Cross-Platform WebRTC

ALEX DUNN

TWITTER: @SUAVE_PIRATE

BLOG: HTTPS://ALEXDUNN.ORG

Agenda

Brief Introduction to WebRTC

How WebRTC Works

Adapting to Native Platforms

Tools and Resources

Look at Some Code

Resources

Finished source: https://github.com/SuavePirate/Xamarin.WebRTC

WebRTC docs: https://webrtc.org

IceLink docs: http://docs.frozenmountain.com/icelink2/

Native libraries: https://webrtc.org/native-code/native-apis/

What is WebRTC

The Basics of WebRTC

Web Real-Time-Communication

New(ish) open-sourced standard for transmitting data from one peer to another

A set of frameworks and tools for creating connections and sending data

Data transmitted over peer-to-peer connections

Agnostic to types of data sent

- Audio buffers
- Video frames
- Raw data
- Files

Important Components and Terms

Peer Connections – the actual connection between two clients

Important Components and Terms

Peer Connections – the actual connection between two clients

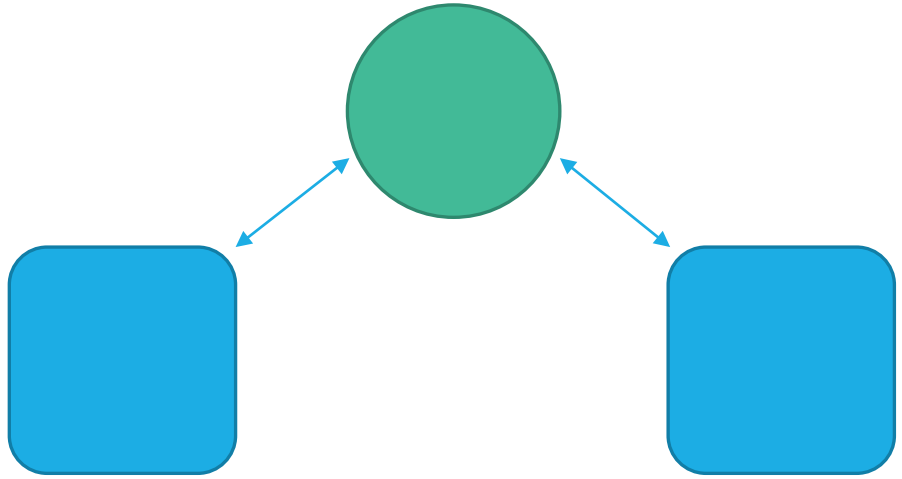
STUN - Session Traversal of User Datagram Protocol [UDP] Through Network Address Translators [NATs]

Important Components and Terms

Peer Connections – the actual connection between two clients

STUN - Session Traversal of User Datagram Protocol [UDP] Through Network Address Translators [NATs]

TURN – Traversal Using Relays around NAT



Important Components and Terms

Peer Connections – the actual connection between two clients

STUN - Session Traversal of User Datagram Protocol [UDP] Through Network Address Translators [NATs]

TURN – Traversal Using Relays around NAT

ICE - Interactive Connectivity Establishment

Vanilla Ice

Important Components and Terms

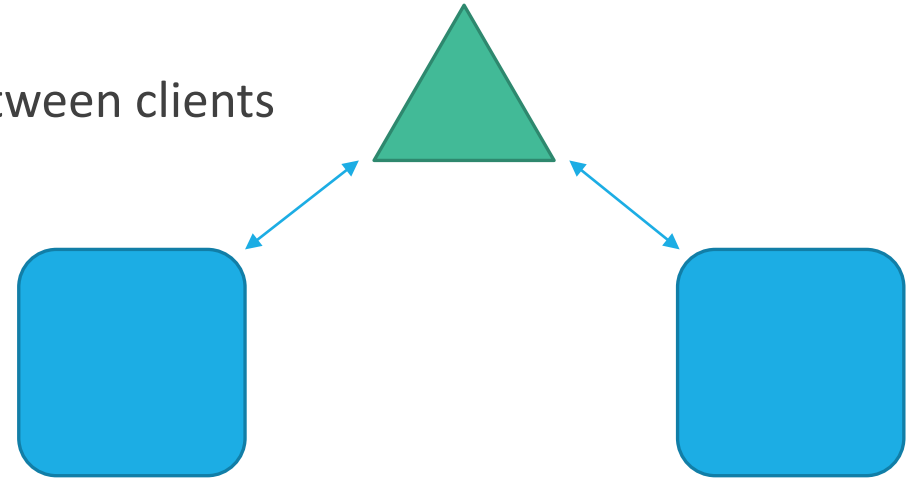
Peer Connections – the actual connection between two clients

STUN - Session Traversal of User Datagram Protocol [UDP] Through Network Address Translators [NATs]

TURN – Traversal Using Relays around NAT

ICE - Interactive Connectivity Establishment

Signaling – a means of communicating connection info between clients



Important Components and Terms

Peer Connections – the actual connection between two clients

STUN - Session Traversal of User Datagram Protocol [UDP] Through Network Address Translators [NATs]

TURN – Traversal Using Relays around NAT

ICE - Interactive Connectivity Establishment

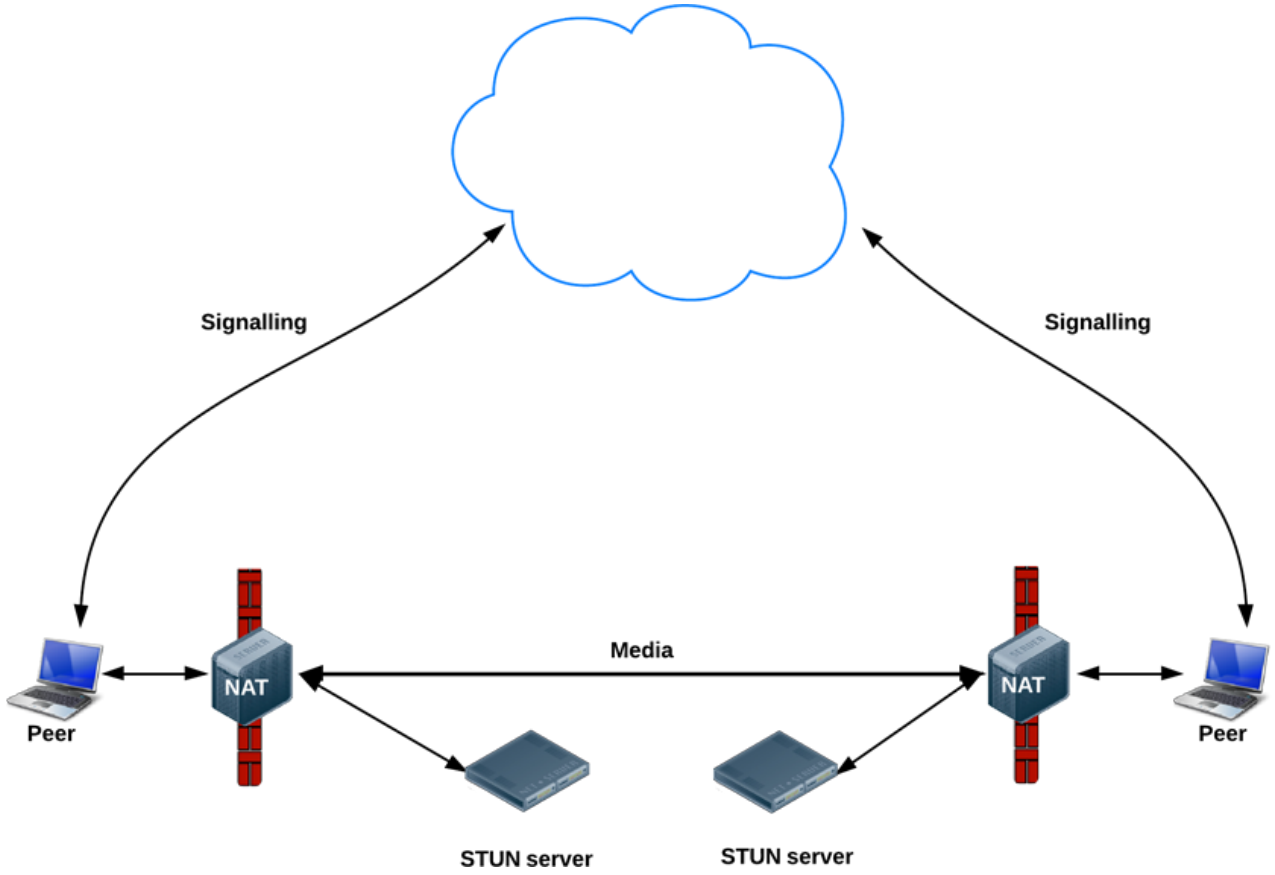
Signaling – a means of communicating connection info between clients

Codecs

- Video
  - VP8
- Audio
  - Opus

Making Peer Connections

Creating a P2P Connection via STUN

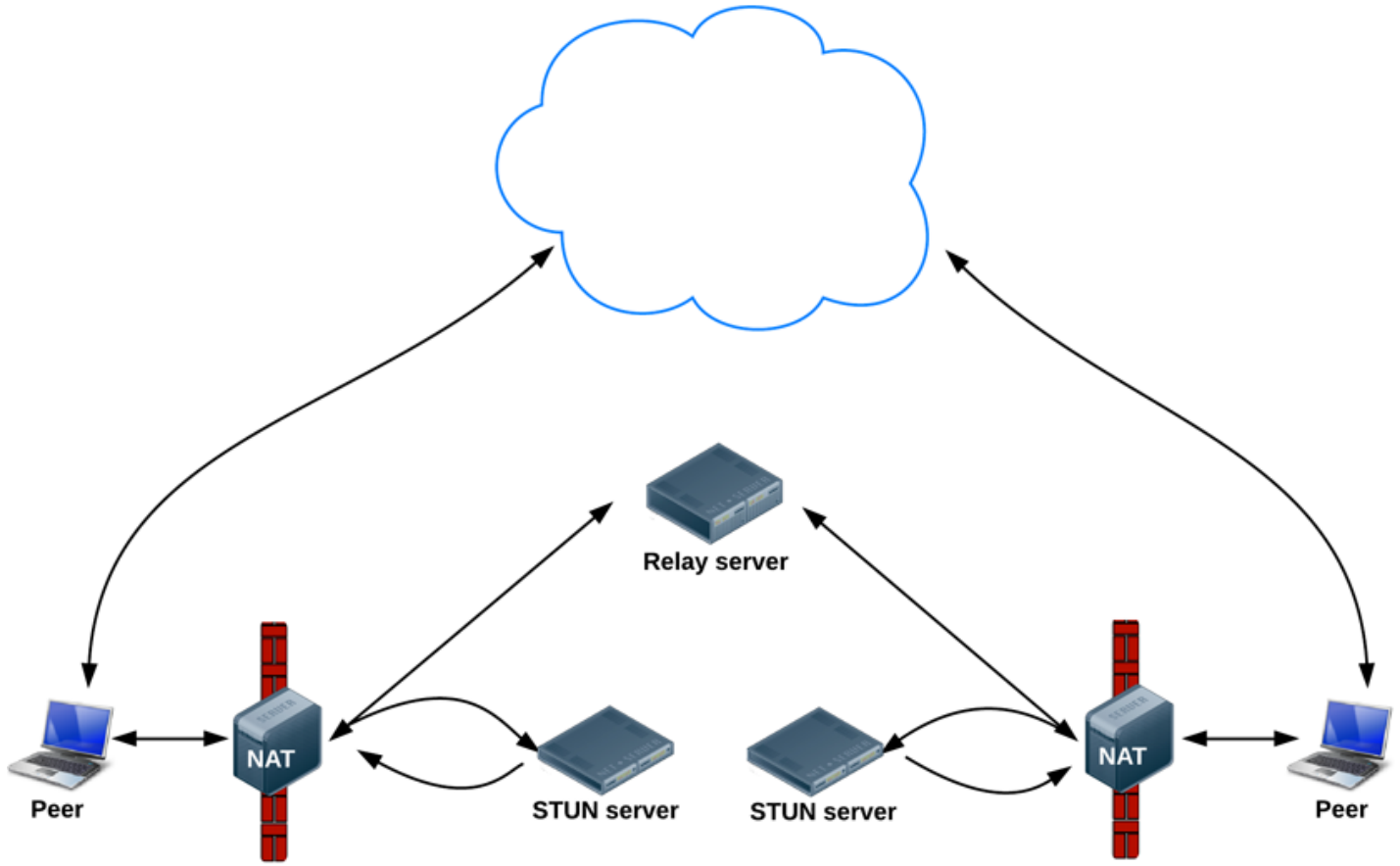


Creating a P2P Connection via STUN

Covers most scenarios

Fails when peers have 2 incompatible NAT types or peers run through heavy firewalls

Creating a P2P Connection via STUN + TURN



Creating a P2P Connection via STUN + TURN

TURN server relays data between two peers

Means no direct peer connection is made

Can have performance implications

Can be expensive

Covers 99.999% of cases

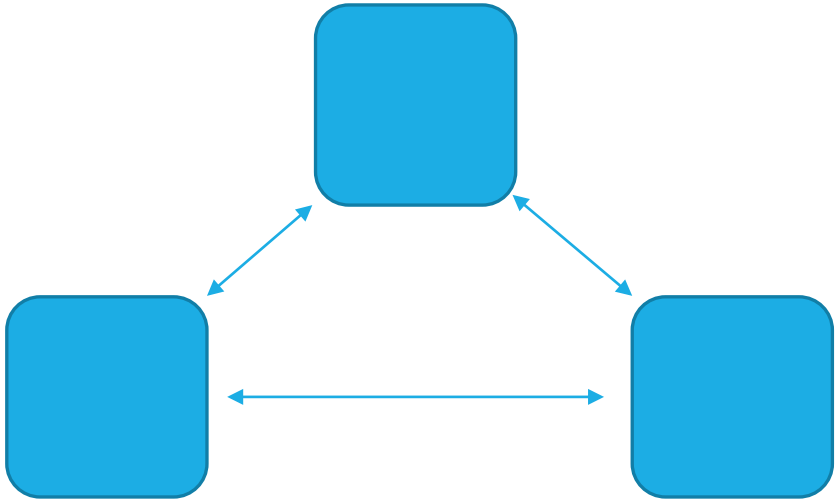
Multi-connection Mesh Model

Each client creates a connection to every other client

Can create unreliable connections

Individual connections can be dropped without dropping them all

Requires extended signaling and accountability



Multi-connection Mixer Model

Each client connects to a single server

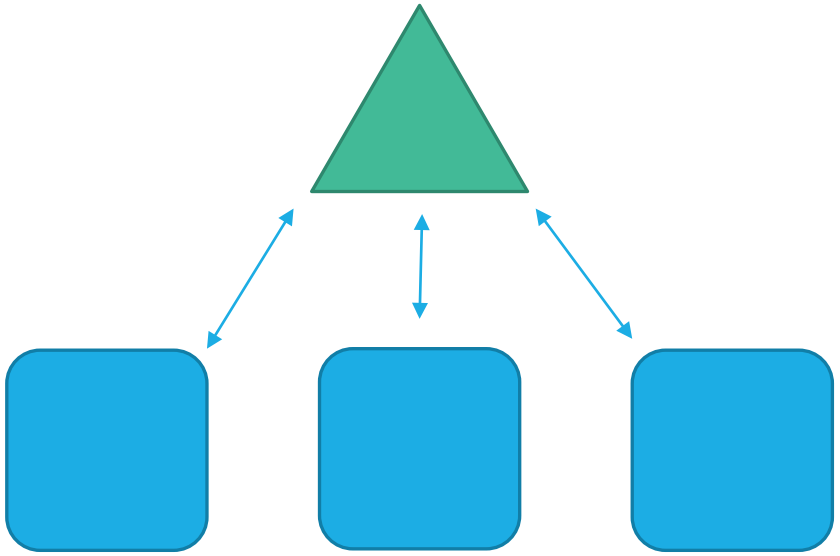
The server mixes audio and video streams per peer

The server relays the mixed streams to each peer

Heavy lifting put on server rather than distributed

Expensive

Hard to scale



Using a Connection

Sending Audio and Video Data

GetUserMedia

- Gets access to device hardware (mic and camera)

Sending Audio and Video Data

GetUserMedia

- Gets access to device hardware (mic and camera)

Encode the media

- Pick a codec
- Encode the data with that codec

Sending Audio and Video Data

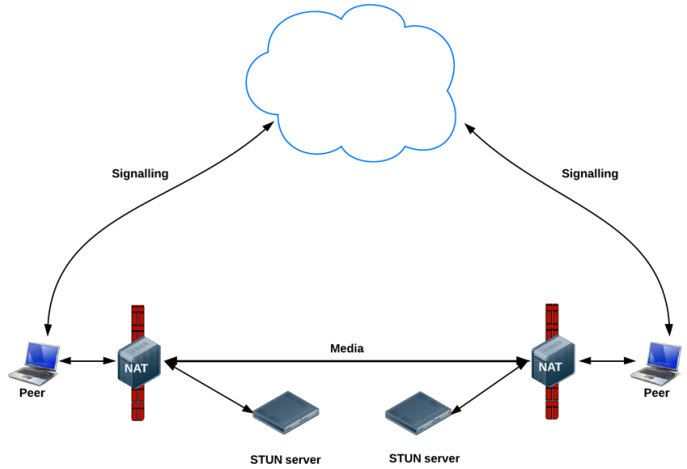
GetUserMedia

- Gets access to device hardware (mic and camera)

Encode the media

- Pick a codec
- Encode the data with that codec

Send encoded media over connection



Sending Audio and Video Data

GetUserMedia

- Gets access to device hardware (mic and camera)

Encode the media

- Pick a codec
- Encode the data with that codec

Send encoded media over connection

Switch media sources (switch camera, switch mic)

Advanced Functionality

Screen streaming

Multi-source streaming

- Streaming multiple video streams from one client
- Streaming multiple audio streams from one client

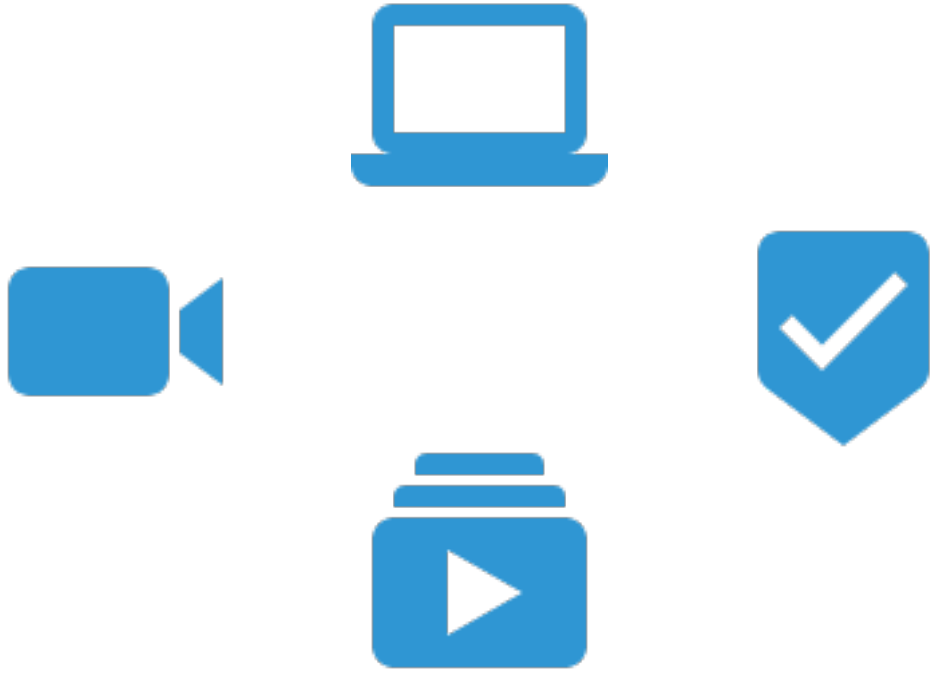
Audio Mixing

Non-TURN relaying

Recording streams

Distributed streaming

Stream signatures



Going Cross-Platform

Platforms Supported

Web*

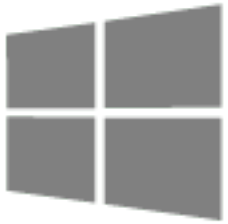
Windows
- WPF
- Windows Forms
- Win8
- UWP*

Windows Phone
- 8.0 Silverlight+
- 8.1 RT*

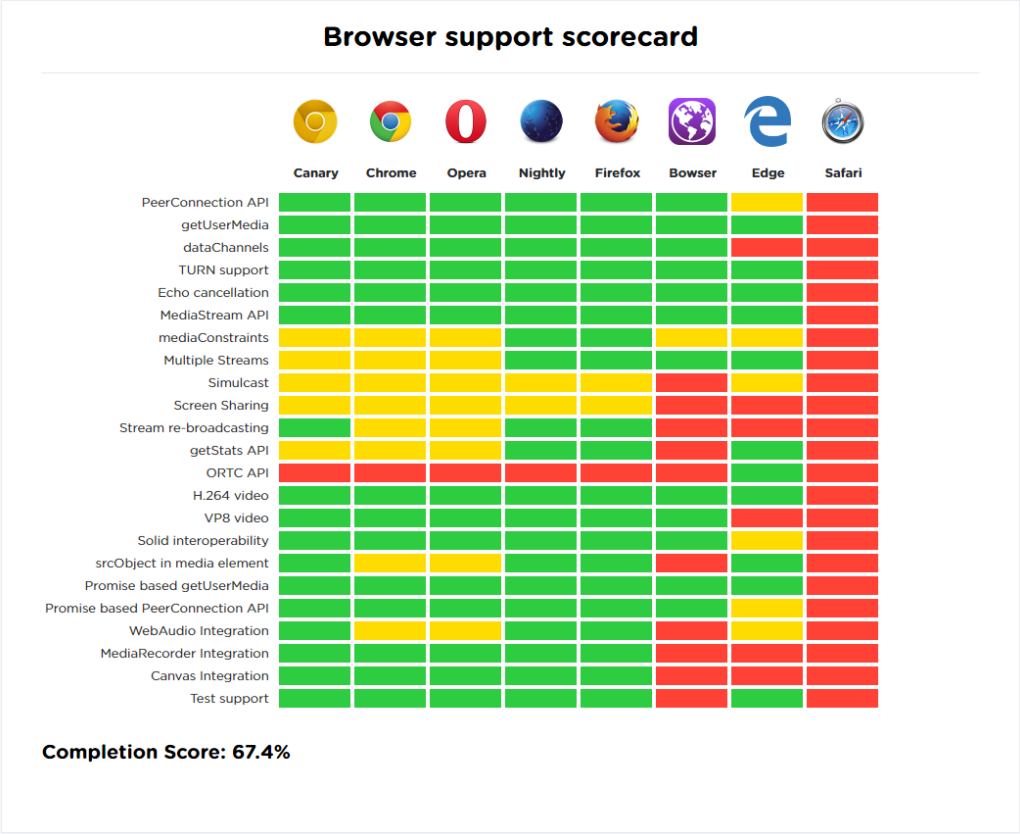
Android

iOS

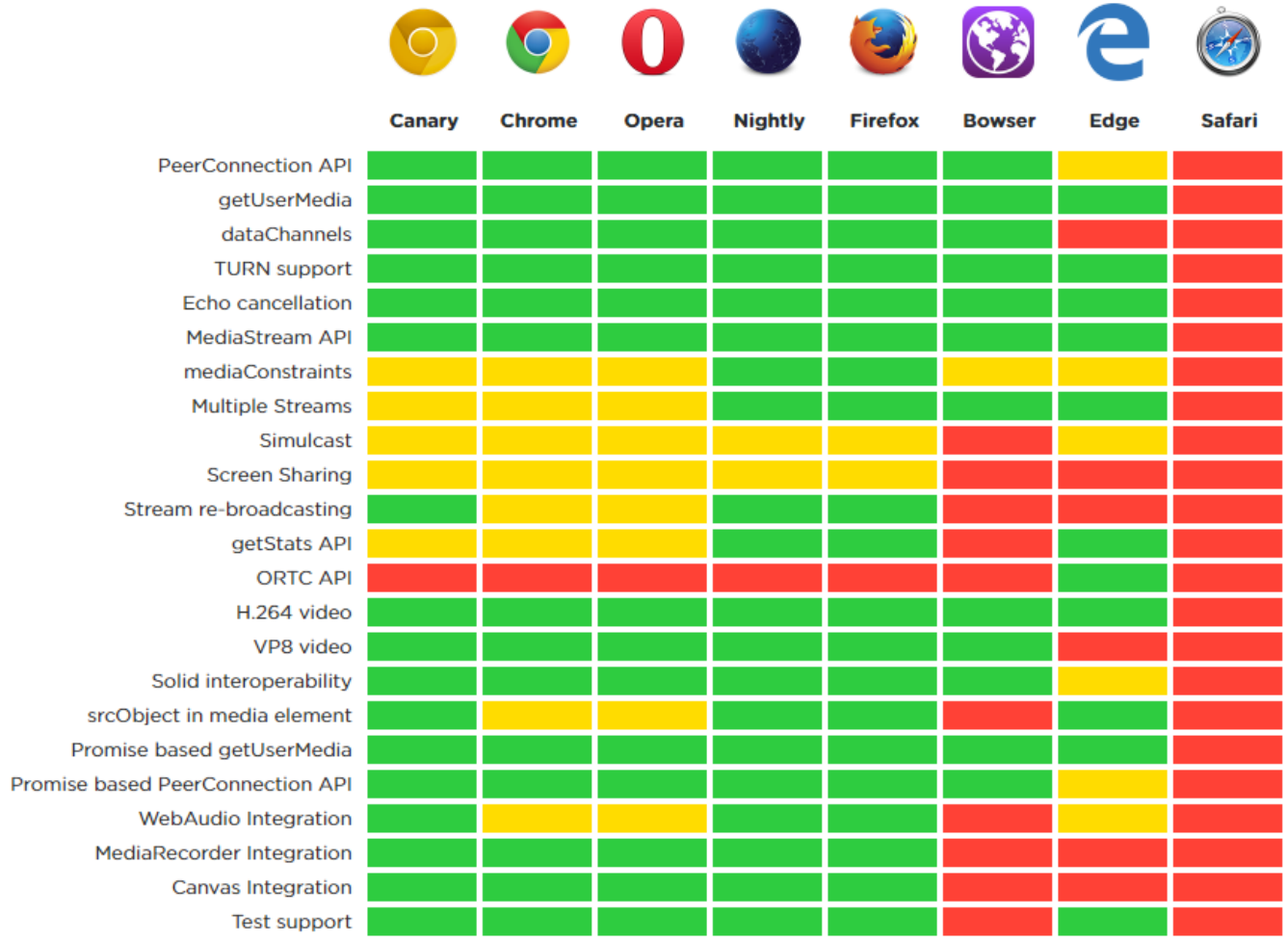
Mac



Web Support


Browser support scorecard
Canary
Chrome
Opera
Nightly
Firefox
Bowser
Edge
Safari
PeerConnection API
getUserMedia
dataChannels
TURN support
Echo cancellation
MediaStream API
mediaConstraints
Multiple Streams
Simulcast
Screen Sharing
Stream re-broadcasting
getStats API
ORTC API
H.264 video
VP8 video
Solid interoperability
srcObject in media element
Promise based getUserMedia
Promise based PeerConnection API
WebAudio Integration
MediaRecorder Integration
Canvas Integration
Test support
Completion Score: 67.4%

Browser support scorecard



Completion Score: 67.4%

Web Workarounds

Npapi plugins

Chrome + Firefox extensions

ActiveX plugins

Java Applet plugins

Still doesn’t catch all

- Edge does not support WebRTC (they are pushing ORTC)
- Edge does not support ActiveX plugins, Java Applets, or npapi

Native Platform Support and Issues

WebRTC is built on an open sourced C++ platform-agnostic framework

- You theoretically can run it anywhere!

Not every platform supports given codecs

VP8 is not native to any Windows SDK!

We need to get a custom build of the native VP8 (C++) codec

Compile the VP8 codec against Visual C++ per platform

Same with Opus

Unlike web browsers that ship with VP8 and Opus codecs as part of the application, we need to include the libraries separately

Building

Windows

- Build VP8 codec
- Build Opus codec
- Wrap calls to C++ library

Android

- Include reference to native codecs built into OS

iOS + Mac

- Add official libraries for native codecs

Xamarin

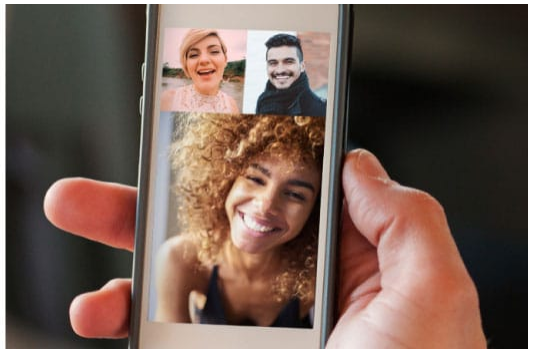
- Add bindings to native libraries for codecs and wrap calls

IceLink

A library and framework wrapping a lot of native WebRTC

Languages and platforms supported

- Android
- iOS
- Windows (WinForms, WPF, Win8, UWP)
- Windows Phone (8, 8.1)
- Web
  - ActiveX control
  - Java Applet
  - Npapi in the works
- Mac
- Xamarin (Android, iOS, Mac)
- Java



Demo