Project Synopsis

on

ENHANCED SECURITY AND BLOCKCHAIN BASED ONLINE MEETING PLATFORM

Submitted as a part of course curriculum for

Bachelor of Technology
in
Computer Science

Submitted by
Kishan Agrawal (2200290120089)
Rishika Agarwal (2200290120135)
Shubham Singh (2200290120170)
Yashasvi Saxena (2200290120199)

Under the Supervision of
Dr. Harsh Khatter
Associate Professor
Department of Computer Science

KIET Group of Institutions, Ghaziabad
Department of Computer Science
Dr. A.P.J. Abdul Kalam Technical University
2024-2025

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the synopsis of the B. Tech Mini Project undertaken during B.Tech. Third Year. We owe a special debt of gratitude to **Dr. Harsh Khatter**, Associate Professor, Department of Computer Science, KIET Group of Institutions, Delhi- NCR, Ghaziabad, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen the light of the day.

We also take the opportunity to acknowledge the contribution of **Dr. Ajay Kumar Shrivastava**, Head of the Department of Computer Science, KIET Group of Institutions, Delhi- NCR, Ghaziabad, for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project.

Last but not the least, we acknowledge our friends for their contribution to the completion of the project.

Signature:

Student's Name: Kishan Agrawal
Roll No: 2200290120089

Signature:

Student's Name: Rishika Agarwal
Roll No: 2200290120135

Signature:

Student's Name: Shubham Singh
Roll No: 2200290120170

Signature:

Student's Name: Yashasvi Saxena
Roll No: 2200290120199

Guide Name & Signature

Dr. Harsh Khatter
Associate Professor
Department of Computer Science
KIET Group of Institutions

Signature
Project coordinator

ABSTRACT

The online meeting system is designed to secure confidential communications between various stakeholders of institutes, organizations, members, and officials. The solution aims to enhance privacy by utilizing both web-based and blockchain-based technologies. Real-time video and audio communication will be enabled ensuring end-to-end encryption. Meeting scheduling, file sharing, and role-based access control will be integrated to provide a comprehensive meeting platform.

For enhanced security, blockchain-based identity verification will be used for secure authentication, and decentralized storage will protect sensitive data and meeting recordings. Smart contracts will automate key processes such as meeting scheduling, access control, and logging, ensuring accountability and transparency. The backend infrastructure will manage user accounts, meeting functionalities, and logs. The frontend will be developed using modern technologies for a seamless user interface.

Key features include real-time communication, secure file sharing, and role-based access to ensure that participants can join, manage, and access meeting resources securely. The system will integrate public APIs for streamlined scheduling. Additional functionalities such as voting systems, screen sharing, chat, and meeting transcription will enhance user engagement.

Robust security measures, including end-to-end encryption of communication and decentralized storage, will protect data in transit and at rest. Blockchain ensures tamper-proof records and identity verification, making the system scalable, secure, and adaptable to needs for confidential, secure, and user-friendly online meetings.

TABLE OF CONTENTS

Page No.

CHAPTER 1 - INTRODUCTION

In today’s digital age, online communication has become an essential part of personal, educational, and professional interactions. With the increasing reliance on virtual platforms for meetings, discussions, and collaborations, the need for secure and efficient systems has grown significantly. Ensuring privacy, protecting data, and enabling seamless communication across various sectors have become paramount concerns. As organizations and individuals share more sensitive information through digital platforms, safeguarding these interactions against cyber threats and unauthorized access is crucial.

Technological advancements such as blockchain, decentralized storage, and end-to-end encryption have provided innovative solutions to these challenges. Blockchain, for instance, offers a decentralized and tamper-proof system for recording information, ensuring transparency and security. Its applications go beyond financial transactions, extending into sectors like healthcare, education, and governance, where data integrity and confidentiality are critical. Similarly, decentralized storage solutions prevent data breaches by distributing files across multiple nodes, making it more difficult for malicious actors to compromise entire systems.

These technological tools, when integrated into communication platforms, can enhance user trust, streamline operations, and protect sensitive data. Whether for government meetings, corporate conferences, or educational sessions, the demand for secure, scalable, and efficient virtual meeting systems is evident. These platforms not only offer real-time communication but also provide advanced features like role-based access control, encrypted file sharing, and seamless scheduling—all of which contribute to a more secure and reliable online environment.

CHAPTER 2 - LITERATURE REVIEW

The literature review includes insights from 3 whitepapers and 7 research papers from diverse sources. These studies were instrumental in selecting the appropriate technologies for implementing various features in the project, as well as understanding the rationale behind those choices.

1.Livepeer Project [1]

In “Livepeer Project” they discuss a decentralized live streaming video protocol based on blockchain using “Ethereum.” The Livepeer project aims to deliver a live video streaming network protocol that is fully decentralized, highly scalable, crypto token incentivized, and results in a solution which is cheaper to an app developer or broadcaster than using traditional centralized live video solutions. Decentralized applications (DApps) to be built in the form of static or infrequently updated web or mobile content, but now DApps still lack the ability to include streaming media and data in an open. and a decentralized way. As the “Livepeer Project” claims, the goal of their project is to decentralize live video broadcast over the internet. Although it is a secure platform based on the International Journal of Computer Applications (0975 – 8887) Volume 183 – No. 16, July 2021 21 blockchain technology, “Livepeer Project” has not suggested any mechanisms to auditability and intrusion detection [1].

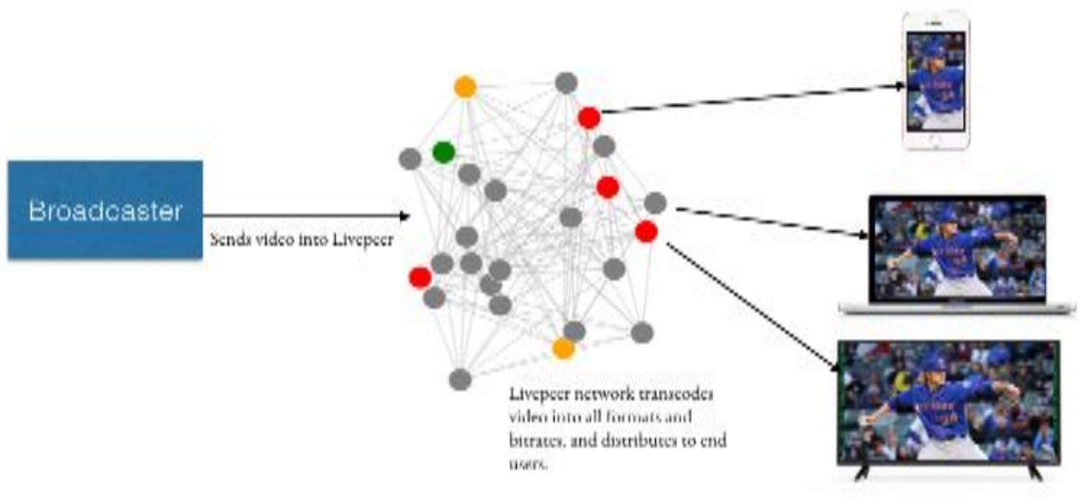
Broadcaster
Sends video into Livepeer
Livepeer network transcodes video into all formats and bitrates, and distributes to end users.

Fig 2.1 A basic working of livepeer project [1]

2.SteemQ [2]

A decentralized platform for STEEM SteemQ is a still on-going project aligned with “Steemit” decentralized social network which is based on blockchain technology. SteemQ is proposed to be a decentralized video platform for user- generated content based next-generation platforms on top of the new Blockchain and P2P technologies. The blockchain of choice is STEEM. This allows the developers to build on top of the same technology that powers Steemit, as well as inherit the benefits of an existing community, currency, and platform. All STEEM social network accounts are automatically SteemQ accounts and vice-versa. The system aims to empower its users to the maximum extent possible while remaining resilient. SteemQ says that it uses IPFS as a core building block of the content distribution system of their prototype. IPFS is a great tool that does a few things well. It provides a robust layer for managing, transporting, referencing, deduplicating, versioning and ensuring the integrity of the content. SteemQ suppose that they can secure the multi hashes by storing them on the immutable blockchain (i.e., STEEM's posts become immutable after the first reward payout. STEEM's transfers are much faster, being immutable and permanent within seconds, when the block is confirmed). This way they suggest that they can simultaneously guarantee the ownership and integrity of the content [2].

3.Prov Chain [3]

A Blockchain-based Data Provenance Architecture in Cloud Environment with Enhanced Privacy and Availability In this paper, researchers have presented a concept called "ProvChain," a blockchain based data provenance architecture to provide assurance of data operations in a cloud storage application, while enhancing privacy and availability at the same time. ProvChain uses the construction of the merkle tree technology for the provenance of data. A list of blockchain transactions will be used to form a block and the block needs to be confirmed by a set of nodes to be included in the blockchain. An attempt to modify a provenance data record will require an adversary to locate the transaction and the block. Blockchain's underlying cryptographic theory will allow modifying a block record only if the adversary can present a longer chain of blocks than the rest of miners' blockchain, which is quite difficult to achieve [3]. As the "ProvChain" claims, the goal of their project is to improve the provenance of data in the IoT based cloud environments. Although it is a secure platform based on the blockchain technology, "ProvChain" has not suggested any mechanisms to auditability and intrusion detection.[3]

4.Blockchain-Based, Decentralized Access Control for IPFS [4]

The weaknesses of blockchain can be overcome by moving the files containing personal data off-chain. This ensures files can be deleted as required and are not causing the chain to grow excessively fast. However, for data sharing between organizations the assurance that files have not been edited or changed must be retained. Moreover, the blockchain tracks which entities have access to which data. IPFS stores files in a distributed way by splitting them into chunks which can be requested and transferred between nodes. Each file is identified by its cryptographic hash. Doing so makes it easy to ensure one has received the correct data by generating the hash of the concatenated file chunks and checking if it matches the hash that was requested. IPFS hashes identifying files that contain personal data can be stored on the blockchain instead of the data itself. Doing so enables compliance with GDPR legislation. The blockchain containing the hash ensures that the file has not been tampered with. The file itself being stored in IPFS means it can be deleted as required, by nodes removing it from their local storage. The hashes of files can be used to associate files with owners and access permissions. Chain growth is reduced as hashes are smaller than the data they represent, if SHA256 is used the on-chain storage required for a file of any size becomes 32 bytes. As such chain growth is vastly reduced. Enforcing Privacy: IPFS is designed to share information as widely as possible and does not attempt to restrict connections or data flow between nodes. The scenario considered herein deals with confidential data. IPFS cannot be allowed to share files with any nodes that request them, it must only share files with those that have permission according to the permissions recorded on the blockchain. As such, the provided solution is a modified version of IPFS that uses a smart contract which can add, removing and updating file ownership and accesses. Entities are identified by their Ethereum public key and files are identified by their unique cryptographic hash. The combination of blockchain assurance and a private network capable IPFS allows for compliance with current data protection and sharing legislation and reduces the cost of doing so.

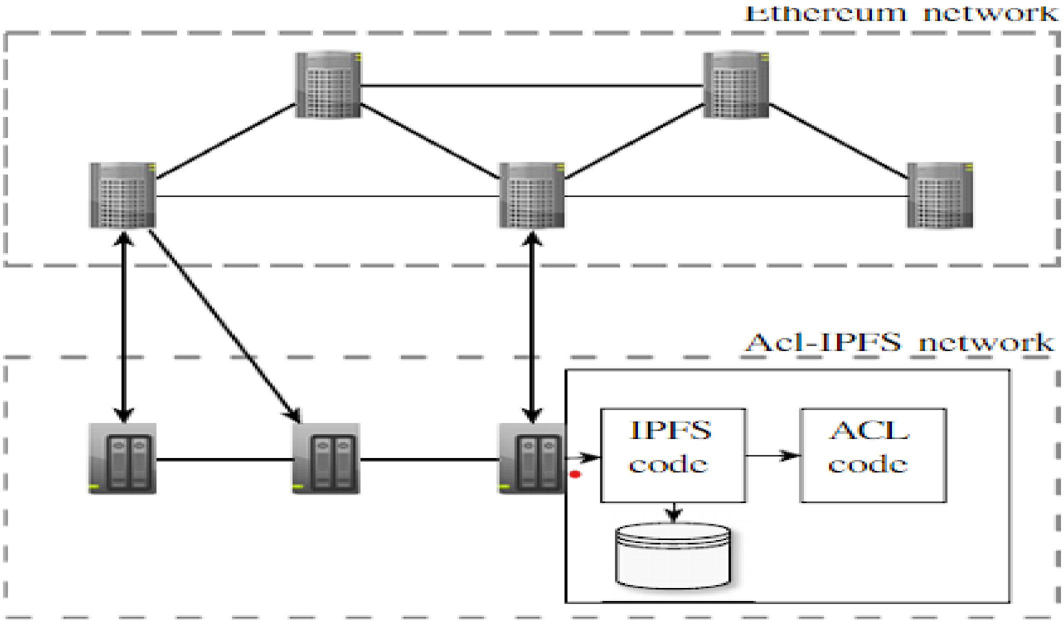

Fig 2.2 The architecture of the ACL-IPFS network [4]

5.Performance Analysis of High-Definition Video Call Over Secure Real Transport Protocol (SRTP) [5]

Several studies have analyzed the effects of implementing security protocols like SRTP on voice and video call performance. Ismail and Yahya conducted experiments comparing HD video calls over secure (using SRTP) and non-secure channels in both wired and wireless LAN environments. They measured key performance indicators including jitter, Mean Opinion Score (MOS), and R-Factor. Their results showed that while the secure channel had slightly higher jitter and lower MOS/R-Factor scores, the differences were minimal and call quality remained acceptable when using SRTP encryption.

These findings align with earlier work by Alexander et al. (2009) [6] who evaluated SRTP's impact specifically on VoIP performance. They found that SRTP added only "negligible overhead" and did not significantly affect VoIP quality metrics like packet inter-arrival time and jitter. Similarly, Adomkus and Kalvaitis (2008) [7] concluded that while SRTP could slightly degrade some VoIP performance parameters, its use was still "necessary" for voice encryption.

However, other researchers have observed more noticeable performance impacts from implementing call security. Sureshkumar and Dutta (2010) [8] measured increases in call setup time, memory utilization, and queue size when enabling security features for

VoIP calls. Their work suggests that additional processing required for encryption/decryption can impact system resources and call initiation.

6.P2P Media Streaming with HTML5 and WebRTC [9]

The potential of WebRTC to support P2P streaming lies in its ability to establish direct communication between browsers using UDP-based protocols. This real-time communication can enable browsers to function as both content consumers and distributors, reducing the dependency on central servers. The application of WebRTC in video streaming, particularly VoD, could decentralize media distribution, thereby reducing infrastructure costs for service providers.

One major performance bottleneck is the computational load associated with cryptographic hashing, such as the generation of MD5 hashes used to verify video file integrity. These operations consume significant CPU resources, which can slow down content delivery and affect the user experience, especially on devices with limited processing power. Performance comparisons between browser versions show a tenfold difference in handling MD5 computations. Improvements in JavaScript implementations and potential standardization of more efficient hashing algorithms could address these issues.

The feasibility of implementing a P2P VoD service using HTML5 and WebRTC has been partially demonstrated through experimental setups. Initial results suggest that handling video streams on desktop and near-future mobile devices is achievable. Still, significant optimization is required to ensure seamless performance across various platforms. HTML5's built-in video elements, combined with the Offline Application Caching API, offer some flexibility by allowing videos to be stored and played even when the user is offline.

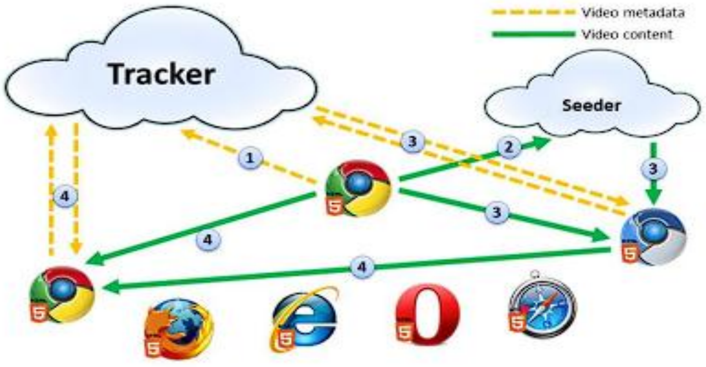

Fig 2.3 Network Architecture for the P2P VoD service [9]

7.Development of a secure video chat based on the WebRTC standard for video conferencing [10]

WebRTC is a standardized technology that enables real-time peer-to-peer communication, allowing browsers to exchange audio, video, and data without the need for additional plugins. Its primary advantage lies in the fact that it ensures security by using protocols such as Secure Real-Time Transport Protocol (SRTP) and Datagram Transport Layer Security (DTLS) for encryption and authentication. These protocols play an essential role in protecting data transmissions, providing privacy, and preventing replay attacks.

The literature highlights the growing relevance of WebRTC in video conferencing applications due to its ability to turn browsers into fully-fledged video communication terminals. This has made WebRTC a crucial tool for modern communication, particularly because of its cross-platform nature and browser support, which eliminates the need for installing extra software.

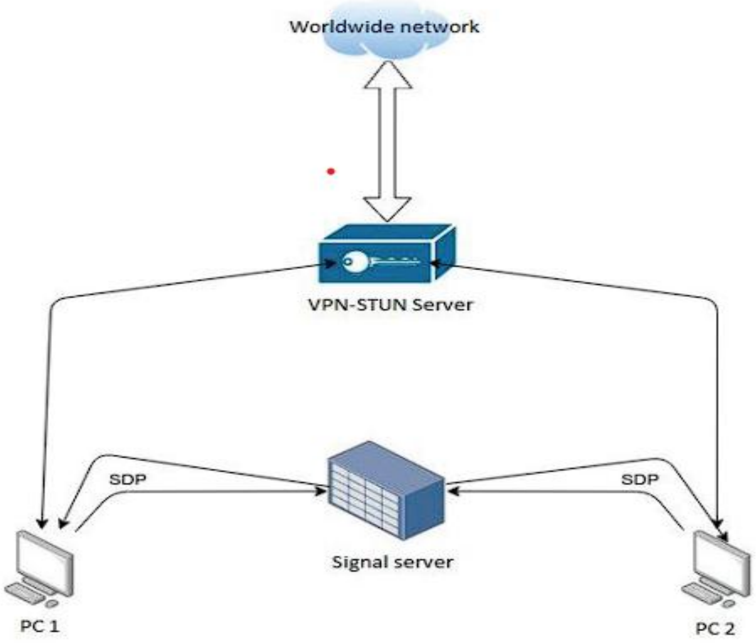

Fig 2.4 Software product architecture (video chat) [10]

8.Secure High-Definition Video Conferencing [11]

The Secure Real-Time Transport Protocol (SRTP) is a profile of RTP designed specifically to address security concerns in real-time media transmission. SRTP employs AES (Advanced Encryption Standard) for encrypting media payloads, ensuring confidentiality without compromising performance. Additionally, it uses Hash-based Message Authentication Codes (HMAC) to safeguard the integrity of RTP packets.

However, SRTP requires a method for exchanging cryptographic keys, which is where the Multimedia Internet Keying (MIKEY) protocol comes into play. MIKEY allows secure key exchange during the media negotiation process, and its integration with SIP makes it an ideal solution for securing video conferencing sessions. MIKEY supports multiple authentication mechanisms, including pre-shared keys, public key encryption, and Diffie-Hellman key exchange, offering flexibility depending on the level of security required.

9.Blockchain-Based E-Voting System [12]

Blockchain's potential for e-voting lies in its ability to ensure:

- **Transparency**: Every vote can be recorded on a public ledger, which is immutable and verifiable.
- **Immutability**: Once votes are cast and added to the blockchain, they cannot be changed or deleted.
- **Security**: Blockchain's decentralized nature reduces the risk of hacking or tampering, as an attacker would need to compromise most nodes in the network.

The authors of the paper highlight several requirements for a blockchain-based e-voting system:

- **Voter Privacy**: The system must prevent any third party from linking a vote to the voter's identity. Technologies such as Zero-Knowledge Proofs (ZKPs) are often proposed as a solution to achieve this.
- **Vote Verification**: Voters should be able to verify that their votes have been recorded correctly without revealing their identity.
- **Coercion Resistance**: The system must protect voters from being forced to vote in a certain way by an external party.
- **Decentralization**: No single entity should control the system; instead, multiple independent nodes should participate in verifying and recording votes.

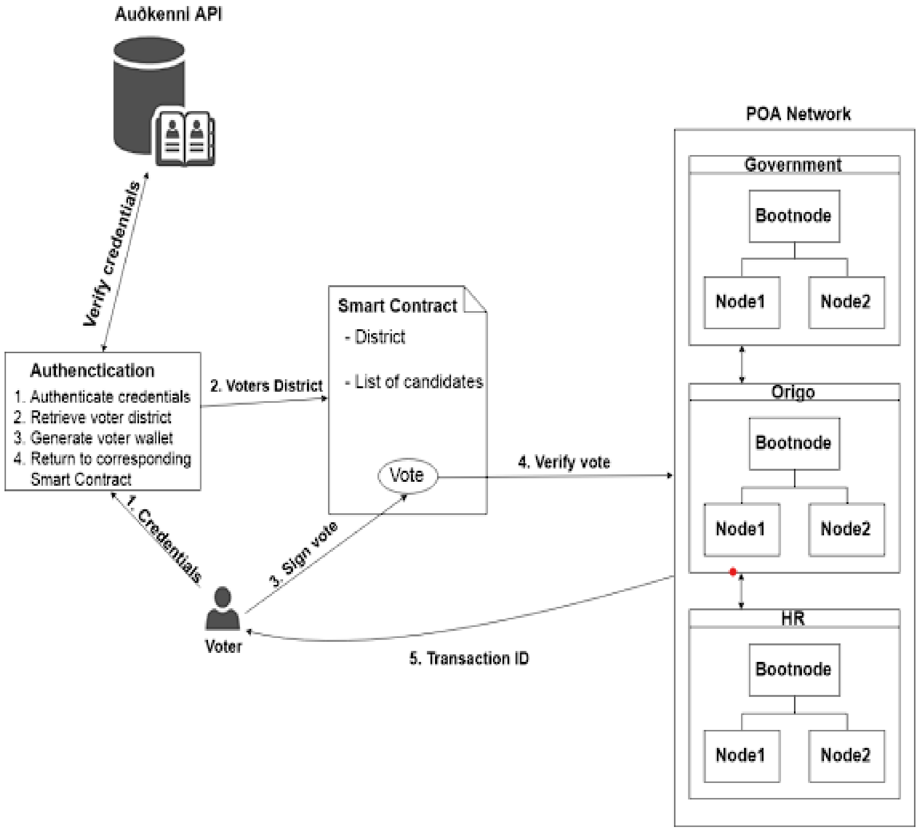

Fig 2.5 Voter authenticates himself [12]

10. Blockchain Private File Storage-Sharing Method Based on IPFS [13]

IPFS is a peer-to-peer file system that stores and shares files in a distributed manner. It uses content-based addressing to locate files based on their unique hash value rather than their physical location. This method increases data availability and reduces redundancy. The integration of blockchain with IPFS enhances security by recording file metadata, including file hash and ownership, on the blockchain while storing the actual file on IPFS. This hybrid model ensures that the file content remains immutable and easily traceable without overburdening the blockchain with substantial amounts of data.

Blockchain-based decentralized file storage solutions, such as Storj and Filecoin, utilize IPFS for file storage and blockchain for maintaining metadata and incentives for storage providers. These systems demonstrate how blockchain can address issues of file integrity and ownership while ensuring scalability and accessibility.

Summary:

SN	Title	Author	Description
1.	**Livepeer Project [1]**	D. Petkanics	The Livepeer Project aims to create a decentralized, cost-effective live video streaming protocol on Ethereum, but lacks security measures like auditability and intrusion detection.
	Link: https://github.com/livepeer/wiki/blob/master/WHITEPAPER.md		
2.	**SteemQ [2]**	Furion	SteemQ is a decentralized video platform built on the STEEM blockchain, designed for user-generated content, utilizing IPFS for content distribution while ensuring ownership and integrity through immutability.
	Link: https://steemit.com/steemq/@furion/steemq-a-decentralized-video-platform-for-steem		
3.	**Prov Chain [3]**	Sachin Shetty, Val Red, Charles Kamhoua, Kevin Kwiat, Laurent Njilla	ProvChain is a blockchain-based data provenance architecture designed for cloud environments, utilizing Merkle tree technology to enhance data assurance, privacy, and availability, but lacks mechanisms for auditability and intrusion detection.

	Link: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/10206/1/Data-provenance-assurance-in-the-cloud-using-blockchain/10.1117/12.2266994.short#_=_		
4.	**Blockchain-Based, Decentralized Access Control for IPFS [4]**	Mathis Steichen, Beltran Borja Fiz Pontiveros, Robert Norvill, Wazen Shbair	The proposed blockchain-based decentralized access control for IPFS enables secure file sharing and compliance with GDPR by storing cryptographic hashes on the blockchain while managing file ownership and permissions through smart contracts, thus enhancing privacy, and reducing chain growth.
	Link: https://www.researchgate.net/publication/327034734_Blockchain-Based_Decentralized_Access_Control_for_IPFS		
5.	Performance Analysis of High-Definition Video Call Over Secure Real Transport Protocol (SRTP) [5]	Nor Rahimah Ismail, Saadiah Yahya	A performance analysis of high-definition video calls over Secure Real-time Transport Protocol (SRTP) shows that while SRTP introduces slight increases in jitter and lower Mean Opinion Scores, the overall call quality remains acceptable, supporting the necessity of encryption despite minor performance impacts.
	Link: https://myjms.mohe.gov.my/index.php/dismath/article/view/13703		
6.	**P2P Media Streaming with HTML5 and WebRTC [9]**	Jukka K. Nurminen, Antony J.R. Meyn, Eetu Jalonen, Yrjo Rajvio, Raul Garcia Marrero	P2P media streaming with HTML5 and WebRTC leverages direct browser communication to decentralize video distribution, reducing infrastructure costs, though performance is hindered by the computational load of cryptographic hashing; optimization and efficient algorithms are needed for seamless user experiences across devices.
	Link: https://ieeexplore.ieee.org/document/6970739?denied=		

7.	**Development of a secure video chat based on the WebRTC standard for video conferencing [10]**	Elena Revyakina	The development of secure video chat using WebRTC enhances real-time peer-to-peer communication by utilizing encryption protocols like SRTP and DTLS, ensuring data privacy and security, and establishing WebRTC as a vital tool for modern cross-platform video conferencing.
	Link: https://www.e3s-conferences.org/articles/e3sconf/abs/2023/26/e3sconf_uesf2023_07017/e3sconf_uesf2023_07017.html		
8.	**Secure High-Definition Video Conferencing [11]**	G. C. Añón	Secure high-definition video conferencing utilizes the Secure Real-Time Transport Protocol (SRTP) for media encryption and integrity protection, paired with the Multimedia Internet Keying (MIKEY) protocol for secure key exchange, ensuring confidentiality and performance during video sessions.
	Link: https://upcommons.upc.edu/handle/2099.1/5334		
9.	**Blockchain-Based E-Voting System [12]**	Friðrik Þ. Hjálmarsson, Gunnlaugur K. Hreiðarsson, Mohammad Hamdaqa, Gísli Hjálmtýsson	A blockchain-based e-voting system ensures transparency, immutability, and security by recording votes on a public ledger, while addressing key requirements such as voter privacy, vote verification, coercion resistance, and decentralization using technologies like Zero-Knowledge Proofs.
	Link: https://ieeexplore.ieee.org/document/8457919/authors#authors		
10.	**Blockchain Private File Storage-Sharing Method Based on IFPS [13]**	Peng Kang, Wenzhong Yang, Jiong Zheng	The blockchain-based file storage method using IPFS ensures secure, immutable metadata and enhances data availability by storing files in a distributed manner without overloading the blockchain.
	Link: https://pubmed.ncbi.nlm.nih.gov/35890780/		

CHAPTER 3 – PROBLEM STATEMENT

Many online meetings are conducted to various stakeholders of institutes, organizations, members, and officials. Lot of confidential data has been shared through these online meetings. To increase security and to make a robust system a personalized online meeting portal is needed.

- Based on the requirements, design the architecture of the online meeting system. Determining the components, such as the server infrastructure, database, APIs, and user interfaces, which will be needed to develop the system.
- Develop the backend infrastructure: Build the backend infrastructure that will handle the core functionalities of the online meeting system. This typically includes user management, meeting scheduling, real-time communication, file sharing.
- Implement video conferencing capabilities: Integrate video conferencing functionality into the system. Web Real-Time Communication technology to facilitate real-time video and audio communication between participants.
- Create a user-friendly interface for participants to join meetings, manage settings, access recordings, and utilize additional features.
- Implement robust security measures to protect the online meetings and user data. This includes encryption of data in transit and at rest, user authentication and access controls, secure storage of meeting recordings, and adherence to privacy regulations.

CHAPTER 4 - OBJECTIVES

- To develop a decentralized online meeting platform that leverages blockchain technology to enhance security and privacy.
- To implement a system for secure, immutable logging of meeting data and actions, ensuring transparency and accountability.
- To provide end-to-end encryption for all meeting communications, including video, audio, and file sharing.
- To create a role-based access control mechanism that utilizes blockchain-based identity verification for secure authentication.
- To enable decentralized storage for meeting recordings and files using platforms like IPFS, ensuring data integrity and availability.
- To ensure the system is scalable, user-friendly, and adaptable to a wide range of use cases and industries.

CHAPTER 5 - PROPOSED SYSTEM

5.1) METHODOLGY:

A) FLOWCHART

This diagram illustrates a comprehensive digital platform's user flow and architecture. It begins with user entry and authentication, offering multiple login methods. Users then access a central dashboard with various functionalities, including personalized settings and core platform features. The system incorporates access control mechanisms for certain areas. Key components include user management, content interaction, and advanced features like real-time communication tools. The flowchart effectively maps out the user journey from entry to engagement with the platform's primary services.

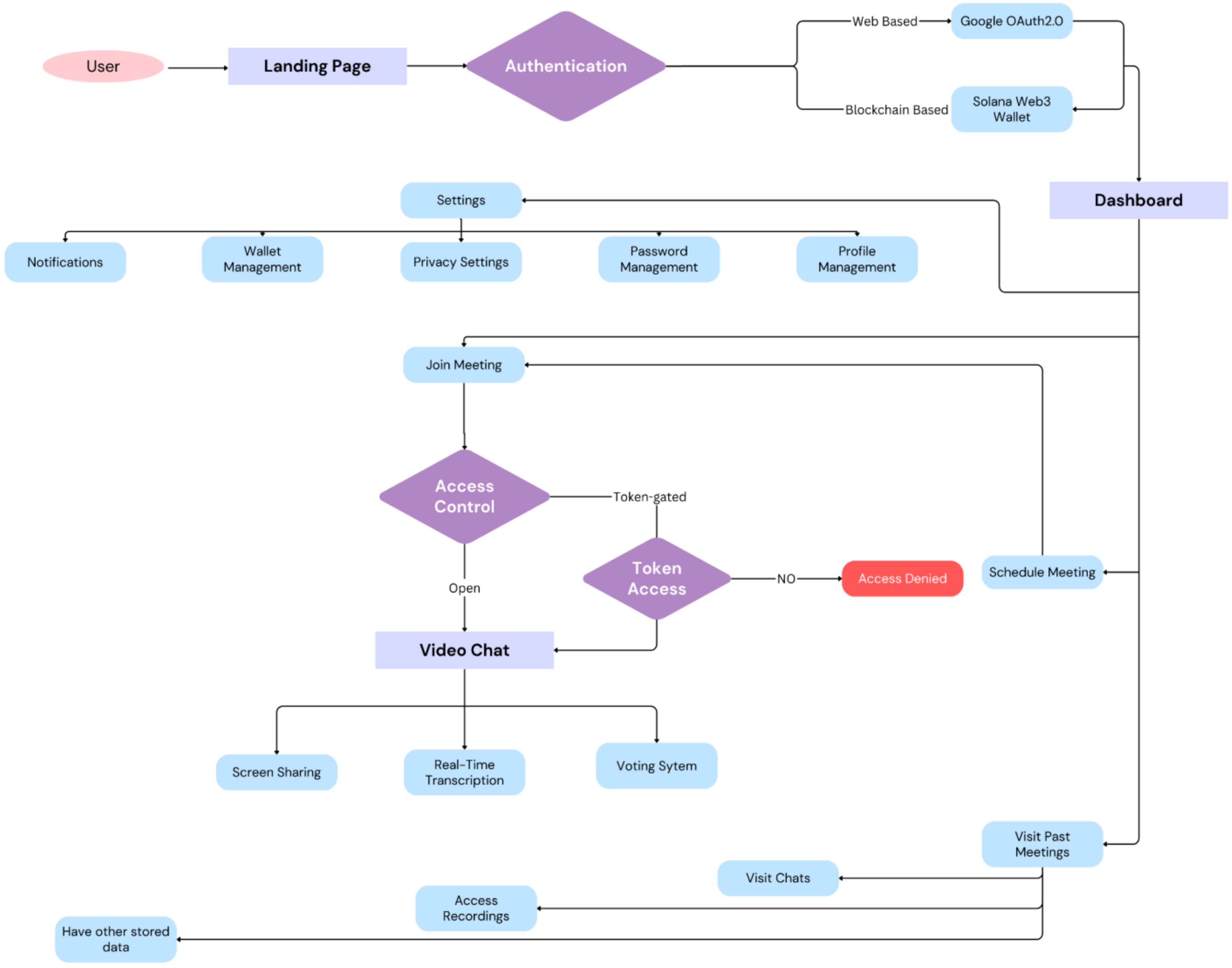

Fig 5.1 Flowchart

B) ER-DIAGRAM

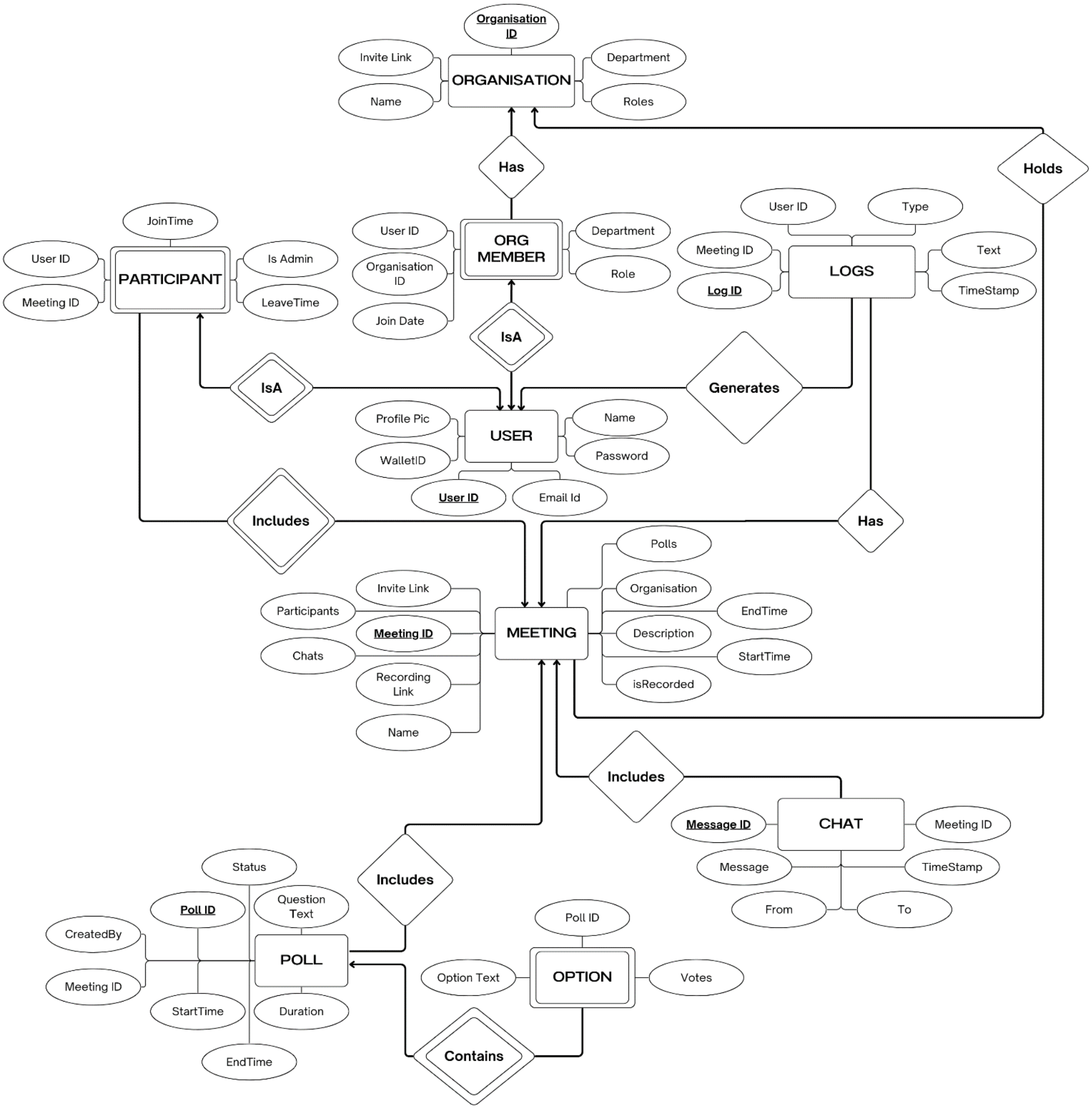

Fig 5.1 ER Diagram

C) IMPLEMENTATION ACTIVITY

Step 1: Server Setup and Initialization

- Set up a scalable server with necessary software (e.g., Nginx, Node.js, etc.).
- Initialize a database (e.g., MongoDB or MySQL) and create schemas for users, meetings, and chat history etc.

Step 2: User Authentication

- Implement secure user registration, login, and logout using JWT or OAuth.
- Set up Role-Based Access Control (RBAC) for managing permissions (admin, host, participant).

Step 3: Blockchain Integration (Solana)

- Use **Solana blockchain** for secure, authentication and to implement RBAC.
- Integrate Solana-based smart contracts ensuring security and traceability.

Step 4: Meeting Management

- Develop APIs for creating, scheduling, and managing meetings.
- Generate unique meeting links and allow users to set meeting parameters (e.g., time, participant list).

Step 5: Invitation and Notification System

- Implement an email or in-app invitation system for inviting participants.
- Enable automated email reminders and in-app notifications for upcoming meetings.

Step 5: Real-Time Communication

- Integrate WebRTC for real-time video and audio communication.
- Set up a signaling server to manage connections between participants.

Step 6: Meeting Room Features

- Implement features for managing participants (e.g., mute/unmute, hand-raising).
- Add support for screen sharing, real-time chat, and voting polls during meetings.

Step 7: Meeting Recording and File Management

- Allow hosts to record meetings and store them securely in cloud storage such as IPFS.
- Provide a file management system for managing shared files such as IPFS.

Step 8: Data Security

- Ensure secure transmission of data using HTTPS (SSL/TLS).
- Implement end-to-end encryption for video/audio streams and encrypt stored data using SRTP.

Step 9: User Interface Development

- Design and implement a user-friendly interface for managing meetings, schedules, and settings.
- Develop a meeting room UI with video controls, chat, and participant management.

Step 10: Testing and Deployment

- Perform unit, integration, and security testing to ensure the platform is reliable.
- Deploy the platform on a live server, set up monitoring tools, and ensure backup and recovery systems are in place.

5.2) TECHNOLOGY USED:

Frontend:

- **React.js:** Core framework for building the user interface, handling component rendering, state management, and user interactions. Its component-based architecture promotes reusability and maintainability of UI elements.

- **JavaScript:** Used throughout the frontend for implementing logic, handling events, and making API calls. It enables dynamic content updates and interactive features without page reloads.

- **Tailwind CSS:** Provides utility classes for rapid UI development, ensuring a consistent and responsive design. Its utility-first approach allows for quick styling and easy customization without leaving your HTML.

- **Shadcn/ui:** UI component library providing pre-built, customizable components that integrate well with Tailwind CSS, speeding up development. It offers accessible and themeable components.

- **Redux Toolkit:** Manages global state in the application, handling user sessions, meeting data, and UI states. It simplifies common Redux use cases, including store setup, defining reducers, and writing immutable update logic.

- **Anchor:** Used to interact with the Solana blockchain from the frontend, allowing operations like viewing meeting logs or verifying data integrity. It provides a convenient abstraction layer for Solana program interactions.

- **Google OAuth 2.0:** Implements secure user authentication using Google accounts. It provides a seamless and trusted login experience while reducing the burden of managing user credentials.

Backend:

- **Node.js:** Runtime environment for server-side code, handling API requests, business logic, and integrations. Its event-driven, non-blocking I/O model makes it ideal for handling concurrent connections in real-time applications.

- **MongoDB:** Stores user profiles, meeting metadata, and other persistent data. Its flexible, document-based structure allows for easy schema evolution as the application's needs change over time.

- **Mongoose:** Provides a schema-based solution for modeling application data and interacting with MongoDB. It offers built-in type casting, validation, query building, and business logic hooks.

- **Morgan:** Logs HTTP requests, useful for debugging and monitoring API usage. It can be configured to log various request details, helping in performance optimization and security auditing.

- **Winston:** Handles application-level logging, providing insights into server operations and potential issues. Its flexibility allows for custom log formats and multiple simultaneous log outputs (console, file, database).

- **Express:** Creates RESTful API endpoints and handles middleware for the backend. Its minimalist structure and robust set of features make it ideal for building scalable web applications and APIs.

Blockchain:

- **Solana:** Used to store immutable records of meeting logs, enhancing security and transparency. Its high-performance blockchain offers fast transaction speeds and low fees, suitable for frequent updates.

- **Web3.js:** Facilitates interaction with the Solana blockchain from the Node.js backend. It provides a comprehensive suite of tools for account management, transaction building, and smart contract interaction.

- **Rust:** Implements secure, decentralized logic for operations like access control or meeting verification on the Solana blockchain. Its performance and safety features make it an excellent choice for writing reliable smart contracts.

- **IPFS:** Stores larger files like meeting recordings or shared documents in a decentralized manner. Its content-addressed storage system ensures data integrity and enables efficient distribution of large files.

APIs:

- **Google Calendar API:** Integrated to allow users to schedule meetings and synchronize them with their Google Calendars. It provides bi-directional synchronization, ensuring that meetings are reflected in users' personal calendars.

- **Calendly API:** Used to implement advanced scheduling features, allowing for easy meeting time selection. It can handle complex availability rules, time zone conversions, and automated reminders.

- **Google Text-to-Speech API:** Provides real-time transcription of meetings, enhancing accessibility and allowing for searchable meeting content. It supports multiple languages and voices, improving the system's inclusivity.

Real-Time Communication:

- **WebRTC:** Enables peer-to-peer audio, video, and data communication directly between browsers. It reduces server load and latency for real-time interactions during online meetings.

- **SRTP:** Secures real-time media streams, ensuring the confidentiality and integrity of audio and video data transmitted during meetings.

- **WebSocket (Socket.io):** Facilitates real-time, bidirectional communication between clients and server. It enables features like instant messaging, live updates, and collaborative tools within the meeting platform.

CHAPTER 6 – EXPECTED OUTCOMES

- **Delivery of a Web Platform for Secure Online Meetings**:

  The primary deliverable is a functional, user-friendly, and secure web-based meeting platform that integrates advanced security measures, blockchain technology, and decentralized storage. This platform will cater to a variety of industries and user bases, offering scalability, adaptability, and robust communication tools.

- **Patent Draft Submission**:

  A comprehensive patent draft will be prepared, detailing the unique aspects of the platform, including the integration of blockchain for immutable records, decentralized storage for secure data management, role-based access control, and real-time communication features. The draft will outline the novel methodologies and architecture that differentiate the platform from existing solutions.

- **Patent Filing and Approval Process**:

  The project includes filing the patent with the relevant authorities, ensuring that the intellectual property rights for the platform's innovative features are protected. This outcome establishes legal ownership and exclusivity over the technology and methods used.

CHAPTER 7 - CONCLUSION

In conclusion, the entire system architecture depicted in the diagram highlights a comprehensive and secure framework for online meetings, combining decentralized technologies with real-time communication tools. The integration of blockchain ensures data immutability, transparency, and security, while the decentralized storage system (such as IPFS) protects meeting recordings and shared files from unauthorized access or data loss. Role-based access control and identity verification further enhance the platform's security by ensuring that only authenticated and authorized users can participate in or access meeting resources.

Moreover, real-time communication via WebRTC facilitates seamless audio and video conferencing with end-to-end encryption, maintaining privacy and confidentiality throughout the meetings. The integration of scheduling APIs and other collaborative features such as file sharing, voting, and transcription tools enhances user engagement and productivity.

By leveraging a combination of blockchain, secure communication protocols, and decentralized storage, the platform offers a scalable and secure solution for online meetings, ensuring that data integrity, privacy, and user accessibility are maintained. The diagram successfully illustrates how these technologies interconnect to create a robust system designed to meet the demands of modern communication while ensuring the highest standards of security and efficiency.

REFERENCES

[1] D. Petkanics, "LivePeer Project overview," 2018.

[2] Furion, "SteemQ - A Decentralized Video Platform for STEEM," 2016.

[3] S. Shetty, V. Red, C. Kamhoua, K. Kwiat, and L. Njilla, "Data provenance assurance in the cloud using blockchain," in Proc. Disruptive Technologies in Sensors and Sensor Systems, vol. 10206, 102060I, 2017.

[4] M. Steichen, B. Fiz, R. Norvill, W. Shbair, and R. State, "Blockchain-Based, Decentralized Access Control for IPFS," in 2018 IEEE International Conference on Blockchain (Blockchain-2018), Halifax, Canada, 2018.

[5] N. R. Ismail and S. Yahya, "Performance Analysis of High-Definition Video Call over Secure Real Transport Protocol (SRTP)," Faculty of Computer Science & Mathematics, University Technology MARA, Shah Alam, Selangor.

[6] A. L. Alexander, A. L. Wijesinha, and R. K. Karne, "An evaluation of Secure Real-Time Transport Protocol (SRTP) performance for VoIP," in Third International Conference on Network and System Security (NSS 2009), Gold Coast, Queensland, Australia, Oct. 2009, pp. 19-21.

[7] T. Adomkus and E. Kalvaitis, "Investigation of VoIP Quality of Service using SRTP Protocol," Telekomunikacijų katedra, Kauno Technologijos Universitetas, 2008.

[8] S. V. Subramanian and R. Dutta, "Comparative Study of Secure vs Non-Secure Transport Protocols on the SIP Proxy Server Performance: An Experimental Approach," in 2010 International Conference on Advances in Recent Technologies in Communication and Computing, 2010.

[9] J. K. Nurminen, A. J. R. Meyn, E. Jalonen, Y. Raivio, and R. G. Marrero, "P2P Media Streaming with HTML5 and WebRTC," Department of Computer Science and Engineering, Aalto University, Finland.

[10] E. Revyakina, "Development of a secure video chat based on the WebRTC standard for video conferencing," Don State Technical University (DSTU), Rostov-on-Don, Russian Federation.

[11] G. C. Añón, Secure High-Definition Video Conferencing, master’s thesis, 2011.

[12] F. Þ. Hjálmarsson, G. K. Hreiðarsson, M. Hamdaqa, and G. Hjálmtýsson, "Blockchain-Based E-Voting System," 2018 IEEE 11th International Conference on Cloud Computing (CLOUD)

[13] P. Kang, W. Yang, and J. Zheng, "Blockchain Private File Storage-Sharing Method Based on IPFS," 2019.