on third-party intermediaries. Additionally, robust cryptography ensures secure data transmission.
2. **WebRTC for Real-Time Communication**: Leveraging WebRTC allows seamless audio and video communication directly between browsers. This technology eliminates the need for central servers, resulting in lower latency and efficient real-time interactions.
3. **Zero Downtime**: By distributing the system across nodes, decentralized applications can achieve high availability and resilience. Even if some nodes go offline, the network remains operational, ensuring uninterrupted communication.
4. **Cryptography**: Smart contracts and user authentication within the system rely on cryptographic techniques. These ensure data integrity, prevent unauthorized access, and maintain trust in the decentralized environment.

3.2 Hardware and Software Requirements

Hardware Requirements:

1. Personal Computer or Laptop: A good-quality computer with a modern processor (e.g., Intel i5/Ryzen 5 or better), at least 8 GB of RAM, and sufficient storage (SSD recommended) to handle development and testing tasks.
2. Internet Connection: Reliable high-speed internet is essential for testing real-time communication features and ensuring smooth video and audio streaming.
3. Web Camera and Microphone: Basic hardware for testing video and audio functionalities.

Software Requirements:

1. Operating System: Any modern OS like Windows, macOS, or Linux.
2. Development Tools:
   Code Editors: Free options like Visual Studio Code or Atom.
   Version Control: Git for source code management, with GitHub or GitLab for hosting repositories.
3. Web Frameworks:
   React.js: For building the user interface.
   Next.js: For server-side rendering and static site generation.
   WebRTC: Open-source technology for real-time communication.
   Blockchain Environment:
   Ethereum Testnet: For smart contract development and testing.
   Solidity: For writing smart contracts.
4. Local Development Servers: Tools like Docker (free version) for containerization and environment management.
   Database: Free options like SQLite or PostgreSQL for managing application data.

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

The proposed system is a decentralized video conferencing web application designed to enhance privacy, security, and efficiency in online communication. Utilizing peer-to-peer (P2P) technology, it establishes direct connections between users, eliminating the need for central servers and thereby mitigating privacy and security issues associated with traditional systems. The application employs Web Real-Time Communication (WebRTC) to enable real-time audio, video, and data sharing directly between browsers, reducing latency and improving user experience.

Additionally, smart contracts are integrated to automate and secure various processes, ensuring transparent and reliable interactions. Developed using React.js and Next.js, the system provides a robust and dynamic user interface while maintaining high performance and scalability. This innovative approach not only addresses the limitations of centralized video conferencing solutions but also sets a new standard for secure and efficient digital communication.

4.1 Selected Methodologies

1. **Peer-to-Peer (P2P) Communication**: This methodology eliminates the need for central servers by establishing direct connections between users, enhancing privacy and security while reducing latency in communication.
2. **Web Real-Time Communication (WebRTC**): Used for real-time audio, video, and data sharing directly between browsers, WebRTC provides a robust framework for seamless and low-latency communication without relying on central servers.
3. **Smart Contracts**: Integrated into the system to automate and secure processes, smart contracts ensure that interactions are transparent and tamper-proof, reducing the need for third-party intermediaries.
4. **React.js and Next.js**: These frameworks are employed to build a dynamic and responsive user interface. React.js handles the front-end development, while Next.js supports server- side rendering and static site generation to improve performance and scalability.
5. **Blockchain Technology**: Utilized for managing smart contracts and ensuring secure, decentralized data management. Blockchain provides a transparent and immutable record of transactions and interactions within the application.

4.2 Architecture Diagram

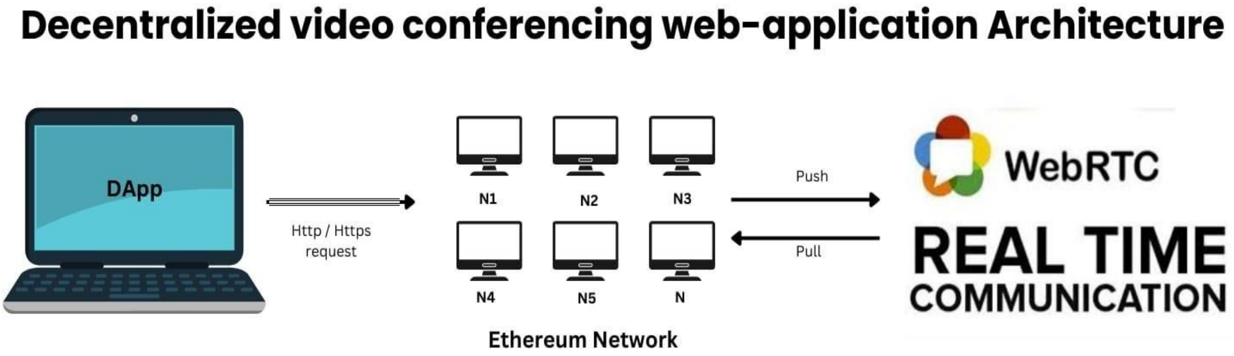

Fig 4.1:Decentralized video conferencing web-application architecture

4.3 Detailed Description of Modules and Workflow

- **User Authentication Module:**

  Description: This module handles user registration, login, and verification processes. It ensures secure access to the application by implementing authentication mechanisms.

  Workflow: Users register and authenticate via a secure login interfac e. Credentials are verified against a decentralized ledger using blockchain technology, ensuring privacy and integrity.

- **Peer-to-Peer Connection Module:**

  Description: Establishes direct connections between users for real-time communication without relying on central servers.

  Workflow: Upon initiating a video call, the system uses WebRTC to negotiate and establish a P2P connection. The signaling process, handled by the application, facilitates the exchange of connection details between peers.

- **Real-Time Communication Module:**

  Description: Manages the streaming of audio, video, and data between users.

  Workflow: Once the P2P connection is established, WebRTC handles the real-time transmission of media and data. The module ensures synchronization and high-quality transmission by continuously monitoring and adjusting for network conditions.

- **Smart Contracts Module:**

  Description: Automates and secures various processes within the application, such as user interactions and transaction records.

  Workflow: Smart contracts, deployed on a blockchain network, execute predefined actions based on contract terms. For instance, they manage access permissions and enforce rules for interactions, ensuring transparency and reducing reliance on intermediaries.

- **User Interface Module:**

  Description: Provides the front-end interface for users to interact with the application.

  Workflow: Developed using React.js and Next.js, this module presents a dynamic and responsive interface. Users can initiate or join video calls, access settings, and manage their profiles. The interface interacts with back-end services and displays real-time updates.

- **Blockchain Management Modules:**

  Description: Oversees the integration of blockchain technology for managing smart contracts and ensuring data integrity.

  Workflow: This module interacts with the blockchain to deploy, execute, and verify smart contracts. It also handles the storage of transaction records and ensures that all interactions are securely recorded on the decentralized ledger.

Overall Workflow:

1. Users authenticate and register through the User Authentication Module.
2. They initiate a video call, which sets up a P2P connection using the Peer-to-Peer Connection Module.
3. Real-time communication is managed by the Real-Time Communication Module via WebRTC.
4. Smart Contracts Module automates and secures interactions, while the Blockchain Management Module maintains a decentralized record.
5. Users interact with the application through the User Interface Module, which dynamically updates based on real-time data and interactions.

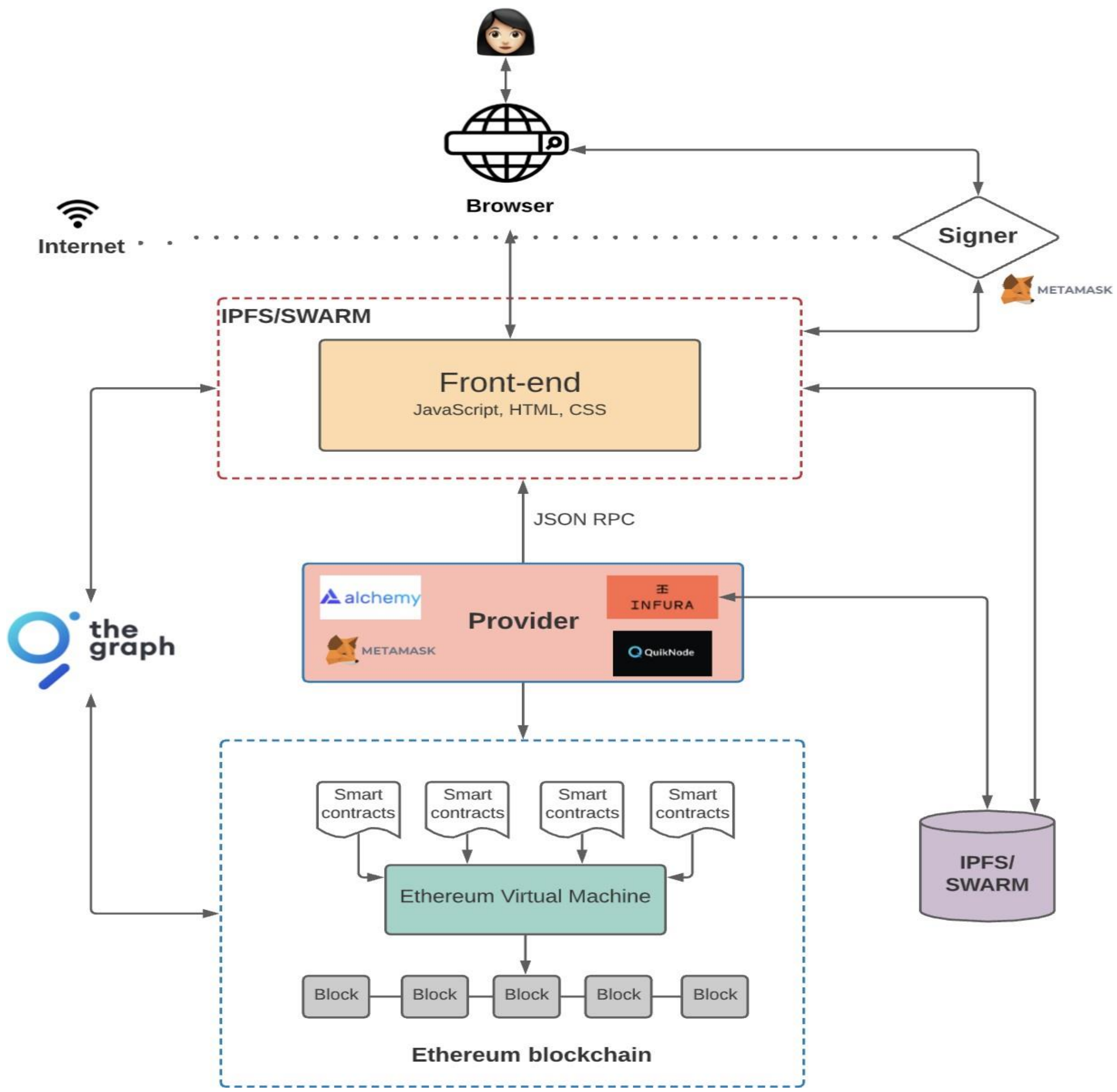

Fig:4.2 workflow diagram

4.4 Estimated Cost for Implementation and Overheads

Development Costs: 0

Tools and Software: Most development tools and libraries are open-source, so costs here are minimal. Expected cost: ₹0.

Blockchain Deployment: 4000 to 5000

Blockchain Platform Fees: Deploying smart contracts on the mainnet will incur transaction (gas) fees. Depending on network congestion, these could range from ₹4,000 to ₹5,000.

Miscellaneous Costs: 1000(Domain and SSL Certificate)

Total - 6000 to 10000 rupee

CHAPTER 5

CONCLUSION

The proposed decentralized video conferencing web application presents a significant advancement in virtual communication technology by addressing critical issues associated with traditional centralized systems. By leveraging peer-to-peer (P2P) communication, the application enhances user privacy, reduces latency, and eliminates single points of failure. The use of WebRTC facilitates direct, real-time audio, video, and data sharing between browsers, improving the overall user experience. Additionally, integrating smart contracts automates and secures various processes, ensuring transparency and minimizing reliance on third-party intermediaries. This approach not only strengthens security but also supports scalable and robust performance. The application's innovative design addresses the shortcomings of existing solutions, providing a more reliable and efficient platform for virtual meetings. By offering a secure, private, and high-performance alternative, this project sets a new standard for future video conferencing systems, making it a valuable contribution to the field of digital communication.

REFERENCES

1. Aditya Trivedi, Niseant Chaubey [2014]. Analysis and Design of Decentralized Conferencing using Wi-Fi based on P2P Architecture. IEEE Xplore.
2. Deeraj Nagothua, Ronghua Xua, Yu Chena, Erik Blaschb, Alexander Avedb (2024). DeFake: Decentralized ENF-Consensus Based DeepFake Detection in Video Conferencing. IEEE.
3. Emmanuel Andre ́, Nicolas Le Breton∗§, Augustin Lemesle∗§, Ludovic Roux and Alexandre Gouaillard [2018]. Comparative Study of WebRTC Open Source SFUs for Video Conferencing. IEEE.
4. Ganesh Vishnu Parbat, Altaz Altaf Daruwala, Omkar Vinay Joshi, Aman Sanjay Singh, Prof. Dr. K. C. Nalavade [2021]. A REVIEW OF MYFRAMES -VIDEO CONFERENCING WEB APPLICATION USING WEBRTC. International Journal of Creative Research Thoughts (IJCRT).
5. Jelena Caiko, Antons Patlins, Arapov Nurlan, Vladimir Protsenko [2020]. Video-conference Communication Platform Based on WebRTC Online meetings. Sciencedirect.
6. Jingren Wei, Shaileshh Bojja Venkatakrishnan. (2023). DecVi: Adaptive Video Conferencing on Open Peer-to-Peer Networks. ACM Digital Library.
7. Karl Andersson, Olov Schelen, Morteza Alizadeh [2022]. DHT- and blockchain-based smart identification for video conferencing. Sciencedirect.
8. Ling Chen, Chong Luo, Hang Li, and Shipeng Li [2018]. DIGIPARTY - A DECENTRALIZED MULTI-PARTY VIDEO CONFERENCING SYSTEM. IEEE International Conference on Multimedia and Expo (ICME).
9. Mourad AMAD, Zahir HADDAD, Lachemi KHENOUS, Kamal KABYL [2019]. A Scalable based Multicast Model for P2P Conferencing Applications. IEEE Xplore.
10. Yeongjae Park, Hyeondo Yoo, Jieun Ryu, Young-Rak Choi, Ju-Sung Kang, Yongjin Yeom (2023). End-to-End PQC Encryption Protocol for GPKI-based Video Conferencing System. IEEE.