INTRODUCTION

One might have noticed a sudden popularity in video chat apps since Covid-19 became a pandemic. Video chat apps like Zoom, Skype, Duo, etc serve as a great tool for visual communication across the world from our fingertips. Small and big businesses have understood the importance of video chatting, and want to create customized video chat app. The corona pandemic requires people to maintain social distancing and to stay at home. So, most start-ups and business firms are now making use of video chat apps for communicating with colleagues and clients. This helps them to keep their businesses running.

A good video chat app results in a collaboration of various activities within the business firm without constant travel or physical meetings. It saves time, money, and can facilitate beneficial activities for the overall growth of your business.

You can speed up decisions even when employees are globally dispersed. Besides, the value of messages is amped up with facilities such app screen sharing, recording meetings, audio conferencing, etc.

Several considerations should be considered to build an efficient, reliable Video chat application. Web Real-Time Communications (WebRTC) is a set of Javascript Application Programming Interfaces (APIs) and it enables web developers to develop Real Time Communication (RTC) features into their web-based application without bother any complexities of plugins. Google launched an open source project for web-based real-time communication and known as WebRTC at May 2011 [1]. However, there are a lot of applications in current market like Skype, Google Hangout, FaceTime and most of the applications are using client-server architecture. Users connect using an agent; this agent could be mobile phone, workstation, and other hardware or software application. Then, the agent connects to a central server. However, client-server architecture using in application will increase a lot of system cost such as configuration and maintenance. Peer-to-peer (P2P) architecture is better as it is scalable and reliable than client-server architecture as single nodes failure will not affect against the whole system. Besides, WebRTC system consists of web servers, browsers with different operating system, workstations, tablets, mobile phone. In addition, WebRTC can interoperate with Session Initiation Protocol (SIP), Jingle and Public Switched Telephone Network (PSTN). Due to its interoperability with VoIP and other video communication system like SIP, Jingle, and PSTN;WebRTC is the best options in implementing P2P audio and video calling application.

**Real-time communication without plugins**

Imagine a world where your phone, TV and computer could all communicate on a common platform. Imagine it was easy to add video chat and peer-to-peer data sharing to your web application. That’s possible with WebRTC.

One of the last major challenges for the web is to enable human communication via voice and video: Real Time Communication, RTC for short. RTC should be as natural in a web application as entering text in a text input. Without it, we're limited in our ability to innovate and develop new ways for people to interact.

Historically, RTC has been corporate and complex, requiring expensive audio and video technologies to be licensed or developed in house. Integrating RTC technology with existing content, data and services has been difficult and time consuming, particularly on the web.

Gmail video chat became popular in 2008, and in 2011 Google introduced Hangouts, which use the Google Talk service (as did Gmail). Google bought GIPS, a company which had developed many components required for RTC, such as codecs and echo cancellation techniques. Google open sourced the technologies developed by GIPS and engaged with relevant standards bodies at the IETF and W3C to ensure industry consensus. In May 2011, Ericsson built [the first implementation of WebRTC](https://labs.ericsson.com/developer-community/blog/beyond-html5-peer-peer-conversational-video).

WebRTC implemented open standards for real-time, plugin-free video, audio and data communication. The need was real:

* Many web services used RTC, but needed downloads, native apps or plugins. Those included Skype, Facebook and Google Hangouts.
* Downloading, installing and updating plugins is complex, error prone and annoying.
* Plugins are difficult to deploy, debug, troubleshoot, test and maintain—and may require licensing and integration with complex, expensive technology. It's often difficult to persuade people to install plugins in the first place!

The guiding principles of the WebRTC project are that its APIs should be open source, free, standardized, built into web browsers and more efficient than existing technologies.