VOICE CHAT WEB APP USING WEBRTC

Avdhesh Kumar Tiwari[1*], Harshit Gupta[2], Sangam Mittal[3], Mayank Rawat[4]

*[*1,2,3,4]Department of computer science, ABES Institute Of Technology, Ghaziabad*

Corresponding Author:

ABSTRACT

Technology improvements have made it possible to communicate more effectively. New technologies have improved existing communication routes. Some real-time communication systems include limitations, such as the need for additional software plugins and downloads to enable real-time communication, as well as security problems. Web Real-time Communication (WebRTC) is a technology that may be able to assist in the resolution of these issues. Thanks to advancements in internet technology, people may now easily access the internet. With the help of technological improvements, the internet is providing an increasing variety of services, all of which can be virtualized. People's internet communication has become ingrained in their daily lives. People used to communicate with one another by exchanging voice chat messages via the internet.In order to achieve scalability, the platform also makes use of cloud computing. The iterative platform architectural design is revealed, as well as some preliminary scalability analysis results.

KEYWORDS: *HTML, CSS, NodeJs, React, Real time communication, WebRTC.*

I. INTRODUCTION

People's connections are more vital than ever in today's society, thanks to the rapid development of the internet, and they're looking for new ways to communicate with one another in real time. The voice chat web app allows users to talk in real time and access a single site where they may join various rooms based on their interests, connect with others, and even create their own rooms. This project can be simplified by using webrtc's rtc data channel api to transfer data directly from one peer to another peer connection.

The voice talk web application grants clients for a consistent casing correspondence and to run over a singular spot through which clients can join different rooms in view of their benefit, can speak with others and besides they are allowed to make their own rooms. This assignment can be relieved by using WebRTC, RTC Data Channel API to move data clearly beginning with one companion then onto the following friend association. Our voice visit Web application is a sound based talk application that licenses clients to sharply partake receipt talks. It focuses towards a horde of individuals that needs adaptability and a space for discussions without the need to impart themselves over video. It gives its clients the versatility to use its organization either as a spot for loosened up discussions or to focus on a live computerized broadcast. It's remarkable from other VoIPs in which it doesn't save any conversations after they are finished. It ensures that each call is uncommon and it encourages its clients to contribute energy on the application or, more then likely they might miss an incredible open door.
WebRTC, which is built on an open standard, allows you to add real-time communication features to your application. By allowing video, audio, and generic data to be sent between peers, it enables developers to design complex speech and video communication systems. The technology is supported by all current browsers and native clients for all major platforms. WebRTC has a lot of advantages over previous forms of real-time interactive multimedia communication. Because it is supported by nearly all modern browsers, users do not need to install or download any additional software. The standard is thoroughly examined, with information on available communication topologies and performance data included.

WebRTC distributes media packets as quickly as possible nowadays. When it comes to large business networks, WebRTC can be a complicated problem. Their firewalls are capable of blocking UDP traffic. To make UDP work well for a wide audience, a lot of work has gone into it.

TCP and UDP are used by the majority of Internet traffic nowadays, not just web pages. Tablets, mobile devices, Smart TVs, and other electronic devices contain them. As a result, it's critical to comprehend how these technologies perform.

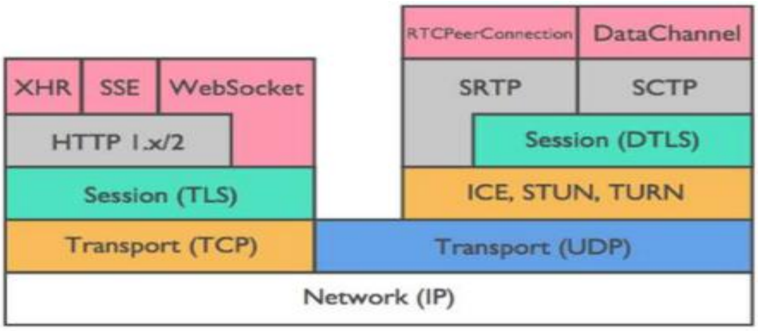

Fig 1. WebRTC Protocol Stack

II. ARCHITECTURE

WebRTC architecture is made up of multiple standards that cater to application and browser APIs. Although WebRTC technology's major goal is to provide real-time communication, it's also meant to work with current communication systems like VOIP, SIP clients, and STN, to mention a few. Web Real-Time Communication (WebRTC) is a collection of standards, protocols, and JavaScript APIs that enable browsers to share audio, video, and data in real time (peers). WebRTC makes real-time communication a common feature that any online application may use via a simple JavaScript API, rather than depending on third-party plug-ins or proprietary software.

A. Media Stream

To send audio from one browser to another, we must first use the Media Stream API to use the microphone (media devices). We may access the microphone or camera stream using the MediaStream API. We can preserve a webcam view inside our web page even if we don't wish to communicate data. Javascript may be used to create all of this. Accessing media, of course, necessitates user consent. This method is invoked every time it is called. The browser displays some notifications or indications in the form of light, indicating that the app is attempting to access the camera. If the user agrees, our application downloads the media stream and attaches it to our page's video element.

B. RTCPeer Connection

After effectively getting the stream or some other information besides, we can now send it to some other program. We'd involve RTCPeerConnection for that. Assuming we are to lay out an association between two programs, Signaling Server

comes to play. This is the way the discussion between the two browsers (let's call them Alpha and Charlie) who need to share stream) and signaling server take place.
The target interface parameters for browser events can be RTC Peer Connection. The RTC Peer Connection features two ports, one for caller and one for responder. The connection is established as follows: first, the caller sends an offer signal to the responder; second, the responder responds with an answer signal; and finally, the caller and responder create a Peer Connection object to save the connection to each other. Both offer and answer signals follow the SDP in this function (Session Description Protocol). SDP is a standard for characterising multimedia connections. Because the majority of users use a LAN (Local Area Network), the SDP package is unable to discover another peer via this network. The signalling server was created to address this issue.

C. RTC Data Channel

The RTC Data Channel interface defines a network channel that can be used to send and receive arbitrary data in a bidirectional peer-to-peer fashion. An RTC Peer Connection is associated with each data channel, and each peer connection can have up to 65,534 data channels in theory (the actual limit may vary from browser to browser). Call the RTC Peer Connection's create Data Channel () function to create a data channel and invite a remote peer to join you. The invited peer must send a data channel event (of type RTC Data Channel Event) to indicate that the data channel has been added to the connection.
RTC Data Channel is utilised among clients in addition to RTC Peer Connection, which is used to transport media streams between clients. RTC Data Channel's port was created to send files in any format between users. To create the RTC Data Channel, we must first create the RTC Peer Connection. The RTC Data Channel requires three steps to transfer a file: First, use the File API in the web browser to acquire file information; secondly, create an RTC Peer Connection between the two users; and finally, use RTC Data Channel to create a data transfer channel.

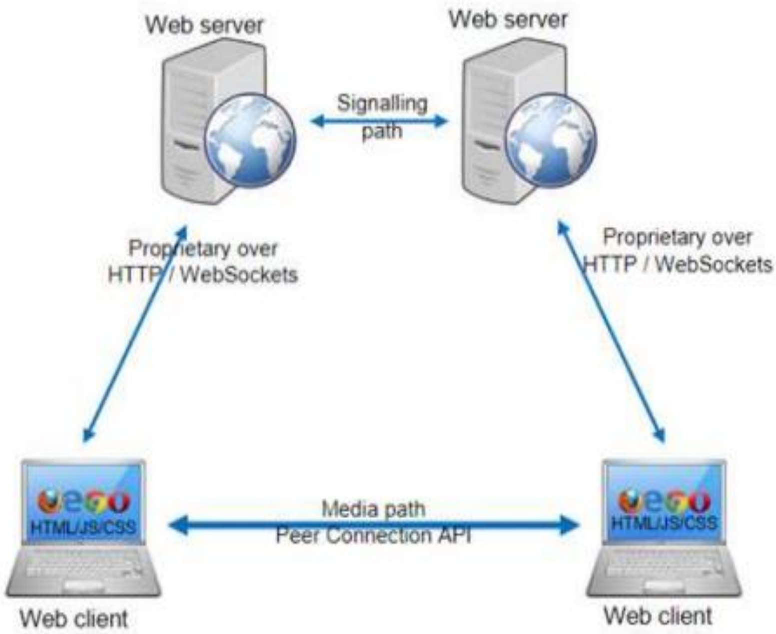

Fig 2. WebRTC Trapezoid

III. PROPOSED METHODOLGY

The objective is to create a web peer to peer real time communication system that allows users to communicate with high-speed data transmission over the communication channel using WebRTC (Web Real-time communication) technology without the need to install any plug-ins or third-party software or apps.
The system is designed to allow audio, with user identification and finding of other system users without the need for installation or setup.
When the user comes on the platform it first authenticates itself by entering the phone number and validates the number by entering the One Time Password generated on that phone number by the mechanism of the system.

The system will generate a token which is valid for a particular time period and the user will need to complete its profile in the provided time. If the user is unable to complete its profile in the given time then the session will expire and the user will have to retry again.
After successful authentication the user will enter into the main page of the web app where the user has two options, either to create its own room or to enter in existing created room by the other users.
For this purpose, we created a Database (using MongoDB) that contains of three collections:

In the first collection it consists of information about users such as username, phone number, profile picture URL, etc.
In the second collection it consists of token which is generated by the system for the user.
Third collection comprises of information about the rooms such as topic of the room, author id, time, etc.
All information sent from the server to the client is protected and encrypted utilising a high-security protocol server that distributes content from various external sources. The server programme will verify each demand's user session before allowing data to flow.

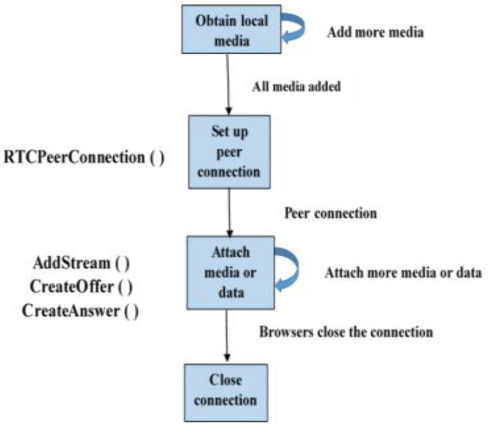

Fig 3. WebRTC Session Establishment

IV. COMPARISON

Methodology	Result
HTML ,Javascript API, Node.js server, Web RTC.	They create a multipoint video conferencing system in which each user is linked to all other users and the same video stream must be supplied to all connections.
Real time communication, WebRTC, Protocols.	Results show that if the conference is floor-controlled, the platform can scale to hundreds or even thousands of users.
WebRTC, QoE management, Quality of experience.	The findings of this study revealed that end users prefer to be assessed using QoS characteristics rather than directly employing QoE methodologies when judging quality.
HTML, WebRTC.	This app employs WebRTC for real-time audio and video streaming, and they usually communicate with cloud servers over HTTPS.
HTML,WebRTC,WebSocket.	The results of the tests show that this conversation system may be used in a variety of network situations, offering a chatting service for a variety of people, and that it is safe, efficient, and simple to maintain and extend.
HTML, Javascript API, Node. js, server, WebRTC.	The results show that the system is stable, fully functional, and secure, and that it can be used in a real-world network to send and receive multimedia data in real time.
WebRTC, Node.js, MongoDB.	As a result, this initiative can be used to initiate conversations amongst friends, and others. This can also be used for a business-related initiative.
Web Real time communication, RTCPeerConnection.	The system was able to eliminate issues with plugins and third-party application downloads, as well as reduce latency and bandwidth utilization, and establish real-time peer-to-peer audio-visual communication.
HTML5, WebRTC.	They came to the conclusion that, in comparison to other systems, this system used many servers and browser-based WebRTC APIs to establish audio and video connections.
WebRTC, Node.js, HTML5, Javascript, multimedia conferencing system.	The result shows that this technology can provide high-level semantics in terms of QoE/QoS indicators, allowing them to build extensive functional test assertions.

V. TECHNOLOGY USED

A. WebRTC

Web Real-Time Communications (RTC) provides a limited set of features to the web browser. Browsers will communicate directly with one another for the first time, resulting in a variety of architectures, including a triangle and trapezoid model. Web RTC's media capabilities are cutting-edge, with a slew of new features. The World Wide Web Consortium (W3C) and the Internet Engineering Task Force are developing the Web RTC underlying specifications (IETF).

B. The Web Browsing Model

The Hypertext Transport Protocol, HTTP, which runs over Transmission Control Protocol, TCP, or in some new Web RTC: APIs and RTCWEB Protocols of the Real-Time Web4implementations, over the Web Socket protocol, provides information transport between the browser and the web server. HTML, which usually includes JavaScript and Cascading Style Sheets, is used to carry the content or application (CSS)

C. MongoDB

MongoDB is designed to be a general-purpose database, therefore it has a long number of unique functions in addition to creating, reading, updating, and deleting data. MongoDB includes unique, compound, geographic, and full-text indexing capabilities, as well as general secondary indexes that enable for a range of quick searches. It has time-to-live collections for data that needs to be discarded after a specified amount of time, such as sessions. Fixed-size collections are also supported, which are handy for storing recent data such as logs. It has a simple mechanism for storing huge files and their metadata. MongoDB lacks some functionality seen in relational databases, such as joins and sophisticated multi-row transactions. Because both of those characteristics are difficult to supply efficiently in a distributed system, omitting them was an architectural decision to allow for greater scalability.

D. Node.js

Node.js is an open-source, cross-stage, back-end JavaScript runtime climate that sudden spikes in demand for the V8 motor and executes JavaScript code outside a web browser. Node.js allows engineers to utilize JavaScript to compose order line instruments and for server-side prearranging running contents server-side to deliver dynamic page content before the page is shipped off the client's internet browser. Subsequently, Node.js addresses a "JavaScript all over" worldview, binding together web-application advancement around a solitary artificial language, instead of various dialects for server-side and client-side contents.

E. React.js

React (also referred to as React.js or ReactJS) could be a free and open-source front-end JavaScript library for building user interfaces supported UI components. it's maintained by Meta (formerly Facebook) and a community of individual developers and firms. React will be used as a base within the development of single-page or mobile applications. However, react is barely concerned with state management and rendering that state to the DOM, so creating React applications usually requires the utilization of additional libraries for routing, additionally as certain client-side functionality.

F. HTML

Hyper Text Markup Language is a short form of Hyper Text Markup Language. It is a markup language for creating web pages. HTML is a markup language that blends markup and hypertext. The link between web sites is referred to as "hypertext." A markup language is used to define the textual data within the tag that defines the structure of web pages. This system is used to annotate content so that it can be understood and manipulated by a computer. Humans can read parts of markup languages (such as HTML). The language includes tags to define the type of text analysis that is required. HTML is a markup language for editing text, images, and other content in your browser.

G. Socket.io

Socket.IO is a JavaScript library that is used by programmers to create real-time "Web Applications." Engine.IO, which implements the transport-based cross-browser/cross-device bi-directional communication layer, is used by Socket.IO. A socket is a single link here between client and a server that enables both the client and the server to send or receive data in real time. Since Library is an occurrence system, it broadcasts and monitors for certain events to be activated.

Conclusion

After going through all the above study and discussion we see that applying WebRTC for real time peer-to-peer communication is more feasible than any other technology. The use of WebRTC technologies has enabled the execution of safe and strong data transmission among users as peer-to-peer or peer-to-group links in real-time communication. As a consequence, anybody can construct their own Website or application such as actual file sharing, real-time communication eco system such as sending messages conversation or video/audio unified communications. By using WebRTC technology, we can build a webPage with the most advanced feature, allowing users to communicate with one another via text message, video/audio calls,and more, all while using simple JavaScript APIs and Node JS.

References:

[1]. George Suciu, Ștefan Ștefănescu , Cristian Beceanu,Marian Ceaparu, WebRTC role in real-time communication and video conferencing, 978-1-7281-6728-2/20 ©2020 IEEE

[2]. Pelayo Nuno, Francisco G. Bulnes, Juan C. Granda, Francisco J. Suarez, Daniel F. Garcıa, A Scalable WebRTC Platform based on Open Technologies, 978-1-5386-4599-4/18 © 2018 IEEE

[3]. Boni García, Micael Gallego, · Francisco Gortázar ,· Antonia Bertolino, Understanding and estimating quality of experience in WebRTC applications, © Springer-Verlag GmbH Austria, part of Springer Nature 2018

[4]. Kundan Singh,John Buford, Developing WebRTC-based Team Apps with a Cross-Platform Mobile Framework, 978-1-4673-9292-1/16 ©2016 IEEE

[5]. Shi Yuzhuo , Hao Kun, Design and Realization of Chatting Tool Based on Web, 978-1-4799-2860-6/13 ©2013 IEEE

[6]. Zinah Tareq Nayyef , Sarah Faris Amer , Zena Hussain, Peer to Peer Multimedia Real-Time Communication System based on WebRTC Technology, 7 (2.9) (2018) 125-130.

[7]. T.Mugilan, L.sathish, Advance web based application for multimedia using P2P connection, International Journal of Modern Trends in Engineering and Science ,2016, ISSN: 2348- 3121

[8]. Edim Azom Emmanuel,Bakwa Dunka Dirting, A Peer-To-Peer Architecture For Real-Time Communication Using Webrtc, Journal of Multidisciplinary Engineering\ Science Studies (JMESS) ISSN: 2458-925X, April - 2017

[9]. Wenpeng Wang, Lingli, A design of multimedia conferencing system based on WebRTC Technology, 978-1-5386-3371-7/17/ ©2017 IEEE

[10]. Boni García, Francisco Gortázar, Luis López-Fernández, Micael Gallego, and Miguel París, WebRTC Testing: Challenges and Practical Solutions, 2471-2825/17/ © 2017 IEEE

[11]. Julian Jang-Jaccard, Surya Nepal, Branko Celler, Bo Yan, WebRTC - based video confrencing service for telehealth, Springer-Verlag Wien ©2014

[12]. Jukka k. Nurminen, Antony J.R. Meyn, Eetu Jalonen, Yrjo Raivio and Raul Garcia Marrero, P2P Media Streaming with HTML5 and WebRTC, 978-1-4799-0056-5/13/$31.00 © 2013 IEEE

[13]. Christian Vogt, Max Jonas Werner, Thomas C. Schmidt, Leveraging WebRTC for P2P Content Distribution in Web Browsers 978-1-4799-1270-4/13/$31.00 © 2013 IEEE

[14]. Boris Grozev, George Politis, Emil Ivov, Thomas Noel, and Varun Singh, Experimental Evaluation of Simulcast for WebRTC, 2471-2825/17/$25.00 © 2017 IEEE

[15]. Alah Johnson, John Yoakum, and Kundan Singh, Avaya Inc, Taking on WebRTC in an Enterprise, 0163-6804/13/$25.00 © 2013 IEEE