Approaching Zero Latency Streaming with WebRTC
Welcome
Hello Sergey,

I'm wondering if you have webRTC support on your road map.

It looks like Aurt once planned it but it doesn't seem to have gotten anywhere.

arut#207

Thanks for your time!

---

I don't think so. webRTC is mind blowing thing to me.
WebRTC helps solve a latency issue but introduces its own set of challenges.
AGENDA

Provide a framework to evaluate when to use WebRTC.
Learn what tools can be used to jump-start your WebRTC deployment.
AGENDA

3

Discuss the benefits of plugin-based servers
AGENDA

4

Introduce the core components of WebRTC
AGENDA

5

Cover how you influence networking
AGENDA

6 Tradeoffs
Jamie Stackhouse
Developer and Hobby Musician

TWITTER
stackhousejs

LINKEDIN
jamie-stackhouse

EMAIL
jamie.stackhouse@ingest.io
Ingest is an encoding engine, and streaming platform for delivering video content.
Support

Visibility into the roadmap and engineers available to talk through the tough problems.
Industry Trends
Video Conferencing
Consumers now dictate how, when, and where to interact with video. Their expectations continue to rise.

- Quality content
- Awesome user experience
- Dependable technology
<table>
<thead>
<tr>
<th>Performance</th>
<th>Cost</th>
</tr>
</thead>
</table>

Defining Success
Defining Success

Performance

Low Latency

Cost
Defining Success

Performance

Low Latency

Cost

Minimize Cost
Defining Success

Performance

Low Latency

Reduce/Eliminate Playback Stalls

Cost

Minimize Cost
Defining Success

- Performance
  - Low Latency
  - Reduce/Eliminate Playback Stalls

- Cost
  - Minimize Cost
  - CapEx
Defining Success

Performance

Low Latency

Reduce/Eliminate Playback Stalls

Concurrent Views Per Stream

Cost

Minimize Cost

CapEx
Defining Success

Performance

Low Latency
Reduce/Eliminate Playback Stalls
Concurrent Views Per Stream

Cost
Minimize Cost
CapEx
OpEx
Defining Success

Performance
- Low Latency
- Reduce/Eliminate Playback Stalls
- Concurrent Views Per Stream

Cost
- Minimize Cost
- CapEx
- OpEx

Drive Engagement
Drive Engagement
But Why WebRTC?

- Low Latency
- Bi-directional and/or Text Communication
- Many streams, single watcher
Existing Tools

DUAL LICENSE
- Asterisk
- Janus

SERVICES
- Talky
- Tokbox

FOSS
- FreeSwitch
- NkMedia
<table>
<thead>
<tr>
<th>Application</th>
<th>Licensing</th>
<th>Support</th>
<th>Features (Use Case)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asterisk</td>
<td>Dual (GPL, Paid)</td>
<td>Community / Company</td>
<td>Telecom Focus - Feature Rich</td>
</tr>
<tr>
<td>Janus</td>
<td>Dual (GPL, Paid)</td>
<td>Community / Company</td>
<td>Gateway - Plugins for individual features</td>
</tr>
<tr>
<td>FreeSwitch</td>
<td>MPL</td>
<td>Community / Company</td>
<td>Telecom Focus - Feature Rich</td>
</tr>
<tr>
<td>Talky</td>
<td>Service</td>
<td>Company</td>
<td>WebRTC service at scale</td>
</tr>
<tr>
<td>Tokbox</td>
<td>Service</td>
<td>Company</td>
<td>WebRTC service at scale</td>
</tr>
<tr>
<td>NkMedia</td>
<td>N/A</td>
<td>Community</td>
<td>Gateway of Gateways, in development</td>
</tr>
</tbody>
</table>
Plug-in Based Development

SIP

STREAMING

VoIP

VIDEO CONFERENCING
WebRTC C++ API (PeerConnection)

Audio Capture / Render

Video Capture

Network I/O

Session management / Abstract Signaling (Session)

iSAC / iBLC Codec

NetEQ for Voice

Echo Canceler / Noise Reduction

VOICE ENGINE

VP8 Codec

Video jitter buffer

Image enhancements

VIDEO ENGINE

SRTP

Multiplexing

P2P STUN + TURN + ICE

TRANSPORT

Audio Capture / Render

Video Capture

Network I/O

WEB API (Edited by W3C WG)

API for web developers

API for browser makers

Overrideable by browser makers

THE WEB

YOUR BROWSER

Your Web App #1

Your Web App #2

Your Web App #3
Javascript Session Establishment Protocol

• Helps communicate with signalling backend

• Generates information using SDP

• Works with the WebRTC W3C APIs
Construct the offer

```javascript
var pc = new RTCPeerConnection(config)
pc.addStream(mediaStream)
pc.createOffer().then(pc.setLocalDescription)
.then(sendToSignalServer)
```

// will fire for each local ICE candidate, // when triggered you need to publish to // the remote signal server to forward to peer
pc.onicecandidate = publishIceCandidateToRemote

Create Answer

```javascript
// create peer connection first on remote // receive msg through signal
var desc = new RTCSessionDescription(msg.sdp)
pc.setRemoteDescription(desc)
// maybe create local stream?
pc.createAnswer().then(pc.setLocalDescription)
```

[Offer]

```
v=0
o=alice 2890844526 2890844526 IN IP4 host.atlanta.example.com
s=
c=IN IP4 host.atlanta.example.com
t=0 0
m=audio 49170 RTP/AVP 0 8 97
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
m=video 51372 RTP/AVP 31 32
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
```

[Answer]

```
v=0
o=bob 2808844564 2808844564 IN IP4 host.biloxi.example.com
s=
c=IN IP4 host.biloxi.example.com
t=0 0
m=audio 49174 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 49170 RTP/AVP 32
a=rtpmap:32 MPV/90000
```
WebRTC C++ API (PeerConnection)
Audio Capture / Render
Video Capture
Network I/O
Session management / Abstract Signaling (Session)
iSAC / iBLC Codec
NetEQ for Voice
Echo Canceler / Noise Reduction
VOICE ENGINE
VP8 Codec
Video jitter buffer
Image enhancements
VIDEO ENGINE
SRTP
Multiplexing
P2P STUN + TURN + ICE
TRANSPORT
Audio Capture / Render
Video Capture
Network I/O
API for web developers
API for browser makers
Overrideable by browser makers
```javascript
var pc = new RTCPeerConnection({
  iceServers: [
    {
      urls: [
        "stun:stun.example.com",
        "stun:stun-1.example.com"
      ]
    }
  ]
});
```
WebRTC is new, and support is still coming in some clients.
The needs of your application should direct what technology is used.
What’s Next?
Thank you

JAMIE STACKHOUSE
Product Owner, Ingest
jamie.stackhouse@ingest.io