
Google
Developers

WebRTC

Plugin-free realtime communication

Justin Uberti - WebRTC Tech Lead, Google
Sam Dutton - Developer Advocate, Google Chrome

Watch this presentation on YouTube

Low cost, high quality audio and video communication

...and data!

“WebRTC is a new front in the long war for an open and unencumbered web”

Brendan Eich
– Mozilla CTO and inventor of JavaScript

WebRTC on Chrome

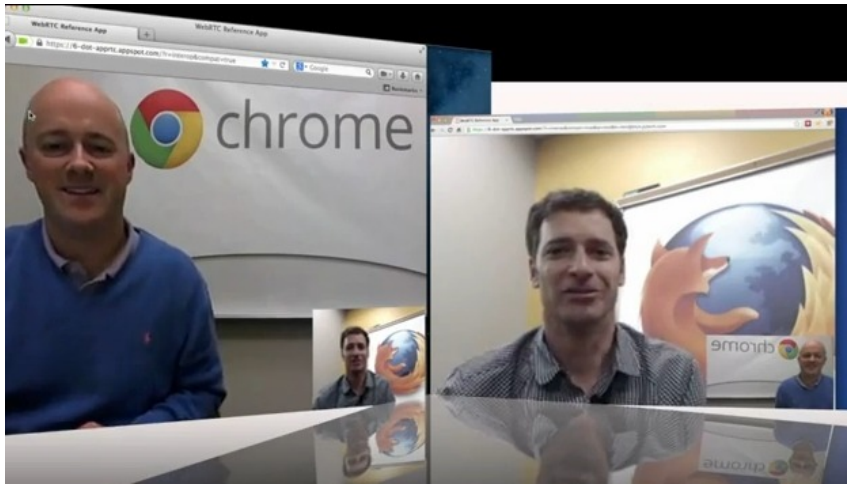
- Bringing real-time communications to the web
- Building a state-of-the-art media stack into Chrome
- Developing a new communications platform

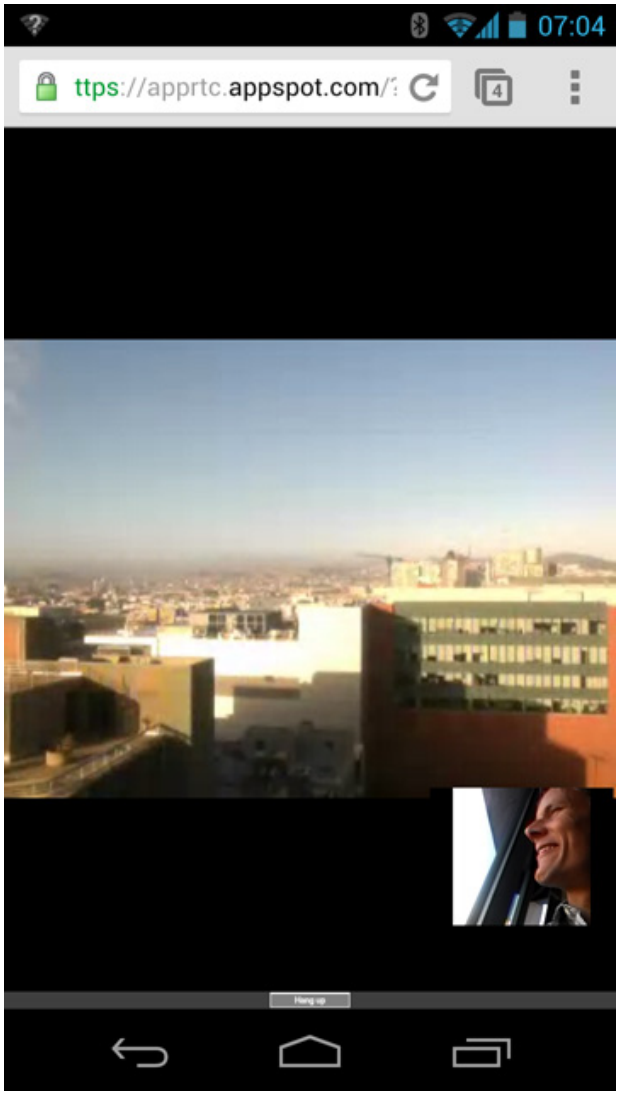
“Voice is just another JS application”

Henning Schulzrinne
– CTO, US FCC

WebRTC across platforms

- Chrome
- Chrome for Android
- Firefox
- Opera
- Native Java and Objective-C bindings





1,000,000,000+

WebRTC Endpoints


Zingaya
CITRIX
conversat.io
Sharefest
bistri
hookflash
thrupoint
TenHands
vLine
real-time video platform
überconference
browser meeting
gocast.it
Meet and Share on Live Video
voxeo
open·tok™
brought to you by tokbox
Meetecho
vidtel

The WebRTC APIs

Three main tasks

- Acquiring audio and video
- Communicating audio and video
- Communicating arbitrary data

Three main JavaScript APIs

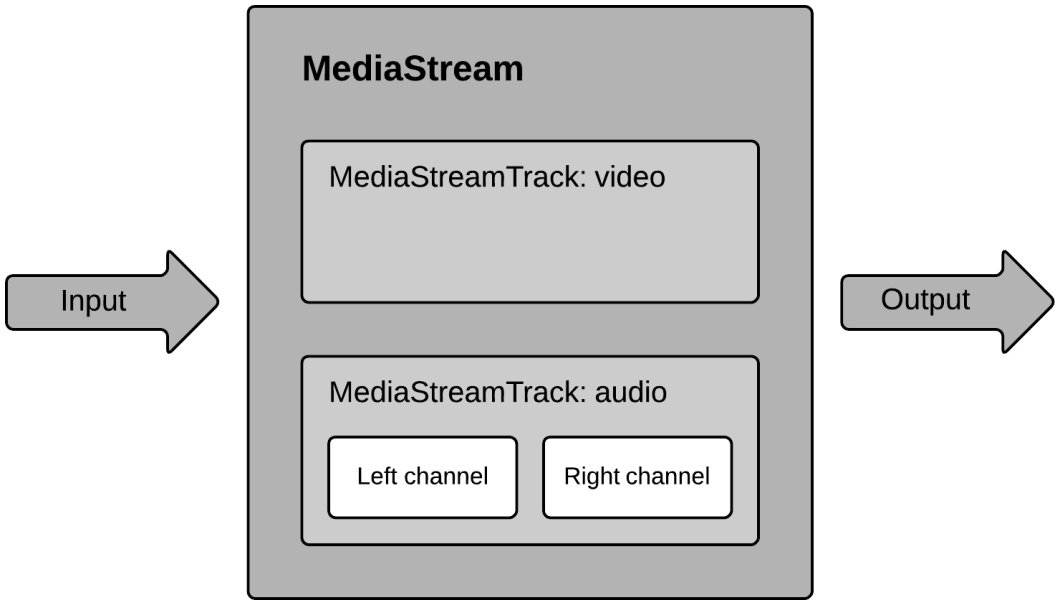
- MediaStream (aka getUserMedia)
- RTCPeerConnection
- RTCDataChannel

MediaStream

Acquiring audio and video

MediaStream

- Represents a stream of audio and/or video
- Can contain multiple 'tracks'
- Obtain a MediaStream with `navigator.getUserMedia()`



MediaStream

aka getUserMedia

JAVASCRIPT

```
var constraints = {video: true};

function successCallback(stream) {
  var video = document.querySelector("video");
  video.src = window.URL.createObjectURL(stream);
}

function errorCallback(error) {
  console.log("navigator.getUserMedia error: ", error);
}

navigator.getUserMedia(constraints, successCallback, errorCallback);
```

simpl.info/gum

idevelop.github.com/ascii-camera

FaceKat

webcamtoy.com

Constraints

- Controls the contents of the MediaStream
- Media type, resolution, frame rate

JAVASCRIPT

```
video: {
  mandatory: {
    minWidth: 640,
    minHeight: 360
  },
  optional [{
    minWidth: 1280,
    minHeight: 720
  }]
}
```

simpl.info/getusermedia/constraints

getUserMedia + Web Audio

JAVASCRIPT

```
// Success callback when requesting audio input stream
function gotStream(stream) {
    var audioContext = new webkitAudioContext();

    // Create an AudioNode from the stream
    var mediaStreamSource = audioContext.createMediaStreamSource(stream);

    // Connect it to the destination or any other node for processing!
    mediaStreamSource.connect(audioContext.destination);
}

navigator.webkitGetUserMedia({audio:true}, gotStream);
```

Make sure to enable Web Audio Input in about:flags!

webaudiodemos.appspot.com/AudioRecorder

gUM screencapture

JAVASCRIPT

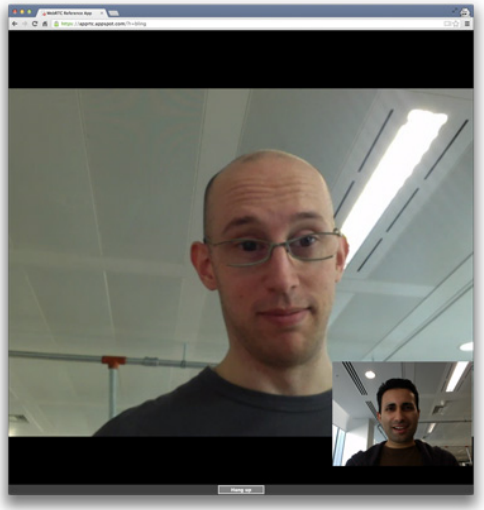
```
var constraints = {
  video: {
    mandatory: {
      chromeMediaSource: 'screen'
    }
  }
};

navigator.webkitGetUserMedia(constraints, gotStream);
```

gUM screencapture

RTCPeerConnection

Audio and video communication between peers

Communicate Media Streams



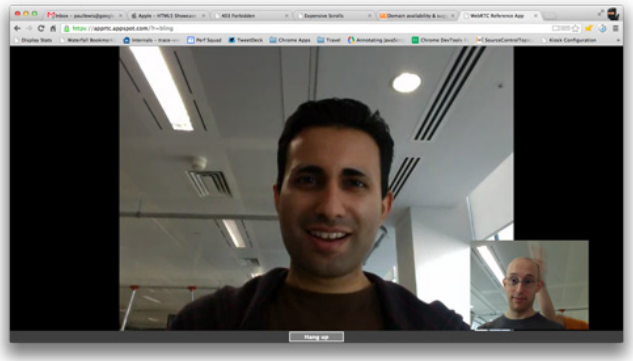
→

getUserMedia

+

RTCPeerConnection

←

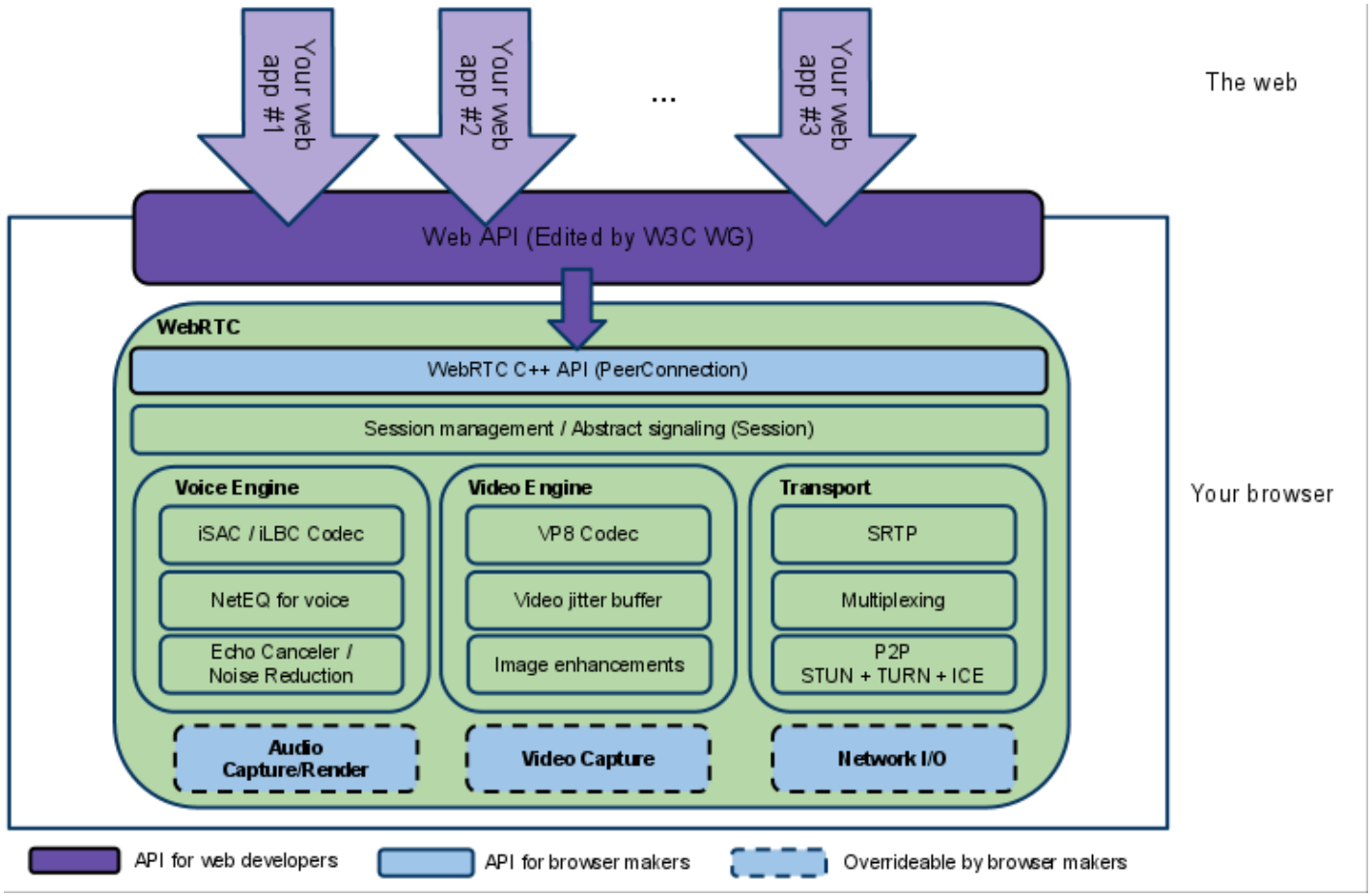


RTCPeerConnection does a lot

- Signal processing
- Codec handling
- Peer to peer communication
- Security
- Bandwidth management

...

WebRTC architecture



RTCPeerConnection sample

JAVASCRIPT

```
pc = new RTCPeerConnection(null);
pc.onaddstream = gotRemoteStream;
pc.addStream(localStream);
pc.createOffer(gotOffer);

function gotOffer(desc) {
  pc.setLocalDescription(desc);
  sendOffer(desc);
}

function gotAnswer(desc) {
  pc.setRemoteDescription(desc);
}

function gotRemoteStream(e) {
  attachMediaStream(remoteVideo, e.stream);
}
```

simpl.info/pc

RTCDataChannel

Bidirectional communication of arbitrary data between peers

Communicate arbitrary data

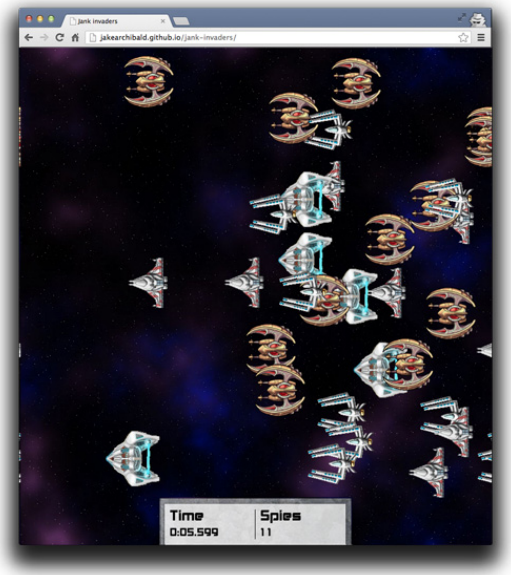
→

RTCDataChannel

+

RTCPeerConnection

←



RTCDataChannel

- Same API as WebSockets
- Ultra-low latency
- Unreliable or reliable
- Secure

RTCDataChannel API

JAVASCRIPT

```
var pc = new webkitRTCPeerConnection(servers,
  {optional: [{RtpDataChannels: true}]});

pc.ondatachannel = function(event) {
  receiveChannel = event.channel;
  receiveChannel.onmessage = function(event){
    document.querySelector("div#receive").innerHTML = event.data;
  };
};

sendChannel = pc.createDataChannel("sendDataChannel", {reliable: false});

document.querySelector("button#send").onclick = function (){
  var data = document.querySelector("textarea#send").value;
  sendChannel.send(data);
};
```

simpl.info/dc

Sharefest

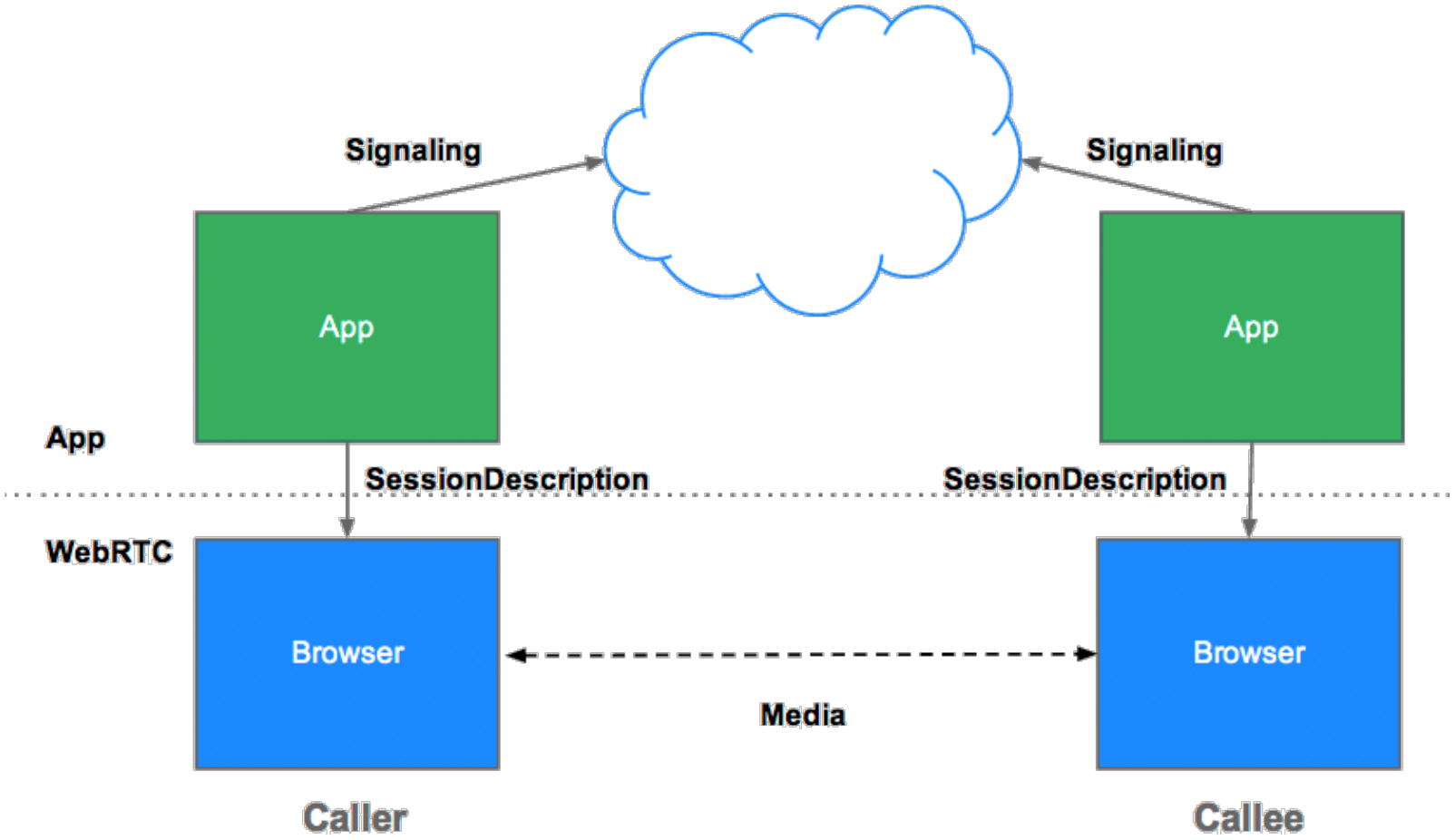
Servers and Protocols

Peer to peer — but we need servers :^\

Abstract Signaling

- Need to exchange 'session description' objects:
  - What formats I support, what I want to send
  - Network information for peer-to-peer setup
- Can use any messaging mechanism
- Can use any messaging protocol

Signaling Diagram



An RTCSessionDescription

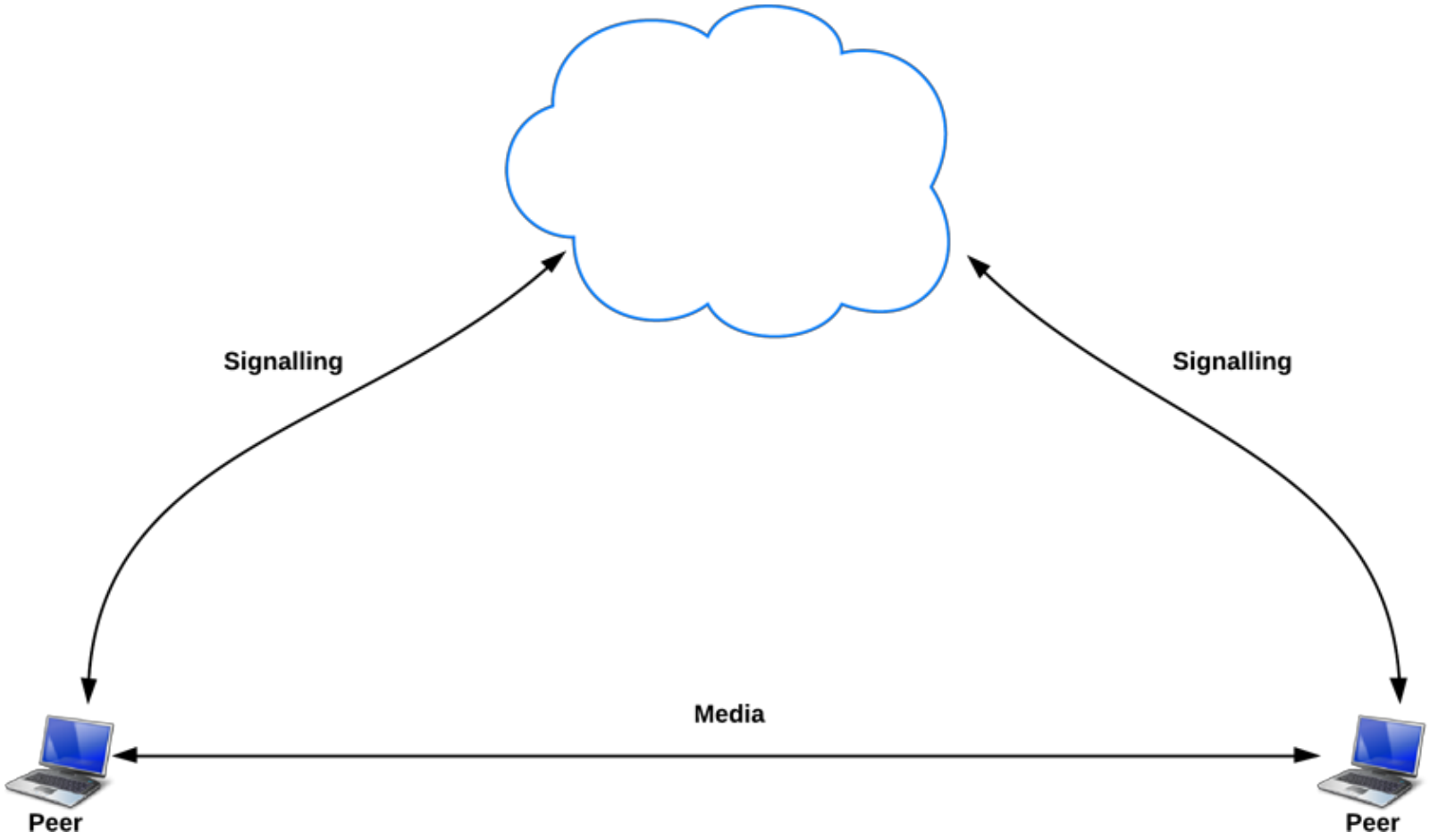
UGH

```
v=0
o=- 7614219274584779017 2 IN IP4 127.0.0.1
s=-
t=0 0
a=group:BUNDLE audio video
a=msid-semantic: WMS
m=audio 1 RTP/SAVPF 111 103 104 0 8 107 106 105 13 126
c=IN IP4 0.0.0.0
a=rtcp:1 IN IP4 0.0.0.0
a=ice-ufrag:W2TGCZw2NZHuwlnf
a=ice-pwd:xdQEccP40E+P0L5qTyzDgfmW
a=extmap:1 urn:ietf:params:rtp-hdrext:ssrc-audio-level
a=mid:audio
a=rtcp-mux
a=crypto:1 AES_CM_128_HMAC_SHA1_80 inline:9c1AHz27dZ9xPI91YNfSlI67/EMkjHHIHORiClQe
a=rtpmap:111 opus/48000/2
...
```

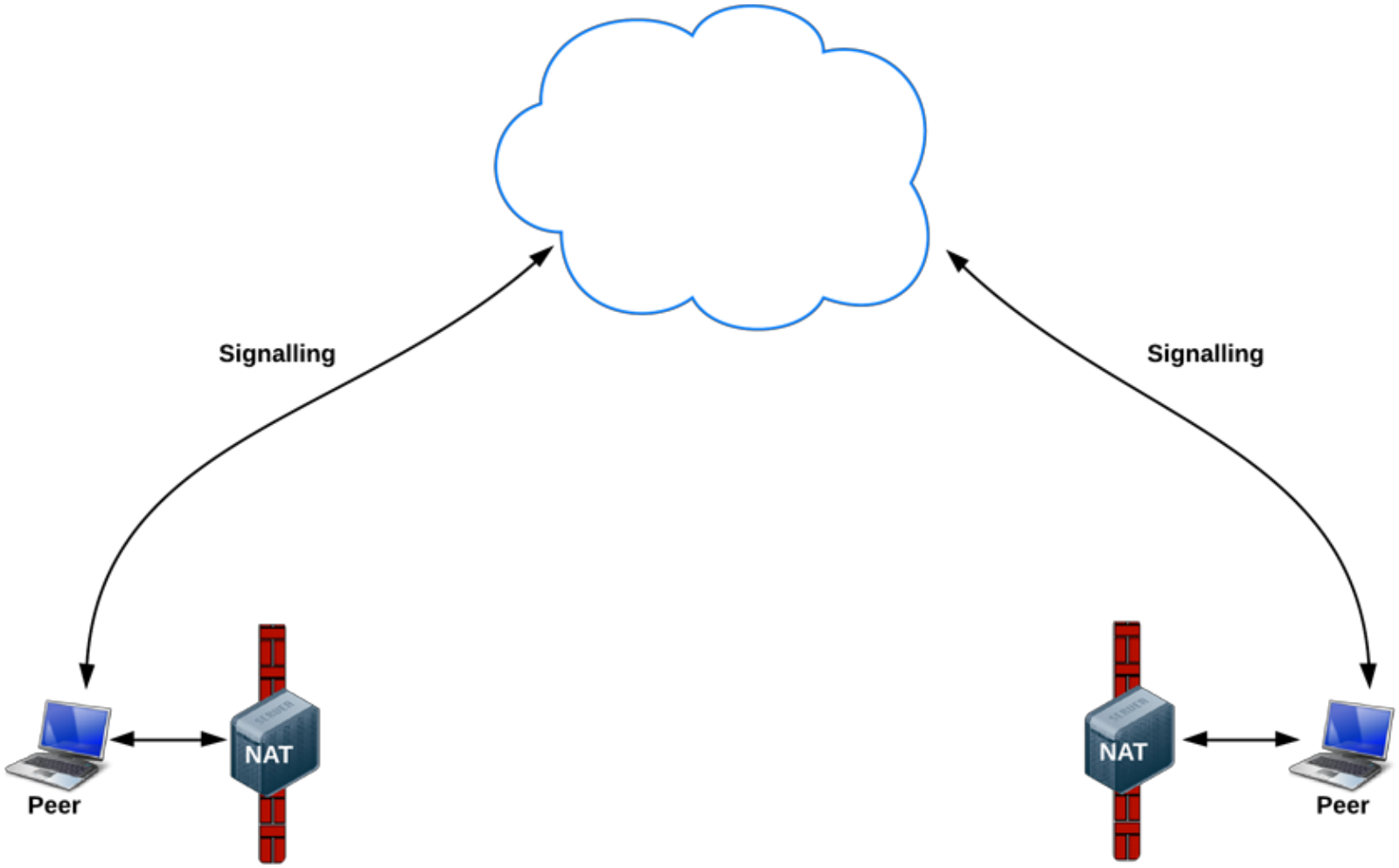
STUN and TURN

P2P in the age of firewalls and NATs

An ideal world



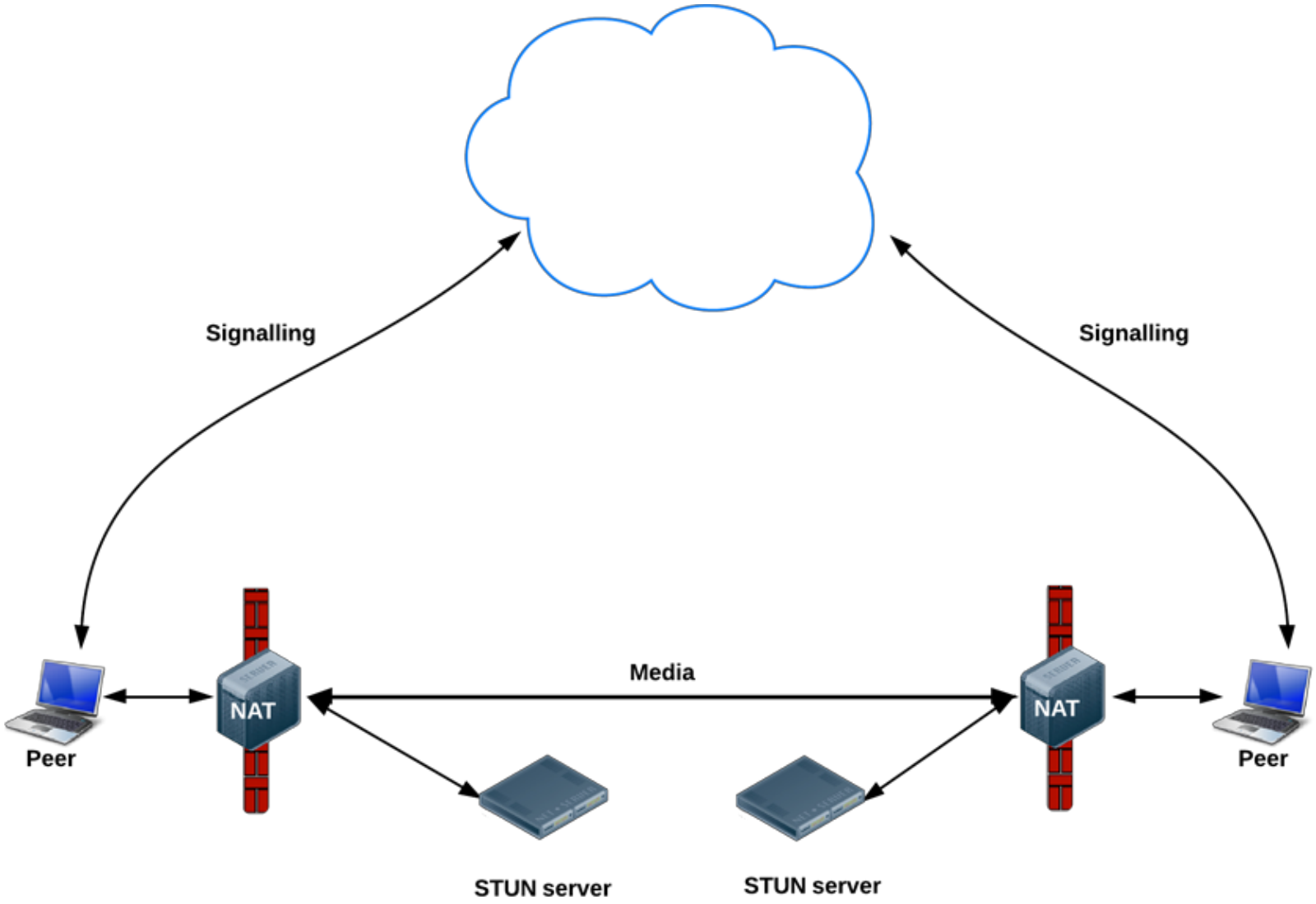
The real world



STUN

- Tell me what my public IP address is
- Simple server, cheap to run
- Data flows peer-to-peer

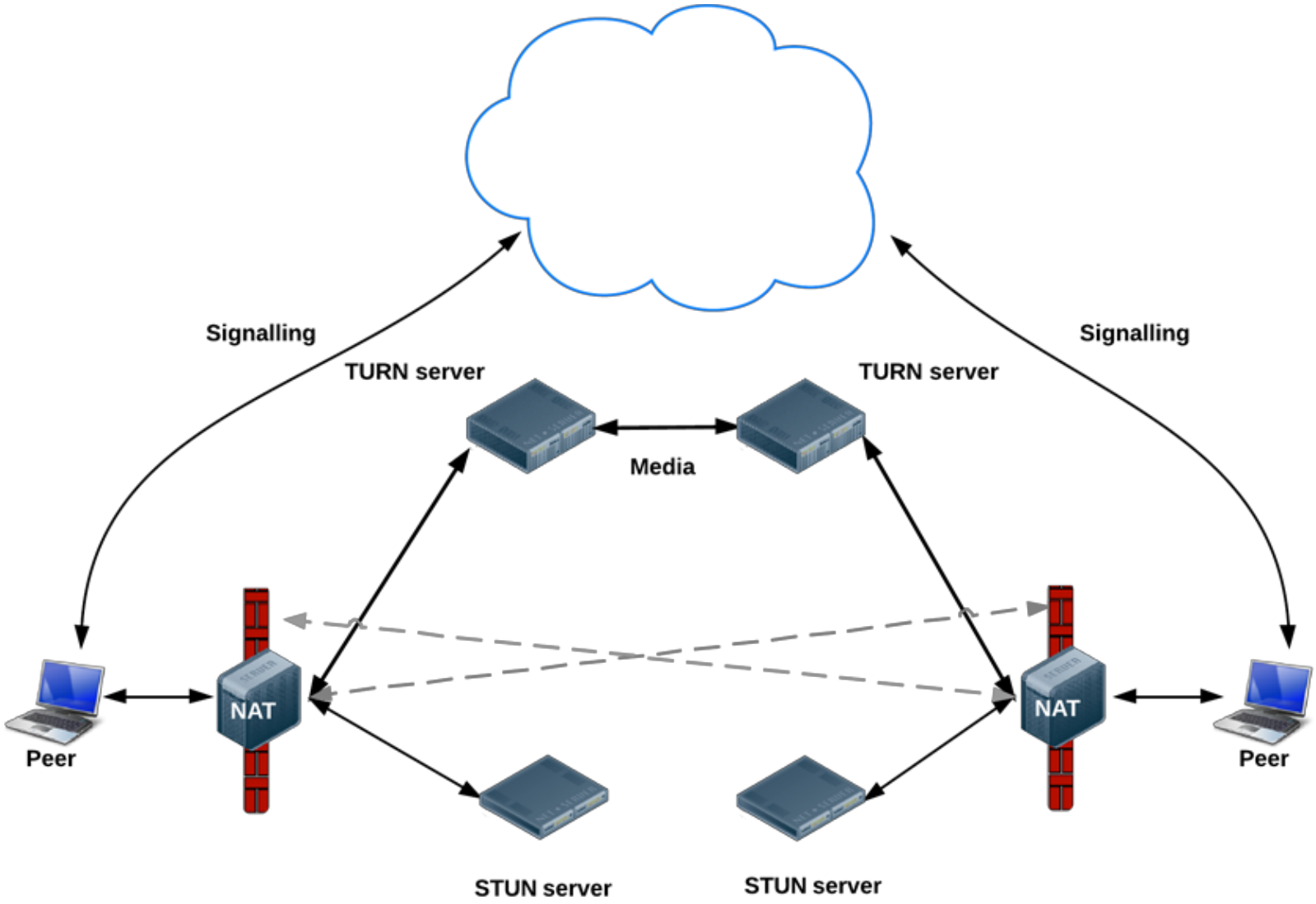
STUN


Signalling
Signalling
Media
NAT
NAT
Peer
Peer
STUN server
STUN server

TURN

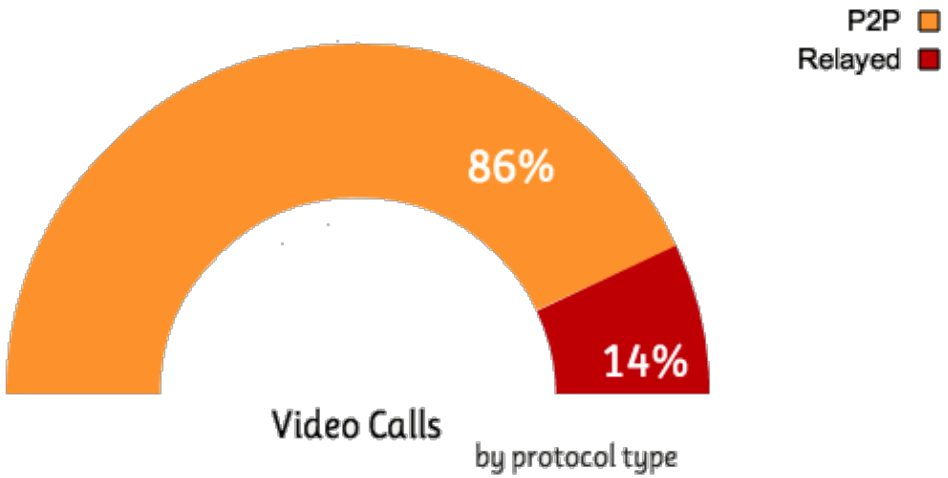
- Provide a cloud fallback if peer-to-peer communication fails
- Data is sent through server, uses server bandwidth
- Ensures the call works in almost all environments

TURN


Signalling
TURN server
TURN server
Signalling
Media
NAT
NAT
Peer
Peer
STUN server
STUN server

ICE

- ICE: a framework for connecting peers
- Tries to find the best path for each call
- Vast majority of calls can use STUN (webrtcstats.com):



Deploying STUN/TURN

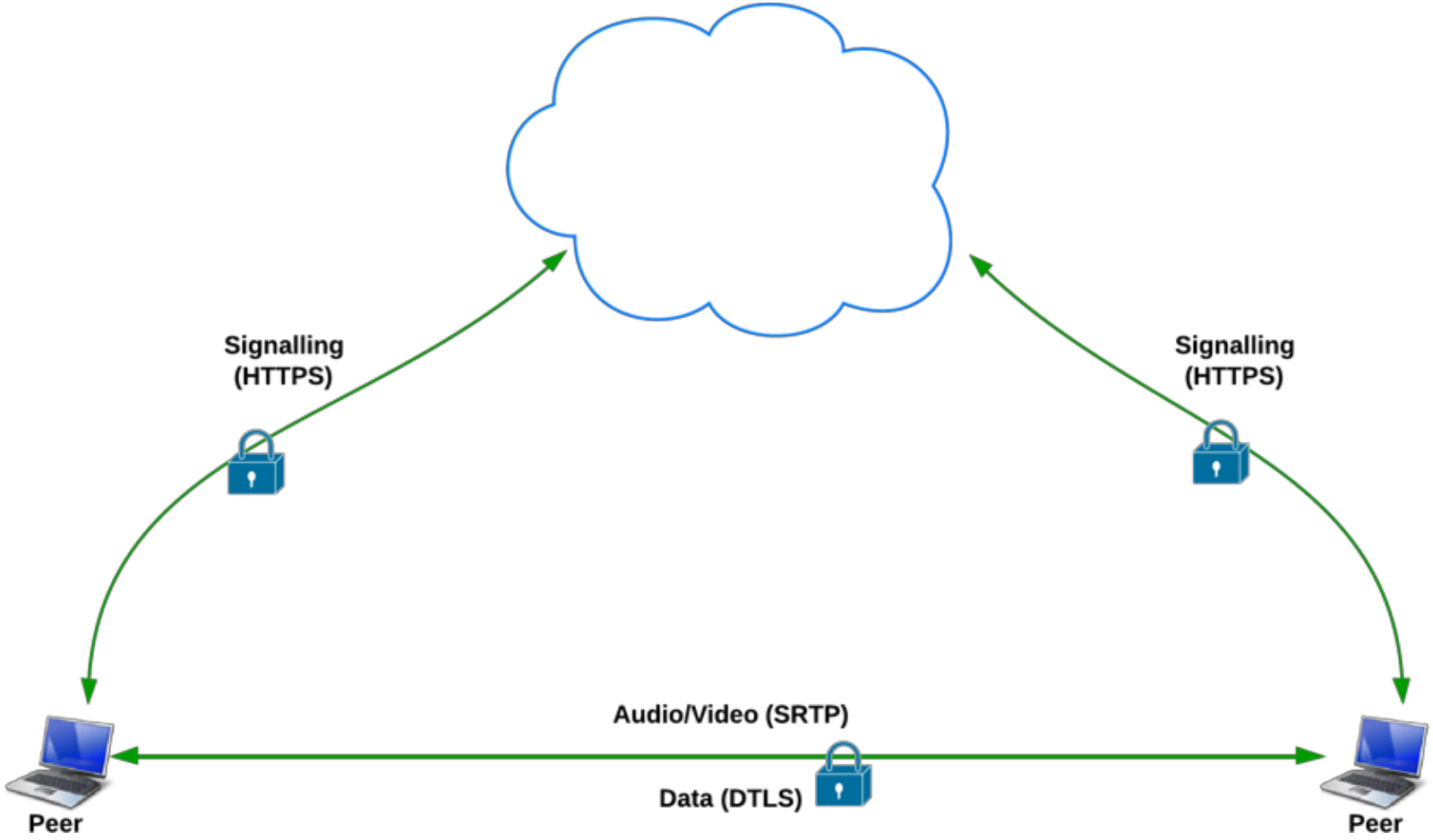
- stun.l.google.com:19302
- WebRTC stunserver, turnserver
- rfc5766-turn-server
- restund

Security

Security throughout WebRTC

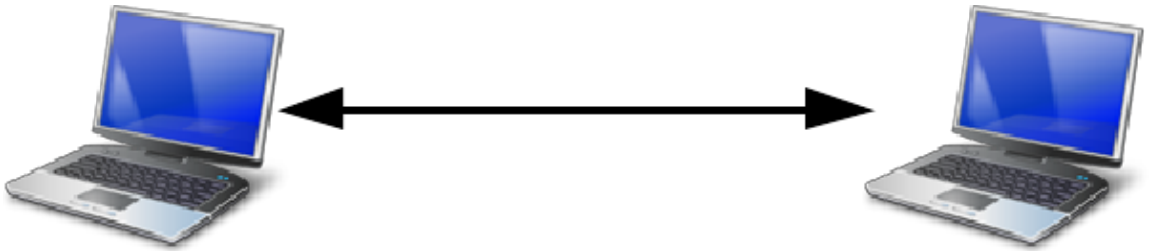
- Mandatory encryption for media and data
- Secure UI, explicit opt-in
- Sandboxed, no plugins
- WebRTC Security Architecture

Secure pathways

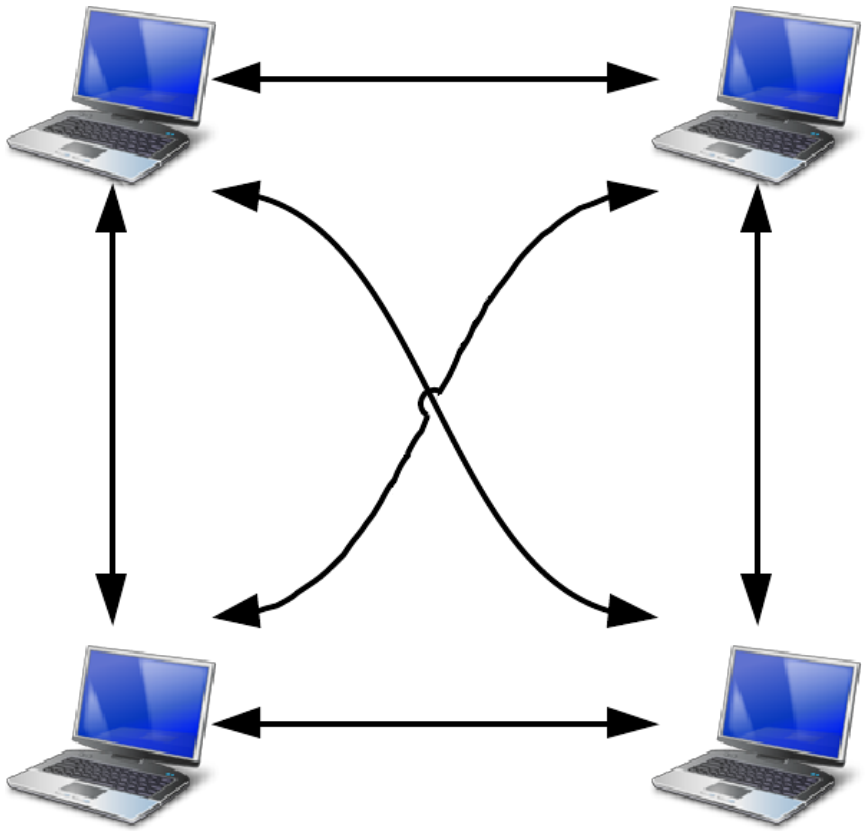


Architectures

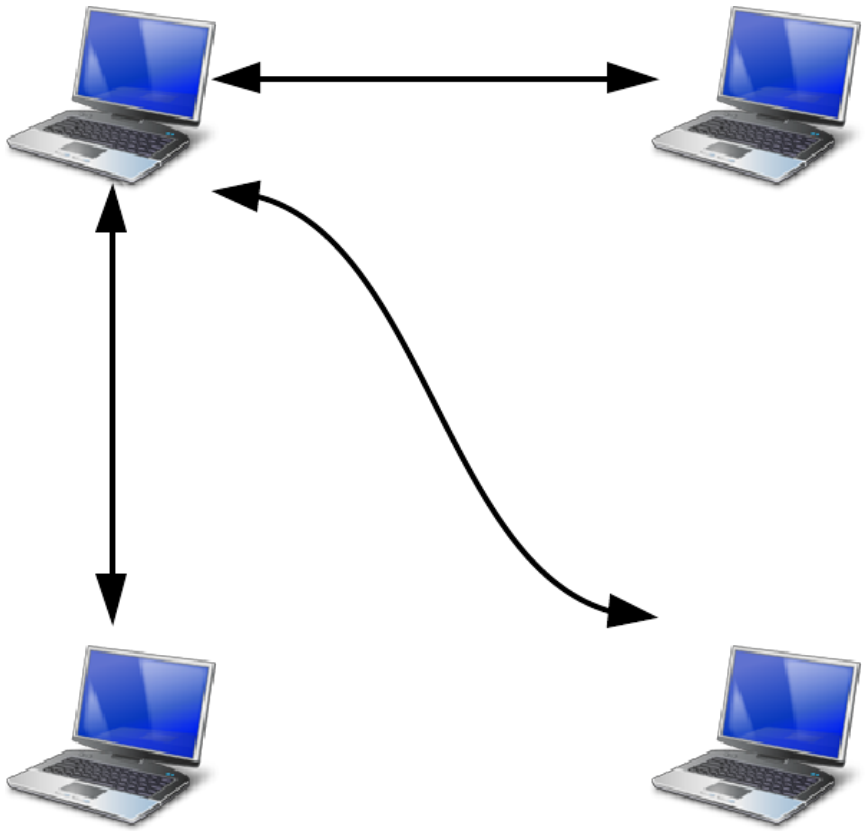
Peer to peer: one-to-one call



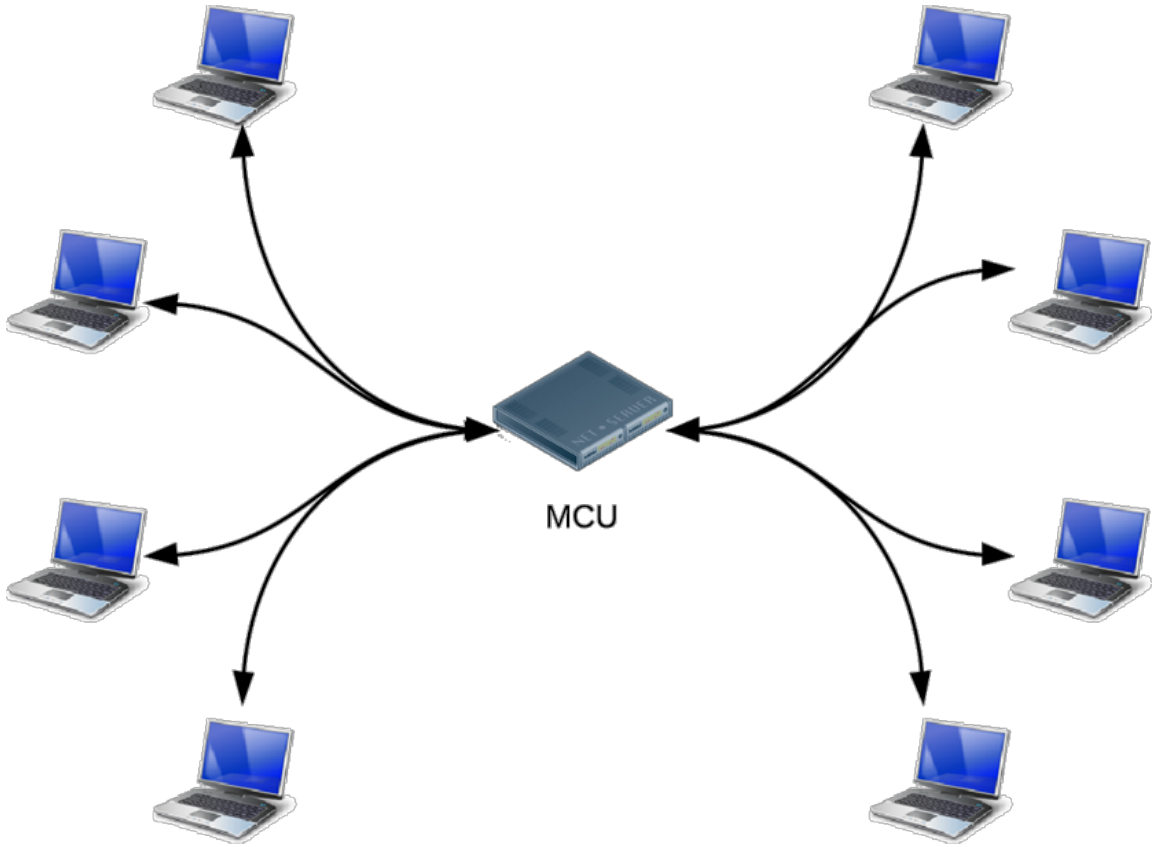
Mesh: small N-way call



Star: medium N-way call



MCU: large N-way call



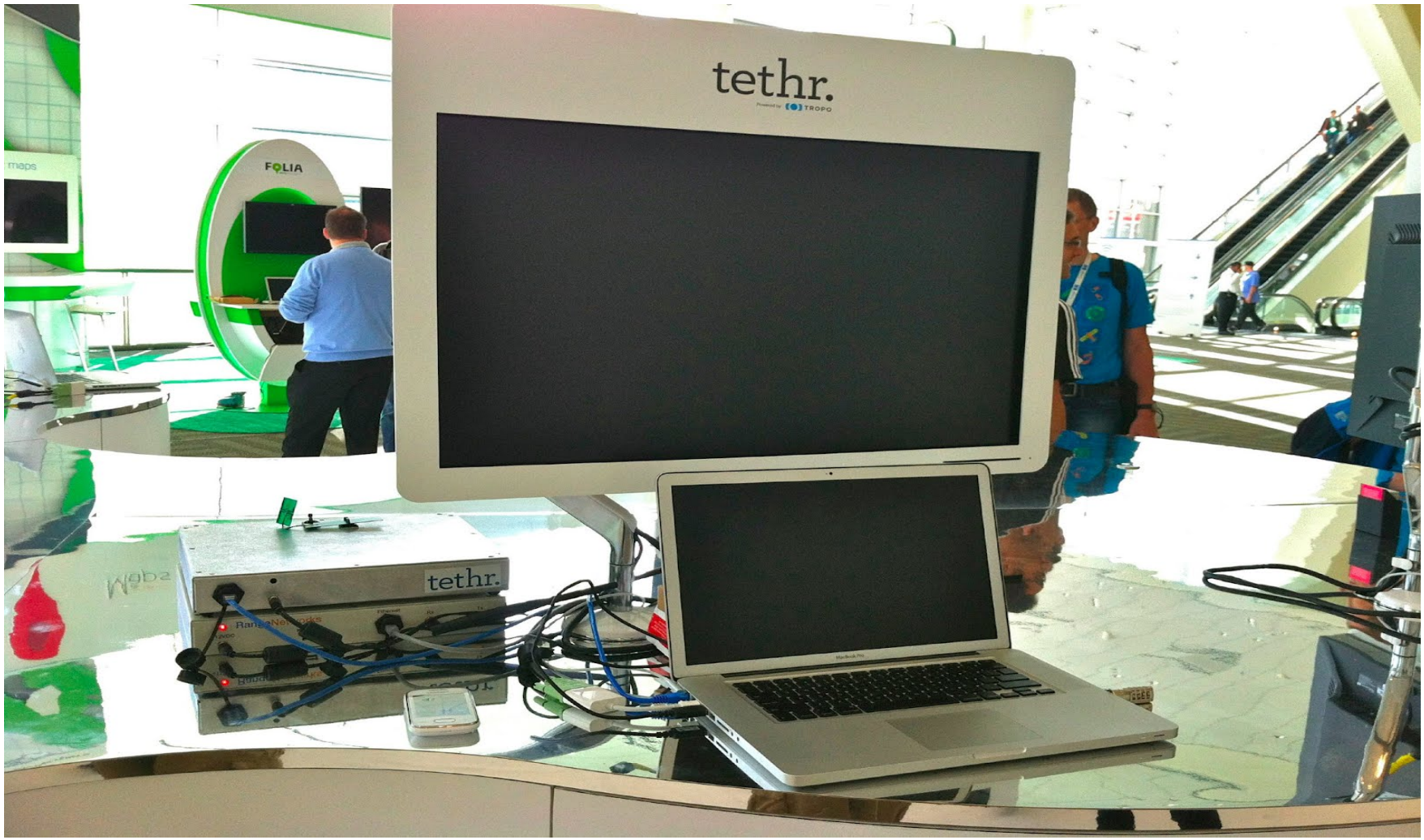
Beyond browsers

Phones and more

- Easy to interoperate with non-browser devices
  - sipML5 open source JavaScript SIP client
  - Phono open source JavaScript phone API
  - Zingaya embeddable phone widget

Zingaya PSTN

Tethr



Building a WebRTC app

chrome://webrtc-internals

adapter.js

Lets you use the same code in all browsers:

- Removes vendor prefixes
- Abstracts Chrome/Firefox differences
- Minimizes effects of spec churn

This is doing my head in.

JavaScript frameworks

- Video chat:
  - SimpleWebRTC
  - easyRTC
  - webRTC.io
- Peer-to-peer data:
  - PeerJS
  - Sharefest

SimpleWebRTC

Easy peer-to-peer video and audio

JAVASCRIPT

```
var webrtc = new WebRTC({
  localVideoEl: 'localVideo',
  remoteVideosEl: 'remoteVideos',
  autoRequestMedia: true
});

webrtc.on('readyToCall', function () {
    webrtc.joinRoom('My room name');
});
```

PeerJS

Easy peer-to-peer data

JAVASCRIPT

```
var peer = new Peer('someid', {key: 'apikey'});
peer.on('connection', function(conn) {
  conn.on('data', function(data){
    // Will print 'hi!'
    console.log(data);
  });
});

// Connecting peer
var peer = new Peer('anotherid', {key: 'apikey'});
var conn = peer.connect('someid');
conn.on('open', function(){
  conn.send('hi!');
});
```

Services

- Complete video services:
  - OpenTok (acquired by Telefonica Digital)
  - vLine

Chris Wilson **LIVE!**

More Information

- Justin Uberti: Google I/O presentation video
- Cullen Jennings video: HTML5 WebRTC
- HTML5 Rocks:
  - Capturing audio and video in HTML5
  - Getting Started With WebRTC
  - Updates
- ...and a book: webrtcbook.com

Contact Us

- webrtc.org
- discuss-webrtc
- +webrtc
- @webrtc
- crbug.com/new

“WebRTC and HTML5 could enable the same transformation for real-time communications that the original browser did for information.”

Phil Edholm
— NoJitter

io13webrtc.appspot.com

<Thank You!>

Justin Uberti - WebRTC Tech Lead, Google
Sam Dutton - Developer Advocate, Google Chrome


Google
Developers