June 17-19, 2014 | Cobb Galleria | Atlanta, Georgia

The Longest Running Global WebRTC Ecosystem Event

www.webrtcexpo.com
WebRTC Plug-ins for Other browsers

Richard Tworek
Tworek, LLC
Strategy > Design > Results
Panelists

• Steven Ness
  – Priologic
• Vladimir Beloborodov
  – Mera
• Dr. Alex Gouaillard
  – Temasys
WebRTC for other browsers

Steven Ness
Priologic
WebRTC for Other Browsers

• WebRTC for browsers without WebRTC support
• Extend WebRTC
  – Host multiple cameras at the same time
    • Forward and backward cameras
  – H.264 support and other codecs (AAC)
  – Fine grained control of WebRTC
    • Optimize WebRTC parameters that are hidden from you
  – Force a certain version of WebRTC
  – Other features
    • Record calls
WebRTC for Other Browsers

• Safari
  – Important to end users that have Macs
  – Not typical enterprise users
  – Objective-C API
  – NPAPI
  – ~4% market share (3.8% May 2014)
WebRTC for Other Browsers

• Many browsers
  – Opera - ~2% market share (1.8% May 2014)
    • (Maxthon, Avant, KHTML, Rockmelt, Torch, ...)

• Few users
  – Varying levels of support for plugin APIs
  – Opera 20 has support for WebRTC
WebRTC for Other Browsers IE

• Internet Explorer
  – Typical enterprise user
  – Broad user base in enterprise
  – Managed desktops
  – ~10% market share (8.9% May 2014)
Ways to do plugins in IE

- Silverlight
  - Unable to create mixed mode assemblies
  - Limited subset of .NET
- NPAPI
  - Another possible approach
  - IE dropped support in IE 5.5SP2
    - Requires an ActiveX wrapper
- Can’t use PPAPI (Chrome only)
- ActiveX
  - WebRTC has a working example of Win32 code
    - Transplant into ActiveX control
  - Our approach
Our implementation

• WebRTC sample client in Win32 code
  – We wrote our own Javascript Foreign Function Interface with ActiveX
  – ATL – C++ is another framework (has a message loop)
  – Wrapped it in an ATL plugin project
  – Started with DirectX surfaces in browser
    • Layout and update issues – Layer an iFrame to make it update
    • iFrame receives the events, hard to get around this
  – Our application requires a thumbnail
    • base64 encoded JPEG
    • Send frames - 1.2MB raw image - 40k data URL- Canvas
    • 3-4x faster than <img> tag
    • All in all about the same CPU as DirectX implementation
Mobile – Go Native

• Android
  – WebRTC in Chrome on Android performs well
  – But: don’t get feel of native app
  – With native we can optimize WebRTC
    • Tweak the internal WebRTC parameters
    • Hardware H.264 codecs

• iOS
  – Safari doesn’t do WebRTC
  – Native app is the only way to go
Plugging in WebRTC

Vladimir Beloborodov
Mera
MERA WebRTC Plug-in in MS IE

It was among early WebRTC plug-ins demoed in 2012
Use Cases for WebRTC Plug-ins

- Adding standard WebRTC if not offered in browser
- Extending interoperability options
- Extending WebRTC capabilities and functionality
WebRTC Plug-ins on Desktops

• For individuals
  – Stuck to specific browsers or versions – *Not many*
  – Using specific Internet services (for interop or extensions)

• For businesses and companies
  – Stuck to specific browsers or versions
  – Integrating WebRTC with existing Telecom solutions
  – Requiring extra functionality
WebRTC Support & Interops

• Adding standard WebRTC to browsers
  – ActiveX for MS Internet Explorer
  – NPAPI for Safari (or older Firefox versions)

• Extending interoperability options
  – Embedded SIP stack or other stacks for signaling
  – Extra codecs for audio (e.g. G.729)
  – Codecs for video
Extensions to WebRTC

- Screen Sharing
- QoS / QoE Monitoring and Control
- Special payloads (e.g. Real-Time Text)
- Non-standard encryptions
- Extensions to ICE
- Accessing specific media devices
- Extra built-in accessibility means
Alternatives to Browser Plug-in

• **Server-side** Implementation
  – Handling different encryptions

• **Client-side JavaScript** Implementation
  – JavaScript SIP stacks
  – Some data payloads (*e.g.* MSRP)
www.merasws.com
Booth #121

@VladimirTechMan
linkd.in/1q0CyPo
History and comparison of webRTC plugins

Dr. Alex Gouaillard
Temasys
## State of the technology

<table>
<thead>
<tr>
<th>Desktop</th>
<th>chrome</th>
<th>firefox</th>
<th>IE</th>
<th>Safari</th>
<th>Opera</th>
<th>ref client</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUM</td>
<td>18+</td>
<td>17+</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC</td>
<td>20+</td>
<td>22+</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DC</td>
<td>26+</td>
<td>22+</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAE</td>
<td></td>
<td></td>
<td>N/A</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(appRTCDesk) (mac, win, linux)

<table>
<thead>
<tr>
<th>Android</th>
<th>firefox</th>
<th>chrome</th>
<th>Opera</th>
<th>ref client</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>24+</td>
<td>29+</td>
<td>18+</td>
<td></td>
</tr>
<tr>
<td></td>
<td>24+</td>
<td>29+</td>
<td>18+</td>
<td>appRTCDEMO</td>
</tr>
<tr>
<td></td>
<td>24+</td>
<td>29+</td>
<td>18+</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N/A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>iOS</th>
<th>safari</th>
<th>firefox</th>
<th>chrome</th>
<th>Opera</th>
<th>ref client</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>against apple policy (2.17)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>appRTCDEMO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Not complete yet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N/A</td>
</tr>
</tbody>
</table>

Brought to you thanks to:

Google Inc.
Intel Corporation
MIPS Technologies
Mozilla Foundation
Opera Software ASA
Vonage Holdings Corp

Temasys Communications
History of webRTC plugins (not exhaustive, not flash)

1st Open Source plugin to support IE

GoCast.it

Nov 11, 2011

2nd Open Source plugin to support IE

Sun, 22 Jul 2012

webrtc4all

2012

3rd Open Source plugin to support IE

15 Jan 2014 - Temasys

16 Mar 2014 - FrozenMountain

05 Feb 2014 - Priologic

04 Apr 2014 - Tokbox

1st

2nd

3rd
Comparison: criteria 1

- **Proprietary vs Generic**

  *Proprietary* plugin either forces you to use a given BE, or forces you to use a given API (or both). One need to rewrite app.

  *Generic* plugin implements the webRTC specs, allowing any site or API/SDK written on top of webRTC to work out of the box. It’s **FREE** as in freedom. It just unlocks webRTC in the browser and you can keep working with your own vendor if you want. One plugin for all, all IT department can validate it once for all vendors.
Comparison: APIs – webRTC is a moving target

- Doubango: complete up to JSEP-02 (2012)
- GoCast.it: complete up to JSEP-03 (2013)
- Temasys: complete latest JSEP-06, webRTC 1.0

Others... well, they have their own proprietary API

Sometimes:

• no getUserMedia,
• no PeerConnection,
• no DataChannel,

but you get “getSelfie()” so .... It’s cool.
Comparison: criteria 2: Security (should be #1, really)

Follow the IETF security document (prompt for hardware access, protect against cross origin attacks, protect against cross plugin instance stream sharing, ....)
To avoid:
- People taking picture or video of you
- People taking screenshot when you’re banking online
- ...

Without your consent! Among other problems.....

These mechanisms are not in libjingle/webRTC, they are in the browsers. If you want to put it in a plugin, you have to rewrite it, wrapping the reference code is NOT ENOUGH!
Comparison: Security – Why screen sharing was pulled out.

Yikes! http://same.io asks users to manually enable the WebRTC screenshare flag, which turns it on for all sites. There are a bunch of potential security concerns here, summarized at http://tools.ietf.org/html/draft-ietf-rtcweb-security-06#section-4.1.1.

Browser vendors and W3C work really hard to make the web a better, more secure, place. They are not happy when some vendors bypass security. You want to develop a plugin? Ask the experts.
Comparison: criteria 3

- **Test Suite:** “if it ain’t tested, it’s broken.”

- webRTC is a moving target,

  ⇒ you have to test with the same standards as the browsers

  ⇒ You have to test against the browsers
Comparison: criteria 4

- **Integration layer**: if you spend more than a day installing, you won’t use it.

- Provide integration layer (adapter.js is a good candidate, everybody copied appRTC)
- Provide examples, tutorial.
- Provide support (yes, to your otherwise competitors as well)

- Ex: jsSIP on safari
  -- with a generic plugin
  -- in less than a day
  -- no change in app code
  -- no support needed
## Comparison: Overview

<table>
<thead>
<tr>
<th>Provider</th>
<th>webRTC API</th>
<th>API version</th>
<th>Security</th>
<th>Middleware</th>
<th>Plugin type</th>
<th>Browser support</th>
<th>OS Support</th>
<th>Test Suite</th>
<th>Integration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temasys</td>
<td>YES</td>
<td>2014 (06)</td>
<td>YES</td>
<td>YES</td>
<td>NPAPI ActiveX</td>
<td>IE and Safari</td>
<td>win, mac</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>GoCast</td>
<td>YES</td>
<td>2013 (03)</td>
<td>YES</td>
<td>YES</td>
<td>NPAPI ActiveX</td>
<td>IE and Safari (not complete)</td>
<td>win, mac</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Doubango</td>
<td>YES</td>
<td>2012 (02)</td>
<td>YES</td>
<td>NO</td>
<td>NPAPI ActiveX</td>
<td>IE and Safari</td>
<td>win</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>Tokbox</td>
<td>NO</td>
<td>-</td>
<td>YES</td>
<td>YES</td>
<td>ActiveX</td>
<td>IE</td>
<td>win</td>
<td>N/A</td>
<td>YES</td>
</tr>
<tr>
<td>Frozen mountain</td>
<td>NO</td>
<td>-</td>
<td>?</td>
<td>NO</td>
<td>JAVA ActiveX (May 2014)</td>
<td>IE</td>
<td>win</td>
<td>N/A</td>
<td>YES</td>
</tr>
<tr>
<td>Priologic</td>
<td>NO</td>
<td>-</td>
<td>NO</td>
<td>NO</td>
<td>ActiveX</td>
<td>IE</td>
<td>win</td>
<td>NO</td>
<td>NO</td>
</tr>
</tbody>
</table>
Next steps:

Build chain of trust  
(37 companies so far)

Help deliver Native implementation faster.
- Bring webRTC to webkit – all can help
- Keep working in the standard committees and in ORTC Community Group to bring MS to the table.

<table>
<thead>
<tr>
<th>Desktop</th>
<th>chrome</th>
<th>firefox</th>
<th>IE</th>
<th>Safari</th>
<th>Opera</th>
<th>ref client</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUM</td>
<td>18+</td>
<td>17+</td>
<td></td>
<td>plugin</td>
<td></td>
<td>webkit</td>
</tr>
<tr>
<td>PC</td>
<td>20+</td>
<td>22+</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DC</td>
<td>26+</td>
<td>22+</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>iOS</th>
<th>safari</th>
<th>firefox</th>
<th>chrome</th>
<th>Opera</th>
<th>ref client</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>webkit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N/A</td>
</tr>
</tbody>
</table>
Thank You

Please remember to complete an evaluation of today’s sessions