What is WebRTC?

WebRTC ("Web Real-Time Communication") is a collection of communications protocols and application programming interfaces that enable real-time communication over peer-to-peer connections. This allows web browsers to not only request resources from backend servers, but also real-time information from browsers of other users.

This enables applications such as video conferencing, file transfer, chat, or desktop sharing without the need of either internal or external plugins.

How WebRTC works?

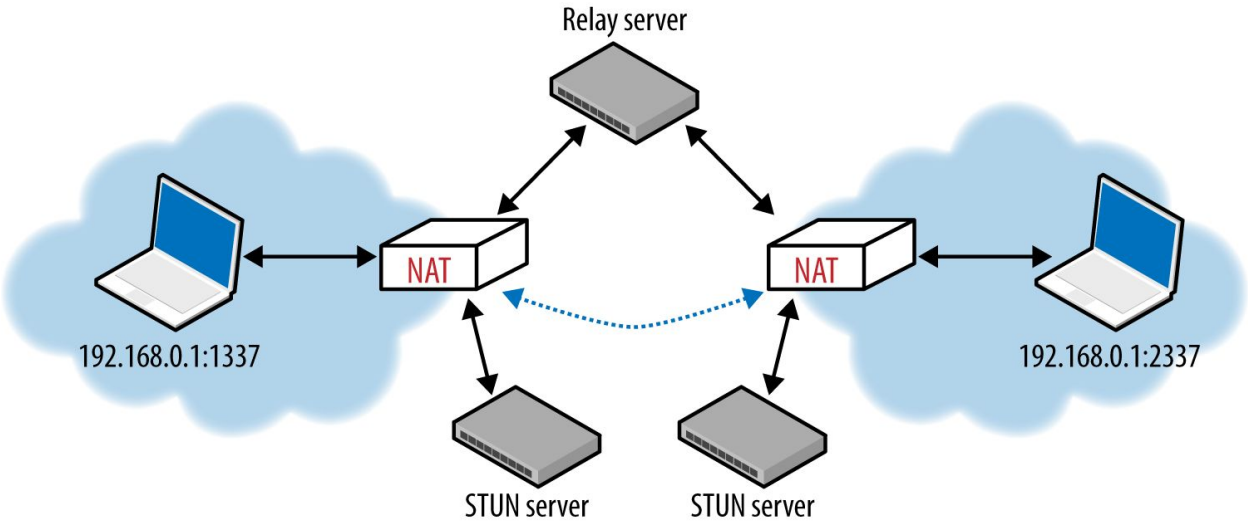
WebRTC has three major components on which it is designed to work.

getUserMedia: This allows browser or the native app gain access to the device's camera and microphone to capture the video.

RTCPeerConnection: This allows audio-video calls set up in the device.

RTCDataChannel: This allows the browser or the native app to get peer-to-peer communication established between the devices.

It is designed to do multiple tasks but setting up this real time peer-to-peer audio video call is the prime advantage.



Therefore, each device gets its public IP. So, as to detect other devices trying to make a connection, signalling data channels are created which support the device to device communication and hence a session is established.

Uses Of WebRTC in our daily use apps-

1. File sharing: Say you're working on a project and want to send massive files to your colleagues. You can directly send it through web browser using WebRTC instead of emailing and uploading it to third-party cloud storage.

2. Multiparty video conferencing: You can create a multi-user video conference using WebRTC and establish a direct peer to peer connection with one another.

3. Screen sharing: You can share your share with your colleagues during meetings and discussions. It helps you to get better and clear ideas for the work in progress and its outcomes.

4. Broadcasting: You may not broadcast with WebRTC, but it enables one-way media transmissions like concerts, speeches, podcasts, and live videos. Few WebRTC platforms also allow you with real-time access to participants attendance.

5. Embedded endpoints: You can connect with live agents when you're on-the-go using WebRTC. You can embed vending machines, ATMs, bus stops, and retail store kiosks with WebRTC engines.

6. Sales enablement: You can help your sales reps using this technology. You can provide ongoing assistance during the purshasing process by integrating your website or application with a WebRTC channel.

7. Emergency response: WebRTC is playing major role in public safety. You can also use location based services and safeguard interactions enabling text communications, audio, and video communications. WebRTC data channel also allows you to have a deeper insight to previously existing communications, when responding to emergency calls.

8. Patient management: You can also reduce patient visits and queue by using WebRTC based solutions for ongoing treatments.

WebRTC platform helps you to give more time to higher priority patients.

Some Popular Apps and Services Which use WebRTC -

Hangouts is an application we use every day and it is powered by WebRTC. SnapChat uses it, and WhatsApp uses it in voice calling service. Bank of America plans to add video chat customer support services, powered by WebRTC, to their ATM machines. PeerCDN uses WebRTC's DataChannel to exchange files across a huge network of clients. An independent developer even used WebRTC to remotely control a robot's movement

Uses Of WebRTC in Liquid Galaxy in the past -

1] multi-user video calls

WebRTC was used to host multi user video calls by Carlos de Dios Felis during Google Summer Of Code 2014.

The project details can be found at - https://www.google-melange.com/archive/gsoc/2014/orgs/lg/projects/cdedios.html

The code to the project can be found at - https://github.com/cdedios/LiquidGalaxyVideoConference

A video regarding it can also be found here - https://youtu.be/-HEx8PQ4EjU

2] JavaScript based Liquid Galaxy

Liquid Galaxy system was built using only the Google Earth Plugin with the views synchronised with JavaScript and WebSockets and/or WebRTC Data Channel. In conjunction with this a "ViewSync.js" library was developed to enable HTML5/WebGL applications to be easily adapted for immersive multi-machine visualisation rigs.

I have also created a simple application using WebRTC -

Here is the link to the repository - https://github.com/programmer-nazi/WebRTC-demo

Here is the live demo - https://codepen.io/programmer_nazi/pen/ppwRyg