Real-Time Communication Testing Evolution with WebRTC

Dr Alex Gouaillard

CoSMo Software Consulting

IMTC, W3C, IETF
RTC Interoperability: Web Apps

Google Allo
Google Duo
Cisco Spark
Chromecast
(JS API) Compliance and Usual Testing

• W3C Standardization process (features supported or not)

• Stand-alone tests

• Single browser (one at a time)
What about WebRTC 1.0?

Why so low test coverage?

- Some parts of specifications were just written
- Tests not needed early on in standardization process
- Many tests need P2P communication / interop testing
Ex. of RTC Feature Not Testable Before: Network and Protocol Testing: ICE

What is my public IP address?

Your public IP address is x.x.x.x:y

STUN server

Client A

Client B
## Compliance and Usual Testing vs. Interoperability Testing

<table>
<thead>
<tr>
<th>Compliance &amp; Usual Testing</th>
<th>Interoperability Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Interaction automation</td>
<td>• Interaction automation</td>
</tr>
<tr>
<td>• Stand-alone tests</td>
<td>• Two <strong>synchronized</strong> instances of the same test</td>
</tr>
<tr>
<td>• Single browser</td>
<td>• 2+ browsers run <strong>in parallel</strong></td>
</tr>
</tbody>
</table>

⇒ New testing tools needed
Browsers Interaction Automation: WebDriver Protocol

- ChromeDriver
- geckodriver
- Microsoft WebDriver
- SafariDriver

Workflow of Selenium WebDriver

Test Scripts

Webdriver

Browsers

Languages: Java, perl, Ruby, PHP, Python, C#
Browsers as a Service: Hosted Selenium Nodes

SAUCE LABS

BrowserStack
In-Progress: WebDriver Needs Update for WebRTC Security & Others

• Permission prompt
  – Geolocation
  – Camera
  – Microphone
  – Bluetooth
  – ...

• Fake Media / Mock Capturer
• ICE Candidate Filtering
• Unsecure origin
Wrap-Up: WebDriver for RTC Tests

• Interaction

• Multiple browsers in parallel

• Synchronization of instances of same test between browsers
Lots of Tools Out There
Addressing a Subset of the Problem

- ir prognosis
- VALID8.com
- WebRTC-test
- Multiplier
- Kurento Testing Framework
- testRTC
- Jitsi-Hammer
- JANVS WebRTC Gateway
- Jattack
OK, Existing Tools are Limited but ... How Bad Can It Be, Really?

Rough Goal: Test all possible pairs of

[Browser] x [Browser Revision] x [OS] x [OS Revision]
What we WANT to test
What we could test today

© Dr. Alex Gouaillard @ Citrix Systems, 2016. Document provided under CC BY-NC 4.0
What we ACTUALLY test today
Ok, It’s REALLY Bad ...
Get Visibility: List and Evaluate
Evaluation of Interop Testing Tools
Dimension 1: Tests Triggers and Frequency

6 months

Weekly (2d)

Daily / Nightly (4h)

Commit-based and fast (20min)

Experimental (manually triggered)
### Evaluation of Interop Testing Tools

**Dimension 2: Type of SE Nodes I Want to Use**

<table>
<thead>
<tr>
<th>Type of SE Nodes</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-Hosted, Pre-configured VM</td>
<td>Apple</td>
</tr>
<tr>
<td>Hosted, as-a-service (e.g. BrowserStack)</td>
<td>Only stable browsers, Extensions and plug-ins, Modified browsers</td>
</tr>
<tr>
<td>Self-Hosted, Ephemeral VM</td>
<td>No manual settings, Extensions and plug-ins, Apple</td>
</tr>
<tr>
<td>Hosted, Ephemeral VM (e.g. Travis)</td>
<td>Limited OS set</td>
</tr>
<tr>
<td>Local / physical machines</td>
<td></td>
</tr>
</tbody>
</table>
Evaluation of Interop Testing Tools

Dimension 3: OS I Want to Support

Different operating systems in Percent

platform

An...id BSD Ch...S Ki...re Kn...le Kn...le Linux Ma...S Robot Un...n Wi...S Wi...e Wi...s Wi...e Wi... iOS
Evaluation of Interop Testing Tools

Dimension 4: Which Browsers?

36+% of Enterprise market (Windows 7)
Which plugin?

WebRTC? What’s your flavor~~?
Permission prompt? Fake Media?

No Apple VM on non-Apple Hardware
WebDriver is more recent and less mature than others

Easy, kind of tested already, great WebDriver support
Evaluation of Interop Testing Tools

Dimension 5: How Early do I Want to Test?

=> How fast do you want your fix in

Ex: Chrome 58 → Chrome 61
Evaluation of Interop Testing Tools

Dimension 6: Freedom, Convenience and Price

- On-premises
- Own cloud
- Hosted
Ok, I See What’s Missing for Each
What’s the Right Design Now?
KITE: Karoshi Interoperability Testing Engine

Because for most of us, manually testing WebRTC interoperability felt like “death by overwork”
KITE: Karoshi Interoperability Testing Engine
Automating RTC P2P tests

•_Generic (any app, any back-end)
•_Reusable (any process, trigger, ...)
•_Easy to maintain (no death 😊 )

Stable / Beta / Dev / Canary

On-premises
Own cloud
Hosted
KITE Design

Easy maintenance by divide and conquer

• Selenium Grid
• Test engine
• Test
• Dashboard
KITE Design

Conductor

- \( \{ N, \text{Test}, \text{Callback} \} \)
- \( \text{list<configs>} \)

Dispatcher

Config \( \Rightarrow \) BrowserObj
Compute \( \text{list< N-Tuple< BrowserObj >}> \)

- \( \{ \text{Test}, \text{Callback} \} \)
- \( \text{list< N-Tuple< BO >}> \)

handle concurrency
if resource: 1 N-tuple \( \Rightarrow \) 1 test runner \( \Rightarrow \) 1 thread

Test Runner

- \( \{ \text{Test}, \text{Callback} \} \)
- 1 N-Tuple< BO >

BrowserObj available
- Cr/FF - Mac - Rem-BB
- BrX - OSX - Rem-BaaS (SauceLabs)
- BrY - OSY - Rem-VM (AWS)

Result Dashboard

\[ \{ \text{N, Test, Callback} \} \]
\[ \text{list<configs>} \]
Configuration File

{
  "name": "local selenium example",
  "callback": "http://localhost:8080/kiteweb/datacenter",
  "remotes": [
    {
      "type": "local",
      "remoteAddress": "http://localhost:4444/wd/hub"
    }
  ],
  "tests": [
    {
      "name": "IceConnectionTest",
      "tupleSize": 2,
      "testImpl": "org.webrtc.kite.IceConnectionTest"
    }
  ],
  "browsers": [
    {
      "browserName": "firefox"
    },
    {
      "browserName": "chrome"
    }
  ]
}
Configuration File

```json
{
    "name": "local selenium example",
    "callback": "http://localhost:8080/kiteweb/datacenter",
    "remotes": [
        {
            "type": "local",
            "remoteAddress": "http://localhost:4444/wd/hub"
        }
    ],
    "tests": [
        {
            "name": "IceConnectionTest",
            "tupleSize": 2,
            "testImpl": "org.webrtc.kite.IceConnectionTest"
        }
    ],
    "browsers": [
        {
            "browserName": "firefox"
        },
        {
            "browserName": "chrome"
        }
    ]
}
```
Configuration File

```json
{
    "name": "local selenium example",
    "callback": "http://localhost:8080/kiteweb/datacenter",
    "remotes": [
        {
            "type": "local",
            "remoteAddress": "http://localhost:4444/wd/hub"
        }
    ],
    "tests": [
        {
            "name": "IceConnectionTest",
            "tupleSize": 2,
            "testImpl": "org.webrtc.kite.IceConnectionTest"
        }
    ],
    "browsers": [
        {
            "browserName": "firefox"
        },
        {
            "browserName": "chrome"
        }
    ]
}
```

Registration of Selenium server
Configuration File

{
  "name": "local selenium example",
  "callback": "http://localhost:8080/kiteweb/datacenter",
  "remotes": [
    {
      "type": "local",
      "remoteAddress": "http://localhost:4444/wd/hub"
    }
  ],
  "tests": [
    {
      "name": "IceConnectionTest",
      "tupleSize": 2,
      "testImpl": "org.webrtc.kite.IceConnectionTest"
    }
  ],
  "browsers": [
    {
      "browserName": "firefox"
    },
    {
      "browserName": "chrome"
    }
  ]
}
Configuration File

```
{
"name": "local selenium example",
"callback": "http://localhost:8080/kiteweb/datacenter",
"remotes": [
{
 "type": "local",
 "remoteAddress": "http://localhost:4444/wd/hub"
}
],
"tests": [
{
 "name": "IceConnectionTest",
 "tupleSize": 2,
 "testImpl": "org.webrtc.kite.IceConnectionTest"
}
]
"browsers": [
{
 "browserName": "firefox"
},
{
 "browserName": "chrome"
}
]
}
```
WELCOME TO KITE DASHBOARD
Local, saucelabs grid
Started at: 27-Sep-2017 10:59:23 - Including 1 test(s)

1. IceConnectionTest

OVERALL

IMPLEMENTATION CLASS: ORG.WEbrtc.KITE.IceConnectionTest

<table>
<thead>
<tr>
<th>Status</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tuple size</td>
<td>2</td>
</tr>
<tr>
<td>Number of test cases</td>
<td>4</td>
</tr>
<tr>
<td>Status</td>
<td>Done</td>
</tr>
<tr>
<td>ETA</td>
<td>27-Sep-2017 11:01:05</td>
</tr>
</tbody>
</table>
Result(s) for: AppRTC ICEConnection Test

Started at: 19-Jun-2017 20:05:51

including 24 test case(s)

*All test cases with problems (not successful) will be displayed as FAILED
Overview: AppRTC ICEConnection Test

<table>
<thead>
<tr>
<th>TOTAL</th>
<th>SUCCESSFUL</th>
<th>FAILED</th>
<th>ERROR</th>
</tr>
</thead>
<tbody>
<tr>
<td>250</td>
<td>243 (97 %)</td>
<td>7 (2 %)</td>
<td>0 (0 %)</td>
</tr>
</tbody>
</table>

Details:
First run: 16-Jun-2017 18:47:42
Last run: 22-Jun-2017 14:45:10

4 run(s):
1. 16-Jun-2017 18:47:42  225 test case(s)
2. 16-Jun-2017 20:20:59  226 test case(s)
3. 19-Jun-2017 20:12:31  24 test case(s)
4. 22-Jun-2017 14:45:10  34 test case(s)

*Showing results for browsers with versions stable, dev and beta only.*
That’s Very Nice
But It Feels Like Something is Missing...
Global RTC Interoperability

- Google Allo
- Google Duo
- Cisco Spark
- Chromecast
Global RTC Interoperability: MOBILE (Browsers)?

- Chrome
- Android
- Google Allo
- iOS
- Google Duo
- Cisco Spark
- Firefox
- Chromecast
Global RTC Interoperability Beyond Desktop Browsers

Appium

• Mobile testing framework

• For native, hybrid and mobile web apps
Global RTC Interoperability
Beyond Desktop Browsers

Electron

• Open source framework

• To build cross platform desktop applications (HTML, CSS, JS)

• Combines Chromium and node.js into a single runtime
Global RTC Interoperability: Beyond 1-1

Jitsi

- Open source multiplatform
- Videoconferencing
- Instant messaging
Global RTC Interoperability: Beyond 1-1

Overview: Jitsi ICEConnection Test

<table>
<thead>
<tr>
<th>TOTAL</th>
<th>SUCCESSFUL</th>
<th>FAILED</th>
<th>ERROR</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>64 (100 %)</td>
<td>0 (0 %)</td>
<td>0 (0 %)</td>
</tr>
</tbody>
</table>

Details:
First run: 18-Jun-2017 21:01:58
Last run: 10-Aug-2017 17:35:30

4 run(s):
1. 19-Jun-2017 21:01:58
   64 test cases
   64 test cases
3. 20-Jun-2017 08:50:07
   64 test cases
4. 10-Aug-2017 17:35:30
   1 test cases

*Showing results for browsers with versions stable, dev and beta only
Global RTC Interoperability
Make it easy: Grid Manager

• Set up a grid through a web app

• Set up Selenium hubs and nodes

• Download and set up
  – Browsers
  – Browsers WebDriver

• Backup everything for follow-up uses

• Tear-down and clean-up after tests
Thank you!