Data Communication using WebRTC

Wasim Thabraze, Nuthi Mukund and Sami Ahmed
Department of Computer Science, MVSR Engineering College

Abstract

Using WebRTC we are trying to develop a product called 'Ping' used for audio, mult-video, file and screen sharing.

- Ping uses WebRTC for the source of data exchange and XMPP Server for signalling and transporting.
- Ping works on Browser to Browser connections instead of naive client server approach.
- Ping guarentees high scalability and upto 60% more efficiency than existing systems.

Introduction

- WebRTC is an upcoming standard that aims to improve real time communication among web browsers in peer to peer fashion.
- Ping uses WebRTC which allows browsers to natively support interactive peer to peer communication and realtime collaboration.
- Ping is a low cost highly efficient solution for data communication.
- Ping improves the state of audio/video communication stack in the browser
- Ping leverages the specification to achieve interoperability among web browsers.
- Unlike existing systems, Ping doesn't require any plugins, extensions and signups.

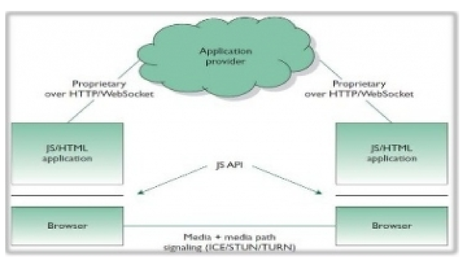

Figure 1: Architecture of Ping

Architecture

We designed an efficient architecture for Ping to withstand high traffic.

- At start browsers do not know each other.
- WebRTC mediates the setup process through the XMPP server of Ping.
- All browsers connected to a particular media object are called to be a part of a swarm and are assigned unique id's.
- Browsers speak to other browsers in the swarm.
- Media flows through the shortest possible path for latency.
- An XMPP server in the backend is used for signalling and initial connection.

Peer Connection

- RTC Peer connection is the WebRTC API that handles stable and efficient communication of streaming data between peers.
- In the real world WebRTC needs servers so the following can happen:
- Browsers discover each other and exchange details like browser name, version, configuration and unique id's.
- WebRTC client applications (Peers) exchange network information.
- Peers exchange data about media such as video format and resolution.

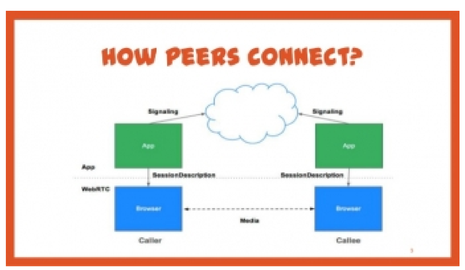

Figure 2: Why WebRTC

Security

- Ping is resistant to several security problems.
- Ping implementations use secure protocols such as DTLS and SRTP.
- Encryption is mandatory for all Ping components including signaling mechanism.
- Ping runs in a browser sandbox and is not a separate process.
- Ping doesn't require separate plugins or components.
- Hashes of the end objects are compared to find any packet loss.

Why WebRTC

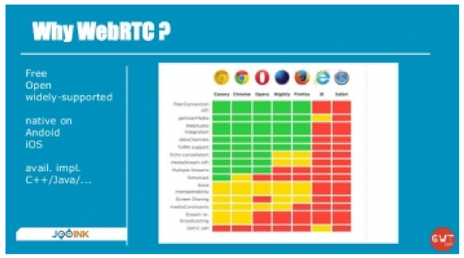

Figure 3: Why WebRTC

- WebRTC is open source so we don't have problems with privacy.
- WebRTC improves connectivity and reduces data costs.
- WebRTC improves latency and is simple to work with.
- WebRTC guarantees high scalability and efficiency even for low bandwidth connections.

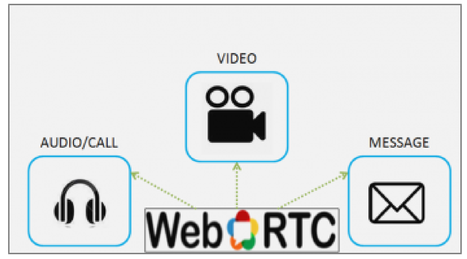

Figure 4: Why WebRTC

Conclusion

- Though there are existing systems like Skype and Hangouts, Ping guarentees around 60% more efficiency.
- Ping also guarentees high scalability. Around 3000 users per second.
- As WebRTC is leveraging communication, Ping uses WebRTC to improved end user experience.

References

- https://webrtc.org
- https://mozilla.org/webrtc
- https://github.com/webrtc

Contact Information

- Web: http://www.pingapp.xyz

wasim@thabraze.me