iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** IV **Month of publication:** April 2022

DOI: https://doi.org/10.22214/ijraset.2022.41326

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue IV Apr 2022- Available at www.ijraset.com

Review of Real-Time Collaboration Frameworks, Libraries and Products

Suvarna Aranjo[1], Neil Pinto[2], Preet Parekh[3], Tanay Modi[4]
[1, 2, 3, 4]*Information Technology Xavier Institute of Engineering*

Abstract: ***The onset of the pandemic brought forth the need to be isolated which prompted companies to adopt the remote working model much more prominently. This sudden switch brought forth a host of problems as people were not used to remote work. To remedy the situation, a host of tools and products gained prominence - Google Meet, Zoom, etc. which allowed people to collaborate remotely. We will be doing a review of the frameworks and technologies that these products use to understand how real-time collaboration works and the advantages and drawbacks of these products.***
Index Terms: WebRTC, MCU, Video-Conferencing, Zoom, Google Meet

I. INTRODUCTION

The need for remote collaboration arose with the arrival of the COVID-19 pandemic which brought We- bRTC to the forefront as people were isolated and had to use tools like Zoom and Google Meet for real-time collaboration. These products use WebRTC as their base to make audio and video communication possible and efficient irrespective of the device streaming audio and video. This paper will give an introduction to WebRTC and its working and also discuss the various models of WebRTC being used. We will also be discussing the challenges faced by WebRTC. Another thing that will be touched upon is a review of collaboration tools that are being currently used.

II. ANALYSIS

A. *Brief Overview of WebRTC*

WebRTC implements three APIs:

1) *MediaStream (aka getUserMedia):* The MediaDe- vices.getUserMedia() method prompts the user for permission to use a media input which produces a MediaStream with tracks containing the requested types of media. That stream can include, for example, a video track (produced by either a hardware or virtual video source such as a camera, video recording device, screen sharing service, and so forth), an audio track (similarly, produced by a physical or virtual audio source like a microphone, A/D converter, or the like), and possibly other track types. [1]
2) *RTCPeerConnection:* The RTCPeerConnection in- terface represents a WebRTC connection between the local computer and a remote peer. It provides methods to connect to a remote peer, maintain and monitor the connection, and close the connection once it's no longer needed. [2]
3) *RTCDataChannel:* The RTCDataChannel interface represents a network channel which can be used for bidirectional peer-to-peer transfers of arbitrary data. Every data channel is associated with an RTCPeerConnection, and each peer connection can have up to a theoretical maximum of 65,534 data channels (the actual limit may vary from browser to browser). [3]
4) *Signaling: session control, network and media in- formation:* WebRTC uses RTCPeerConnection to com- municate streaming data between browsers (aka peers), but also needs a mechanism to coordinate communi- cation and to send control messages, a process known as signaling. Signaling methods and protocols are not specified by WebRTC: signaling is not part of the RTCPeerConnection API.

Instead, WebRTC app developers can choose whatever messaging protocol they prefer, such as SIP or XMPP, and any appropriate duplex (twoway) communication channel. Signaling is used to exchange three types of information:

a) Session control messages: to initialize or close communication and report errors.
b) Network configuration: to the outside world, what's my computer's IP address and port?
c) Media capabilities: what codecs and resolutions can be handled by my browser and the browser it wants to communicate with? [4]

©IJRASET: All Rights are Reserved

The acquisition and exchange of network and me- dia information can be done simultaneously, but both processes must have completed before audio and video streaming between peers can begin. The offer/answer architecture described above is called JSEP, JavaScript Session Establishment Protocol.

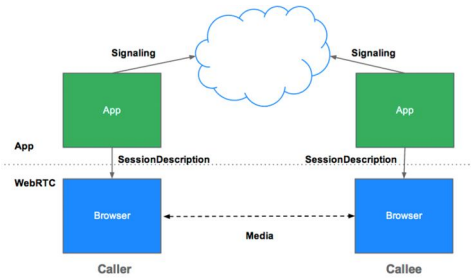


Fig. 1: JSEP Architecture [4]

Once the signaling process has completed success- fully, data can be streamed directly peer to peer, between the caller and callee—or if that fails, via an intermediary relay server.

RTCPeerConnection is the WebRTC component that handles stable and efficient communication of streaming data between peers. Below is a WebRTC architecture diagram showing the role of RTCPeerConnection.

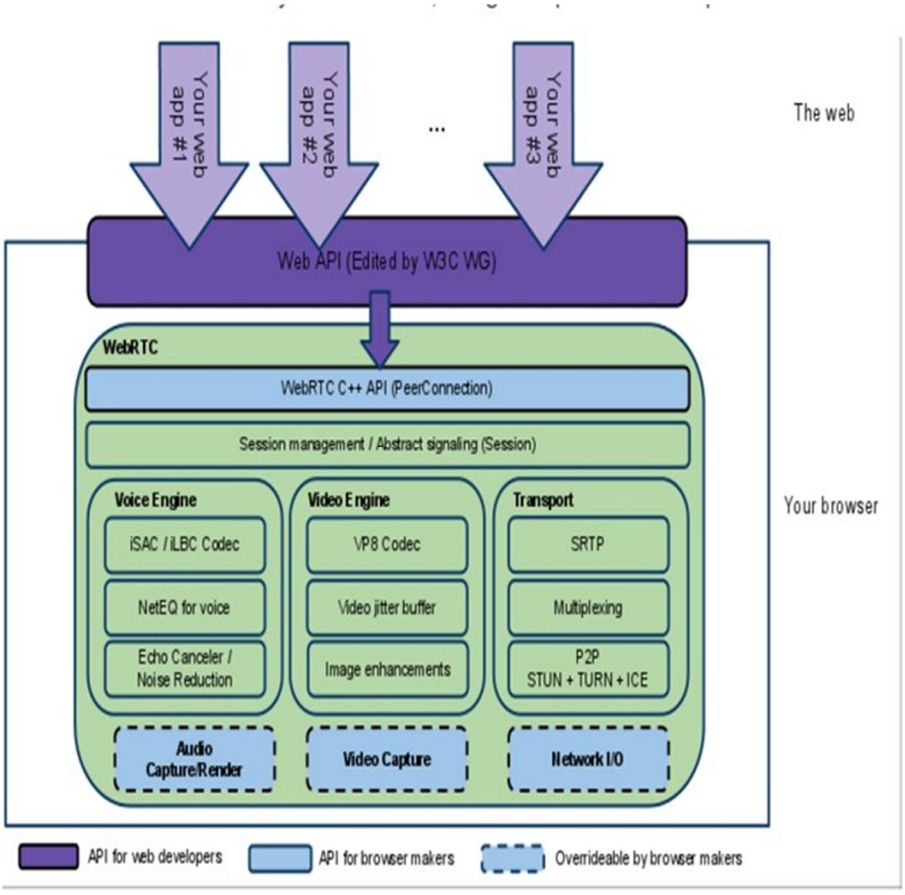


Fig. 2: RTCPeerConnection in WebRTC [4]

B. Models of Multimedia Conferences using WebRTC

1) Centralized model: In this model, dialog signaling and media mixing are both carried out by only one user agent i.e. the Conference Focus (CF). This CF takes media from users who participate on the conference, mixes them, and sends out separately the appropriately mixed stream to each participant.

2) *Fully-meshed model:* To make the conference in- dependent from the departure of the CF, distributed signaling and administration model can be used to sup- port small group of multi-party communication. In this approach, every endpoint directly communicates with every other one. All the parties in the conference are ”equal” and no participant is topologically special or has any additional rights or abilities beyond those of the others. Any conference member can, at any time, invite another user to participate in conference.
3) *Centralised signaling and fully-coupled media:* On the same time where 2-B2 is more robust and can resist to any participant departure or failure, the use of decentralized signaling approach should make the conference administration more complex. In many cases, it's important to have one or many ”super” conference members that hold the conference moderator privileges. These privileges will allow conference creation, modi- fication and termination as well as conference admin- istration by applying conference policy when it comes to manage participant adding/removing operations for example. Conference policy rules can include a black list, pre-authorized participant, conference creation in adhoc mode, etc. In all cases, having a single view of these rules is very important and should make conference management more trivial while enforcing conference access security.
4) *Centralised signaling and tightly-coupled media:* Fully distributing media require from each participant within N-party conference to mix N-1 media flow and to send N-1 media flow. This topology is not usually adapted for handheld devices with limited computation power and bandwidth. Another model that distribute media in tree-based model can be proposed while the sig- naling part of the communication is centralized around the CF. This model take in consideration the fact that conference participant are not equal in terms of network characteristics (bandwidth) and hardware capabilities (computational power). Therefore, participants should not support the same media load within the same N-party conference. This tightly model is using the tree-based distribution and affect media mixing/distribution to a set of selected end-points while the other participant will be connected to the conference in ”heavy” way without having to mix or to distribute media flow. [5]

C. Deploying WebRTC Based Browsers in Small Con- ference Group

The model where media load is fully coupled and signaling in centralized on the web server should comply with small scale conference based on WebRTC browsers since each browser in conference with N participants will create very limited number of media connections. These connections can be managed within the same HTTP session established with the Web Server and using adapted JavaScript code. [5]

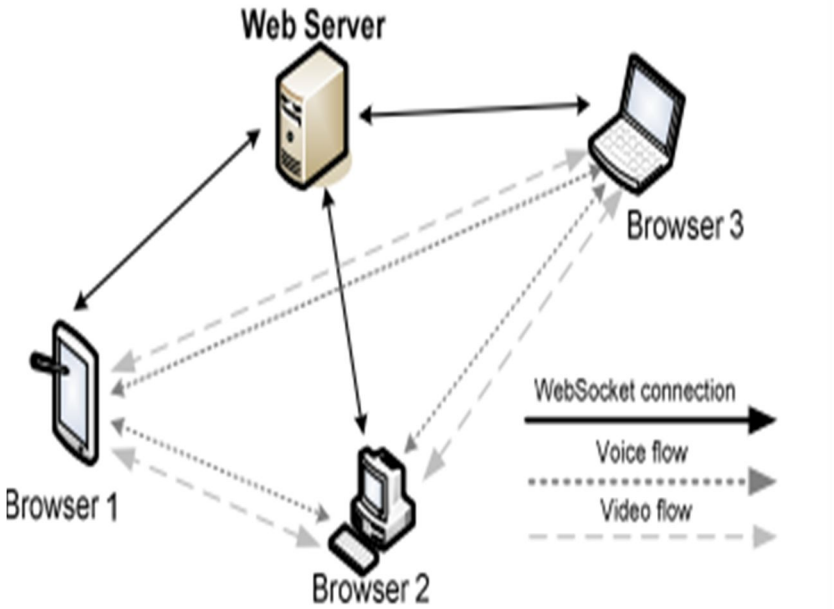


Fig. 3: WebRTC interconnection for small conference [5]

D. Signaling Mechanism

WebSocket based connection create P2P bi-directional communication between the Web Server and each ac- tive conference participant. To join conference, Browser should send HTTP request using the appropriate URI. This URI can be obtained using email, Instant messaging system and phone or even by publishing it publically on forums or social web sites. The URI concern the specific conference room and HTTP request will include related information about current participant membership. [5]

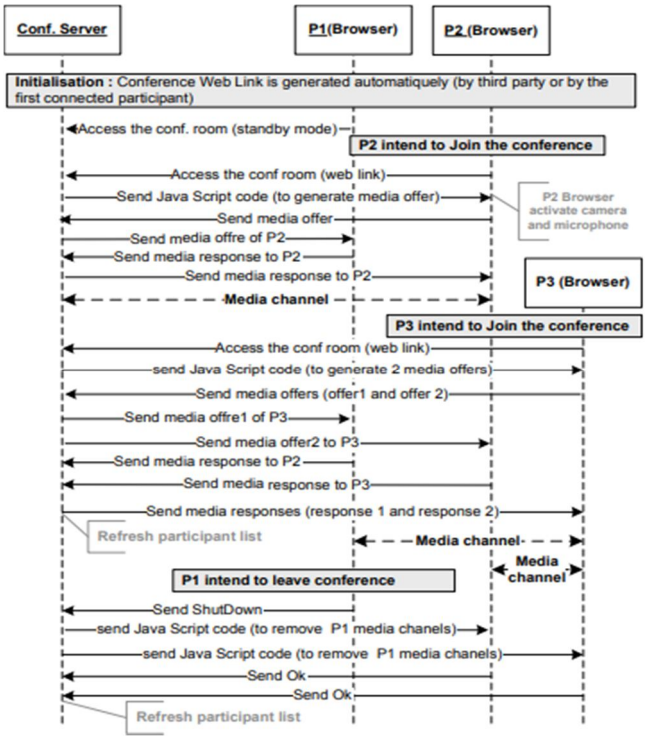


Fig. 4: Example of signaling mechanism [5]

E. Issues faced in WebRTC Video Conferencing

When one goes about applying one of the recog- nized technologies within a totally different ecosystem, it always brings a lot of problems and challenges. The challenges that we are going to discuss, belong to entire web ecosystem where preferably various types of the equipment and device efficiently interact and provide high level services which might be needed in compli- cated scenarios e.g. education

1) *Diverse Access Devices and Networks:* With the passage of time, the diversity of devices that access the web has reasonably increased. There was a time when people could access WWW though personal computers only,, but now there is a wide range of devices with many different capacities and access networks. For example, consider smart phones that have only a limited process power as compared to desktops or laptops, and yet they access the WWW via certain cellular networks.
2) *Screen Size Issues:* When it comes to screen resolutions and quality of the images, smaller screens would not show the ex- act amount of information compared to big screens. Same goes for videoconferencing, sending a high quality video to a person who uses a small screen, would be simply useless since the user will not be able to feel the difference. In complicated scenarios, we can take multi-conferencing as an example; it is not possible to show all the participants at the same time using small screen.
3) *CPU Problems:* WebRTC videoconferencing is a real-time commu- nication; it demands the processor to do decoding, encoding and dispensing of the video and audio at the same period of time. We define this as CPU stress and it relies on many factors e.g. used codec's, quality of the video, audio and their re- spective sizes. The problem gets worse in case of cell phones; not only the cell phone CPUs impose a limitation at how the processes can be carried out, but also a battery drain due to CPU stress.
4) *Availability of Bandwidth and Latency:* When testing with wired Ethernet connection for desktops computers and 3G for cell phones. These differences might not seem important, but they must be considered if communication optimization is our goal. To make things interesting, let us consider a 3G networks; their currently available bandwidth without any notice may vary from previously avail- able bandwidth.

F. Solution

To address the above-mentioned challenges, we are trying to work out the implementation of MCU (Multi- point Control Unit) capable of adapting, redirecting and translating streams of media. While this paper is being written, WebRTC IETF group has already defined almost all of the features of the videoconferencing at the level of protocol and media negotiation. Likewise, the implementation of MCU has to be one of the safe routes since not only it works at the transport layer but also the layers underneath it. [6]

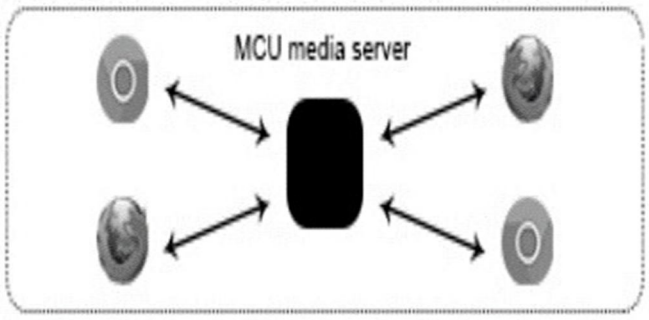


Fig. 5: MCU Browser Interaction [6]

A Multiple Communication Unit has to redirect the media flows among the people taking part in commu- nication, by only enabling them to gather the data from other participants, acting as the middle point of a matrix. As all of the data will have to go through that Multiple Communication Unit, there can be many operations providing the latest services for the video-conference- based communication. MCU acts as centralized bridge to facilitate the real time communication between various devices from different places.

III. COMPARATIVE ANALYSIS OF REAL-TIME COLLABORATION TOOLS

In today's world, there are many news relating which platform to use for video conferencing. Video conferenc- ing apps are witnessing a huge rise in downloads due to lock-downs across the globe. Video conferencing users are seeing an exponential growth as millions of peo- ple are working remotely due to COVID-19 pandemic. According to a report, video conferencing platforms downloads topped 62 million in the month of March 2020. [7]

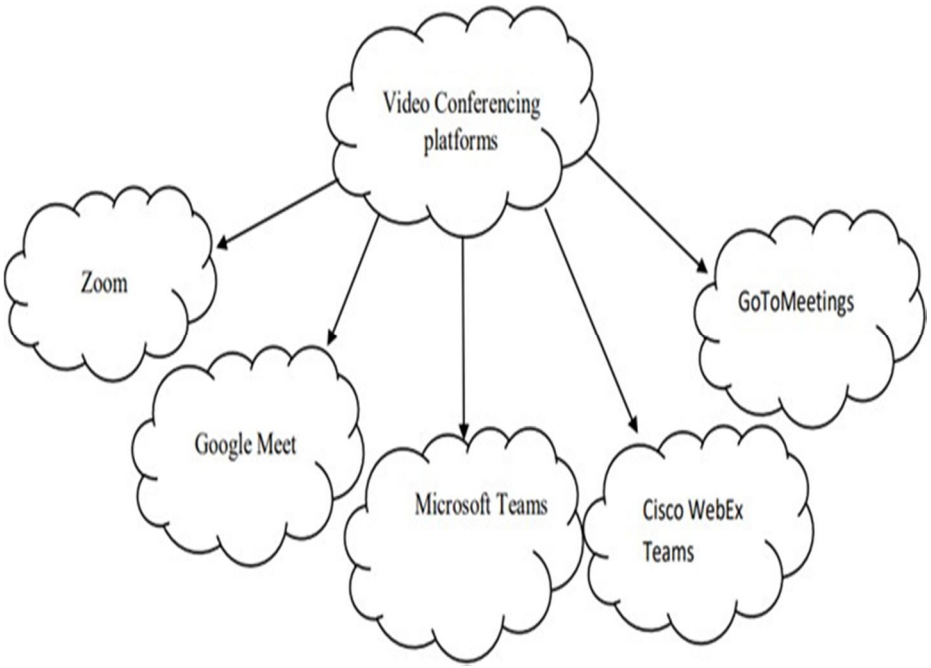


Fig. 7: Video Conferencing Platforms [7]

A. Way to a Safe Video Call

1) *Password Protect your Meetings:* Password protection is always been a best option for anything to hide. The user can prevent unwanted attendees from entering into the meeting as these will be password protected.
2) *Locking up the Meetings:* After the meeting starts, locking up the meeting prevents unwanted users to participate to join-randomly
3) *Switch off Participant Screen Sharing:* This feature is the most important one as many of the users such issues that anything suddenly appears on the screen without even authorising by that particular reason.

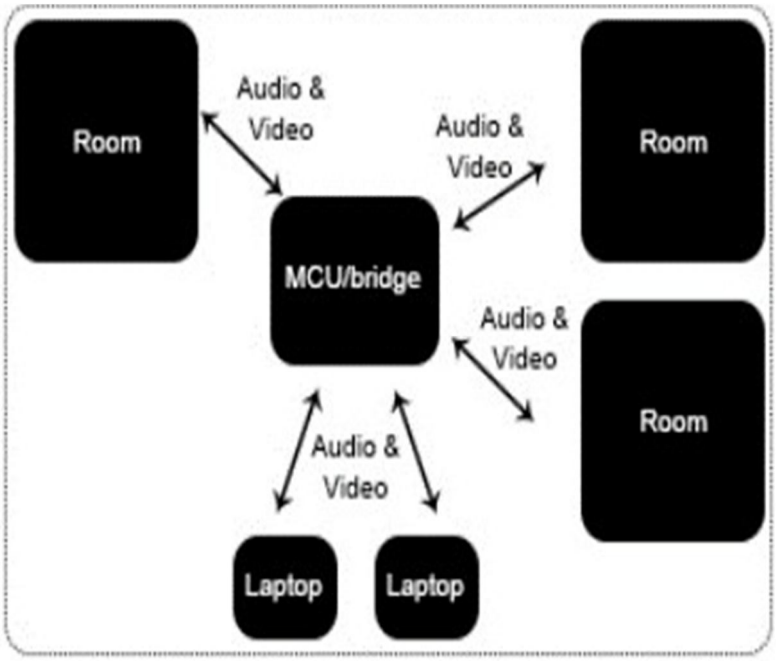


Fig. 6: MCU Bridge for WebRTC [6]

B. Features Comparison

Para m-eters	Zoom	Google Meet	Microsoft Teams	Cisco WebEx Teams	GoTo Meet-ings
No. of par-tici-pants	50-500 ,10000 (webinar)	25, 100 (for Gsuite users)	250-10000 (paid)	1 to 3000	1 to 1000
Auth e-ntica-tion mech-a-nism	Poor authen-tication mecha-nism	Uses SHA-1 HMAC for authenti-cation	Uses OAuth 2.0 for authenti-cation	NA	NA

Auth-or-isation mech-a-nism	No	Yes	Yes	NA	NA
Encry-pted Com-mu-nica-tion	No	Yes	Yes	256-bit AES,PKI	Yes
Compat-ibility	IOS, Android, Mi-crosoft, Linux(parti al)	IOS, Android, Mi-crosoft, Linux	IOS, Android, Mi-crosoft, Linux	IOS, Android, windows, Mac OS, Not Linux	IOS, An-droid, Win-dows, Mac OS, Linux
White-board	Yes	No	Yes	Yes	Yes
Recor-ding Ca-pa-bili-ties	Yes	Yes	Yes (paid)	Computer and services based	Yes
Video qual-ity	HD	HD	VGA, HQ	VGA, HQ, HD	VGA, HD
Chat sup-port	Yes	Yes	Yes	Yes	Yes
Desk-top Shar-ing	Yes	Yes	Yes	Yes	Yes

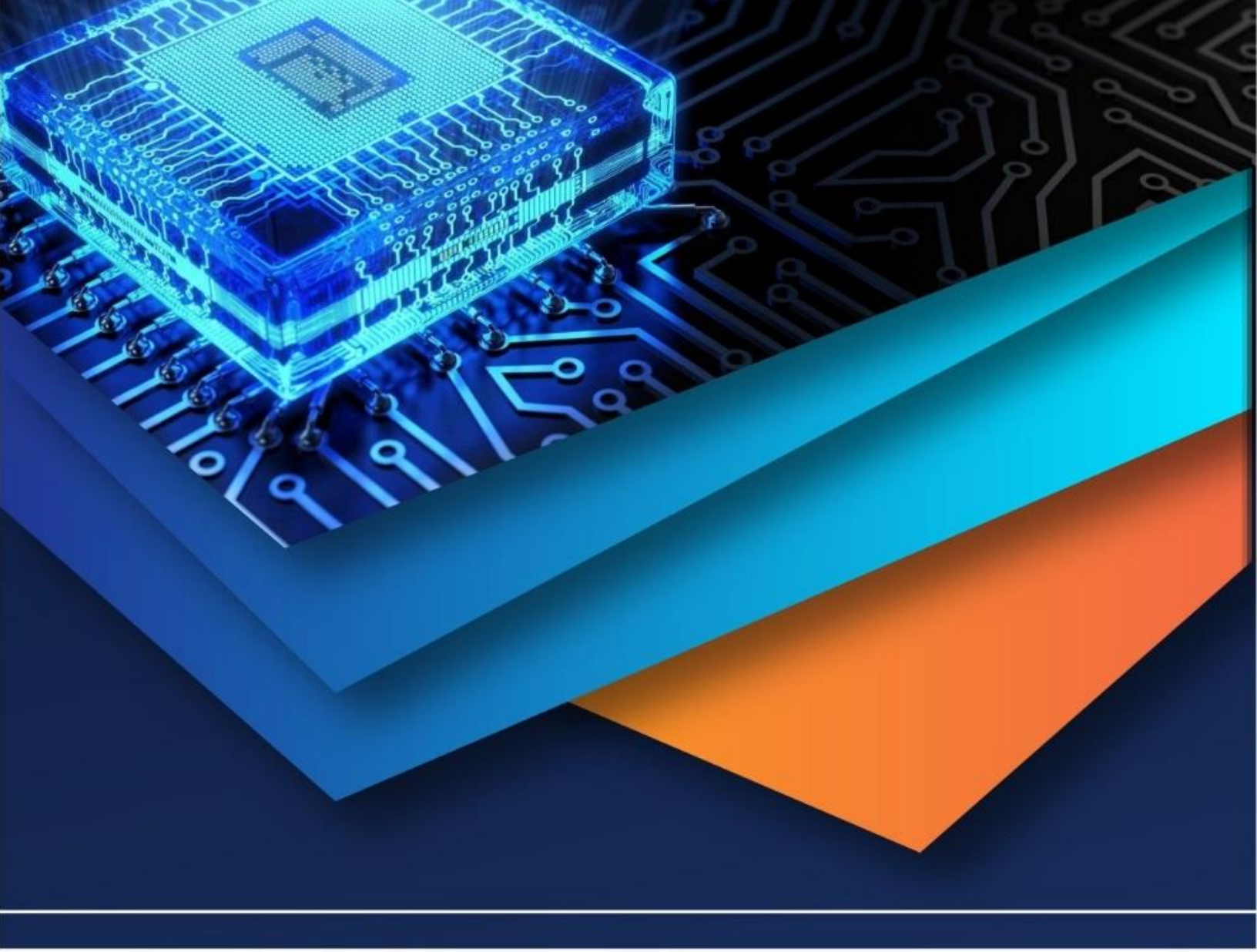
TABLE I: Comparison of Features [7]

IV. CONCLUSION

Thus, we have analysed the working of the WebRTC framework which is the primary tool used to facilitate real-time collaboration and have also looked at its draw- backs and tried to provide a simple solution to overcome the problem. Also, the various tools being currently used for video conferencing have been analysed and compared based on the features they offer.

REFERENCES

[1] "Mediadevices.getusermedia() - web apis: Mdn." [On- line]. Available: https://developer.mozilla.org/en-US/docs/Web/API/MediaDevices/getUserMedia
[2] "Rtcpeerconnection - web apis: Mdn." [Online]. Available: https:
[3] //developer.mozilla.org/en-US/docs/Web/API/RTCPeerConnection
[4] "Rtcdatachannel - web apis: Mdn." [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/RTCDataChannel
[5] S. Dutton et al., "Getting started with webrtc," HTML5 Rocks, vol. 23, 2012.
[6] W. Elleuch, "Models for multimedia conference between browsers based on webrtc," in 2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). IEEE, 2013, pp. 279–284.
[7] M. Pasha, F. Shahzad, and A. Ahmad, "Analysis of challenges faced by webrtc videoconferencing and a remedial architecture," arXiv preprint arXiv:1701.09182, 2017.
[8] R. Singh and S. Awasthi, "Updated comparative analysis on video conferencing platforms-zoom, google meet, microsoft teams, webex teams and gotomeetings," *EasyChair Preprint*, vol. 4026, pp. 1–9, 2020.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)