INTRODUCTION

In response to the escalating demand for secure digital communication, the project, "Secure Video & Multimedia Sharing App using WebRTC, employs cutting-edge WebRTC technology. This innovative platform seamlessly integrates multimedia sharing, high-quality video calls, and collaborative screen sharing while prioritizing robust security measures. By allowing users to engage in real-time digital interactions with exclusive room-based access control, the project addresses the evolving requirements of private and focused communication. Through this synthesis of WebRTC capabilities, the system aims to redefine online interactions, offering a trusted environment for users to connect, collaborate, and exchange information seamlessly.

1. MEDIA SHARING

Firstly, the platform facilitates Media Sharing, allowing users to seamlessly exchange various forms of media, fostering a dynamic and interactive communication experience. Whether it's images, documents, or multimedia content, this feature ensures fluid sharing among participants.

2. VIDEO CONFERENCING

Secondly, leveraging the real-time capabilities of WebRTC, the platform enables crystal-clear Video Calls. This feature enhances remote communication by providing a high-quality video interface, bridging geographical distances for users.

3. SCREEN SHARE Thirdly

The Screen Share functionality empowers users to share their screens, fostering effective presentations, collaborative work, and knowledge transfer. Participants can share their window, or entire screen with audio or without audio of the device. This feature enables real time sharing of content, enhancing collaboration among participants.

4. ROOM-BASED ACCESS CONTROL

A notable aspect ensuring security and privacy is the implementation of Room-based Access Control. Each room accommodates only two participants, establishing an exclusive and secure environment for video conferencing. This limitation not only bolsters security but also fosters focused and intimate collaboration spaces. Within these rooms, participants can engage in video conferencing, share screens, and exchange media across their devices seamlessly. The utilization of WebRTC technology is instrumental in enabling these functionalities, facilitating peer-to-peer communication in real-time without additional software or plugins. This project report endeavours to comprehensively explore the technical architecture, implementation nuances, encountered challenges, and devised solutions. By showcasing the efficacy of WebRTC in crafting secure, multimedia-driven collaboration tools, the report aims to contribute to the evolving landscape of digital communication, enabling users to connect, collaborate, and exchange information

OBJECTIVE

The core objective of this project is to envision, develop, and evaluate an innovative media sharing application utilizing the powerful capabilities of WebRTC. The primary goals revolve around creating a seamless platform that empowers users to engage in real-time audio, video, and data sharing directly within their web browsers. The project unfolds through distinct phases: Initiating with the conceptualization and design of an intuitive media sharing application interface that prioritizes user experience. Following this, the application will be built, integrating WebRTC standardized APIs and protocols to enable efficient communication. The architecture will ensure secure data transmission by incorporating encryption mechanisms and effective network traversal solutions. Multi-party communication capabilities will be implemented, enabling users to engage in collaborative interactions with ease. Additionally, the application will undergo rigorous testing to ensure reliability and consistent performance. The project's ultimate goal is to reshape the landscape of online communication, providing users with a dynamic platform that transcends geographical limitations and redefines media sharing experiences. Through successful implementation and user feedback, this project aims to contribute to the evolution of online interactions by leading in a new era of real-time multimedia communication.

SCOPE

The "Secure Video & Multimedia Sharing App using WebRTC" project is envisioned to create a dynamic and secure platform that revolutionizes digital communication. The project aims to seamlessly integrate multimedia sharing, high-quality video calls, and collaborative screen sharing within a fortified environment. It emphasizes stringent security measures, including room-based access control, limiting video conferencing sessions to two participants for exclusive and secure collaboration spaces. User-centric features such as customizable profiles, real-time chat, and cross-device compatibility further enhance the overall communication experience. With a commitment to continuous improvement, the project considers future enhancements, contributing to the evolving landscape of secure, real-time multimedia communication.

FEATURES

The features of the "Secure Video & Multimedia Sharing App using WebRTC project encompasses various dimensions to address the evolving needs of digital communication. The project will focus on:

1. Media Sharing Capabilities:

Develop a platform that seamlessly facilitates the exchange of diverse multimedia content, including images, documents, and other formats, ensuring a dynamic and interactive communication experience.

2. Real-time Video Calls:

Leverage WebRTC real-time capabilities to enable high-quality video calls, bridging geographical distances and enhancing remote communication for users.

3. Collaborative Screen Sharing:

Implement a screen-sharing functionality that empowers users to share their screens in real-time, facilitating effective presentations, collaborative work, and knowledge transfer.

4. Room-Based Access Control:

Ensure security and exclusivity by implementing room-based access control, restricting video conferencing sessions to only two participants within each room.

5. User Profiles and Customization:

Allow users to create and customize profiles, managing their preferences and privacy settings within the platform.

6. Real-time Chat System:

Integrate a real-time chat system for users to communicate through text, enhancing the collaborative experience alongside multimedia sharing.

7. Security Measures:

Implement encryption mechanisms and robust security protocols to safeguard user data and communications, addressing privacy concerns.

8. Cross-Device Compatibility:

Design the application to be responsive, ensuring seamless functionality across various devices and screen sizes, providing a consistent user experience.

9. Testing and Quality Assurance:

Conduct rigorous testing to ensure the reliability, security, and consistent performance of the platform under diverse scenarios, addressing potential issues and optimizing user experience.

10. Future Enhancements Consideration:

The project will lay the groundwork for potential future enhancements, such as group video calls or advanced multimedia editing features, to meet evolving user needs.

11. Contribution to Digital Communication Landscape:

The project aims to contribute significantly to the evolution of online interactions by showcasing the efficacy of WebRTC in crafting secure and feature-rich collaboration tools.

12. User Feedback and Iterative Development:

Gather user feedback to understand the effectiveness of the platform and iterate on the development process, ensuring continuous improvement and user satisfaction.

TECHNOLOGY USED

The technology stack for the "Secure Video & Multimedia Sharing App using WebRTC" project includes:

1. Frontend:

HTML, CSS, JavaScript React/Angular/Vue.js)

2. Backend:

Node.js, Django, or a suitable backend framework

3. Database:

MongoDB, MySQL, or PostgreSQL

4. Authentication:

JSON Web Tokens (JWT) or OAuth

5. Real-time Communication:

WebRTC and Web Sockets

6. Notification System:

Implementing technologies for real-time notifications

WebRTC

WebRTC, or Web Real-Time Communication, is an open-source project transforming real-time communication on the web. Developed by the W3C and IETF, it allows direct peer-to-peer communication within web applications without requiring additional plugins. Key components include the Media Stream API for multimedia device access, RTC Peer Connection for establishing communication channels, and RTC Data Channel for low-latency data exchange. Beyond conventional audio and video communication, WebRTC supports peer-to-peer data sharing and adaptive bitrate streaming for optimized media delivery. Emphasizing security, it implements encryption for confidential communication. Compatible with major browsers, WebRTC finds applications in video conferencing, live streaming, collaborative tools, browser-based voice calling, and online gaming platforms. Its versatility and robust features position WebRTC as a foundational technology, enabling developers to innovate and redefine real-time communication experiences on the web.

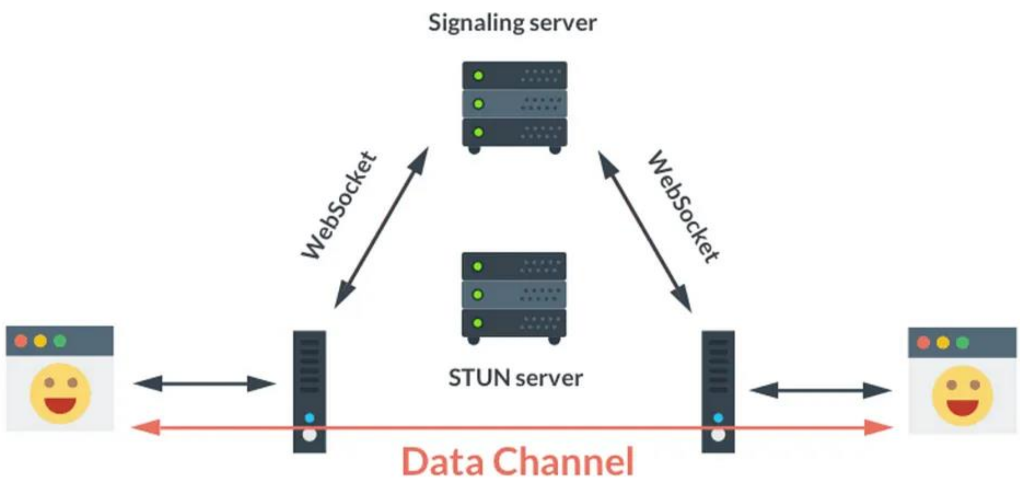

Fig.1. WebRTC Simple Explanation

WORKING OF WEBRTC

WebRTC is a protocol which allows us to do bi-directional, real-time, peer-to-peer, media exchange between 2 devices. Real-time means there is no lag. Peer-to-peer means 2 devices connected to each other directly. Media exchange is where 2devices can send/receive video and audio to each other.

Each client has to go through these steps for successful WebRTC communication.

A. Signalling

Before 2 devices can start a P2P communication, they need to know about each other, this is done with the help of Signalling. Both devices agree upon a common central server through which they can talk and exchange information about each other. This server is called signalling server. There is no standard for connecting to Signalling server, so devices can use protocols like HTTP, REST, web sockets, MQTT etc. to connect to the server.

Devices exchange information like:

1. IPs and Ports that the Device is reachable on (candidates).

2. How many audio and video tracks the device wishes to send.

3. What audio and video codecs each device supports.

4. Values used for securing connection (certificate fingerprint).

B. Connecting

Once 2 devices know each other's details, they try to find the best possible way to connect to each other using ICE protocol. ICE (Interactive Connectivity Establishment) is a protocol which is used to find the best ways to connect to a device called ICE candidate. ICE uses STUN or TURN servers to find the best possible way to connect. STUN server is a simple server, which on request provides the public IP and port of the connecting device. TURN server is a Relay server, i.e. it acts as a central server through which the data passes through. While connecting the 2 devices check if the ICE candidate is valid.

C. Securing

Once devices know how to connect to each other, they secure it using 2 protocols DTLS and SRTP. DTLS (Datagram Transport Layer Security) which is just TLS over UDP. The TLS protocol provides HTTPS security. SRTP (Secure Real Time Transport Protocol) is RTP with packets encrypted. Devices then do a handshake using DTLS protocol. They check if the certificates match the fingerprint passed in the SDP during signalling. For Audio/Video transmission devices use the same certificates to encrypt the data and send it via RTP.

D. Communicating

Two Devices can communicate 2 things between each other:

1: Media: Audio, Video.

2: Data Messages (simple string messages)

WebRTC specifies 2 different protocols for each of the above type. For Media, devices use RTP (Real-time Transport Protocol) encrypted with SRTP. For Data, devices use the SCTP (Stream Control Transmission Protocol)

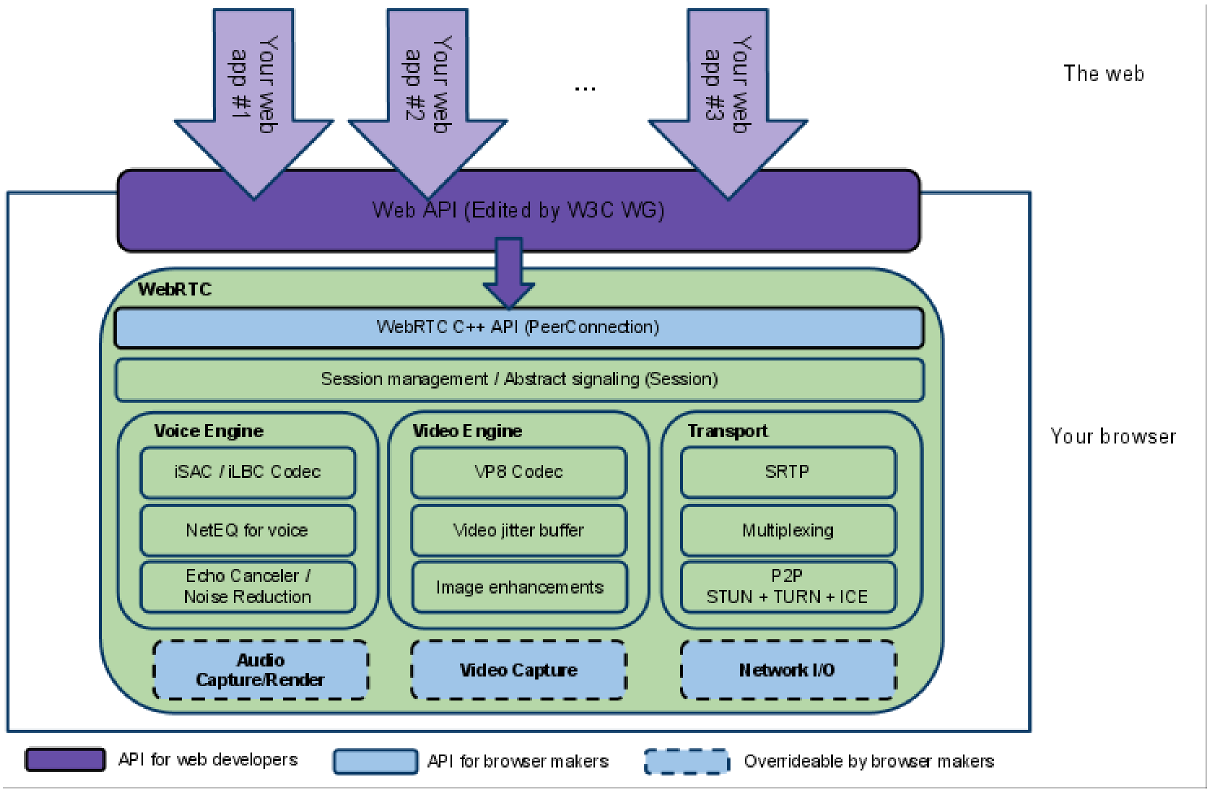

Fig. 2. Architecture of WebRTC

DEVELOPMENT STEPS

Setting up the Development Environment:

Install and configure the necessary tools and frameworks for both frontend and backend development.

User Authentication:

Implement a secure user authentication system using technologies like JWT (JSON Web Tokens) or OAuth to ensure authorized access to the platform.

Integration of WebRTC:

Integrate WebRTC technology to harness its real-time communication capabilities, enabling seamless and direct peer-to-peer connections for video calls and multimedia sharing.

Multimedia Sharing Features:

Develop features for multimedia sharing, allowing users to exchange various forms of media, including images, documents, and multimedia content.

Real-Time Chat System:

Integrate a real-time chat system to enable users to communicate through text during multimedia sharing and video calls, enhancing collaborative interactions.

User Profiles:

Create user profiles with customization options, enabling users to manage their settings, preferences, and privacy controls within the platform.

Notification System:

Implement a notification system to alert users about new content, messages, or interaction requests, improving overall user engagement and experience.

Testing and Deployment:

Conduct thorough testing of the platform to ensure reliability, security, and consistent performance under various scenarios. Once testing is complete, deploy the application to a hosting platform, considering scalability and performance require

CHALLENGES ENCOUNTERED DURING DEVELOPMENT

Real-time Communication Complexity:

Addressing the intricacies of real-time communication using WebRTC, ensuring smooth and synchronized interactions without latency or delays.

Scalability Concerns:

Managing the platform's scalability to handle a potentially large user base, particularly during peak usage periods for video calls and multimedia sharing.

Security Implementation:

Implementing robust security measures, including encryption mechanisms, to protect user data and communication channels, mitigating potential vulnerabilities.

Cross-Browser Compatibility:

Ensuring consistent functionality and performance across various web browsers, considering the nuances of browser-specific implementations of WebRTC.

User Authentication Challenges:

Overcoming challenges related to user authentication, including securing user credentials and implementing secure login and registration processes.

Optimizing Multimedia Sharing:

Optimizing the multimedia sharing features to handle diverse formats efficiently, ensuring a seamless and responsive user experience.

Chat System Synchronization:

Synchronizing the real-time chat system with other features, such as multimedia sharing and video calls, to maintain a cohesive and synchronous collaborative environment.

User Profile Customization:

Developing a user-friendly interface for profile customization, allowing users to manage settings and privacy controls intuitively.

Notification System Implementation:

Implementing an effective and real-time notification system, overcoming challenges related to timely alerts and ensuring a reliable notification mechanism.

Deployment and Hosting Challenges:

Addressing challenges related to deployment and hosting, including optimizing the platform for the selected hosting environment and ensuring reliable performance.

Testing for Reliability:

Rigorously testing the application for reliability under various scenarios, uncovering and addressing potential bugs, glitches, or performance bottlenecks.

User Feedback Incorporation:

Incorporating user feedback iteratively to enhance the platform's features, user interface, and overall performance based on user experience.