mediasoup

Cutting Edge WebRTC Video Conferencing

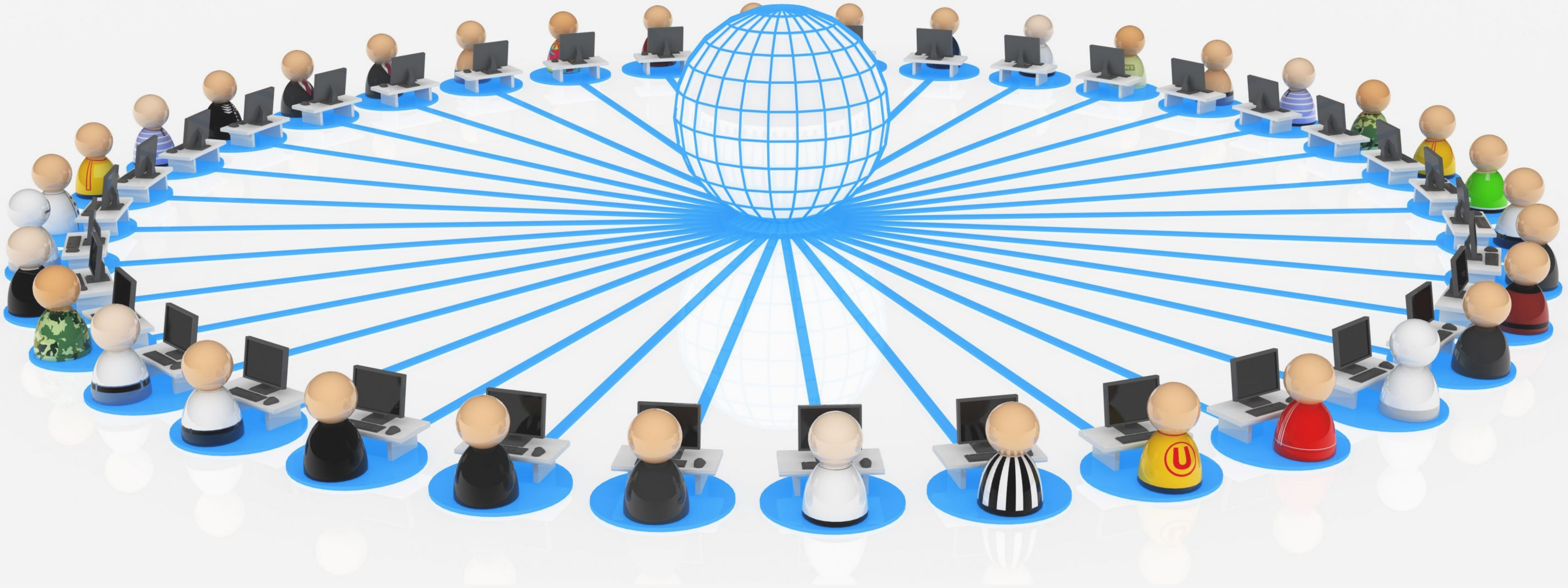
OPENSIPS SUMMIT 2017

IÑAKI BAZ CASTILLO

WHAT IS MEDIASOUP?

A multi-party video solution for Node.js

- For WebRTC endpoints
- Not a standalone media server
- A server-side Node.js module
- JavaScript ES6 API (core written in C++)
- ISC License (do whatever you want)



MULTI-PARTY VIDEO CONFERENCING

TOPOLOGIES

TOPOLOGIES

FULL MESH

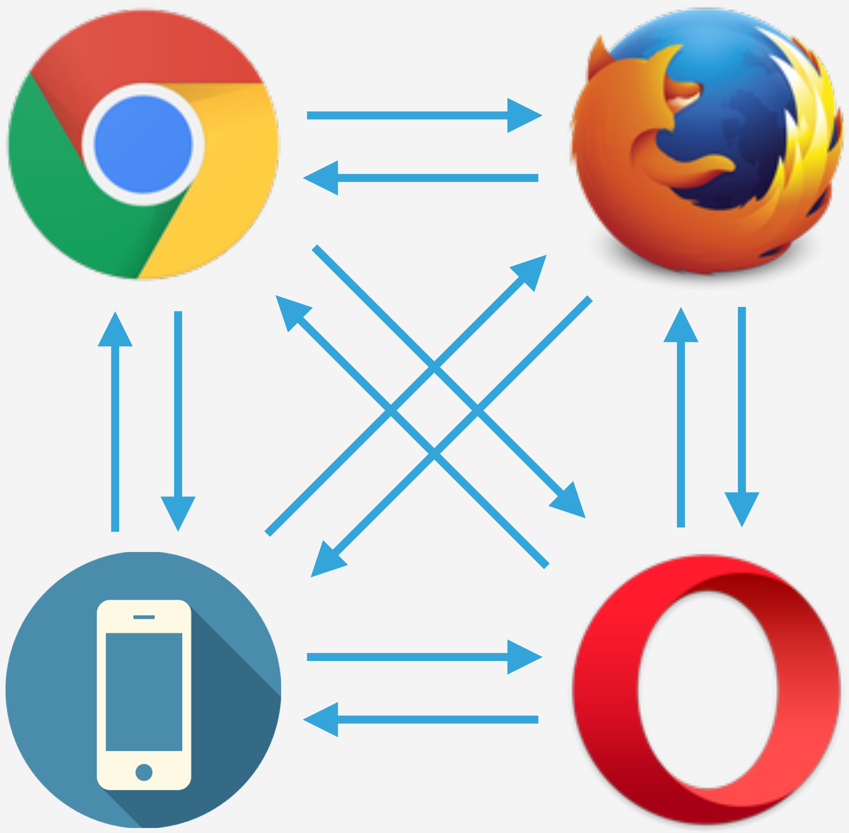
- Each participant sends his audio and video to all other participants

✓ No media server needed

✓ Low latency

✗ Lot of encodings in each participant

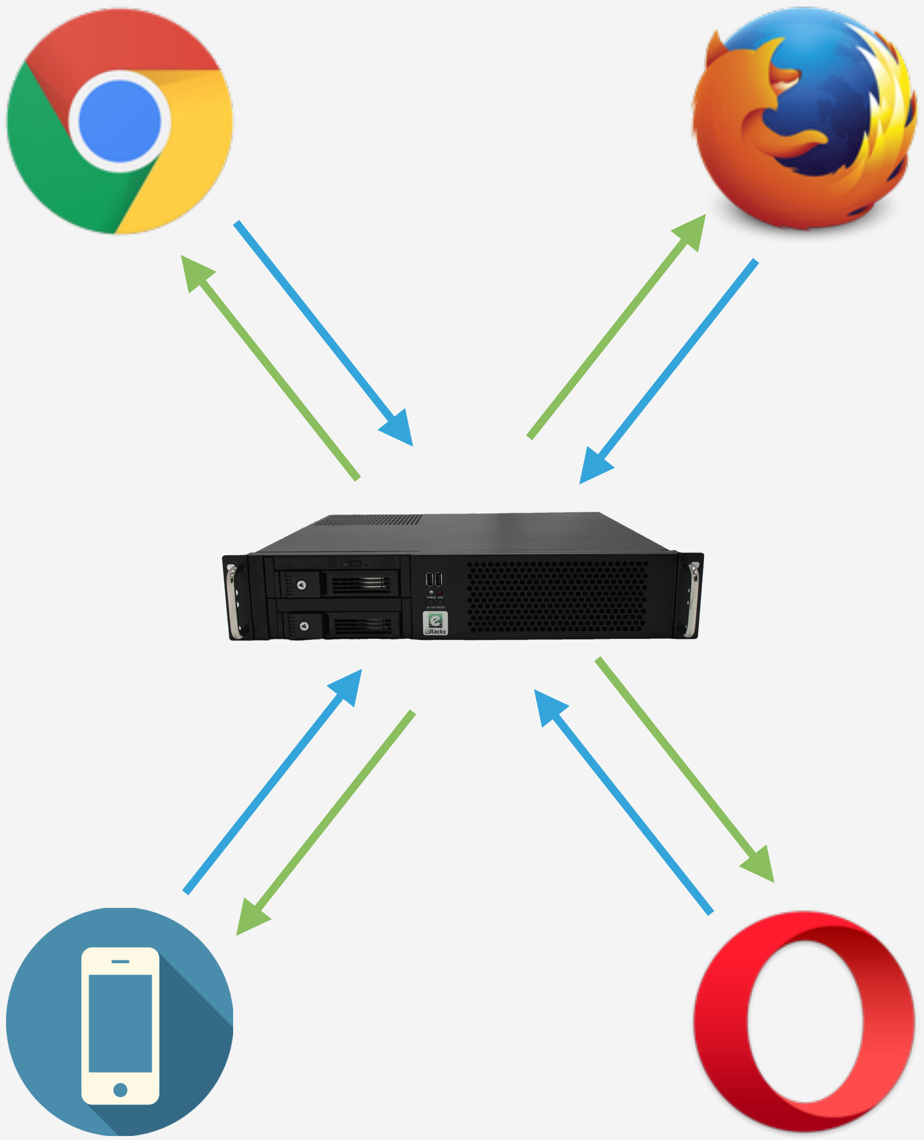
✗ High uplink bandwidth required



TOPOLOGIES

MCU: MULTIPOINT CONTROL UNIT

- Each participant sends his streams to a media server
- The server mixes all the streams and composes a single one
- ✓ Simple at client side: single audio/video mixed stream from the server
- ✓ Low bandwidth required
- ✗ CPU expensive decoding/encoding in server side (high latency)
- ✗ Non flexible client side applications



SFU: SELECTIVE FORWARDING UNIT

- The media server routes the streams of a participant to all the others

✓ High throughput, low latency

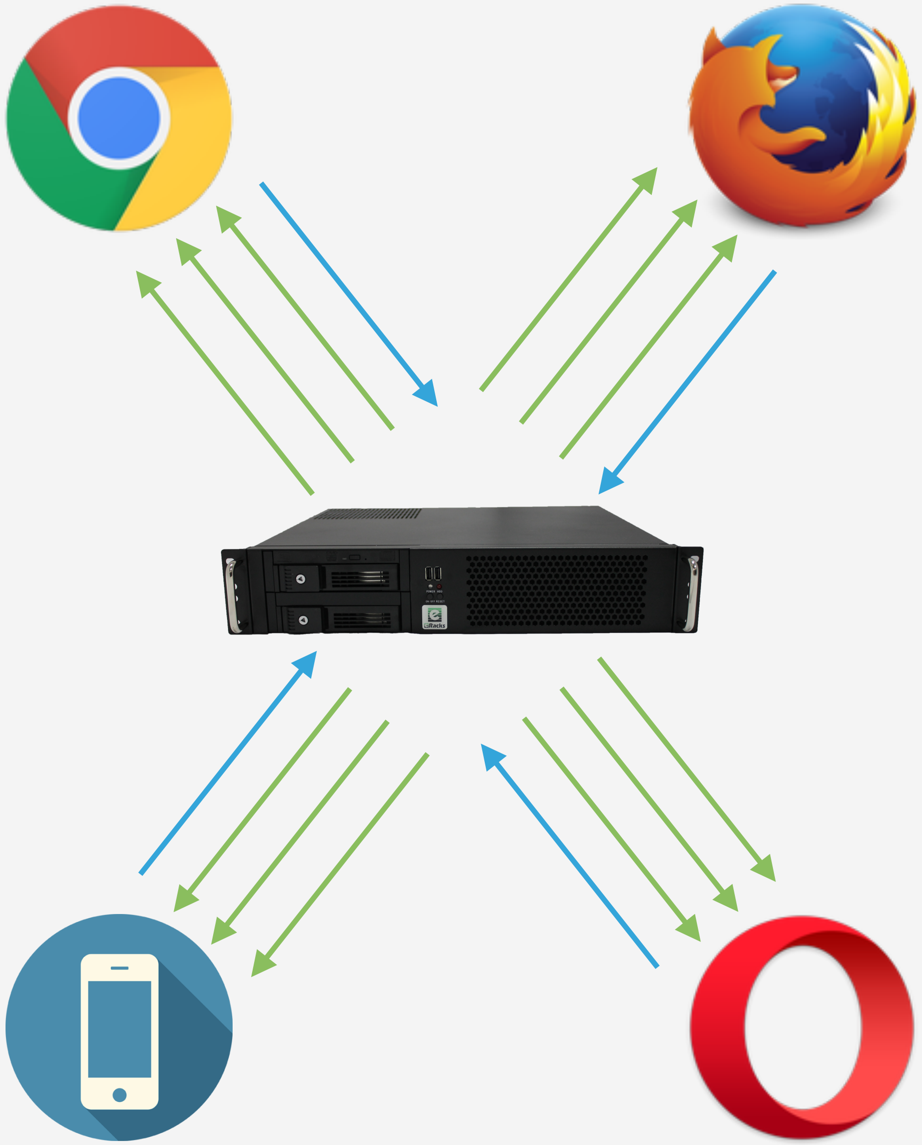
✓ Low CPU at server side (no decoding/encoding)

✓ Good uplink bandwidth usage

✓ Flexible layout at client side

✗ Simulcast/SVC required for real scenarios

✗ Difficult to interoperate with non WebRTC endpoints



TOPOLOGIES

	FULL MESH	MCU	SFU
Client uplink	Very high	Low	Low
Client downlink	Very high	Low	High
Client CPU usage	Very high	Low	Medium
Server CPU usage	-	Very high	Very low
Latency	None	High	Low
Can transcode	-	Yes	No
Requires simulcast/SVC	-	-	Yes
Flexible video layout	Yes	No	Yes

YES!

MEDIASOUP IS AN SFU

MEDIASOUP IS MINIMALIST

MEDIASOUP IS MINIMALIST

WHAT MEDIASOUP IS *NOT*

- It is NOT a standalone server
- It does NOT have “init” scripts for Debian or CentOS
- It does NOT have a “config” file
- It does NOT implement the SIP protocol
- It does NOT implement ANY signaling protocol

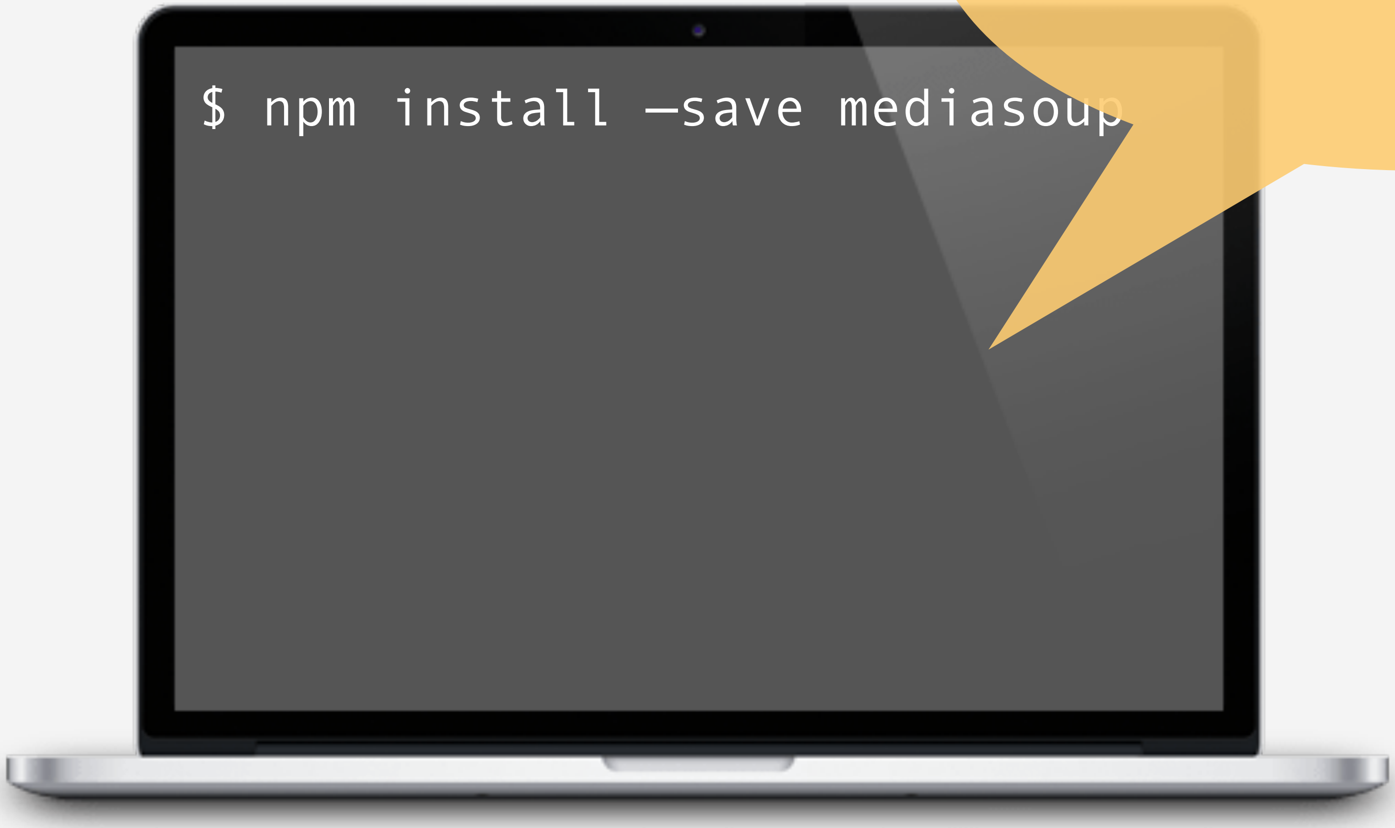
REMEMBER

> MEDIASOUP IS A NODE.JS MODULE

Yours truly

A NODE.JS MODULE

- A Node.js module is a dependency within a project
- Create your Node.js based application
- Add mediasoup as a module into it



```
$ npm install —save mediasoup
```

At the end, mediasoup is yet another dependency entry in the package.json of your Node.js application

PACKAGE.JSON EXAMPLE

```
{
 "name": "my-amazing-multiconference-server-app",
 "version": "1.0.0",
 "description": "Enterprise conferencing application",
 "main": "index.js",
 "author": "My Great Company",
 "license": "UNLICENSED",
 "dependencies": {
   "express": "^4.14.0",
   "mediasoup": "^1.1.0",
   "socket.io": "^1.5.0"
 }
}
```

MEDIASOUP EXPOSES A JAVASCRIPT API

```javascript
// Load mediasoup module
const mediasoup = require('mediasoup');

// Create a mediasoup Server
const server = mediasoup.Server();

// Options for the mediasoup Room
const roomOptions = {
    mediaCodecs: [
        {
            kind      : 'audio',
            name      : 'audio/opus',
            clockRate : 48000
        },
        {
            kind      : 'video',
            name      : 'video/vp8',
            clockRate : 90000
        }
    ]
};

// Create a mediasoup Room
server.createRoom(roomOptions)
    .then((room) => {
        // Got the Room instance
        handleRoom(room);
    });
```

MEDIASOUP API

LOW LEVEL API

mediasoup exposes a low level API based on ORTC

```
// Create a Peer
const peer = room.Peer('alice');

// Create a ICE+DTLS Transport
peer.createTransport(options)
    .then((transport) => {
        transport.setRemoteDtlsParameters(data);
    });

// Create a RtpReceiver to handle audio from the browser
const audioReceiver = peer.RtpReceiver('audio', transport);

// Create a RtpReceiver to handle video from the browser
const videoReceiver = peer.RtpReceiver('video', transport);
```

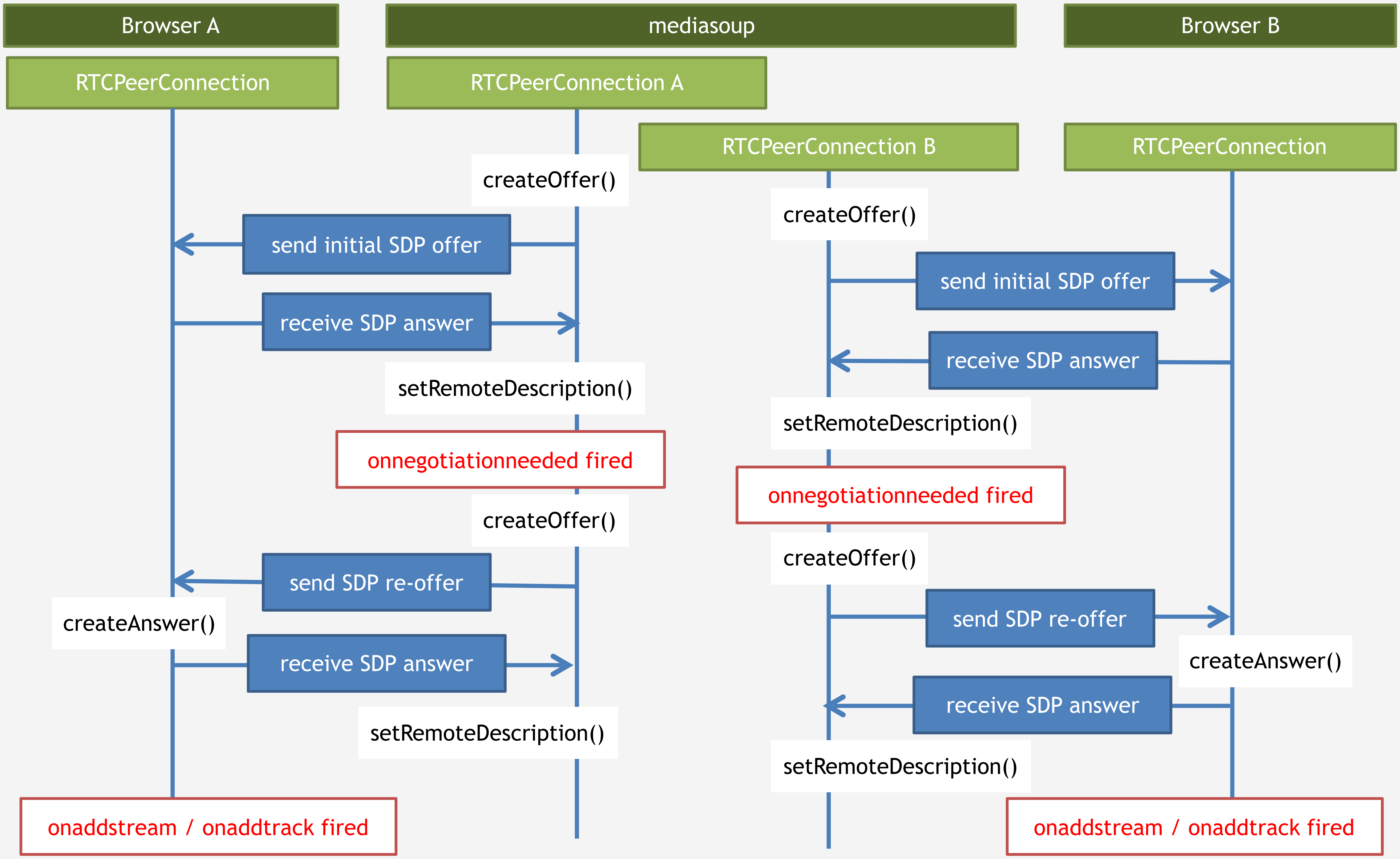
MEDIASOUP API

HIGH LEVEL API

mediasoup also exposes a high level API – WebRTC 1.0 compliant

```
// Create a PeerConnection
const peerconnection = new mediasoup.webrtc.RTCPeerConnection(options);

// Create an initial SDP offer
peerconnection.createOffer()
    .then((offer) => {
        // Apply the obtained SDP offer
        return peerconnection.setLocalDescription(offer);
    })
    .then(() => {
        // Send the SDP offer to the browser
        return browser.sendOffer(peerconnection.localDescription);
    })
    .then((answer) => {
        // Upon receipt of the answer set it into the PeerConnection
        return peerconnection.setRemoteDescription(answer);
    });
```



Browser A

mediasoup

Browser B

RTCPeerConnection

RTCPeerConnection A

RTCPeerConnection B

RTCPeerConnection

createOffer()

send initial SDP offer

receive SDP answer

setRemoteDescription()

onnegotiationneeded fired

createOffer()

send SDP re-offer

createAnswer()

receive SDP answer

setRemoteDescription()

onaddstream / onaddtrack fired

createOffer()

send initial SDP offer

receive SDP answer

setRemoteDescription()

onnegotiationneeded fired

createOffer()

send SDP re-offer

createAnswer()

receive SDP answer

setRemoteDescription()

onaddstream / onaddtrack fired

REMEMBER

> MEDIASOUP JUST FORWARDS MEDIA

Yours truly

MEDIASOUP API

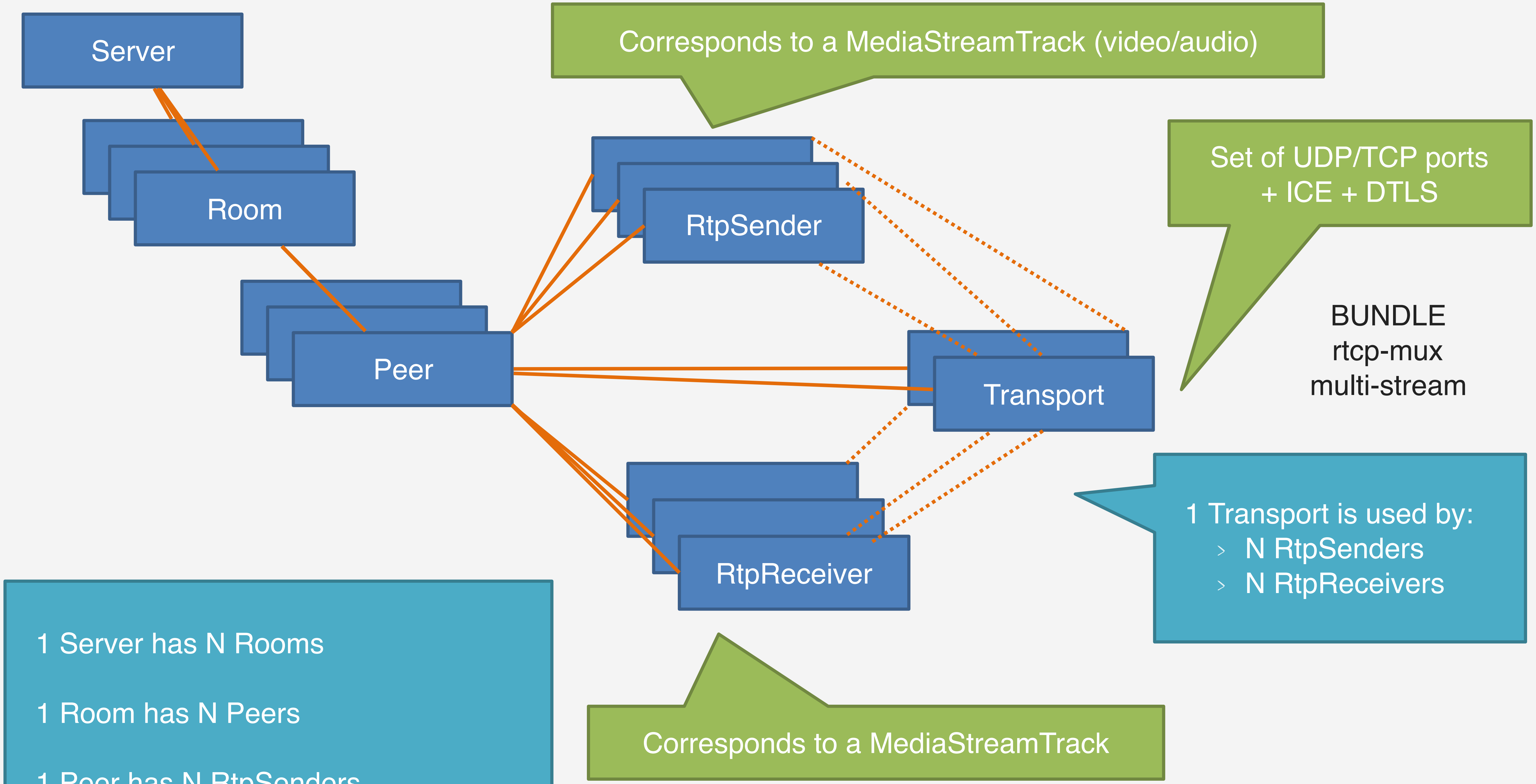
JUST MEDIA

- mediasoup does NOT talk the SIP protocol
- ...nor it talks ANY signaling protocol
- You can use socket.io (for example) to communicate with browsers/endpoints via WebSocket
- ...or build your own protocol
- including SIP!

Similar to the WebRTC API for browsers, mediasoup just handles the media layer

MEDIASOUP-WORKER

INTERNALS



Corresponds to a MediaStreamTrack (video/audio)

Set of UDP/TCP ports + ICE + DTLS

BUNDLE
rtcp-mux
multi-stream

1 Transport is used by:

- N RtpSenders
- N RtpReceivers

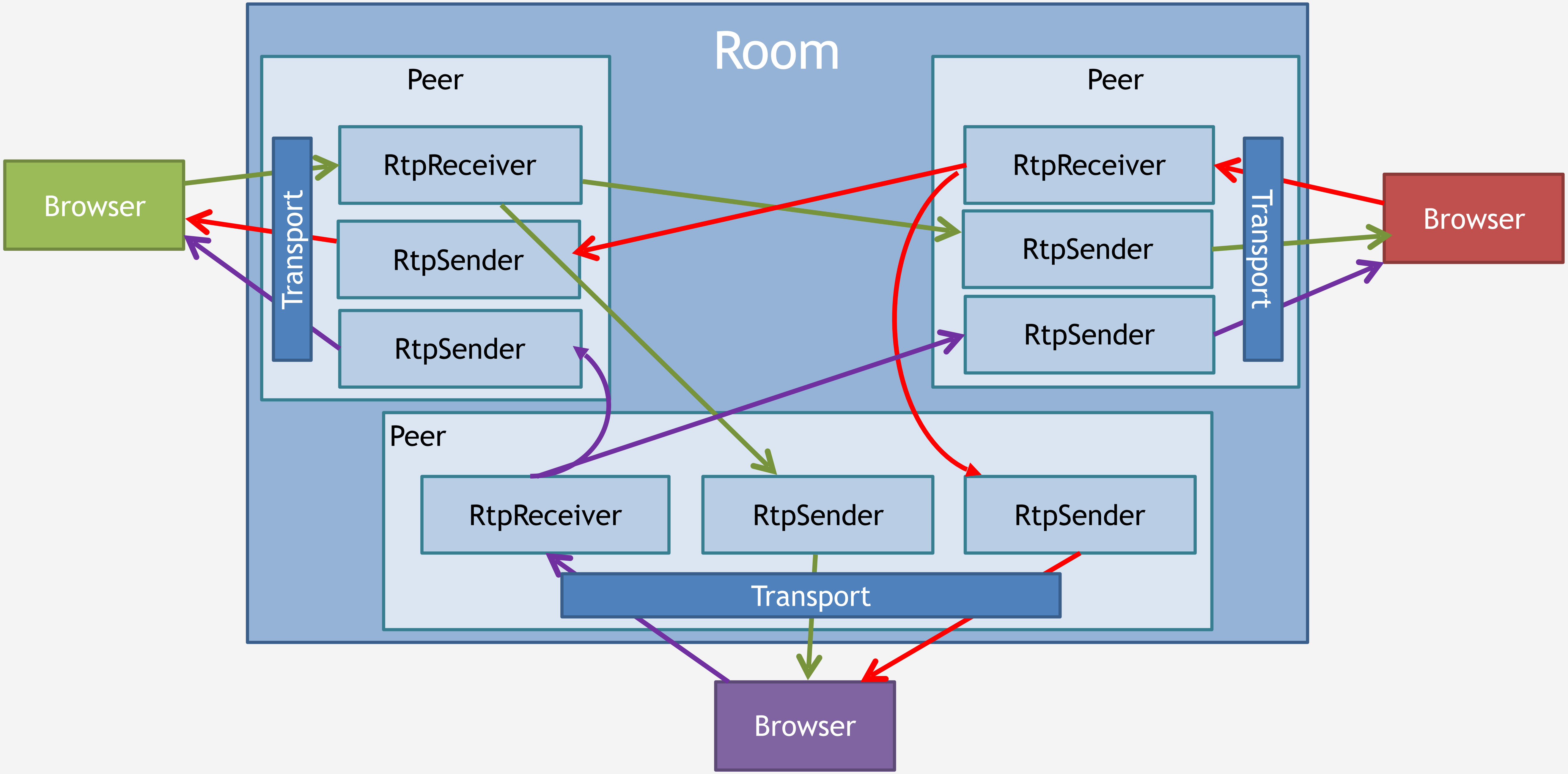
Corresponds to a MediaStreamTrack

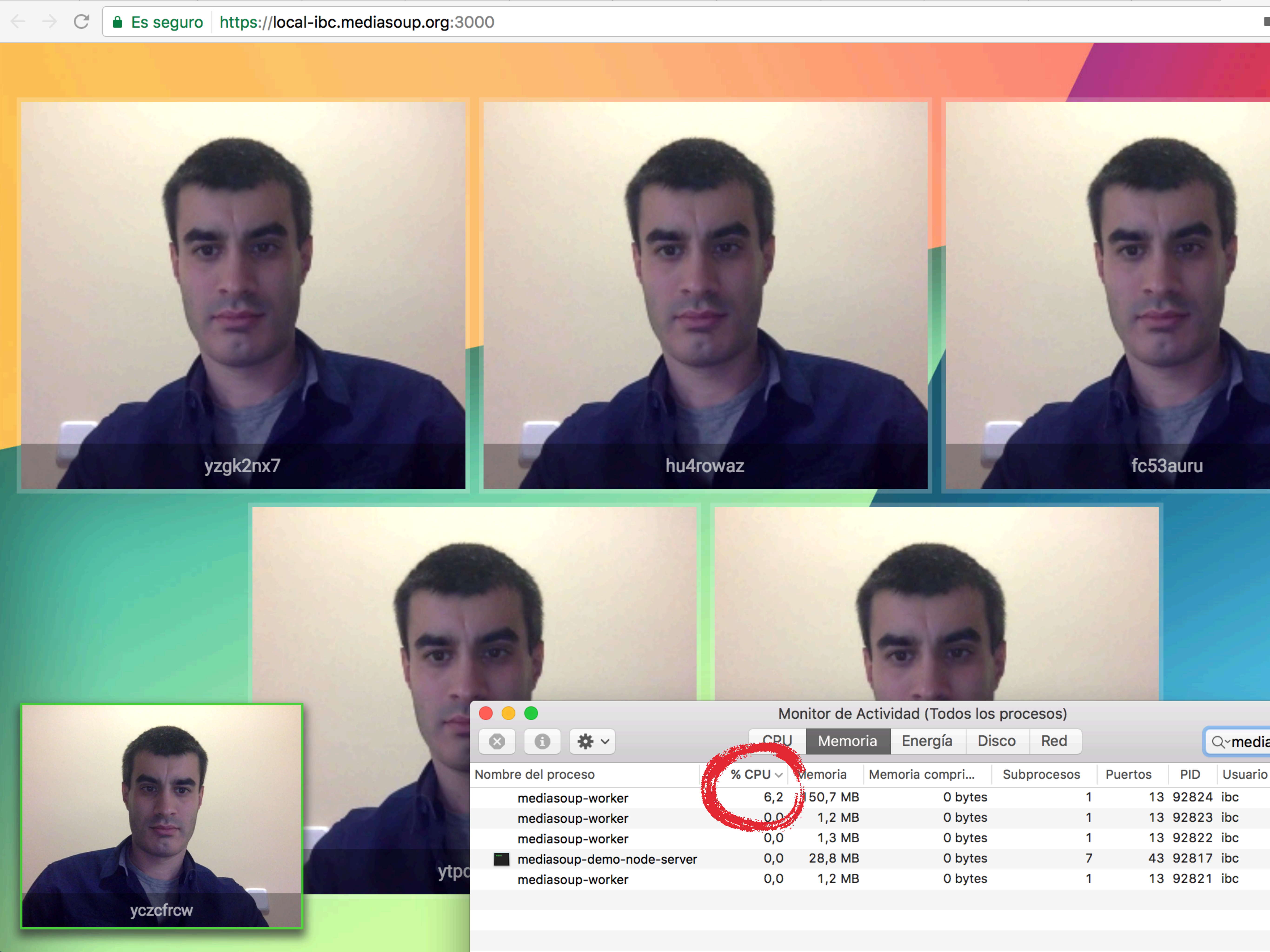
1 Server has N Rooms

1 Room has N Peers

1 Peer has N RtpSenders
1 Peer has N RtpReceivers
1 Peer has N Transports

Thanks to Massie_G for the graphics


Room
Peer
Transport
RtpReceiver
RtpSender
RtpSender
Browser
Peer
RtpReceiver
RtpSender
RtpSender
Transport
Browser
Peer
RtpReceiver
RtpSender
RtpSender
Transport
Browser



Monitor de Actividad (Todos los procesos)

Nombre del proceso	% CPU	Memoria	Memoria compri...	Subprocesos	Puertos	PID	Usuario
mediasoup-worker	6,2	150,7 MB	0 bytes	1	13	92824	ibc
mediasoup-worker	0,0	1,2 MB	0 bytes	1	13	92823	ibc
mediasoup-worker	0,0	1,3 MB	0 bytes	1	13	92822	ibc
mediasoup-demo-node-server	0,0	28,8 MB	0 bytes	7	43	92817	ibc
mediasoup-worker	0,0	1,2 MB	0 bytes	1	13	92821	ibc

MOVING FORWARD

WEBRTC STATUS IN BROWSERS

MULTI-STREAM

- Chrome: Plan-B
  - Single m=audio/video SDP section with an SSRC per remote stream
- Firefox: Unified-Plan
  - A m=audio/video SDP section per remote stream
- **Good news!** mediasoup handles both for you

WebRTC SDP with 6 participants

```
v=0
o=mediasoup 46688400 5 IN IP4 0.0.0.0
s=-
t=0 0
a=ice-lite
a=fingerprint:sha-256 7A:90:A4:BF:F3:C3:DB:F1:F5:96:1B:C4:EF:EE:1B:B0:ED:B4:1C:9B:F4:E3:93:4C:73:0E:6F:8E:4B:B0:E4:0D
a=msid-semantic: WMS *
a=group:BUNDLE recv-audio-track-1 recv-video-track-1 6td9gi9c 39r1spyv 7quqh7q9 vvac3e8h k9kc8f94 9rd4x636 cpe155cu rrmsfy8
m=audio 7 RTP/SAVPF 100
c=IN IP4 127.0.0.1
a=rtpmap:100 opus/48000/2
a=fmtp:100 maxplaybackrate=48000;stereo=1;useinbandfec=1
a=extmap:1 urn:ietf:params:rtp-hdrext:ssrc-audio-level
a=setup:actpass
a=mid:recv-audio-track-1
a=recvonly
a=ice-ufrag:igycbg1all10adgu
a=ice-pwd:7lzxklmreaa9nqdjk242di7g33jacji4
a=candidate:udpcandidate 1 udp 1078862079 94.23.86.78 44082 typ host
a=candidate:tcpcandidate 1 tcp 1078862079 94.23.86.78 43368 typ host tcptype passive
a=end-of-candidates
a=rtcp-mux
a=rtcp-rsize
m=video 7 RTP/SAVPF 101
c=IN IP4 127.0.0.1
a=rtpmap:101 VP8/90000
a=fmtp:101 max-fr=60;max-fs=12288
a=rtcp-fb:101 nack
a=rtcp-fb:101 nack pli
a=rtcp-fb:101 ccm fir
a=rtcp-fb:101 goog-remb
a=setup:actpass
a=mid:recv-video-track-1
a=recvonly
a=ice-ufrag:igycbg1all10adgu
a=ice-pwd:7lzxklmreaa9nqdjk242di7g33jacji4
a=candidate:udpcandidate 1 udp 1078862079 94.23.86.78 44082 typ host
a=candidate:tcpcandidate 1 tcp 1078862079 94.23.86.78 43368 typ host tcptype passive
a=end-of-candidates
a=rtcp-mux
a=rtcp-rsize
```

```
m=audio 7 RTP/SAVPF 100
c=IN IP4 127.0.0.1
a=rtpmap:100 opus/48000/2
a=fmtp:100 minptime=10;useinbandfec=1
a=extmap:1 urn:ietf:params:rtp-hdrext:ssrc-audio-level
a=setup:actpass
a=mid:6td9gi9c
a=msid:8WxEvK2zVVoXFPxbCr4mLCW2G81xIYZ8pVE5 eee732df-a0df-4d35-8942-1e0a60a40c36
a=sendonly
a=ice-ufrag:igycbg1all10adgu
a=ice-pwd:7lzxklmreaa9nqdjk242di7g33jacji4
a=candidate:udpcandidate 1 udp 1078862079 94.23.86.78 44082 typ host
a=candidate:tcpcandidate 1 tcp 1078862079 94.23.86.78 43368 typ host tcptype passive
a=end-of-candidates
a=ssrc:1286266882 cname:28wNswR3FUHxwzj0
a=rtcp-mux
a=rtcp-rsize
m=video 7 RTP/SAVPF 101
c=IN IP4 127.0.0.1
a=rtpmap:101 VP8/90000
a=rtcp-fb:101 ccm fir
a=rtcp-fb:101 nack
a=rtcp-fb:101 nack pli
a=rtcp-fb:101 goog-remb
a=setup:actpass
a=mid:39r1spyv
a=msid:8WxEvK2zVVoXFPxbCr4mLCW2G81xIYZ8pVE5 93b6b706-fa25-4e37-bf1b-bdedbad89780
a=sendonly
a=ice-ufrag:igycbg1all10adgu
a=ice-pwd:7lzxklmreaa9nqdjk242di7g33jacji4
a=candidate:udpcandidate 1 udp 1078862079 94.23.86.78 44082 typ host
a=candidate:tcpcandidate 1 tcp 1078862079 94.23.86.78 43368 typ host tcptype passive
a=end-of-candidates
a=ssrc:845231579 cname:28wNswR3FUHxwzj0
a=rtcp-mux
a=rtcp-rsize
```

```
m=audio 7 RTP/SAVPF 100
c=IN IP4 127.0.0.1
a=rtpmap:100 opus/48000/2
a=fmtp:100 minptime=10;useinbandfec=1
a=extmap:1 urn:ietf:params:rtp-hdrext:ssrc-audio-level
a=setup:actpass
a=mid:7quqh7q9
a=msid:iGN1C3GMq4WPt1KT31UTngKciO8m8kMnBGvb a785da51-570d-43be-afc4-8e7a1d7fe1be
a=sendonly
a=ice-ufrag:igycbg1all10adgu
a=ice-pwd:7lzxklmreaa9nqdjk242di7g33jacji4
a=candidate:udpcandidate 1 udp 1078862079 94.23.86.78 44082 typ host
a=candidate:tcpcandidate 1 tcp 1078862079 94.23.86.78 43368 typ host tcptype passive
a=end-of-candidates
a=ssrc:4207168435 cname:+/8VfCudrb3JfFgs
a=rtcp-mux
a=rtcp-rsize
m=video 7 RTP/SAVPF 101
c=IN IP4 127.0.0.1
a=rtpmap:101 VP8/90000
a=rtcp-fb:101 ccm fir
a=rtcp-fb:101 nack
a=rtcp-fb:101 nack pli
a=rtcp-fb:101 goog-remb
a=setup:actpass
a=mid:vvac3e8h
a=msid:iGN1C3GMq4WPt1KT31UTngKciO8m8kMnBGvb 1d60cd3e-4fb0-4302-a682-01fd78dcdb83
a=sendonly
a=ice-ufrag:igycbg1all10adgu
a=ice-pwd:7lzxklmreaa9nqdjk242di7g33jacji4
a=candidate:udpcandidate 1 udp 1078862079 94.23.86.78 44082 typ host
a=candidate:tcpcandidate 1 tcp 1078862079 94.23.86.78 43368 typ host tcptype passive
a=end-of-candidates
a=ssrc:1414267784 cname:+/8VfCudrb3JfFgs
a=rtcp-mux
a=rtcp-rsize
```

```
m=audio 7 RTP/SAVPF 100
c=IN IP4 127.0.0.1
a=rtpmap:100 opus/48000/2
a=fmtp:100 minptime=10;useinbandfec=1
a=extmap:1 urn:ietf:params:rtp-hdrext:ssrc-audio-level
a=setup:actpass
a=mid:k9kc8f94
a=msid:J92bKibbUUNt4YCVSPmYZ5hf99roTh0ZWsz7 ecf0bec4-30e1-4032-8166-86b9c337a05e
a=sendonly
a=ice-ufrag:igycbg1all10adgu
a=ice-pwd:7lzxklmreaa9nqdjk242di7g33jacji4
a=candidate:udpcandidate 1 udp 1078862079 94.23.86.78 44082 typ host
a=candidate:tcpcandidate 1 tcp 1078862079 94.23.86.78 43368 typ host tcptype passive
a=end-of-candidates
a=ssrc:3189387350 cname:jbGCvnwerAH2WScE
a=rtcp-mux
a=rtcp-rsize
m=video 7 RTP/SAVPF 101
c=IN IP4 127.0.0.1
a=rtpmap:101 VP8/90000
a=rtcp-fb:101 ccm fir
a=rtcp-fb:101 nack
a=rtcp-fb:101 nack pli
a=rtcp-fb:101 goog-remb
a=setup:actpass
a=mid:9rd4x636
a=msid:J92bKibbUUNt4YCVSPmYZ5hf99roTh0ZWsz7 75df520e-daad-4eb2-b8f1-2a7e6fa778d8
a=sendonly
a=ice-ufrag:igycbg1all10adgu
a=ice-pwd:7lzxklmreaa9nqdjk242di7g33jacji4
a=candidate:udpcandidate 1 udp 1078862079 94.23.86.78 44082 typ host
a=candidate:tcpcandidate 1 tcp 1078862079 94.23.86.78 43368 typ host tcptype passive
a=end-of-candidates
a=ssrc:1674535480 cname:jbGCvnwerAH2WScE
a=rtcp-mux
a=rtcp-rsize
```

```
m=audio 7 RTP/SAVPF 100
c=IN IP4 127.0.0.1
a=rtpmap:100 opus/48000/2
a=fmtp:100 minptime=10;useinbandfec=1
a=extmap:1 urn:ietf:params:rtp-hdrext:ssrc-audio-level
a=setup:actpass
a=mid:cpe155cu
a=msid:5LWkclP8CcvGjlwlDSwvVDVxlBymio46LOg0 365ba5ad-fef4-4eac-8416-2ffc8729d51e
a=sendonly
a=ice-ufrag:igycbg1all10adgu
a=ice-pwd:7lzxklmreaa9nqdjk242di7g33jacji4
a=candidate:udpcandidate 1 udp 1078862079 94.23.86.78 44082 typ host
a=candidate:tcpcandidate 1 tcp 1078862079 94.23.86.78 43368 typ host tcptype passive
a=end-of-candidates
a=ssrc:2214209061 cname:4PJw/3GgaWOdyWVW
a=rtcp-mux
a=rtcp-rsize
m=video 7 RTP/SAVPF 101
c=IN IP4 127.0.0.1
a=rtpmap:101 VP8/90000
a=rtcp-fb:101 ccm fir
a=rtcp-fb:101 nack
a=rtcp-fb:101 nack pli
a=rtcp-fb:101 goog-remb
a=setup:actpass
a=mid:rrmsfy8t
a=msid:5LWkclP8CcvGjlwlDSwvVDVxlBymio46LOg0 14b06ad1-b7d8-409a-9a7f-cf2b57e37c50
a=sendonly
a=ice-ufrag:igycbg1all10adgu
a=ice-pwd:7lzxklmreaa9nqdjk242di7g33jacji4
a=candidate:udpcandidate 1 udp 1078862079 94.23.86.78 44082 typ host
a=candidate:tcpcandidate 1 tcp 1078862079 94.23.86.78 43368 typ host tcptype passive
a=end-of-candidates
a=ssrc:2390228262 cname:4PJw/3GgaWOdyWVW
a=rtcp-mux
a=rtcp-rsize
```

EOF

WEBRTC STATUS IN BROWSERS

SIMULCAST & SVC

- Chrome:
  - Own non-standard and non-documented simulcast mechanism
  - SVC in VP8 and VP9
- Firefox:
  - Implements draft-ietf-mmusic-sdp-simulcast
  - No SVC yet

MEDIASOUP ROADMAP

MEDIASOUP ROADMAP

npm v1.1.0

1.X.Y

- ✓ ICE / DTLS / RTP over UDP / TCP on IPv4 / IPv6
- ✓ Multi-stream over a single transport
- ✓ Plan-B (Chrome) and Unified-Plan (Firefox)
- ✓ Video congestion control via REMB

2.X.Y

- JavaScript SDK for client and server side
- Simulcast & SVC
- Microsoft Edge support

> Connected, feel free to talk now

You: DAD

You: ISTHATYOU?!?!?!?!?!?!!?!

THE CORNERSTONE

YOUR APPLICATION

YOUR APPLICATION

THINGS YOU CAN DO WITH MEDIASOUP

- Online poker game
- e-learning web application
- Customer assistance app
- Social app for meeting new people
- ...and yes, a boring conferencing app for enterprises

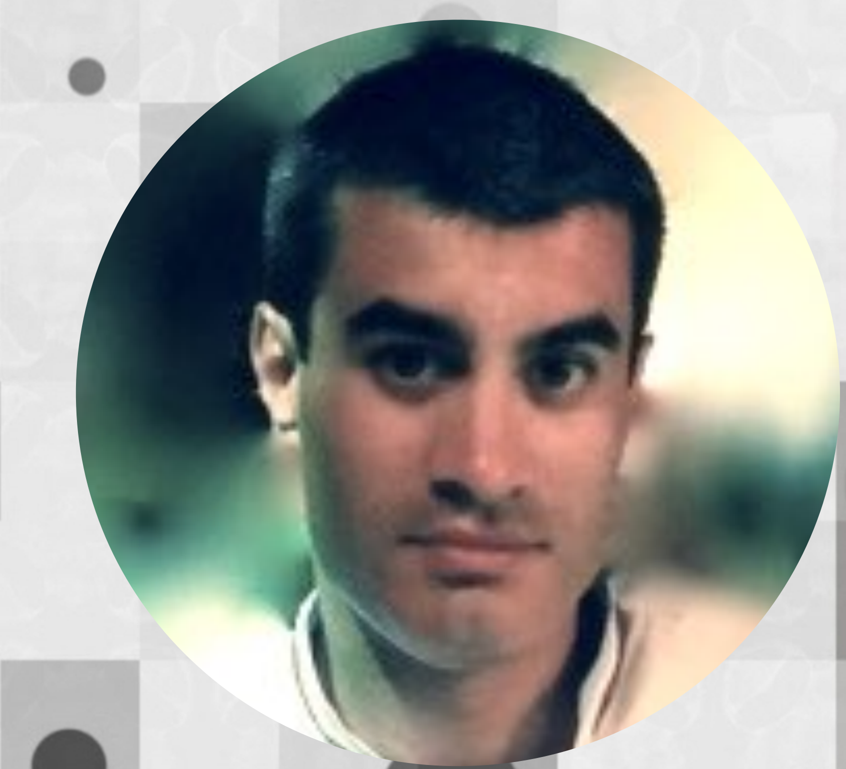
LET’S DEMO

LET'S CONNECT

- WiFi SSID: **Radisson_Guest**
- Use Chrome or Firefox (desktop or Android)
- Open this URL:

  https://demo.mediasoup.org#room-id=opensips

- Say “yes” to everything



THE TEAM

José Luis Millán

Iñaki Baz Castillo

THE END

THANKS !!!

Iñaki Baz Castillo

https://mediasoup.org

https://inakibaz.me