

KITE for webrtc testing

04-2018 Update

W3C WebRTC interim meeting

WebRTC testing original sins

Original sin => complexify (add features inside) instead of modularity

-> trying to make wpt selenium aware

-> trying to make wpt tests asymmetric and synchronized

-> trying to force a tool (travis)

-> trying to force one signalling

KITE allows to run the unmodified wpt tests in any configurations and to send the result to anything you want, without assumption.

Overview

Goal: testing p2p communication between two webrtc-capable [browsers]

W3C Goal: test webrtc APIs to proceed to standard (compliance)

Browser vendors: fasten implementation, reference test (compliance)
Regression testing, early warning

All: interoperability between webrtc capable [clients], and [back-ends]
Using any signaling protocol.

Open Source code with currently two W3C Members contributing: Google and CoSMo.

KITE Interop: webdriver extension work (w Apple)

Permissions

Special case of GUM as a permission proxy

- Network / ice candidates

- Media device and listing

Fake Media

HTTPS origin

...

KITE Interop SE Grid - Browser configs

(without saucelab, without Mobile, Without Electron)



21 MARCH
2018



23 MARCH
2018



21 MARCH
2018



21 MARCH
2018



23 MARCH
2018



23 MARCH
2018



23 MARCH
2018



23 MARCH
2018



21 MARCH
2018



21 MARCH
2018



21 MARCH
2018



21 MARCH
2018



21 MARCH
2018



23 MARCH
2018



23 MARCH
2018



23 MARCH
2018

KITE Interop: Whatever app, whatever back-end

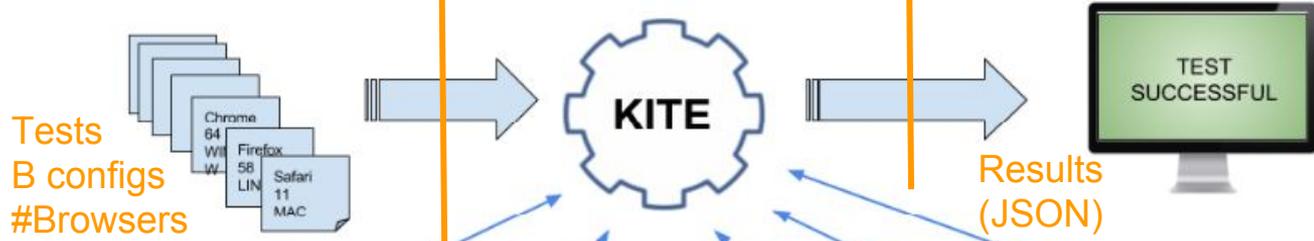
Webrtc does not define signalling, nor back-end. How to test all cases? You also need to be able to have discovery and handshake before testing connections.

=> KITE is app / back-end independant, and does not assume anything about the test. The test is the part where which app and which back-end will be used is decided.

- Run any client against same back end (end-to-end interop)
- reuse tests/runs against prod, staging, dev, or local back-end
- Reuse test/runs against different apps all together.

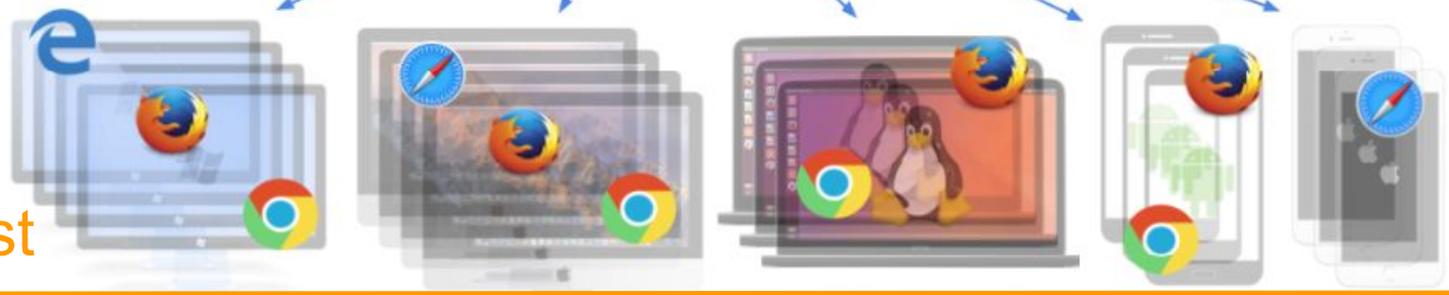
Examples to come: WPT, apprtc, jitsi,

KITE Modular Design



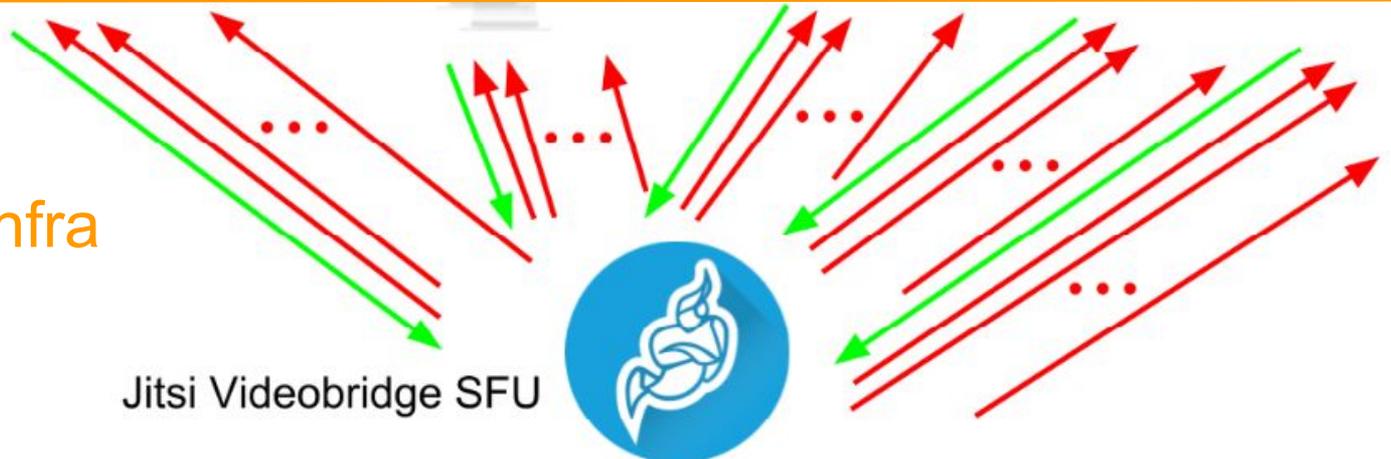
SE Grid

Running test



Test own Infra

Jitsi Videobridge SFU



KITE 1-browser : auto WPT runs

(2d of work for all available configs)

WPT / WEBRTC	23 MARCH 2018	23 MARCH 2018	27 MARCH 2018	23 MARCH 2018	27 MARCH 2018	27 MARCH 2018	23 MARCH 2018	23 MARCH 2018	27 MARCH 2018							
iceConnectionState	0/2	0/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2
RTCPeerConnection-iceGatheringState	0/3	0/3	2/3	2/3	2/3	2/3	2/3	2/3	2/3	2/3	2/3	2/3	2/3	2/3	1/3	1/3
RTCPeerConnection-onDataChannel	0/3	0/3	2/3	2/3	2/3	2/3	2/3	2/3	1/3	1/3	1/3	2/3	2/3	2/3	1/3	1/3
RTCPeerConnection-onnegotiationneeded	0/7	0/7	3/7	3/7	3/7	3/7	3/7	3/7	7/7	7/7	7/7	7/7	7/7	7/7	5/6	5/6
RTCPeerConnection-ontrack.https	0/5	0/5	1/5	1/5	1/5	1/5	1/5	1/5	5/5	5/5	5/5	4/5	5/5	5/5	2/5	2/5
RTCPeerConnection-peerIdentity	0/6	0/6	0/6	0/6	0/6	0/6	0/6	0/6	0/6	0/6	0/6	0/6	0/6	0/6	0/6	0/6
RTCPeerConnection-removeTrack.https	0/12	0/12	0/12	0/12	0/12	0/12	0/12	0/12	9/12	9/12	9/12	4/6	9/12	9/12	2/12	2/12
RTCPeerConnection-setDescription-transceiver	0/5	0/5	0/5	0/5	0/5	0/5	0/5	0/5	4/5	4/5	4/5	4/5	4/5	4/5	0/5	0/5
RTCPeerConnection-setLocalDescription-answer	0/5	0/5	1/6	1/6	1/6	1/6	1/6	1/6	2/6	2/6	2/6	2/6	2/6	2/6	4/6	4/6
RTCPeerConnection-setLocalDescription-offer	0/5	0/5	2/7	2/7	2/7	2/7	2/7	2/7	3/7	3/7	3/7	3/7	3/7	3/7	3/7	3/7
RTCPeerConnection-setLocalDescription-pranswer	0/4	0/4	4/7	4/7	4/7	4/7	4/7	4/7	0/7	0/7	0/7	0/7	0/7	0/7	6/7	6/7

web-platform-tests dashboard

Latest Run Recent Runs About GitHub Source

Search test files, like cors/allow-headers.htm

WPT /

Showing results for run 4a0df34066.

[View latest run](#)

Data below are intended for web platform implementers and do not contain useful metrics for evaluation or comparison of web platform features. Also note that tested Edge and Safari are not pre-release versions (#109, #110).

Spec	 chrome 66.0.3359.117 linux 4.4 @4a0df34066 Apr 18 2018	 edge 15 windows 10 @4a0df34066 Apr 20 2018	 firefox 59.0.2 linux 4.4 @4a0df34066 Apr 18 2018	 safari 11.0 macos 10.12 @4a0df34066 Apr 18 2018
/2dcontext	1924 / 1999	1232 / 1503	1861 / 1999	1654 / 1880
/BackgroundSync	35 / 35	8 / 18	17 / 35	17 / 35
/FileAPI	827 / 842	504 / 708	752 / 827	650 / 707
/IndexedDB	1828 / 1833	464 / 881	1809 / 1833	1531 / 1613
/WebCryptoAPI	40606 / 43207	791 / 17251	14541 / 27366	36328 / 41493
/WebIDL	455 / 470	165 / 254	461 / 470	422 / 470
/accelerometer	23 / 103	21 / 101	23 / 103	23 / 103
/acid	101 / 103	0 / 3	101 / 103	101 / 103
/ambient-light	19 / 57	17 / 55	19 / 57	19 / 57
/apng	2 / 3	1 / 3	3 / 3	2 / 3
/audio-output	18 / 20	3 / 12	4 / 15	4 / 15
/background-fetch	59 / 181	59 / 180	59 / 181	60 / 181
/battery-status	44 / 46	3 / 9	4 / 10	4 / 10
/beacon	51 / 117	71 / 85	100 / 107	22 / 107

KITE 2-browsers: appRTC runs and Stats integration

Daily runs on webrtc.org !

Analysis and bug reports against Browsers, webdrivers impl, ...

ICE stats:

[chrome 65.0 ANDROID] [MicrosoftEdge 16.16299 WINDOWS 10]

LOCAL CANDIDATE	REMOTE CANDIDATE	ICE STATE	NOMINATED	PRIORITY	BYTES SENT	BYTES RECEIVED
RTCIceCandidate_YMvfhdZ	RTCIceCandidate_gWp4gO4C	succeeded	true	9079290933605827000	831542	123116

SDP

Offer Answer

Offer

```
v=0
o=- 48895174166018984 0 IN IP4 127.0.0.1
s=-
t=0 0
a=group:BUNDLE audio video
a=msid-semantic: WMS 5882B0EC-B490-44E4-A31D-9244428F4357
m=audio 50434 UDP/TLS/RTP/SAVPF 111 0 8 126
c=IN IP4 192.168.1.201
a=rtcp:9 IN IP4 0.0.0.0
a=candidate:1 1 udp 2130706431 192.168.1.201 50434 typ host generation 0
a=candidate:2 1 udp 33553919 2001:0:9d38:6abd:2098:14ae:bd9f:336c 61862 typ host generation 0
a=ice-ufrag:UWNR
a=fingerprint:sha-256 66:28:E0:08:AC:93:8E:91:3C:A2:40:CD:04:28:EC:5D:07:EB:B3:97:88:93:A0:D8:EB:7D:D0:08:B1:DC:5D:62
a=setup:active
a=mid:audio
```

Audio (Packets lost: 4 | Avg jitter: 0.00522)

Bytes (avg S/R: 3/5 kB/s) Packets (avg S/R: 61/61 /s)

Video (Packets lost: 0 | Avg jitter: 0.00000)

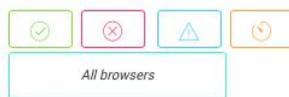
Bytes (avg S/R: 40/0 kB/s) Packets (avg S/R: 47/1 /s)

IceConnectionTest (total: 288)

Started at: 11 Apr, 2018 10:23:43 AM

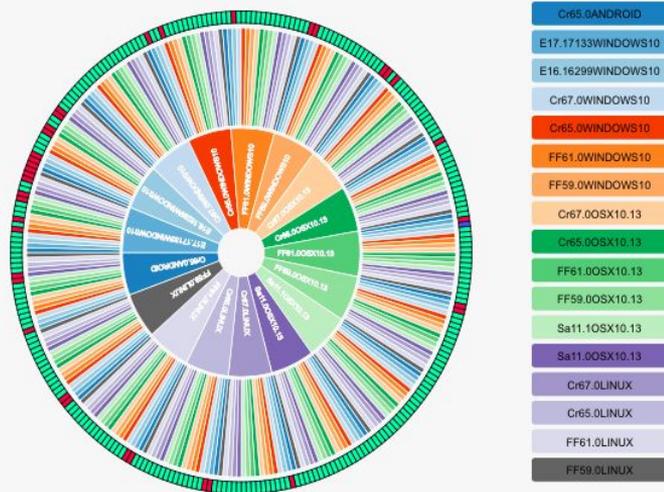


Filter:



STATS	RESULT	DURATION	BROWSER(S)
0	⊗	30s	65.0.ANDROID 17.17133.WINDOWS 10
1	⊙	39s	65.0.ANDROID 16.16299.WINDOWS 10
2	⊙	38s	65.0.ANDROID 67.0.WINDOWS 10
3	⊙	35s	65.0.ANDROID 65.0.WINDOWS 10
4	⊙	34s	65.0.ANDROID 61.0.WINDOWS 10

SUCCESSFUL



KITE 3-browsers: Jitsi runs

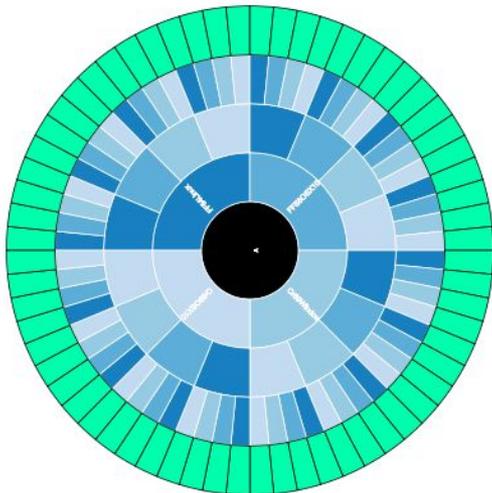
Result(s) for: Jitsi ICEConnection Test

Started at: 20 Jun, 2017 9:56:07 AM

including 64 test case(s)



- FF54Linux
- FF53OSX10.12
- Cr59Windows8.1
- Cr59OSX10.11



STATUS	DURATION	BROWSER(S)
✓	15s	54Linux 54Linux 54Linux
✓	25s	54Linux 54Linux 53OS X 10.12
✓	12s	54Linux 54Linux 59Windows 8.1
✓	19s	54Linux 54Linux 59OS X 10.11
✓	13s	54Linux 53OS X 10.12 54Linux
✓	27s	54Linux 53OS X 10.12 53OS X 10.12
✓	31s	54Linux 53OS X 10.12 59Windows 8.1
✓	26s	54Linux 53OS X 10.12 59OS X 10.11
✓	15s	54Linux 59Windows 8.1 54Linux
✓	28s	54Linux 59Windows 8.1 53OS X 10.12
✓	32s	54Linux 59Windows 8.1 59Windows 8.1
✓	22s	54Linux 59Windows 8.1 59OS X 10.11
✓	26s	54Linux 59OS X 10.11 54Linux

KITE Update: new tests for multiparty, multistream, simulcast ...

Multiparty -> jitsi test. No limit in the number of parties ... in KITE :-)

Multistream -> testing against Unified Plan, helping chrome delivering faster
(see next slide)

Simulcast -> Requires SFU. Only sender->SFU defined in webrtc 1.0, logic for layer switching not defined. Lots of arbitrary decisions for now. See next slides.

KITE: multistream and Unified Plan

Multi-stream (Unified plan):

- It runs a local signaling server (node), currently with and without adapter.js
- Using streams from html video elements for peer connection.
- The test verifies :
 - that the generated SDPs are compliant with the unified plan format (sender).
 - the received stream ids against ids announced in the SDP offer (receiver).
 - that media is flowing (with canvas sum as usual).

Results (4/12):

- Firefox: it works as expected, even without adapter.js
- Chrome: there are some errors, and the flags provided do not seem to work yet. In Progress.

KITE: Simulcast

Simulcast

- the dedicated app runs over https and is available at <https://simulcast-test.dev.cosmosoftware.io/>
- The test verifies the following (not in order):
 - echoed stream is displayed from loopback peerconnection.
 - stream is sent back from SFU (validates it received it, format was correct, and it could extract the right layer).
 - access to SDP offer/answer from peerconnection object.
 - SDP offer/answer format.

Results (4/12):

- it works as expected on Firefox only, but using the SDP format from the older draft. However, that should be considered a failure for strict compliance testing, since the SDP is not using the latest specs format.

KITE: Load and Infra Testing - Config file

Different configuration file with the notion of time/frequency, and multiplier:

For example, to run a test at midnight with 50,000 Chrome at a time only need one entry to be defined:

```
[ 0 0 0, cr 53 win 10, 50000 ]
```

The variations across time are equally easy to write “crontab-style”, e.g. add one Chrome 53 on Windows 10 every 20 seconds:

```
[ */20 * *, cr 53 win 10, 1 ]
```

KITE: Load Testing - Grid Manager

- Scalability:
 - Multiple SE hubs
 - Allowed for Multi cloud vendors
- On-demand spawning/closing of hubs/nodes
- On-Demand network configuration / instrumentation (with callstats.io)
 - Test bandwidth estimation
 - Test simulcast / SVC / bitrate adaptation vs bandwidth scenarii
 - Test RTX / RED / FEC behavior against network (jitter, packet loss, ...)
- Tested against most open-source webrtc SFU, results proposed to IPTComm
- Tested with a customer up to 50,000 concurrent clients (streaming use case)

