W3C WebRTC WG Meeting

October 22, 2018
Lyon, FR

Chairs: Bernard Aboba
Harald Alvestrand

W3C WG IPR Policy

- This group abides by the W3C Patent Policy https://www.w3.org/Consortium/Patent-Policy/
- Only people and companies listed at https://www.w3.org/2004/01/pp-impl/47318/status are allowed to make substantive contributions to the WebRTC specs

Welcome!

- Welcome to the TPAC meeting of the W3C WebRTC WG!
- During today's sessions, we hope to:
  - Make progress on open issues in screen sharing, audio output, media capture and streams, webrtc-pc, webrtc-stats and other current specifications.
  - Discuss the status of implementations and interoperability testing
  - Discuss how to remove roadblocks to bringing WebRTC to Proposed Recommendation (PR)
  - Discuss next generation use cases and potential additional work items.
- Will update editors drafts after the meeting

About this Meeting

Information on the meeting:

- Meeting info:
  - https://www.w3.org/2011/04/webrtc/wiki/October_22-23_2018
- Links to latest drafts:
  - https://w3c.github.io/mediacapture-main/
  - https://w3c.github.io/mediacapture-output/
  - https://w3c.github.io/mediacapture-screen-share/
  - https://w3c.github.io/webrtc-pc/
  - https://w3c.github.io/webrtc-stats/
  - https://www.w3.org/TR/mst-content-hint/
  - https://w3c.github.io/webrtc-nv-use-cases/
  - https://w3c.github.io/webrtc-dscp-exp/
- Link to Slides has been published on WG wiki
- Scribe? IRC http://irc.w3.org/ Channel: #webrtc
- The meeting is being recorded.
- Hangouts info has been sent to you in email.

Meeting Schedule at TPAC
October 22, 2018 (Morning)

8:30 AM - 9:00 AM State of the WEBRTC WG (Harald)
Status of specifications and implementations.

9:00 AM - 10:00 AM: Capture and Output (Jan-Ivar)
Screen Capture: https://w3c.github.io/mediacapture-screen-share/
MediaCapture & Streams: https://w3c.github.io/mediacapture-main/
Audio Output Devices API: https://w3c.github.io/mediacapture-output/

10 AM - 10:30 AM Break

10:30 AM - noon WebRTC-PC (Harald)
Reference: https://w3c.github.io/webrtc-pc/

12:00 PM - 1:00 PM Lunch

Meeting Schedule at TPAC
October 22, 2018 (Afternoon)

1:00 PM - 1:30 PM WPT Test Process (Bernard)

1:30 PM - 2:00 PM WPT Testing (Dr. Alex and team)

2:00 PM - 2:30 PM KITE Testing (Dr. Alex and team)

2:30 PM - 3:00 PM Next Steps toward bringing WebRTC-PC to PR (Bernard)

3:00 PM - 3:30 PM Break

3:30 PM - 4:10 PM WebRTC NV use cases (Bernard)
Reference: https://w3c.github.io/webrtc-nv-use-cases/

4:10 PM - 4:30 PM Performance issues with CV/ML (OpenCV.js) Use Cases (Ningxin Hu)

4:30 PM - 5:00 PM WebRTC-ICE (Peter Thatcher)
Reference: https://w3c.github.io/webrtc-ice/

Meeting Schedule at TPAC
October 23, 2018 (Morning)

8:30 AM - 9:30 AM Scalable Video Coding Extension for WebRTC (Bernard)
Reference: https://rawgit.com/aboba/webrtc-sim/master/svc.html

9:30 AM - 10 AM Access to Raw Media (Harald)
Reference: https://alvestrand.github.io/audio-worklet/

10:00 AM - 10:30 AM Break

10:30 AM - 11 AM Data Channel and WHAT WG Streams (Jan-Ivar)

11:00 AM PM - 11:30 AM QUIC and WHATWG Streams (Peter Thatcher)
Reference: https://w3c.github.io/webrtc-quic/

11:30 AM - noon Second Screen WG (Peter Thatcher)
Reference: https://www.w3.org/2014/secondscreen/charter-2018.html

Noon - 1 PM Lunch

Meeting Schedule at TPAC
October 23, 2018 (Afternoon)

1:00 PM - 2:00 PM Workers and Worklets (Youenn Fablet)

2 PM - 3 PM Remaining WebRTC issues and Other current specifications (Varun)
WebRTC-Stats: https://w3c.github.io/webrtc-stats/
Identity: https://w3c.github.io/webrtc-identity/identity.html
Content-Hints: https://w3c.github.io/mst-content-hint/
DSCP: https://www.w3.org/TR/webrtc-dscp/

3 PM - 3:30 PM Break

3:30 PM - 4:00 PM End-to-End Encryption (Emad?)

4:00 PM - 4:30 PM Media over QUIC (Peter Thatcher)

4:30 PM - 5:30 PM Wrapup and Next Steps (Harald)

State of the WebRTC WG
Harald Alvestrand (30 minutes)

What we're chartered to do

- Finish WebRTC 1.0 (HIGH PRIORITY)
- Define an object-oriented API (based on ORTC)
- Describe requirements for new use cases
- Address those use cases
  - New protocols (and associated APIs)
  - New data access functions

What our environment demands

- WebRTC 1.0 should “just work”
  - Across all browsers
  - In all networks
- Low level data access
  - In a performant manner (example: link)
- Son of ORTC
  - Although the pressure seems to have decreased

Media Capture and Streams

- Candidate Recommendation (Oct 17)
- 20 open issues
  - 12 of which are > 3 months old
- Interoperability matrix shows lots of things working in ¾ of browsers
- Community sense seems to be “works”
- Promise: PR in Q4 2018 (that’s now!)

WebRTC-1.0

- Candidate Recommendation
  - Renewed Sep 18 (separated Identity spec)
- 46 open issues
  - 31 are > 3 months
- Interoperability matrix shows lots of issues, but also lots of interoperability
- Confluence map shows implementation progress (see RTCPeerConnection entries).
- Community sense “in development”?
- Promise: PR in Q3 2019

WebRTC-Identity

- Candidate Recommendation (split sept)
- 23 open issues
  - 22 older than 3 months
- Test suite has not been separated
- Promise: PR in Q3 2019 (as for webrtc-pc)
- Community sense: “Not much happening”

Resources available to WG

- Editors: 2 editors (Jan-Ivar and Henrik) currently active on mediacapture-streams, screen-capture and webrtc-pc
  - Some others contribute PRs - THANK YOU!
- Adam, Taylor and Dan have left the editor team since last TPAC
  - THANK YOU for all your efforts!
- Other drafts managed by other editors

Where resources come from

- People are motivated to get stuff done that they care about
- Organizations sponsor people to get stuff done that they care about
- W3C is a “gift economy” - to make something happen, volunteer to work on it!
- Careful balance of “polish” vs “new work” needed - otherwise, new work goes elsewhere

Other documents - active

- Screenshare - active work
  - Triggered by external event (chrome app store)
  - Push to bring functionality up to par with existing implementations based on gUM.
  - Security still troublesome, but can’t live without
- Recorder - heavy use, updates
  - Also one suggested path for “media access”
- Stats identifiers - updates
  - Linked to webrtc-pc

Other documents - quiet

- Depth - quieted down?
- Audio output devices - in use, little activity
- Content hints - released, PRs merged, only two (minor) open issues.
- DSCP - no code, no activity

We should eventually kill or finish these.

Attention focus for this meeting

- Finish 1.0
  - Get all the bugs resolved
  - Figure out how to get to interop across the board
- Look at new APIs
  - Where what we have is not enough
  - Use cases and requirements are key!
- Attend to Raw Media
  - Because that’s where we’re being asked to go

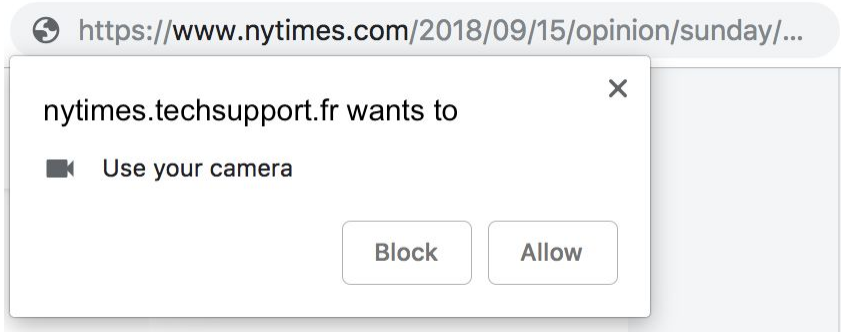
Capture and Output (60 minutes)

For Discussion In This Session

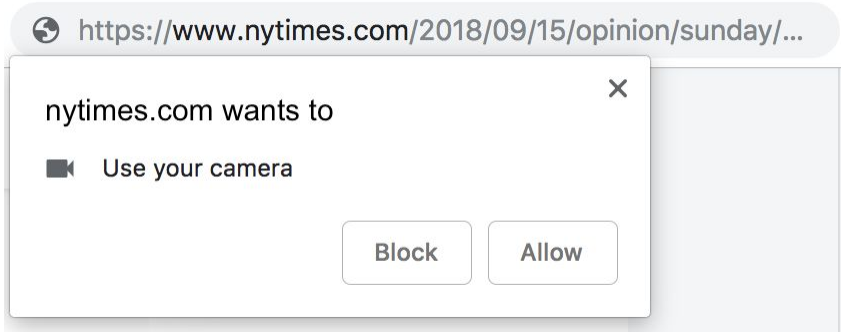
- **Media Capture and Streams**
  - **Issue 532: What does it mean to combine origins? (Jan-Ivar)**
  - **Issue 540: Should getUserMedia be functional in SecureContext only? (Youenn)**
- **Audio Output**
  - **Issue 78: Should setSinkId be functional in SecureContext only? (Jan-Ivar)**

Issue 532/PR 548: What's it mean to combine origins? (jib)

- "Combine" language added in 2016 in #309 to solve iframe gUM permissions.
  Grant to "nytimes.com+techsupport.fr", but NOT to nytimes.com or techsupport.fr alone.
  But users don't understand iframes.



- Overtaken by **Feature Policy**'s `allow`.
  Grant to "nytimes.com" which is responsible for delegating w/ `<iframe allow="camera">` or caller gets `NotAllowedError`.
  (Partly implemented in Chrome & Firefox 64 behind pref)



Mock from w3c/permissions#185 (not yet implemented)

- Conclusion: We can remove "combine" language once #546 is merged.

Issue 532/PR 548: What's it mean to combine origins? (jib)

PR 548 removes *originIdentifier* outright:

- 6. Let originIdentifier be the current settings object's responsible browsing context's [HTML52]
- top-level browsing context's active document's origin.
- 7. If the current settings object's origin is different from originIdentifier, set originIdentifier to the result
- of combining originIdentifier and the current settings object's origin.

9.4. ~~For the origin identified by originIdentifier,~~ Request permission to use a PermissionDescriptor...

...because it was never actually used by the request permission to use algorithm, which instead gets the current settings object's permission state, which now says (PR 163):

+ 3. If there exists a policy-controlled feature identified by descriptor.name and settings has an associated
+ Document named document, run the following step:
+ 1. If document is not allowed to use the feature identified by descriptor.name return "denied".

Q: Good riddance, or was there intent here to dictate something about scope or UX?

Issue 540: Should getUserMedia be functional in SecureContext only? (Jan-Ivar)

Proposal: Limit getUserMedia to [SecureContext] only. This means:

```
console.log('getUserMedia' in navigator.mediaDevices); // false in http
```

Or, limit navigator.mediaDevices itself to SecureContext only?

```
console.log('mediaDevices' in navigator);                // false in http
console.log('getUserMedia' in navigator.mediaDevices); // TypeError in http
```

The latter gets rid of `enumerateDevices()` and `ondevicechange` as well.

Issue 78: Should setSinkId be functional in SecureContext only? (Jan-Ivar)

Proposal: Limit setSinkId to [SecureContext] only. This means:

```
console.log('setSinkId' in document.createElement('audio')); // false in http
console.log('setSinkId' in document.createElement('video')); // false in http
console.log('sinkId' in document.createElement('audio'));    // false in http
console.log('sinkId' in document.createElement('video'));    // false in http
```

For Discussion In This Session (cont'd)

- **Screen Capture**
  - Issue 29: Full screen needs handling (Henrik)
  - Issue 31: Define behavior of existing constraints (Jan-Ivar)
  - Issue 35: Handling source device pixel ratio (Jan-Ivar)
  - Issue 37: Limiting browser sharing to a list of domain/URLs (Jan-Ivar)
  - Issue 71: Unclear how to aggregate windows or handle multiple windows/monitors (Henrik)
  - Issue 79: Constraint to exclude application audio (echo) (Henrik)
  - Issue 81: The user agent should be allowed to change sources after getDisplayMedia resolves (Henrik)
  - Issue 82: Should getDisplayMedia() be moved to navigator.mediaDevices (Youenn)

Issue 29: Full screen needs handling (Henrik)

A shared application may enter fullscreen. Unclear what happens to the track.

- Different ways of entering fullscreen!
- Unclear to the User Agent what the intent is.

Proposal:

- If “window” enters fullscreen: our track is resized.
- If sharing “window”, and new fullscreen window is spawned: we MUST NOT share it; our track MAY become inaccessible (muted).
- If sharing “application”, windows are aggregated. No problem.

UA clarify to user what’s shared. Related issue 81: Allow changing sources!

Issue 31 / PR 84: Define behavior of existing constraints (jib)

The following new and existing MediaStreamTrack Constrainable Properties are defined to apply to the user-selected video display surface, with the following behavior:

Property Name	Type	Behavior
width	`ConstrainULong`	The width or width range, in pixels. As a capability, max *MUST* reflect the display surface's width, and min *MUST* reflect the width of the smallest aspect-preserving representation available through downscaling by the user agent.
height	`ConstrainULong`	The height or height range, in pixels. As a capability, max *MUST* reflect the display surface's height, and min *MUST* reflect the height of the smallest aspect-preserving representation available through downscaling by the user agent.
frameRate	`ConstrainDouble`	The frame rate (frames per second) or frame rate range. As a capability, max *MUST* reflect the display surface's frame rate, and min *MUST* reflect the lowest frame rate available through frame decimation by the user agent.

Issue 31 / PR 84: Define behavior of existing constraints (jib)

aspectRatio	`ConstrainDouble`	The exact aspect ratio (width in pixels divided by height in pixels, represented as a double rounded to the tenth decimal place) or aspect ratio range. As a setting, represents `width / height`. As a capability, min and max both *MUST* be the current setting value, rendering this property immutable from the application viewpoint.
resizeMode	`ConstrainDOMString`	This string (or each string, when a list) should be one of the members of `VideoResizeModeEnum`. As a setting, `none` means the `MediaStreamTrack` contains all bits needed to render the display in full detail, which if `window.devicePixelRatio > 1`, means `width` and `height` will be larger than the display's appearance from an end-user viewpoint would suggest, whereas `crop-and-scale` means the `MediaStreamTrack` contains an aspect-preserved representation of the display surface that has been downscaled by the user agent, but not cropped. As a capability, the values `none` and `crop-and-scale` both *MUST* be present.

displaySurface	ConstrainDOMString	This string (or each string, when a list) should be one of the members of DisplayCaptureSurfaceType. As a setting, indicates the type of display surface that is being captured. As a capability, the setting value *MUST* be the lone value present, rendering this property immutable from the application viewpoint.
logicalSurface	ConstrainBoolean	As a setting, a value of true indicates capture of a logical display surface, whereas a value of false indicates a capture capture of a visible display surface. As a capability, this same value *MUST* be the lone value present, rendering this property immutable from the application viewpoint.
cursor	ConstrainDOMString	This string (or each string, when a list) should be one of the members of CursorCaptureConstraint. As a setting, indicates if and when the cursor is included in the captured display surface. As a capability, the user agent *MUST* include only the set of values from CursorCaptureConstraint it is capable of supporting for this display surface.

Issue 31 / PR 84: Define behavior of existing constraints (jib)

When inherent properties of the underlying source of a user-selected display surface change, for example in response to the end-user resizing a captured window, and these changes render the capabilities and/or settings of one or more constrainable properties outdated, the user agent *MUST* queue a task to run the following step:

1. Update all affected constrainable properties at the same time.

   If this causes an "overconstrained" situation, then the user agent *MUST* ignore the culprit constraints for as long as they overconstrain. The user agent *MUST NOT* mute the track, and the user agent *MUST NOT* fire the overconstrained event.

> ***NOTE***
> ***While min and exact constraints produce TypeError on getDisplayMedia(), this specification does not alter the track.applyConstraints() method. Therefore, they may instead produce OverconstrainedError or succeed depending on values, and therefore potentially be present to cause this "overconstrained" situation. The max constraint may also cause this, e.g. with aspectRatio. This spec considers these to be edge cases that aren't useful.***

§ 5.4.1 Downscaling and Frame Decimation

For the purposes of the SelectSettings algorithm, the user agent *SHOULD* consider all possible combinations of downscaled dimensions that preserve the aspect ratio of the original display surface (to the nearest pixel), and frame rates available through frame decimation, as available settings dictionaries.

The downscaling and decimation effects of constraints is then effectively governed by the fitness distance algorithm.

The intent is for the user agent to produce output that is close to the ideal `width`, ideal `height`, and/or ideal `frameRate` when these are specified, while at all times preserving the aspect ratio of the original display surface.

The user agent *SHOULD* downscale by `window.devicePixelRatio` by default, unless otherwise directed by applied constraints.

The user agent *MUST NOT* crop the captured output.

The user agent *MUST NOT* upscale the captured output, or create additional frames.

> ***NOTE***
> ***The max constraint type lets a web application provide a maximum envelope for constrainable properties like width and height. This is helpful to limit extreme aspect ratios, should the end-user resize a window or browser surface to such an extreme while it is being captured.***

Issue 31 / PR 84: Define behavior of existing constraints (jib)

Live demo:
https://jsfiddle.net/jib1/sLbnk4aj

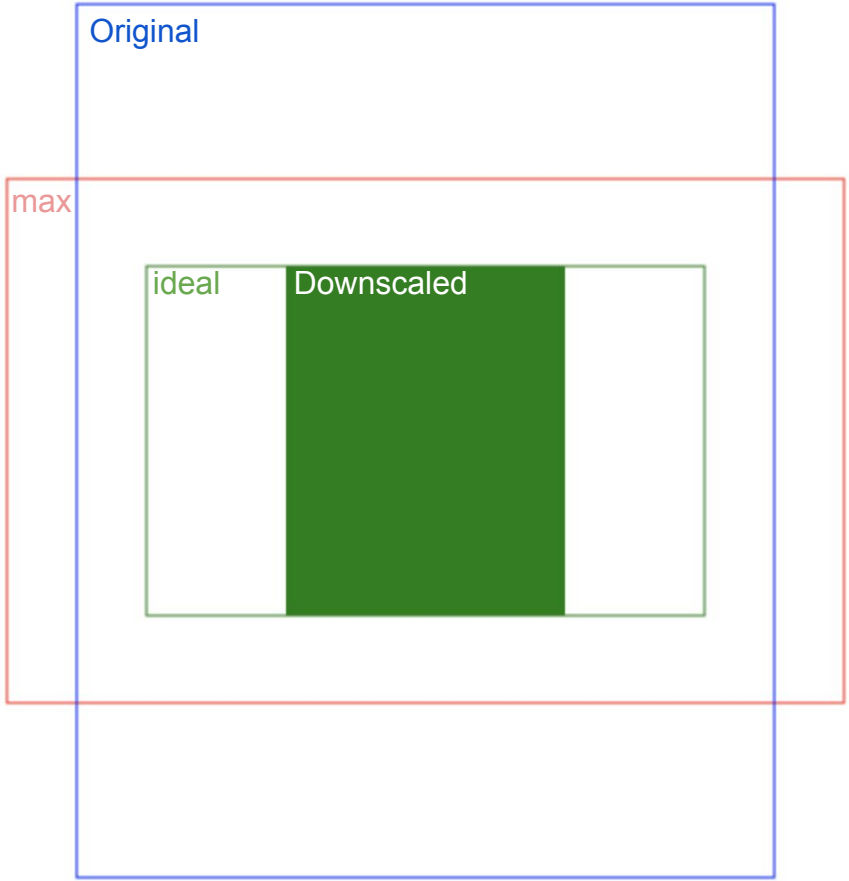
Click anywhere to resize the original.
Max is unused in the demo and is for show.

Handles extreme aspects better than expected.

Grows outside of ideal for squarish sources (possibly from multiple best candidates?) so maybe good to keep max constraint around?

UAs may be able to avoid these outliers, by picking equal candidates closer to either ideal.

As with gUM, UAs have final say with ideal.



Issue 31: Define behavior of existing constraints (Jan-Ivar)

Recap slide #1 (in case of questions about existing behavior):

Behavior differs from cameras. Constraints for downscaling only, ***not*** discovery. Therefore, min, exact, and advanced are disallowed outright:

```
await navigator.getDisplayMedia({video: {width: {min: 320}}});         // TypeError
await navigator.getDisplayMedia({video: {width: {exact: 320}}});       // TypeError
await navigator.getDisplayMedia({video: {advanced: [{width: 320}]}}); // TypeError
```

But the ideal and max constraints are allowed (more on this later).

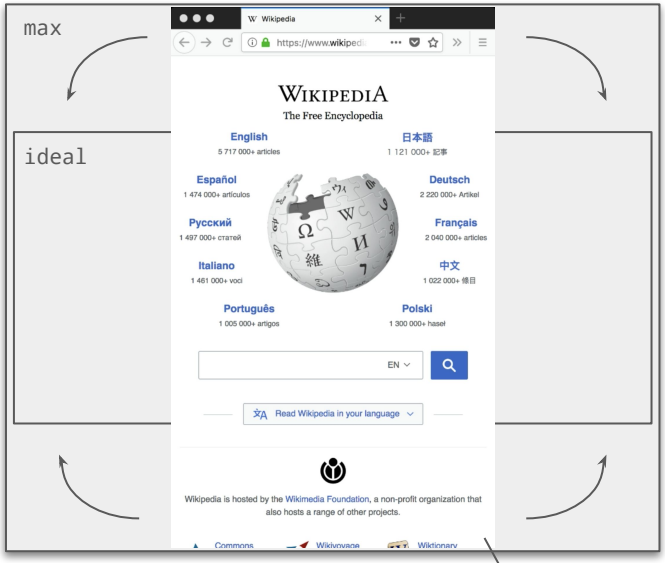
This eliminates the risk of post-prompt `OverconstrainedError` and discovery.

Issue 31: Define behavior of existing constraints (Jan-Ivar)

Recap slide #2 (if needed): Use max constraint to define outer bounds

Why? End-user might resize window extremely tall or wide during live capture.

```
await gDM({video: {width: {ideal:320, max:320}, height: {ideal:200, max:340}}});
```



User resizes live. Video stays inside bounds

Having two forms of constraints allows room for aspect changes within limits.

If end-user makes window any taller, it's always downscaled to fit within outer bounds.

Issue 35 / PR 84: Handling source device pixel ratio (Jan-Ivar)

Or: **How to handle Retina™ displays?** (`window.devicePixelRatio > 1`)

- Q: What do width/height constraint numbers deal in? Answer: size of *data*.
- Cue from: How to canvas-draw in Retina? Answer: Double it & scale down.

Proposal: How to screen-capture in Retina? Answer: Double it & scale down:

> *"UAs SHOULD downscale by* `devicePixelRatio` *by default."*

```
const [track] = (await navigator.getDisplayMedia({video: true})).getVideoTracks();
const isRetina = track.getSettings().resizeMode == 'crop-and-scale';

await track.applyConstraints({resizeMode: 'none'}); // get every bit of data
const cap = track.getCapabilities();
await track.applyConstraints({width: cap.width.max, height: cap.height.max});
```

No new API needed.

Issue 37: Limiting browser sharing to domain/URL list (jib)

- Issue: *Please consider adding another constraint which will be relevant for "browser". A commercial product may (and will) need to limit screen sharing to only those tabs which were open from a white-list of domains or even urls. And vice versa - it will be very useful to support a black list of domains/urls (never share contents if a tab is navigated to this address).*

- Unfortunately, this is a form of influencing user selection, which a malicious site may use to direct users to sharing a web surface under attacker control.

Issue 37: Limiting browser sharing to domain/URL list (jib)

- Spec expressly forbids it:
  - *”The user agent MUST let the end-user choose which display surface to share out of all available choices every time, and MUST NOT use constraints to limit that choice. Instead, constraints MUST be applied to the media chosen by the user, only after they have made their selection. This prevents an application from influencing the selection of sources”*
  - *“UAs are encouraged to warn users against sharing browser display devices and monitor display devices where browser windows are visible, or otherwise try to discourage their selection on the basis that these represent a significantly higher risk when shared.”*

- It’s why getDisplayMedia() didn’t allow constraints until a few months ago. Constraints were reinstated thanks to the above strong language.

Issue 37: Limiting browser sharing to domain/URL list (jib)

- Reasons outlined in Security and Permissions:
  - *“Display capture presents a less obvious risk to the cross site request forgery protections offered by the browser sandbox. Display and capture of information that is also under the control of an application, even indirectly, can allow that application to access information that would otherwise by inaccessible to it directly.” [...]*
  - *“This issue is discussed in further detail in [RTCWEB-SECURITY-ARCH] and [RTCWEB-SECURITY]. Display capture that includes browser windows, particularly those that are under any form of control by the application, risks violation of these basic security protections.” [...] ”It is strongly advised that elevated permissions be required to access any display surface that might be used to circumvent cross-origin protections for content.”*
- Firefox plan is to remove “screen” vs “window” distinction in legacy API in transition.
- Proposal: close

Security concerns

Full-screen/browser sharing is scary!

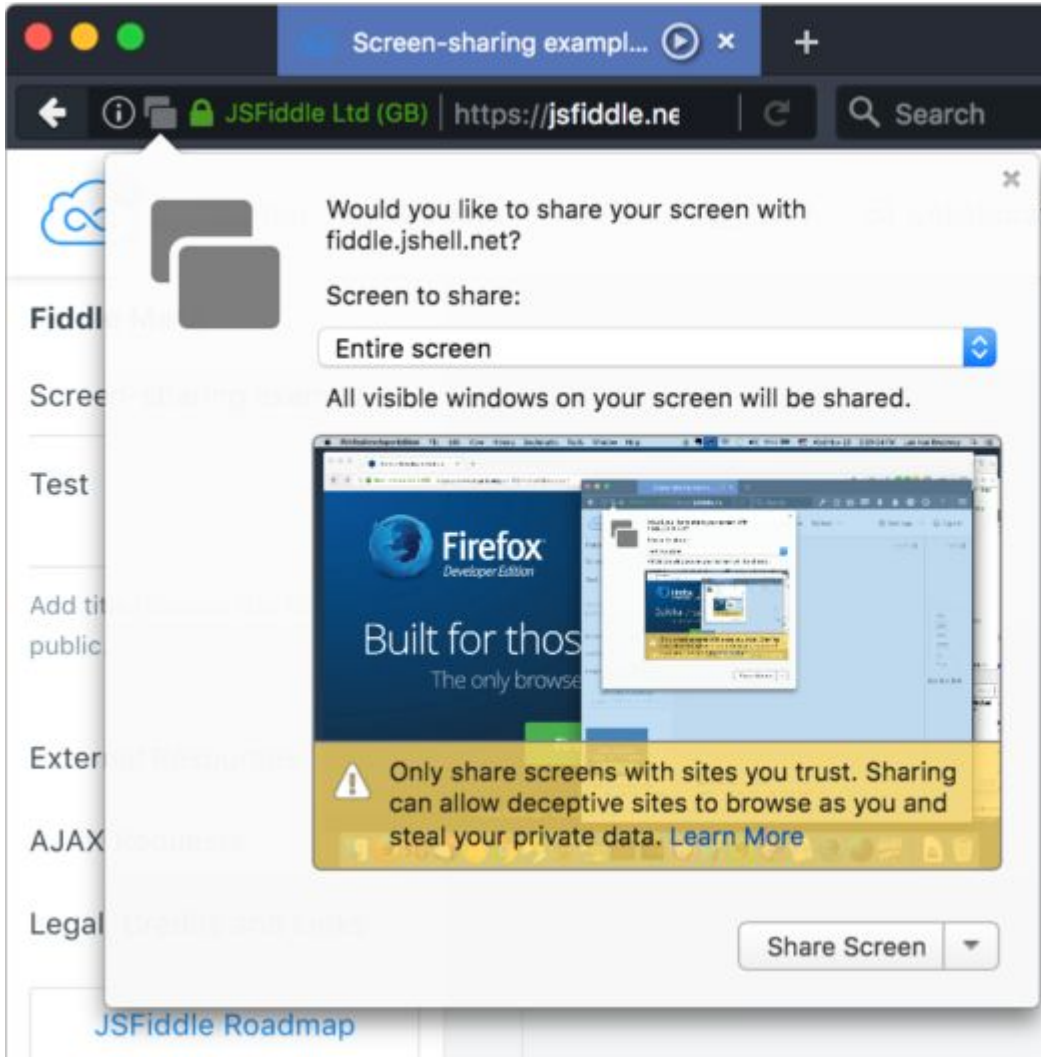
Not just passive threats.

If a web surface under site control is captured, that website has keys to the car, and can iframe-navigate as the logged-in user effectively.

Sidesteps cross-origin protections.

Firefox warns, but hard to explain ☞

Google “share screen trust” for more



Issue 71: Unclear how to aggregate windows or handle multiple windows/monitors (Henrik)

Spec: *Multiple* monitors *can be aggregated into a single logical* ***monitor****; multiple* windows *can be aggregated into a single* ***application*** *surface.*

Does not say how to aggregate. Proposal:

- Relative window sizes MUST be maintained.
- Area between windows MUST NOT leak other application data (including desktop icons); it SHOULD be filled in ~~black~~.
- Up to the User Agent how to position windows relative to each other. (Imagine two small windows on the opposite side of the screen.)

Issue 61/PR 78: Mention capture of (system) audio (Henrik)

getDisplayMedia({audio:true,video:true}) to allow sharing audio+video!

- Mixing audio sources is complicated.
- Availability of audio sources is platform-dependent.
- Audio is complementary; makes sense to share even if audio unavailable.

Proposal / Recap of September Virtual Interim decisions (minutes):

- {audio:true} lets the User Agent choose sources.
- {audio:true} is ignorable; getDisplayMedia() can succeed even if no audio can be produced.
- If audio cannot be produced for the duration of the stream, an audio track must not be created.
- (PR discussion): Audio-only requests must be rejected.

Issue 79: Constraint to exclude application audio (echo) (Henrik)

getDisplayMedia({audio:true,video:true}) in WebRTC conference. Echo!

- Remote participants hearing themselves, other participants twice.
- Audio source mixing is complicated and platform-dependent.
  - We already settled “audio” is optional (PR 78).

Need to avoid “problematic” audio. Need to be easy to implement.

- Constraint to exclude source application’s audio.
  - Any sources that don’t include source app satisfies the constraint.
    - E.g. “tab audio” satisfies constraint.
    - Can always satisfy by not supplying audio.

Issue 81: The user agent should be allowed to change sources after getDisplayMedia resolves (Henrik)

Allow User Agents to implement UI to change capturing sources on-the-fly.

I shouldn't have to stop presenting and start presenting again.

- Use case: I want to change which tab/window I'm capturing.
- Use case: PowerPoint spawns a new fullscreen window, I want to share it instead of the original "presentation notes" window.

Proposal:

- Remove language that says source MUST NOT change.
  Clarify DisplayCaptureSurfaceType and etc. settings may change.

Issue 82: Should getDisplayMedia() be moved to navigator.mediaDevices (Youenn)

- navigator.mediaDevices.getDisplayMedia is better for consistency
- Potential compatibility issue with shipping implementations
  - But getDisplayMedia is still evolving
    - SecureContext, Constraint changes

Break (see you at 10:30 AM)

Webrtc-PC (60 minutes)

WebRTC Issues for Discussion

- Issue 2005/1718: Regarding “a=msid” (Henrik)
- Issue 1930: Rename sender.transport.transport to sender.transport.iceTransport? (Jan-Ivar)
- Issue 1940: transceiver.direction doesn’t respond, if out of sync (Jan-Ivar)
- Issue 1981: RTCIceTransport selected candidate behavior when changing state (Steve Anton)
- Issue 2004: No procedure for the ICE failed state (Steve Anton)
- Issue 1982: Missing normative steps for determining codecs (Jan-Ivar)
- Issue 2006: setCodecPreferences and Direction (Henrik)
- Issue 2009: Clarify how codecs should be prioritized (Henrik)
- Issue 2008: Using codecPayloadType with addTransceiver() (Henrik)
- Issue 1964: Effect of RTCRtpSendParameters on simulcast (Bernard)
- Issue 1827: RTCDataChannel.send() during ‘closing’ state (Bernard)

Issue 2005: Regarding “a=msid” (Henrik)

Issue 1718: “a=msid” line should contain sender/receiver IDs, not track IDs

- Problem: Local and remote track IDs typically don’t match, signaling them is confusing.
- Problem: Multiple identical “a=msid” lines not permitted, “`addTransceiver(track); addTransceiver(track);`” would yield illegal SDP.

[Update] A JSEP PR removed track ID from “a=msid” lines.

- Transceivers are correlated with “mid”, use that instead.
- Stream IDs are still signaled as “a=msid:{streamId}”, this allows you to know which track is which without knowing “mid” (e.g. at “ontrack”).

Issue 1930: Rename sender.transport.transport to sender.transport.iceTransport? (Jan-Ivar)

Two nested attributes of the same name is unintuitive / hard to read:

```
pc.getTransceivers()[0].sender.transport.transport; // whah?
```

Can we rename it?

```
pc.getTransceivers()[0].sender.transport.iceTransport; // ah!
```

Edge already implements the old one, and would be affected.

But with WebRTC for ORTC already shimmed in adapter, is this fixable? Shim:

```
sender.transport.iceTransport || sender.transport.transport;
```

Issue 1940: transceiver.direction is no-op, if out of sync (jib)

TL;DR: `transceiver.direction = newValue; // may sometimes not work`

The reason is complicated, but boils down to this line:

> 6. If newDirection is equal to transceiver's [[Direction]] slot, abort these steps.

Idempotent, except [[Direction]] is what you set it to last, NOT [[CurrentDirection]].

This would be fine if you're the lone control point, but as offerer, the remote end may reduce [[currentDirection]] on you in SRD(answer) (e.g. sendrecv => sendonly / recvonly / inactive).

If this happens, you won't be able to set it back, without first setting it to something else.

Proposed solution: Set [[Direction]] when setting [[CurrentDirection]] as part of SRD(answer).

Issue 1981: RTCIceTransport selected candidate behavior when changing state (Steve Anton)

TLDR: Can we tie getSelectedCandidatePair() to the RTCIceTransport state, giving stronger guarantees?

Currently: Starts null. Updated “when the ICE Agent indicates that the selected candidate pair … has changed”.

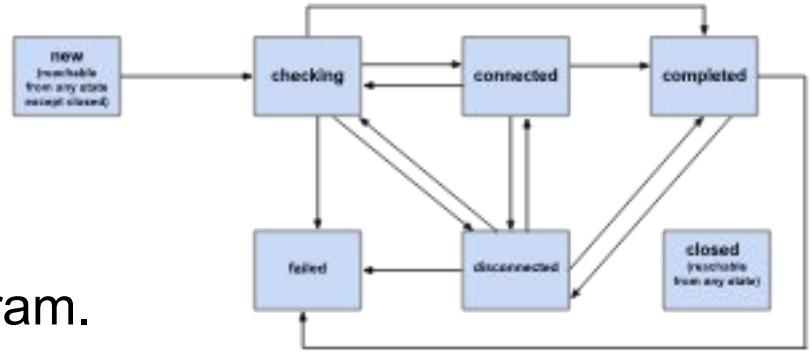
Questions:

- Does state = ‘connected’ imply a selected pair? (“The `RTCIceTransport` has found a usable connection”)
- Should selected pair = null for ‘disconnected’ and ‘failed’? (ORTC says ‘yes’)
- Should selected pair = null when ‘closed’?
- Should a selectedcandidatepairchange event fire if the state change would imply it?

Proposal:

- getSelectedCandidatePair() = null unless state = ‘connected’ or ‘completed’.
- selectedcandidatepairchange event only fires if the selected candidate pair changes to non-empty and the state is already ‘connected’ or ‘completed’.

Issue 2004: No procedure for the ICE failed state (Steve Anton)



Currently we have all of the following:

- `failed:` … This is a terminal state.
- No transition from ‘failed’ on the diagram.
- An example state transition: (`disconnected` or `failed`, ICE restart occurs): `checking`
- ***Performing an ICE restart is recommended when `iceConnectionState` transitions to "`failed`".***

Proposal:

- Change **`failed`** to a non-terminal state.
- Add transition from **`failed`** to **`checking`** (by ICE restart).
- Add non-normative language explaining that media/data channels recover or may disconnect (e.g., due to timeouts).

Issue 1982: Missing normative steps for determining codecs (Jan-Ivar)

sender.getParameters today:*"The codecs sequence is populated based on the codecs that have been negotiated for sending, and which the user agent is currently capable of sending"*

receiver.getParameters today: *"The codecs sequence is populated based on the codecs that the receiver is currently prepared to receive"*

But these are synchronous methods. Proposal:

sender.getParameters: *"*codecs *is set to the value of* ***[[SendCodecs]]****"*

receiver.getParameters *"*codecs *is set to the value of* ***[[ReceiveCodecs]]****"*

...and have SRD(Answer) and SLD(Answer):

"Set ***[[SendCodecs]]*** *to the codecs that have been negotiated for sending, and which the user agent is currently capable of sending" and*

"Set ***[[ReceiveCodecs]]*** *to the codecs that have been negotiated for receiving, and which the user agent is currently prepared to receive" ?*

E.g.:

```
console.log(sender.getParameters().codecs.length); // 0
await pc.setRemoteDescription(msg.offer);
console.log(sender.getParameters().codecs.length); // 0
await pc.setLocalDescription(await pc.createAnswer());
console.log(sender.getParameters().codecs.length); // 3
```

Issue 2006: setCodecPreferences and Direction (Henrik)

- RTCRtpTransciever.setCodecPreferences() takes as input codecs from RTCRtpSender.getCapabilities() and RTCRtpReceiver.getCapabilities().
- Problem:
  - Within an m-line the meaning of a listed codec depends on direction (sendonly/recvonly/sendrecv) as defined in RFC 3264.
  - Sender and receiver capabilities may be different.
  - Example: A user-agent might support AV1 for decoding but not encoding. So in an Offer, AV1 would be included in a recvonly m-line but not in a sendrecv or sendonly m-line.
  - Effect of setCodecPreferences() depends on *direction*.
    - When called, should codecs not supported for *direction* be ignored?
    - What happens if *direction* changes?

Issue 2006: Proposal for setCodecPreferences (cont'd)

- Alternative 1: Leave setCodecPreferences() as a transceiver method, but better define its operation:
  - When direction = "sendrecv" codecs included in createOffer/createAnswer are filtered by the intersection of Receiver and Sender.getCapabilities().codecs[]
  - When direction = "sendonly" codecs included in createOffer/createAnswer are filtered by sender.getCapabilities().codecs[]
  - When direction = "recvonly" codecs included in createOffer/createAnswer are filtered by receiver.getCapabilities().codecs[]

Issue 2006: Proposal for setCodecPreferences (cont'd)

- Alternative 2: Move setCodecPreferences() to sender and receiver.
  - sender.setCodecPreferences() sets preferences between codecs in RTCRtpSender.getCapabilities()
  - receiver.setCodecPreferences() sets preferences between codecs in RTCRtpReceiver.getCapabilities().
  - When direction = "sendrecv" only codecs in the intersection of Receiver/Sender.getCapabilities() are included.
  - What happens if there are conflicts between the sender and receiver codec preferences?
    - Sender's codec preferences win?

Issue 2009: Clarify how codecs should be prioritized (Henrik)

Offerer codec order is only preserved if the answerer does not modify the codec order. It seems that the offerer's priority is only respected if answerer did not use setCodecPreferences().

Alternative Proposals:

- 1) Clarify that this is intended.
- 2) Score codecs based on position, e.g: VP8, VP9, H264 = 3, 2, 1.

  Score = offerer score + answerer score + score of worst position

```
offer: [VP8, VP9, H264], answer: [H264, VP9, VP8]
VP8 = 3 + 1 + 1 = 5 pts  // 1st (3pts) and 3rd (1pts) place
VP9 = 2 + 2 + 2 = 6 pts  // 2nd (2pts) and 2nd (2pts) place, the winning compromise
H264 = 1 + 3 + 1 = 5 pts // 1st (3pts) and 3rd (1pts) place
```

Issue 2008/PR 2010: Using codecPayloadType with addTransceiver() (Henrik)

- codecPayloadType is a member of RTCRtpEncodingParameters, not currently ‘read-only’
  - ORTC: codecPayloadType enables simulcast with different codecs, allows encodings to be validated based on the specified codec.
- Assigned codec payload types are not provided in RTCRtpCodecCapabilities.
- Assigned codec payload types are only known when they become available in sender.getParameters() via codecs[].payloadType.
- Problems:
  - How can the application say “I want to send VP9” if it doesn’t yet know the codecPayloadType for VP9?
  - Without knowing the codec, the validity of sendEncodings is not fully assessable when addTransceiver() is called.
    - The maximum number of simulcast streams might vary by codec.
    - Future desired encoding capabilities (such as supported scalabilityMode values) may vary by codec.

Issue 2008/PR 2010: Using codecPayloadType with addTransceiver() (cont'd)

- Approach 1: make codecPayloadType read-only in RTCRtpEncodingParameters.
  - sendEncodings are considered to apply to any codec.
  - Lack of an exception in addTransceiver does not indicate that the desired sendEncodings can be applied, only that they cannot immediately be determined to be invalid.
    - sendEncodings may not be applied without an error indication.
    - Calling setCodecPreferences can reduce probability of astonishing results.
  - When negotiation has completed, selected codecs and encodings can be determined from sender.getParameters().

Issue 2008/PR 2010: Using codecPayloadType with addTransceiver() (cont'd)

- Approach 2 (PR 2010):
  - Add RTCRtpCodecCapability.preferredPayloadType. Allows valid payload type values to be obtained from RTCRtpSender.getCapabilities().codecs[].
  - codecPayloadType enables extended encoding validity checks
    - No guarantee codecPayloadType is negotiated, though setCodecPreferences may help.
  - Negotiated codecs and applied encodings can be determined when negotiation has completed by calling sender.getParameters().

Lunch (see you at 1:00 PM)

WPT Test Process (Bernard, 30 minutes)

web-platform-tests/webrtc Status

https://github.com/w3c/web-platform-tests/pulls?q=is%3Aopen+is%3Apr+label%3Awebrtc

- Issue Status
  - 8 open issues, 6 open > 6 months
  - Limited progress on issues with major effect on overall “red” status
- PR Status
  - 124 WebRTC-labeled WPT PRs merged since October 23, 2017
    - 78 from “chromium export”
    - 8 from “mozilla:gecko-sync”
  - 8 Open PRs awaiting review, some since November 2015!
    - See: https://github.com/web-platform-tests/wpt/pulls?q=is%3Aopen+is%3Apr+label%3Awebrtc

Can We Get to PR on Current Path?

- WEBRTC WG is chartered for 18 more months.
  - At current velocity, we would merge 186 WPT PRs in that time.
- 87 Existing WPT WEBRTC tests, more needed
  - If Issue density is >2 per test (seems likely), we won't converge in time.
- Solutions:
  - Increase in PR review velocity
    - Requires improvement in review process and/or more reviewers
  - Increase in PR submission rate
    - Can be achieved by increasing test authors

WPT Ownership (Soares from 4/26/18)

- Current owners of webrtc in WPT are volunteers
- Time to manage tests are limited
  - Keep track of spec changes
  - Update tests
  - Review PRs
  - Discussions on what should be the correct behavior
- Lack of time -> unmerged PRs
  - PRs submitted by non-owners are not reviewed by owners
  - PRs submitted by owners are rarely reviewed + no other owners to approve
- Need more owners for WPT
  - People who can commit time to manage tests in the long run
- Resolutions from April 2018 test meeting:
  - Browser vendors to nominate test reviewers/authors
  - Results?

Status of Key WPT Issues

https://github.com/w3c/web-platform-tests/pulls?q=is%3Aopen+is%3Apr+label%3Awebrtc

- Issue 7424: Need mock MediaStream data for some WebRTC tests
  - PR 10764 enabled use of procedurally generated media streams.
  - Is this being used as widely as it might be?
- Issue 9213: Parts of WebRTC require generating RTP to test
  - Still open.
- Issue 10622: replaceTrack tests are incorrect
  - Still open.
- Issue 10981: Firefox doesn't load H.264 codec
  - Still open.
- Issue 836871: WebRTC Tests are leaking heavy resources
  - Fixed.

Status of Key WPT Issues (cont'd)

https://github.com/w3c/web-platform-tests/pulls?q=is%3Aopen+is%3Apr+label%3Awebrtc

- Dependency Issues
  - Issue 9111/PR 9424: RTCIceTransport.html : dependency on SctpTransport
    - PR never reviewed, Issue still open.
  - Issue 9110/PR 9424: RTCDtlsTransport-getRemoteCertificates.html : dependency on SctpTransport
    - PR never reviewed, Issue still open.
  - PR 10566: addTrack: split up tests and reduce dependencies
    - Closed - went another route.

Issue 9213: Parts of WebRTC require generating RTP to test

- Tests requiring RTP generation include:
  - Contributing sources: https://w3c.github.io/webrtc-pc/#dom-rtcrtpcontributingsource-audiolevel (depends on the mixer-to-client header extension defined in RFC 6465)
  - Simulcast tests (only in KITE)
- To test this would require a server (mixer or SFU)
  - Similar in concept to wptserve (HTTP server) or pywebsocket (WebSockets server)
  - Server controls what gets sent to the browser on the network.
  - Prerequisites: STUN/TURN, DTLS, etc.
- What (open source) mixers or SFUs can be used for these tests?

Test Before Commit

- At TPAC 2017, the WEBRTC WG adopted a “test before commit” policy.
- How well has that been working?
  - Of WebRTC-PC PRs, only PR 1886 has had “Needs Test” label applied.
  - PRs submitted by non-owners are not being reviewed.
  - Successful “Test Before Commit” requires a functioning review process.

WG decisions to be made

- Should we continue with “test as you commit”?
- If so, how to encourage progress toward fixing Issues?
  a. Do we need to fix potholes in the road? Focus on WPT test gaps and fundamental issues?
  b. Particularly a problem for Simulcast where WPT tests currently don’t exist.
- Process improvements
  a. How do we recruit additional reviewers?
  b. Should we schedule a bi-weekly WPT Issue & PR review meeting?
- How to test getContributingSources & simulcast? (more later)

Webrtc WPT

TPAC 2018 Lyon

Soares and Dr Alex, CoSMo Soft.

Data Channel Tests (`lgrahl`)

- WPT PR #13499 intends to add any missing data channel tests
- Updated all ~60 existing test cases
- Added ~200 new test cases (including a workaround to run them in Webkit - please fix bug 184688)
- Thesis will be released soon that contains a full evaluation of the results for Chromium, Firefox and Safari (sorry, Edge, better luck next time)

Data Channel Tests (`lgrahl`)

Review process seems too lengthy for external contributions (PR has been created in April… but also originally introduced controversial changes to `testharness.js`).

Combination of…

- disabled global timeout,
- `promise_test`, and
- local timeouts for each test

has proven to be a good workaround to prevent resource exhaustion and false positives due to the global timeout firing.

WPT Issues July - Oct 2018

66 PRs, 2 Issues

Google	35
Mozilla	9
lukebjerring	7
fippo	5
youennf	2
WPT	5
Others	4

By Contributors

webrtc-pc	33
mediacapture-main	9
mediacapture-screen-share	5
mst-content-hint	2
webrtc-ice	5
webrtc-quic	4
wpt	5
mediacapture-output	2
mediacapture-fromelement	2

By Specs

WebRTC WPT results - Oct 2018

Path	Chrome 70 Linux 18.04 @d44bc3ed38 Oct 20 2018	Edge 17 Windows 10 @d44bc3ed38 Oct 21 2018	Firefox 62 Linux 18.04 @d44bc3ed38 Oct 20 2018	Safari 11.1 macOS 10.13 @d44bc3ed38 Oct 21 2018
mediacapture-depth/	6 / 6	0 / 1	6 / 6	6 / 6
mediacapture-fromelement/	42 / 45	0 / 5	22 / 45	32 / 45
mediacapture-image/	129 / 177	0 / 20	64 / 177	82 / 173
mediacapture-record/	49 / 72	0 / 2	56 / 72	2 / 72
mediacapture-streams/	207 / 249	0 / 30	186 / 249	5 / 34
screen-capture/	8 / 21	0 / 2	8 / 21	7 / 11
webrtc/	579 / 1318	0 / 88	700 / 1318	226 / 555
webrtc-stats/	4 / 5	0 / 1	4 / 5	4 / 5

Path	Tests Passing in *X* / 4 Browsers				
	0 / 4	1 / 4	2 / 4	3 / 4	4 / 4
mediacapture-depth/	0 / 6	0 / 6	0 / 6	6 / 6	0 / 6
mediacapture-fromelement/	1 / 45	11 / 45	20 / 45	13 / 45	0 / 45
mediacapture-image/	48 / 177	46 / 177	34 / 177	49 / 177	0 / 177
mediacapture-record/	31 / 95	23 / 95	40 / 95	1 / 95	0 / 95
mediacapture-streams/	29 / 293	137 / 293	126 / 293	1 / 293	0 / 293
screen-capture/	13 / 21	0 / 21	1 / 21	7 / 21	0 / 21
webrtc/	509 / 1318	302 / 1318	387 / 1318	120 / 1318	0 / 1318
webrtc-stats/	1 / 5	0 / 5	0 / 5	4 / 5	0 / 5

Coverage Status - TPAC 2017

- PR #8051: Add coverage report and tools for WebRTC tests
- Coverage = (total - todo) / total

```
$ cd webrtc/tools
$ node scripts/overview.js
Overall Coverage
====================
todo          |    248
tested        |    315
trivial       |    173
untestable    |     79
====================
total         |    815
coverage      | 69.57%
====================
```

4. Peer-to-peer connections	67.83%
5. RTP Media API	67.01%
6. Peer-to-peer Data API	71.87%
7. Peer-to-peer DTMF	93.54%
8. Statistics Model	100.00%
9. Identity	86.04%
10. Media Stream API Extensions for Network Use	35.71%

Number of WPT tests and coverage

webrtc/	1318
webrtc-stats/	5
mediacapture-streams/	249
mediacapture-fromelement/	45
screen-capture/	21

Total Tests

	2016	**2017**	**2018**
webrtc/	293	1296	1318
coverage	<10% (manual)	69.57%	N/A

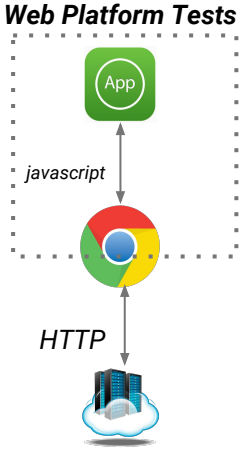
Yearly Progress

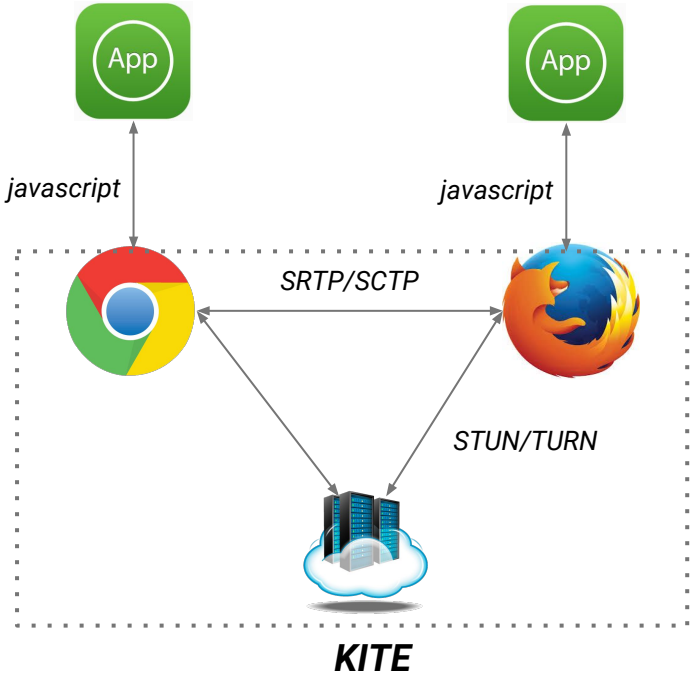
Ongoing: Separation of tests by specs

- Tests for new extension specs, e.g. webrtc-quic & webrtc-ice, are placed in the same webrtc/ directory
- Discussions for separating the tests into their own subdirectory
- Issue management - spec labels in addition to the “wg-webrtc” label?
- Good time to apply to all other specs?

WPT: Some things remain difficult to test automatically

- Permission prompt
  - Origin display
  - Remember decision
- getDisplayMedia
  - Device / network discon.
  - System audio
- Simulcast
  - Not P2P, must test against SFU
- Interoperability of 2+ browsers over the wire
  - The elephant in the room
  - More on that with KITE





Webrtc webdriver status Update

TPAC 2018 Lyon

Soares and Dr Alex, CoSMo Soft.

Specific WebRTC Testing Issues

(Stockholm meeting 2018)

1. Permission prompt

- Those prompts are not part of the DOM (UA), cannot be access by JS in purpose.
- Those prompts are not modal in nature, no existing webdriver API to manipulate them
- All browsers have a by-pass mechanism (except edge), which all differ
  - Registry entry (edge)
  - Persistent choice (edge, manual once)
  - Profile (mz)
  - Command line argument (cr)
  - Dev Menu / command line (safari)

2. Media

GetUserMedia Permission prompt

- Permissions Automation mainly by Bocoup.
  - Introduce "Automation" section #151
    - Merged 22 Dec 2017

- WebDriver implementation status
  - All browsers have it (audience?)
  - Microsoft just added it to october insider release

Specific WebRTC Testing Issues (Stockholm meeting 2018)

2. **Media creation: How to**

- To test specific video/audio capture HW on specific devices [do not forget!]
- To test peer connection or other apis with programmatically generated media
  - Front end / back end, audio/video sync, degradation, resolutions, ….
- To test peer connection or other apis on VMs or devices without capture HW
  - CL arg (cr)
  - Mock capturer automatically made default capturer under automation (safari)
  - Read from file (cr)
  - Other JS API to generate media (webAudio, from Canvas, ...)
  - Virtual device (registering as OS device driver) (bocoup proposal)

Need mock devices for getUserMedia() tests #12046

- Alternative - allow usage of internal APIs in WPT?
  - Safari - internal API available
  - Chrome - command line flags
  - Firefox - fake media device constraint?

- Screensharing testing

WPT: automation

- Originally a manual test suite, huge effort for automation
- Chose to make WPT webdriver-aware instead of having an external test-runner instrumenting browsers to make them run the WPT tests (KITE).
  - Rewrite the tests
  - Add dependency to webdriver binary, version, …. To the tests
- Adding missing items to webdriver protocol
- Adding a selenium grid (task-cluster, ….) to run on all configs

- STATUS: Desktop browsers, more or less all nowadays, webdriver permitting.
  - Recent addition of Safari Tech Preview

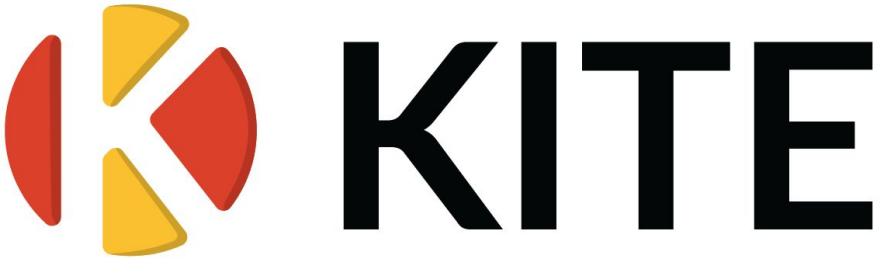
WPT automation & KITE WPT test

Path	Chrome 70 Linux 18.04 @d44bc3ed38 Oct 20 2018	Edge 17 Windows 10 @d44bc3ed38 Oct 21 2018	Firefox 62 Linux 18.04 @d44bc3ed38 Oct 20 2018	Safari 11.1 macOS 10.13 @d44bc3ed38 Oct 21 2018
mediacapture-depth/	6 / 6	0 / 1	6 / 6	6 / 6
mediacapture-fromelement/	42 / 45	0 / 5	22 / 45	32 / 45
mediacapture-image/	129 / 177	0 / 20	64 / 177	82 / 173
mediacapture-record/	49 / 72	0 / 2	56 / 72	2 / 72
mediacapture-streams/	207 / 249	0 / 30	186 / 249	5 / 34
screen-capture/	8 / 21	0 / 2	8 / 21	7 / 11
webrtc/	579 / 1318	0 / 88	700 / 1318	226 / 555
webrtc-stats/	4 / 5	0 / 1	4 / 5	4 / 5

Webrtc Interop (2+ browsers) status Update

TPAC 2018 Lyon

Soares and Dr Alex, CoSMo Soft.



KITE Interop SE Grid - Browser configs

(without saucelab, without UWP, Without Electron [comm])

KITE 2-clients, beyond browsers: appRTC(mobile)

Allow for more generic p2p interop tests:

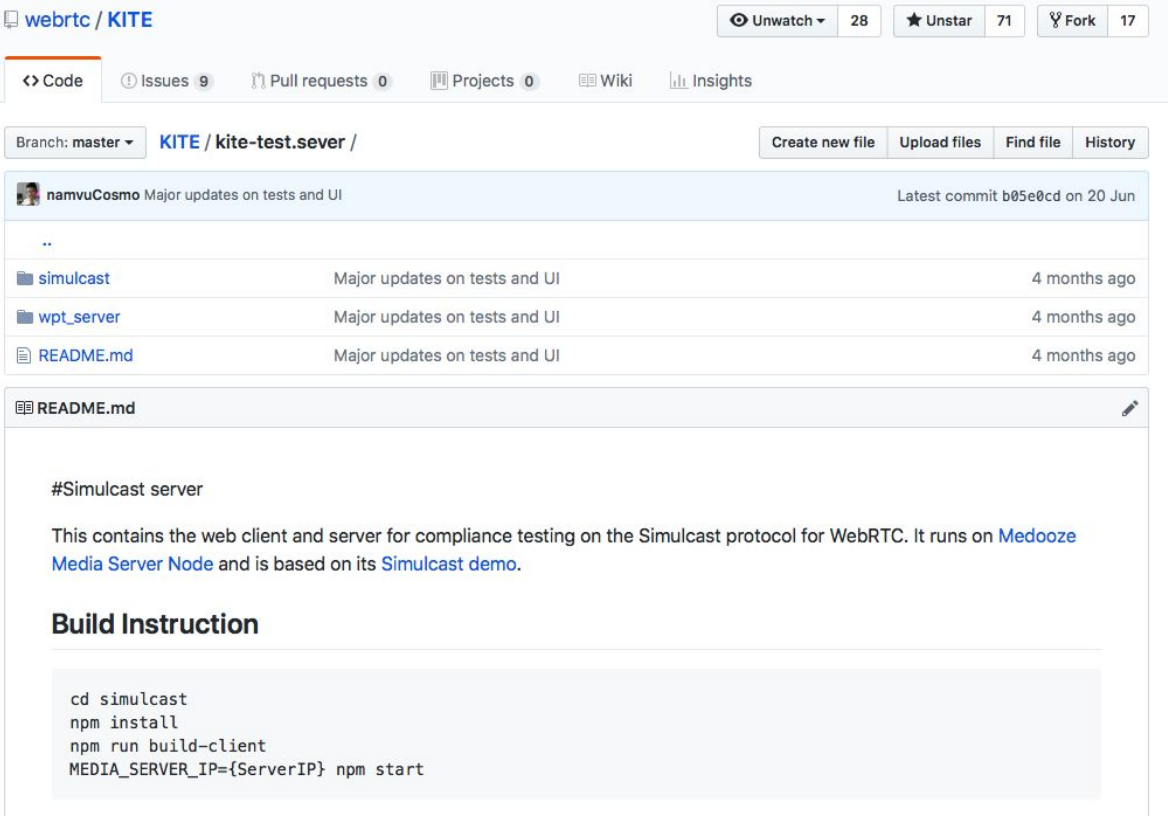
- Ios
- Android
- Since M71: mac desktop

- UWP (in progress, to be shared by MS)
- Edge support (in progress, to be shared by CoSMo)

KITE: Simulcast (stockholm 2018)

Simulcast

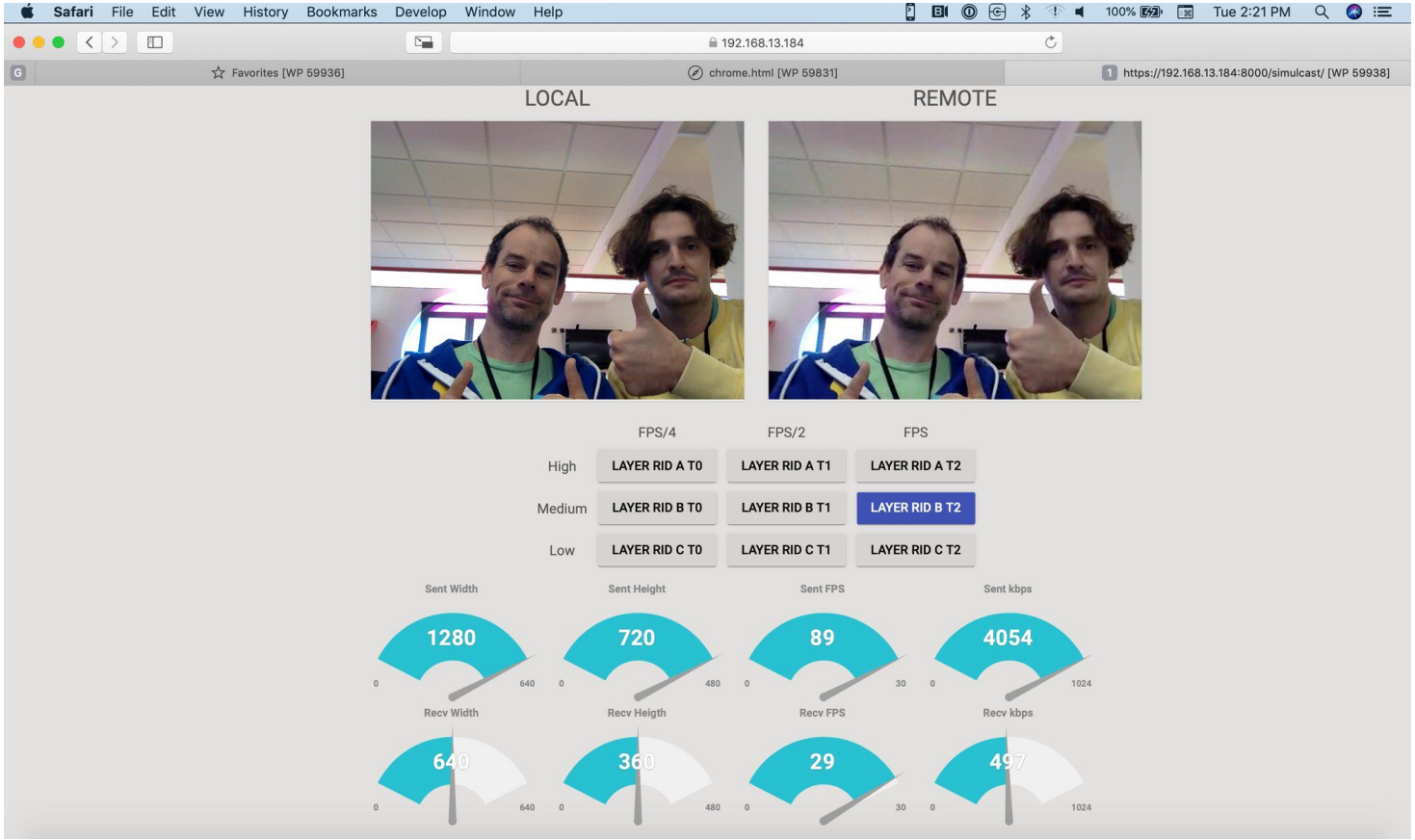
- the dedicated app runs over https
  - -- is available hosted
  - -- is run locally for KITE test
  - -- open source test:
- The test verifies the following:
  - echoed stream is displayed.
  - stream received from SFU
    - it received it,
    - format was correct,
    - it could extract the right layer
  - access to SDP offer/answer.
  - SDP offer/answer format.



KITE Simulcast Update 06/2018: Some recent results

- The majority of the failed cases are Edge's. There are 2 reasons for this:
  - webdriver mismatch for Edge insider:
  - Edge's webRTC implementation is slightly different from the others browsers. (here)
- Firefox crashed when tested against safari. (here)
- Chrome bug related to multi-stream and unified plan (here)
- Electron webdriver hanging, fixed.
- Edge does not enumerate virtual capture devices (manycams, vlc2vcam with Magic Camera, ….)

KITE Simulcast Update 10/2018: Apple (H.264)



KITE Update 06/2018: Network Instrumentation

Goal: Evaluate the behavior of the following algorithm types:

- Bitrate adaptation and degradation preferences (Q, spatial, temporal)
- Simulcast / SVC layer control
- Bandwidth estimation
- Congestion control
- Error correction
- Jitter correction

Mean: Instrumentation and programmatic control the following
(1) independently for each client (2) or server (3) on each OS a browser is available:

- Network Bandwidth and corresponding variations across time
- Network Quality and corresponding variations across time
  - Jitter
  - Packet loss
- NAT settings,
- Firewall settings,

Original collaboration proposal by CallStats.io, which had a solution without (3).

KITE Update 10/2018: Network Instrumentation: Verify

WebRTC Testing: INTEL contributions

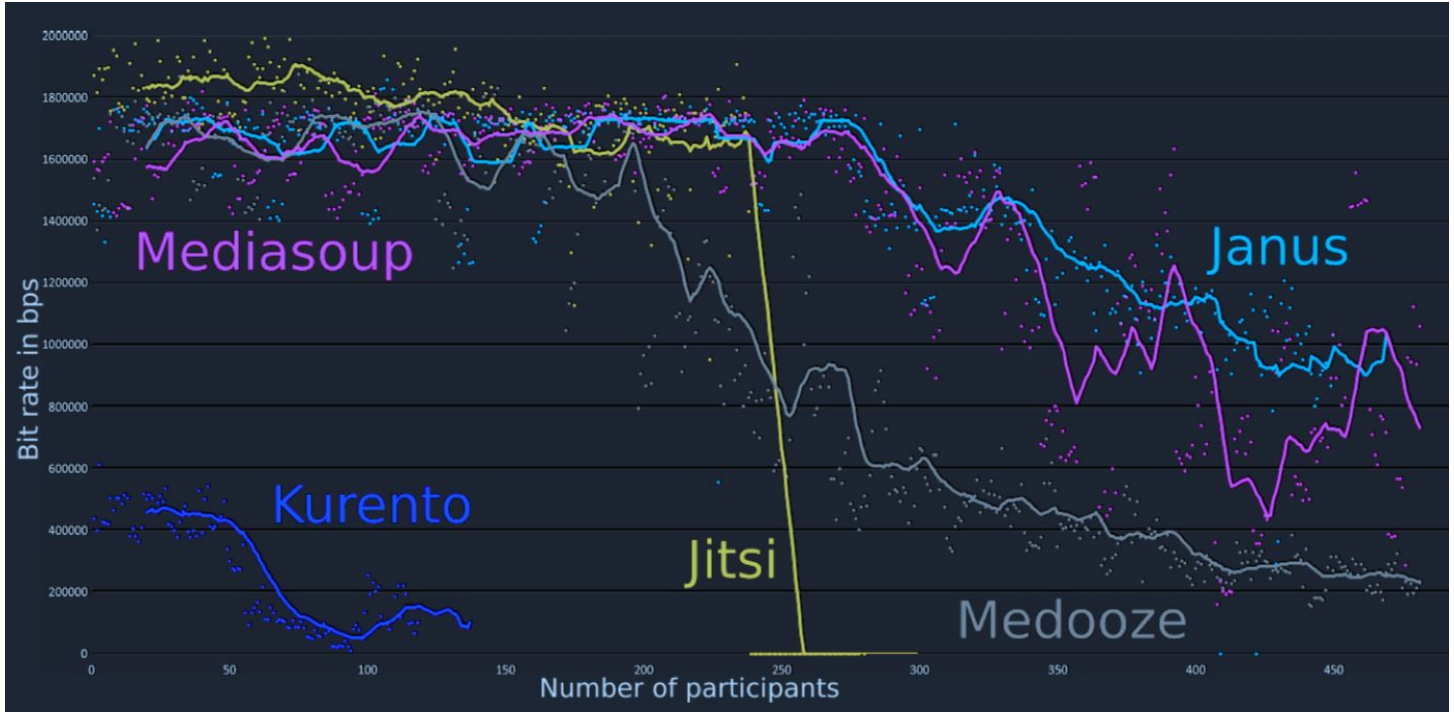
(see Jianju ZHU)

- **IATF - Interactive API Testing Framework**
  - Firstly introduced in GTAC 2016
  - Open source ETA Q1, 2019
  - Video: here
  - Slides: Here

- **QoS testing – WebRTCBench**
  - Co-developed with UCI in 2015. Original version open-sourced then, but not updated.
  - Current internal version to Open source ETA Q1, 2019
  - Paper: https://ieeexplore.ieee.org/document/7351769/

WebRTC Testing: Interesting scientific publications (1/2)

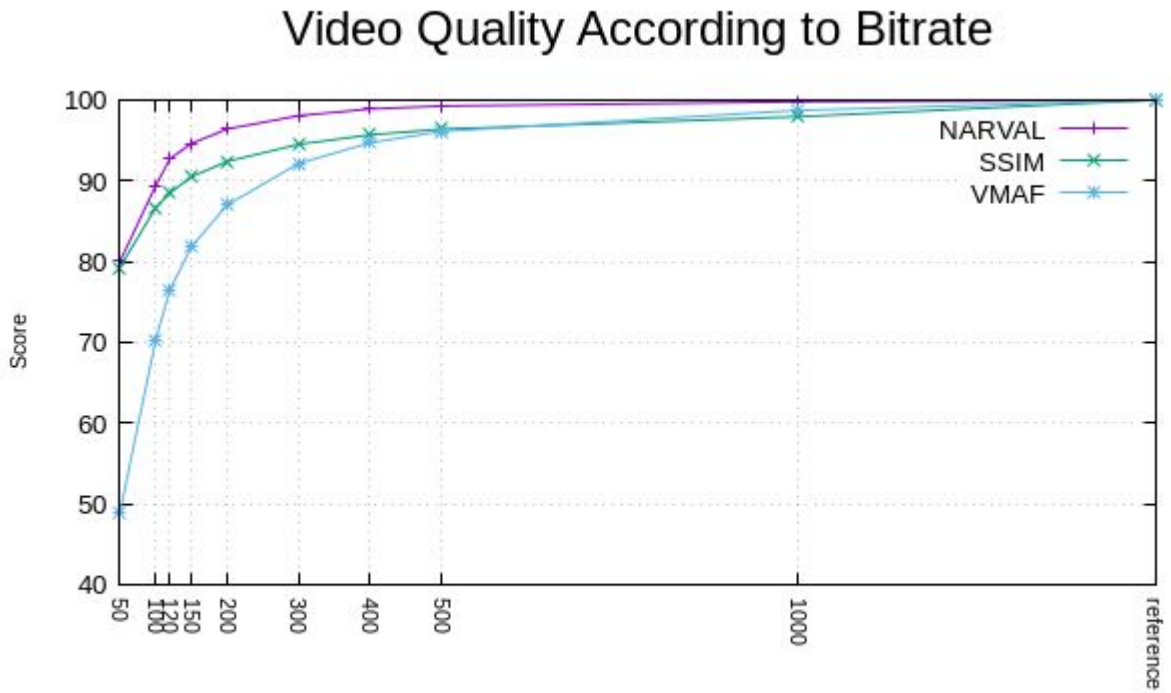
Comparative Study of WebRTC Open Source SFUs for Video Conferencing,
Emmanuel André, Nicolas Le Breton, Augustin Lemesle, Ludovic Roux and Alex. Gouaillard
in Proceedings of IIT Real-Time Communications, Illinois Institute of Technology, Chicago, USA, October 2018



WebRTC Testing: Interesting scientific publications (2/2)

NARVAL, A No-Reference Video Quality Tool for Real-Time Communications, Augustin Lemesle, Alexis Marion, Ludovic Roux and Alexandre Gouaillard

in Proceedings of Human Vision and Electronic Imaging, Burlingame, California, USA, January 2019



Webrtc: next steps to PR for WebRTC-PC
(Bernard, 30 minutes)

W3C Requirements for PR

- Process: https://www.w3.org/2018/Process-20180201/#rec-pr
- Criteria:
  - *must* show adequate implementation experience except where an exception is approved by the Director,
  - *must* show that the document has received wide review,
  - *must* show that all issues raised during the Candidate Recommendation review period other than by Advisory Committee representatives acting in their formal AC representative role have been formally addressed,
  - *must* identify any substantive issues raised since the close of the Candidate Recommendation review period by parties other than Advisory Committee representatives acting in their formal AC representative role,
  - *may* have removed features identified in the Candidate Recommendation document as "at risk" without republishing the specification as a Candidate Recommendation.
- How can we remove the obstacles to reaching PR?

WebRTC Issues

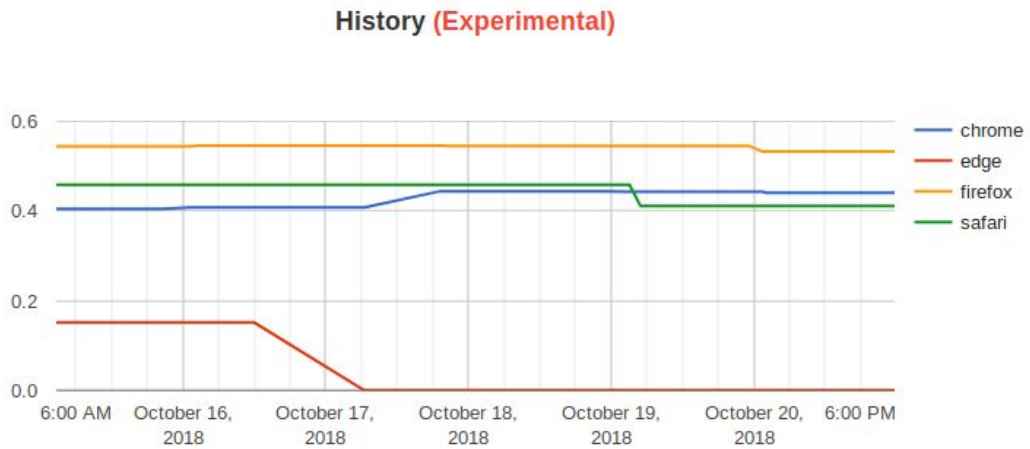
- 46 Open Issues. Labels:
  - TPAC: 13
  - Editorial: 8
  - PR exists: 4
  - Needs submitter/assignee action: 4
  - Question: 4
  - Simulcast: 4 (2 non-TPAC)
  - Enhancement: 3
  - Icebox: 1
  - Miscellaneous: 7
- New issue velocity: 7/month
- Current fix velocity: 10/month
- Seems possible to reach zero Issue Bounce in 18 months with current resources, sooner with more editors.

Simulcast: The Final Frontier

- Number of Issues labeled “simulcast” growing
  - Effect of encoding parameters under-specified
  - Fixes may require substantial discussion
    - Implementations differ significantly so changes may be needed.
  - Interactions with other under-implemented functionality (e.g. setCodecPreferences, getCapabilities) have been encountered.
  - “Issue Mountain” looks bigger the closer we get, hard to estimate “glide path” on the other side before we reach the summit
- Potential solutions
  - Virtual interim(s) devoted to simulcast?
  - Simulcast hackathon?

WPT/WebRTC Status

- WPT status: https://wpt.fyi/webrtc
- Greener... but still mostly red/pink.
- Currently, no tests for simulcast
  - Can “simulcast playground” approach help?
- Still false negatives due to dependencies.
- History (what does this mean?)



WPT Status: Pink is the New Yellow

Path	Chrome 70 Linux 18.04 @d44bc3ed38 Oct 20 2018	Edge 17 Windows 10 @d44bc3ed38 Oct 20 2018	Firefox 62 Linux 18.04 @d44bc3ed38 Oct 20 2018	Safari 11.1 macOS 10.13 @d44bc3ed38 Oct 20 2018
RTCCertificate.html	5 / 6	0 / 1	1 / 6	1 / 6
RTCConfiguration-bundlePolicy.html	16 / 16	0 / 1	8 / 16	14 / 16
RTCConfiguration-iceCandidatePoolSize.html	10 / 10	0 / 1	1 / 10	10 / 10
RTCConfiguration-iceServers.html	33 / 76	0 / 1	25 / 76	20 / 76
RTCConfiguration-iceTransportPolicy.html	14 / 17	0 / 1	11 / 17	17 / 17
RTCConfiguration-rtcpMuxPolicy.html	14 / 14	0 / 1	1 / 14	1 / 14
RTCDTMFSender-insertDTMF.https.html	5 / 8	0 / 1	7 / 8	0 / 1
RTCDTMFSender-ontonechange-long.https.html	1 / 2	0 / 1	1 / 2	0 / 1
RTCDTMFSender-ontonechange.https.html	3 / 14	0 / 1	11 / 14	0 / 1
RTCDataChannel-bufferedAmount.html	1 / 5	0 / 1	1 / 5	0 / 5
RTCDataChannel-id.html	3 / 3	0 / 1	1 / 3	1 / 3
RTCDataChannel-send.html	7 / 11	0 / 1	11 / 11	1 / 11
RTCDataChannelEvent-constructor.html	5 / 5	0 / 1	5 / 5	5 / 5
RTCDtlsTransport-getRemoteCertificates.html	1 / 2	0 / 1	1 / 2	1 / 2
RTCIceCandidate-constructor.html	2 / 18	0 / 1	4 / 18	8 / 18
RTCIceTransport-extension.https.html	1 / 28	0 / 1	1 / 28	0 / 1
RTCIceTransport.html	1 / 3	0 / 1	1 / 3	0 / 3
RTCPeerConnection-add-track-no-deadlock.https.html	2 / 2	0 / 1	2 / 2	0 / 1
RTCPeerConnection-addIceCandidate.html	14 / 24	0 / 1	14 / 24	1 / 24
RTCPeerConnection-addTrack.https.html	4 / 10	0 / 1	5 / 10	0 / 1
RTCPeerConnection-addTransceiver.https.html	2 / 13	0 / 1	11 / 13	0 / 1
RTCPeerConnection-canTrickleIceCandidates.html	1 / 4	0 / 1	4 / 4	1 / 4
RTCPeerConnection-connectionState.html	1 / 3	0 / 1	1 / 3	1 / 3
RTCPeerConnection-constructor.html	21 / 24	0 / 1	22 / 24	19 / 24
RTCPeerConnection-createAnswer.html	3 / 4	0 / 1	4 / 4	4 / 4
RTCPeerConnection-createDataChannel.html	19 / 31	0 / 1	18 / 31	21 / 31
RTCPeerConnection-createOffer-offerToReceive.html	3 / 16	0 / 1	16 / 16	4 / 16
RTCPeerConnection-createOffer.html	3 / 8	0 / 1	7 / 8	4 / 8
RTCPeerConnection-generateCertificate.html	7 / 9	0 / 1	7 / 9	1 / 9
RTCPeerConnection-getDefaultIceServers.html	1 / 2	0 / 1	1 / 2	1 / 2
RTCPeerConnection-getIdentityAssertion.sub.html	1 / 13	0 / 1	1 / 13	1 / 13
RTCPeerConnection-getStats.https.html	6 / 14	0 / 1	5 / 14	0 / 1
RTCPeerConnection-getTransceivers.html	2 / 2	0 / 1	2 / 2	2 / 2
RTCPeerConnection-iceConnectionState.html	2 / 3	0 / 1	2 / 3	2 / 3
RTCPeerConnection-iceGatheringState.html	3 / 4	0 / 1	3 / 4	1 / 4
RTCPeerConnection-ondatachannel.html	3 / 4	0 / 1	3 / 4	1 / 4
RTCPeerConnection-onnegotiationneeded.html	4 / 8	0 / 1	8 / 8	5 / 8
RTCPeerConnection-ontrack.https.html	2 / 6	0 / 1	6 / 6	0 / 1
RTCPeerConnection-peerIdentity.html	1 / 7	0 / 1	1 / 7	1 / 7
RTCPeerConnection-remote-track-mute.https.html	0 / 5	0 / 1	1 / 5	0 / 1
RTCPeerConnection-removeTrack.https.html	5 / 13	0 / 1	11 / 13	0 / 1
RTCPeerConnection-setDescription-transceiver.html	1 / 6	0 / 1	6 / 6	1 / 6
RTCPeerConnection-setLocalDescription-answer.html	4 / 6	0 / 1	2 / 6	4 / 6
RTCPeerConnection-setLocalDescription-offer.html	4 / 6	0 / 1	2 / 6	2 / 6
RTCPeerConnection-setLocalDescription-pranswer.html	3 / 5	0 / 1	1 / 5	4 / 5
RTCPeerConnection-setLocalDescription-rollback.html	1 / 5	0 / 1	3 / 5	3 / 5
RTCPeerConnection-setLocalDescription.html	4 / 4	0 / 1	2 / 4	4 / 4
RTCPeerConnection-setRemoteDescription-answer.html	4 / 4	0 / 1	2 / 4	4 / 4
RTCPeerConnection-setRemoteDescription-offer.html	5 / 6	0 / 1	2 / 6	4 / 6
RTCPeerConnection-setRemoteDescription-pranswer.html	5 / 5	0 / 1	1 / 5	5 / 5
RTCPeerConnection-setRemoteDescription-replaceTrack.https.html	6 / 7	0 / 1	4 / 7	0 / 1
RTCPeerConnection-setRemoteDescription-rollback.html	1 / 4	0 / 1	3 / 4	2 / 4
RTCPeerConnection-setRemoteDescription-tracks.https.html	11 / 15	0 / 1	4 / 15	0 / 1
RTCPeerConnection-setRemoteDescription.html	4 / 6	0 / 1	5 / 6	5 / 6
RTCPeerConnection-track-stats.https.html	18 / 19	0 / 1	3 / 19	0 / 1
RTCPeerConnection-transceivers.https.html	4 / 43	0 / 1	31 / 43	0 / 1
RTCPeerConnectionIceEvent-constructor.html	6 / 9	0 / 1	7 / 9	1 / 9
RTCQuicStream.https.html	1 / 14	0 / 1	1 / 14	0 / 1
RTCQuicTransport.https.html	1 / 17	0 / 1	1 / 17	0 / 1

WPT Status (cont'd)

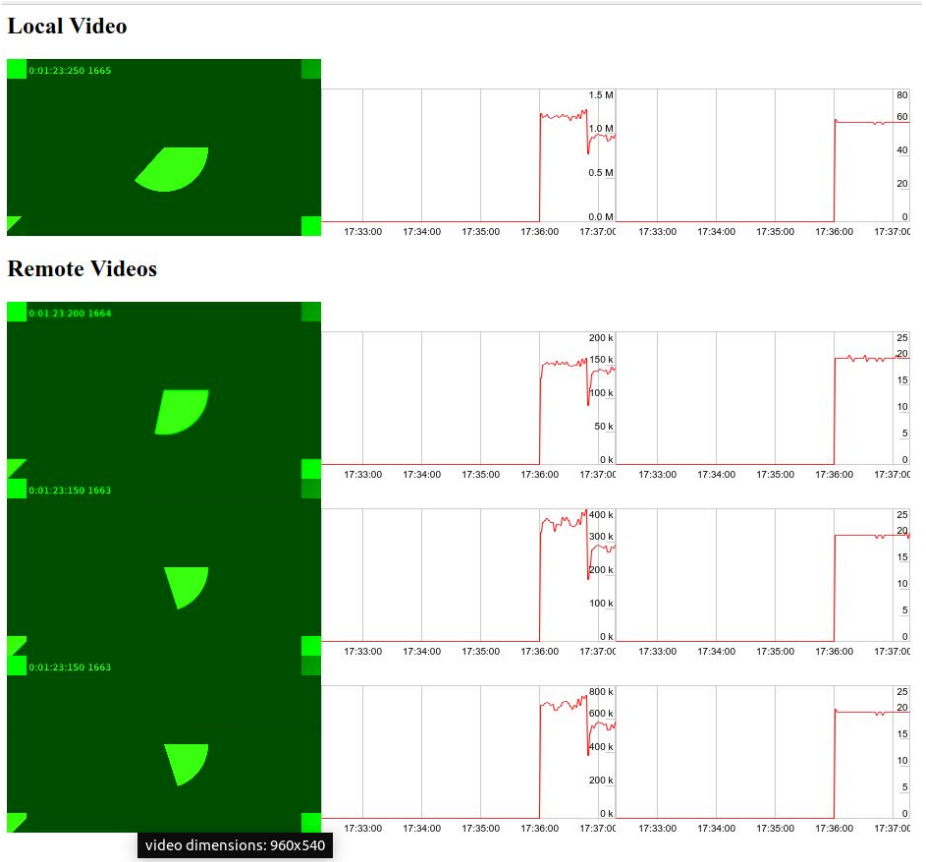
RTCRtpParameters-codecs.html	1 / 7	0 / 1	1 / 7	1 / 7
RTCRtpParameters-degradationPreference.html	1 / 3	0 / 1	1 / 3	1 / 3
RTCRtpParameters-encodings.html	1 / 23	0 / 1	1 / 23	1 / 23
RTCRtpParameters-headerExtensions.html	1 / 2	0 / 1	1 / 2	1 / 2
RTCRtpParameters-rtcp.html	1 / 3	0 / 1	1 / 3	1 / 3
RTCRtpParameters-transactionId.html	1 / 6	0 / 1	1 / 6	1 / 6
RTCRtpReceiver-getCapabilities.html	4 / 4	0 / 1	1 / 4	1 / 4
RTCRtpReceiver-getContributingSources.https.html	1 / 2	0 / 1	1 / 2	0 / 1
RTCRtpReceiver-getParameters.html	1 / 2	0 / 1	1 / 2	1 / 2
RTCRtpReceiver-getStats.https.html	1 / 3	0 / 1	1 / 3	0 / 1
RTCRtpReceiver-getSynchronizationSources.https.html	1 / 2	0 / 1	1 / 2	0 / 1
RTCRtpSender-getCapabilities.html	4 / 4	0 / 1	1 / 4	1 / 4
RTCRtpSender-getStats.https.html	1 / 3	0 / 1	1 / 3	0 / 1
RTCRtpSender-replaceTrack.https.html	1 / 10	0 / 1	10 / 10	0 / 1
RTCRtpSender-setParameters.html	1 / 2	0 / 1	2 / 2	1 / 2
RTCRtpTransceiver-setCodecPreferences.html	1 / 10	0 / 1	1 / 10	1 / 10
RTCRtpTransceiver-setDirection.html	1 / 4	0 / 1	1 / 4	2 / 4
RTCRtpTransceiver.https.html	0 / 38	0 / 1	24 / 38	0 / 1
RTCSctpTransport-constructor.html	1 / 3	0 / 1	1 / 3	1 / 3
RTCSctpTransport-maxMessageSize.html	1 / 6	0 / 1	1 / 6	1 / 6
RTCTrackEvent-constructor.html	1 / 8	0 / 1	8 / 8	7 / 8
datachannel-emptystring.html	1 / 2	0 / 1	1 / 2	0 / 2
getstats.html	2 / 2	0 / 1	1 / 2	0 / 2
historical.html	7 / 16	0 / 1	7 / 16	16 / 16
idlharness.https.window.html	218 / 495	0 / 1	297 / 495	0 / 1
no-media-call.html	2 / 2	0 / 1	2 / 2	0 / 2
promises-call.html	2 / 2	0 / 1	2 / 2	0 / 2
protocol/	4 / 4	0 / 1	1 / 4	0 / 1
simplecall.https.html	2 / 2	0 / 1	2 / 2	0 / 1

Simulcast Playground

- Single browser tests for simulcast, written by Fippo.
  - Enables testing of simulcast operation without a conferencing server.
  - Can determine if maxBitrate, maxFramerate, active is having the desired effect.
    - Assuming the specification defines the “desired effect”!
  - WebRTC Hacks article: https://webrtchacks.com/a-playground-for-simulcast-without-an-sfu/
- Repo: https://github.com/fippo/simulcast-playground
  - Separate page for each browser, because simulcast isn’t interoperable enough (yet)
- Could be extended to allow tests between two browsers without a conferencing server.
  - Interoperability progress needed to make this feasible.

Testing simulcast in-browser

- Local + remote videos
- Bandwidth (sent, received)
- Framerate
  - Chrome adds up framerate…
- Synchronization?
- (bandwidth drop due to CPU constrained)



Testing simulcast in-browser

- No external dependencies
- Test that simulcast works
  - basic integration test
- Test that the browser is sending requested number of spatial layers
- Enabling/disabling layers and observing effect on streams/stats
  - setParameters()
- Exposes statistics at receiver
  - stats currently not exposed at sender level, see webrtc-stats#348, webrtc-stats#318
  - allows finding out target bitrate per-layer

How?

- SDP munging
  - during ‘signaling’
  - different in Chrome/Safari and Firefox
- Remote description creates three different tracks for three SSRCs
  - ontrack fires three times
  - Simulcast is three different streams without dependency
- Relies on SSRC demuxing
  - not going to work with MID/RID? Don’t signal it!
  - not going to work for VP9-SVC (why not?)
- source @ https://github.com/fippo/simulcast-playground
  - or using SDES + co in the Chrome tree

Why?

- Cheap
- Coverage
- Find bugs now…
- Removes potential issues introduced by SFU implementations
    - Recent SFU benchmarking study by Cosmo Consulting discovered some “interesting” behavior

Why not?

- Simulcast done differently and non-spec
- Sender stats may expose all the necessary information
- Low complexity input may not reach high bitrates, making tests flaky
- Non-spec code in WPT

Confluence Status

- Web-platform-tests dashboard “does not contain useful metrics for evaluation or comparison of web platform features”
- Web confluence project:
  - Looks at properties and methods exposed by browsers:
  - https://web-confluence.appspot.com/#!/
  - Caveat: no guarantee that a widely-supported API is interoperable in its details, or will remain part of the web platform.
  - Tool that extracts data from the confluence tracker: https://dontcallmedom.github.io/webrtc-impl-tracker/?webrtc

WebRTC-PC: Functionality “Red” in Confluence

- Falling below the “2 implementations” bar (pink/red in confluence)
  - Simulcast: 3 implementations, but not interoperable (at API or protocol level)
  - Methods: pc.getDefaultIceServers, pc.onicecandidateerror, pc.sctp, pc.onstatsended, pc.idpErrorInfo, RTCCertificate.getSupportedAlgorithms, RTCRtpTransceiver.setCodecPreferences
    - Assumption: Some of these will eventually be implemented.
  - Attributes: RTCIceCandidate properties
    - How much implementer interest is there?
  - Events: RTCPeerConnectionIceErrorEvent, RTCPeerConnectionIceEvent.url
    - How much implementer interest is there?
  - Objects with only 1 implementation: IceTransport, DtlsTransport, Identity (removed)
  - Objects with no implementations: SctpTransport
  - Outlook for under-implemented objects?
- Issues:
  - Data on RTCIdentity* is incorrect, probably because these interfaces are only exposed in the non-default global
  - New methods in Chrome “Unified Plan” implementation not covered.

Confluence Status (cont'd)

Interface	Member		Chrome	Edge	Firefox	Safari
RTCPeerConnection	createOffer	4/4	40.0.2214.93+	15.15063+	45.0+	11.0.3+
	createAnswer	4/4	40.0.2214.93+	15.15063+	45.0+	11.0.3+
	setLocalDescription	4/4	40.0.2214.93+	15.15063+	45.0+	11.0.3+
	localDescription	4/4	43.0.2357.65+	15.15063+	45.0+	11.0.3+
	currentLocalDescription	2/4			57.0+	11.0.3+
	pendingLocalDescription	2/4			57.0+	11.0.3+
	setRemoteDescription	4/4	40.0.2214.93+	15.15063+	45.0+	11.0.3+
	remoteDescription	4/4	43.0.2357.65+	15.15063+	45.0+	11.0.3+
	currentRemoteDescription	2/4			57.0+	11.0.3+
	pendingRemoteDescription	2/4			57.0+	11.0.3+
	addIceCandidate	4/4	40.0.2214.93+	15.15063+	45.0+	11.0.3+
	signalingState	4/4	43.0.2357.65+	15.15063+	45.0+	11.0.3+
	iceGatheringState	4/4	43.0.2357.65+	15.15063+	45.0+	11.0.3+
	iceConnectionState	4/4	43.0.2357.65+	15.15063+	45.0+	11.0.3+
	connectionState	1/4				11.0.3+
	canTrickleIceCandidates	2/4		15.15063+	47.0+	
	getDefaultIceServers	0/4				
	getConfiguration	3/4		15.15063+	45.0+	11.0.3+
	setConfiguration	2/4	58.0.3029.81+			11.0.3+
	close	4/4	40.0.2214.93+	15.15063+	45.0+	11.0.3+
	onnegotiationneeded	4/4	43.0.2357.65+	15.15063+	45.0+	11.0.3+
	onicecandidate	4/4	43.0.2357.65+	15.15063+	45.0+	11.0.3+
	onicecandidateerror	0/4				
	onsignalingstatechange	4/4	43.0.2357.65+	15.15063+	45.0+	11.0.3+
	oniceconnectionstatechange	4/4	43.0.2357.65+	15.15063+	45.0+	11.0.3+
	onicegatheringstatechange	4/4	59.0.3071.86+	15.15063+	53.0+	11.0.3+
	onconnectionstatechange	1/4				11.0.3+
	generateCertificate	2/4	49.0.2623.75+		45.0+	
	getSenders	3/4	64.0.3282.119+		45.0+	11.0.3+
	getReceivers	3/4	59.0.3071.86+		45.0+	11.0.3+
	getTransceivers	2/4			59.0+	11.0.3+
	addTrack	3/4	64.0.3282.119+		45.0+	11.0.3+

Confluence Status (cont'd)

Interface	Member		Chrome	Edge	Firefox	Safari
	addTrack	3/4	64.0		9+	
	removeTrack	3/4	64.0.3282.119+		45.0+	11.0.3+
	addTransceiver	2/4			59.0+	11.0.3+
	ontrack	3/4	64.0.3282.119+		46.0+	11.0.3+
	sctp	0/4				
	createDataChannel	3/4	40.0.2214.93+		45.0+	11.0.3+
	ondatachannel	3/4	43.0.2357.65+		45.0+	11.0.3+
	getStats	4/4	40.0.2214.93+	15.15063+	45.0+	11.0.3+
	onstatsended	0/4				
	setIdentityProvider	1/4			45.0+	
	getIdentityAssertion	1/4			45.0+	
	peerIdentity	1/4			45.0+	
	idpLoginUrl	1/4			45.0+	
	idpErrorInfo	0/4				
RTCSessionDescription	type	4/4	43.0.2357.65+	15.15063+	45.0+	11.0.3+
	sdp	4/4	43.0.2357.65+	15.15063+	45.0+	11.0.3+
	toJSON	4/4	43.0.2357.65+	15.15063+	45.0+	11.0.3+
RTCIceCandidate	candidate	4/4	43.0.2357.65+	15.15063+	45.0+	11.0.3+
	sdpMid	4/4	43.0.2357.65+	15.15063+	45.0+	11.0.3+
	sdpMLineIndex	4/4	43.0.2357.65+	15.15063+	45.0+	11.0.3+
	foundation	0/4				
	component	0/4				
	priority	0/4				
	ip	0/4				
	protocol	0/4				
	port	0/4				
	type	0/4				
	tcpType	0/4				
	relatedAddress	0/4				
	relatedPort	0/4				
	usernameFragment	0/4				
	toJSON	4/4	43.0.2357.65+	15.15063+	45.0+	11.0.3+
RTCPeerConnectionIceEvent	candidate	3/4	56.0.2924.76+	15.15063+	45.0+	
	url	0/4				

Confluence Status (cont'd)

RTCPeerConnectionIceErrorEvent	hostCandidate	0/4				
	url	0/4				
	errorCode	0/4				
	errorText	0/4				
RTCCertificate	expires	2/4	49.0.2623.75+		45.0+	
	getSupportedAlgorithms	0/4				
	getFingerprints	1/4	61.0.3163.79+			
RTCRtpSender	track	4/4	64.0.3282.119+	13.10586+	45.0+	11.0.3+
	transport	1/4		13.10586+		
	rtcpTransport	1/4		13.10586+		
	getCapabilities	1/4		13.10586+		
	setParameters	1/4			46.0+	
	getParameters	2/4			46.0+	11.0.3+
	replaceTrack	3/4	65.0.3325.146+		45.0+	11.0.3+
	getStats	1/4			55.0+	
	dtmf	2/4	66.0.3359.117+		52.0+	
RTCRtpReceiver	track	4/4	59.0.3071.86+	13.10586+	45.0+	11.0.3+
	transport	1/4		13.10586+		
	rtcpTransport	1/4		13.10586+		
	getCapabilities	1/4		13.10586+		
	getParameters	1/4				11.0.3+
	getContributingSources	3/4	59.0.3071.86+	13.10586+	59.0+	
	getSynchronizationSources	1/4			59.0+	
	getStats	1/4			55.0+	
RTCRtpTransceiver	mid	2/4			59.0+	11.0.3+
	sender	2/4			59.0+	11.0.3+
	receiver	2/4			59.0+	11.0.3+
	stopped	2/4			59.0+	11.0.3+
	direction	2/4			59.0+	11.0.3+
	currentDirection	1/4			59.0+	
	stop	2/4			59.0+	11.0.3+
	setCodecPreferences	0/4				

Confluence Status (cont'd)

RTCDtlsTransport	transport	1/4		13.10586+		
	state	1/4		13.10586+		
	getRemoteCertificates	1/4		13.10586+		
	onstatechange	0/4				
	onerror	1/4		13.10586+		
RTCIceTransport	role	1/4		13.10586+		
	component	1/4		13.10586+		
	state	2/4		13.10586+		11.0.3+
	gatheringState	1/4				11.0.3+
	getLocalCandidates	0/4				
	getRemoteCandidates	1/4		13.10586+		
	getSelectedCandidatePair	0/4				
	getLocalParameters	0/4				
	getRemoteParameters	1/4		13.10586+		
	onstatechange	0/4				
	ongatheringstatechange	0/4				
	onselectedcandidatepairchange	0/4				
RTCTrackEvent	receiver	3/4	64.0.3282.119+		46.0+	11.0.3+
	track	3/4	64.0.3282.119+		46.0+	11.0.3+
	streams	3/4	64.0.3282.119+		46.0+	11.0.3+
	transceiver	2/4			59.0+	11.0.3+
RTCSctpTransport	transport	0/4				
	state	0/4				
	maxMessageSize	0/4				
	onstatechange	0/4				

Confluence Status (cont'd)

RTCDataChannel	label	3/4	56.0.2924.76+		60.0+	11.0.3+
	ordered	3/4	56.0.2924.76+		60.0+	11.0.3+
	maxPacketLifeTime	1/4				11.0.3+
	maxRetransmits	2/4	56.0.2924.76+			11.0.3+
	protocol	3/4	56.0.2924.76+		60.0+	11.0.3+
	negotiated	2/4	56.0.2924.76+			11.0.3+
	id	3/4	56.0.2924.76+		60.0+	11.0.3+
	priority	0/4				
	readyState	3/4	56.0.2924.76+		60.0+	11.0.3+
	bufferedAmount	3/4	56.0.2924.76+		60.0+	11.0.3+
	bufferedAmountLowThreshold	3/4	56.0.2924.76+		60.0+	11.0.3+
	onopen	3/4	56.0.2924.76+		60.0+	11.0.3+
	onbufferedamountlow	3/4	56.0.2924.76+		60.0+	11.0.3+
	onerror	3/4	56.0.2924.76+		60.0+	11.0.3+
	onclose	3/4	56.0.2924.76+		60.0+	11.0.3+
	close	3/4	56.0.2924.76+		60.0+	11.0.3+
	onmessage	3/4	56.0.2924.76+		60.0+	11.0.3+
	binaryType	3/4	56.0.2924.76+		60.0+	11.0.3+
	send	3/4	56.0.2924.76+		60.0+	11.0.3+
RTCDataChannelEvent	channel	3/4	56.0.2924.76+		45.0+	11.0.3+
RTCDTMFSender	insertDTMF	2/4	66.0.3359.117+		52.0+	
	ontonechange	2/4	66.0.3359.117+		52.0+	
	canInsertDTMF	1/4	66.0.3359.117+			
	toneBuffer	2/4	66.0.3359.117+		52.0+	
RTCDTMFToneChangeEvent	tone	3/4	66.0.3359.117+	13.10586+	52.0+	
RTCStatsEvent	report	0/4				
RTCIdentityProviderGlobalScope	rtcIdentityProvider	0/4				
RTCIdentityProviderRegistrar	register	0/4				
RTCIdentityAssertion	idp	0/4				
	name	0/4				
RTCErrorEvent	error	0/4				

WG decisions to be made

- What do we do to accelerate simulcast testing?
- When do we remove features that fall below the bar?
  - In a year? In 18 months? Never?
  - Separating functionality can be non-trivial (worked with Identity, but not Simulcast)
- How do we measure interoperability?
  - Using WPT tests running natively on browsers?
  - Using WPT tests running on adapter.js shim?
  - Using KITE?

Break (see you at 3:30 PM)

Webrtc-NV Use Cases (Bernard, 30 minutes)

WebRTC Next Version Use Cases

- Intended as a followup to RFC 7748 “WebRTC Use Cases”
- Goal: to document the use cases motivating development of “WebRTC Next Version” APIs and the requirements that arise from them.
- Document: https://w3c.github.io/webrtc-nv-use-cases/

Questions to Ask About Use Cases

- Deployment likelihood
  - Can you imagine an established company or startup investing resources to implement this use case?
  - Have developers already implemented (or attempted to implement) this use case using WebRTC 1.0 or ORTC?
    - Has that implementation garnered a substantial audience?
  - Is this use case widely implemented outside the Web (e.g. in native applications)?
- Barriers to implementation in WebRTC 1.0
  - Are there major limitations to implementation of this use case in WebRTC 1.0 or can it already be done “well enough”?

Improving Use Cases in RFC 7748

- Multiparty online game with voice communications (Section 2.1).
  - Requirements: ICE improvements, bandwidth limits, early media.
  - Multiple implementations based on ORTC
- Mobility (Section 2.2)
  - Requirements: ICE improvements
  - Multiple implementations based on ORTC
- Video conferencing with a central server (Section 2.3)
  - Requirements: SVC, differential protection, receivers without corresponding senders, no SDP
  - Multiple implementations based on ORTC

New Use Cases Implemented Using WebRTC 1.0 or ORTC

- File Sharing (Section 3.1).
  - Requirements: transfer of large files, back pressure, better congestion control, server support, support for workers.
  - Has motivated WebRTC 1.0 bug fixes, API extensions to be discussed tomorrow (e.g. WHATWG streams)
- Internet of Things (Section 3.2)
  - Requirements: ICE improvements, detailed control of data transport behavior, support for ordered/unordered, reliable/unreliable.
  - Many (powered) implementations based on WebRTC 1.0 and ORTC (spinning classes with remote instructors, speech-driven appliances, etc.)
  - Is there a real world demand for WebRTC data exchange in low power IoT devices? Examples?

Use Cases That Developers Have Implemented Natively

- Where use cases are implemented and deployed, motivation can be quantified.
- Funny Hats (Section 3.3).
  - Lots of native mobile applications shipping this today.
  - In some cases, performance not adequate on the web (e.g. background blur)
- Machine Learning (Section 3.4)
  - Integration of webrtc and machine learning in native applications is becoming very popular.
  - Widespread interest in tensorflow.js but in some use cases performance is insufficient. See: https://modeldepot.github.io/tfjs-yolo-tiny-demo/
  - Requirements: See next presentation
- Virtual Reality Gaming (Section 3.5)
  - Multi-player VR games are typically developed as native applications.
  - Requirements: Ability to synchronize data with audio/video/depth
- Secure communications (Section 3.6)
  - Worlds of entertainment and RTC are converging.
  - Entertainment scenarios (e.g. sports, game streaming) now require low latency.
  - E2E scenarios not just about untrusted cloud, but also content protection.
  - Requirements: See Emad's presentation tomorrow.

Open Issues

- Total: 16
- Breakdown by topic:
  - Clarification required/editorial: 5
  - ICE: 4
  - Raw media: 3
  - One-way communications/Sender-Receiver: 2
  - Miscellaneous: 2
- Breakdown by Submitter:
  - Lennart Grahl: 6
  - Harald: 5
  - Astojilj: 2
  - Others: 3

WG decisions to be made

- Adopt WebRTC NV Use Cases as a WEBRTC WG work item?

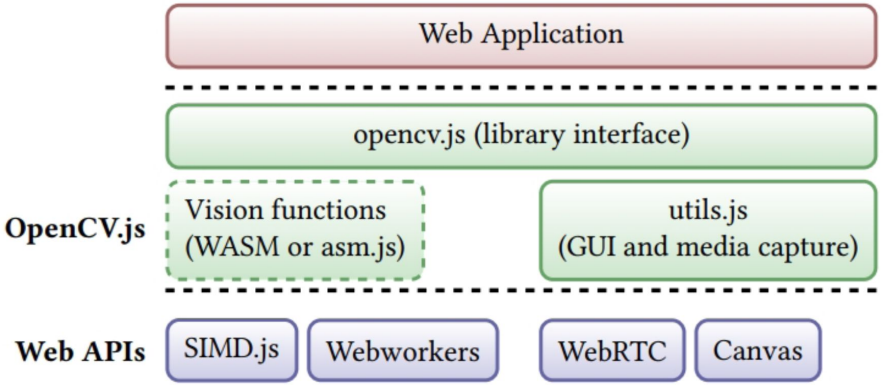
Performance Challenges of OpenCV.js video pipeline

Ningxin Hu, Intel
(30 minutes)

Acknowledgements:
Moh Haghighat, Intel
Sajjad Taheri, UC Irvine

OpenCV.js Overview

- Introduce js module of OpenCV
- Convert 200+ most commonly used vision functions into WASM
- Interact with media stream, video, image and canvas on Web
- Expose web developer friendly JavaScript API



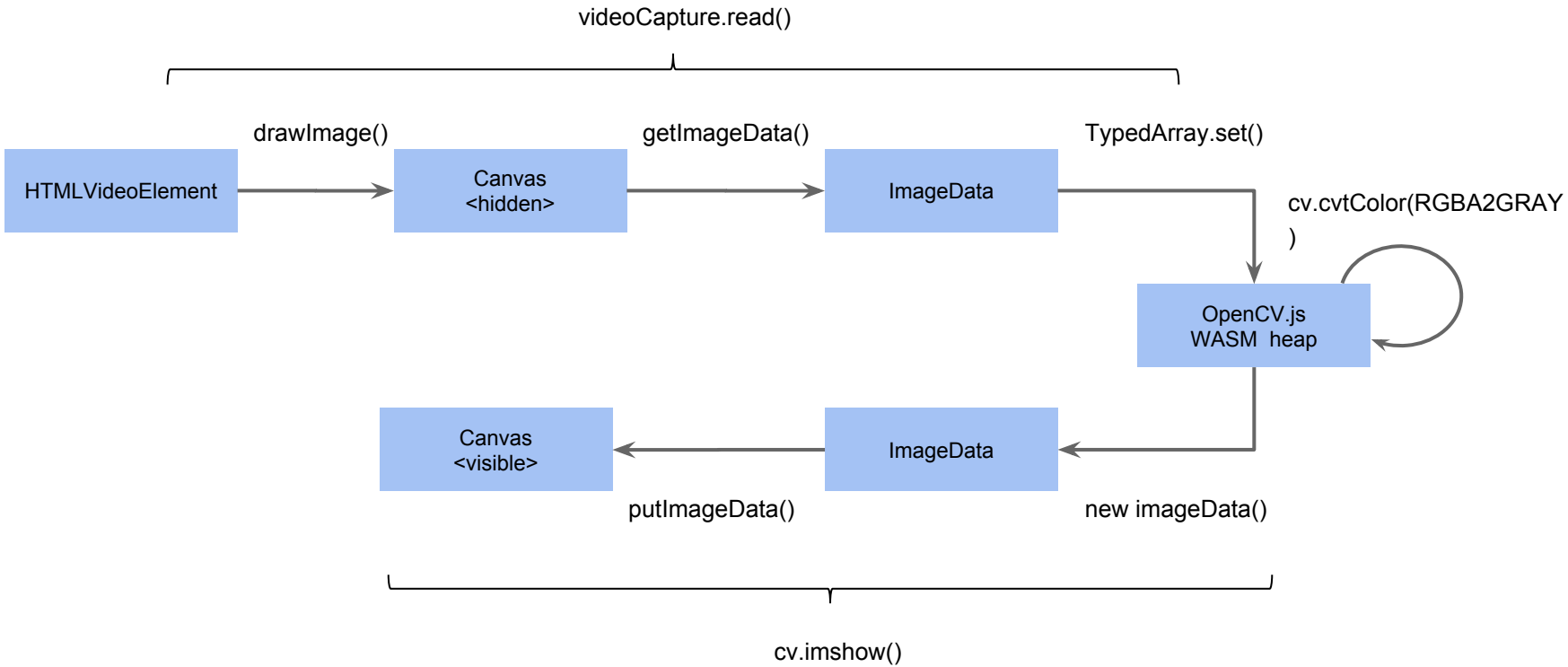
- 30+ tutorials and demos on docs.opencv.org
- Derived from an Intel funded research at the UC Irvine
- Developed by UC Irvine, Intel, OpenCV.org, and Google Summer of Code
- More references: EETimes, Intel Parallel Universe

Video I/O Example of OpenCV.js

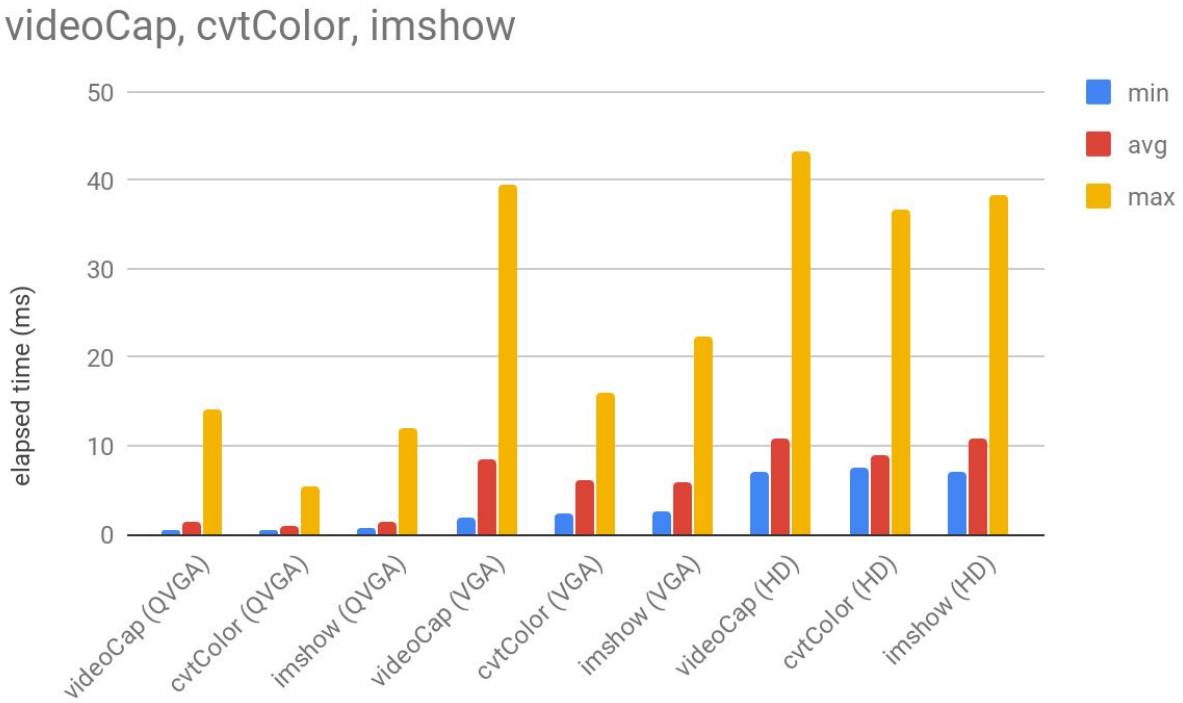
```
let src = new cv.Mat(height, width, cv.CV_8UC4);
let dst = new cv.Mat(height, width, cv.CV_8UC1);
let cap = new cv.VideoCapture(videoSource);
const FPS = 30;
function processVideo() {
    let begin = Date.now();
    cap.read(src);
    cv.cvtColor(src, dst, cv.COLOR_RGBA2GRAY);
    cv.imshow("canvasOutput", dst);
    let delay = 1000/FPS - (Date.now() - begin);
    setTimeout(processVideo, delay);
}
setTimeout(processVideo, 0);
```

https://docs.opencv.org/3.4.3/dd/d00/tutorial_js_video_display.html

Interaction with WebRTC, video and canvas



Performance



Test: https://codepen.io/huningxin/pen/ReMezx

Data collected on Chrome 69, Win 10, i5-7300U, HD620

Challenges

- Performance:
  - Memory copies: No WASM memory mapping support, have to copy memory in/out OpenCV.js WASM heap
  - GC jitters: due to new ImageDatas from WASM heap
  - Color conversions: Only RGBA, but OpenCV works with BGR (default), YUV and Grayscale
  - Off-main-thread processing: the overhead of transferring data from/to web worker
- Features:
  - Set capture properties: brightness, contrast, saturation, gain, exposure etc.,
  - Get camera intrinsics: for camera pose estimation use case
  - Support cv.VideoWriter

Note: Mozilla FoxEye project did early exploration for video pipeline efficiency.

WebRTC-ICE
(Peter Thatcher, 30 minutes)

Reminder of Purposes

Of NV-style RTCIceTransport (free from PC):

- Needed by everything else NV-style (SctpTransport, RtpXer, QuicTransport)

Of FlexICE:

- wifi/cell control
- check activity/frequency control
- "relay first" checking
- continual gathering and
- network switching control
- forking

Reminder of status 4 months ago (Stockholm f2f)

- Consensus on doing NV-style IceTransport (w/o PC)
- Consensus on doing FlexICE (generally)
  - With observation that it could apply to PC-style IceTransport

Spec Progress

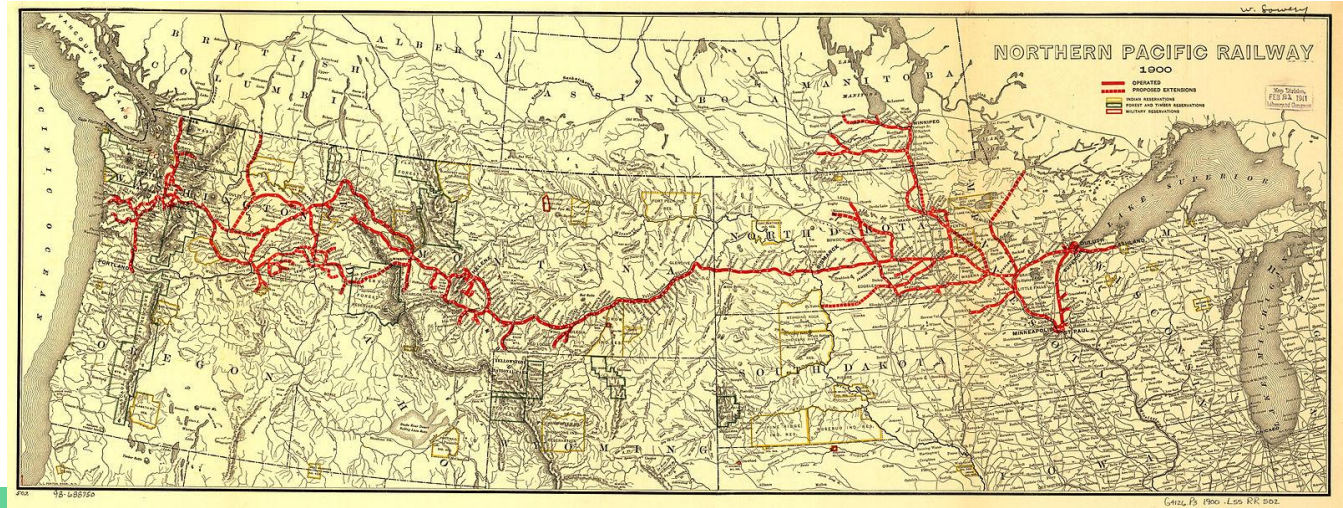
- Present already
  - `IceTransport.gather(IceGatherOptions)`
- Present In recent PR (https://github.com/w3c/webrtc-ice/pull/22)
  - `IceCandidate.networkInformation`
  - `IceCandidate.networkId`
  - `RTCIceGatherOptions.networkIds`
  - `IceTransport.retainLocalCandidate(IceLocalCandidate)`
  - `IceTransport.removeLocalCandidate(IceLocalCandidate)`
  - `IceTransport.getCandidatePairs()`
  - `IceCandidatePair.setMinCheckInterval(seconds)`
  - `IceCandidatePair.setFrozen(bool)`
  - `IceCandidatePair.select()`
  - `IceCandidatePair.nominate()`
  - `IceCandidatePair.waitForReceiveTimeout(seconds)`
  - `IceTransport.onchecksent`
  - `IceCheck.response`
  - `RTCIceCandidatePair.setCheckPriority(priority)`
  - `IceTransport.fork()`

Impl Progress

- Implementation of NV-style IceTransport in Chrome (no FlexICE)
- Implementation of ORTC-style (similar to NV) in Edge (no FlexICE)

Next Steps

- (Finish) implementations of NV-style
- Land the FlexICE PR (it's big and a little rough)
- Implement NV-style data channels on top (topics tomorrow)
- Implement FlexICE (Is there an urgent customer?)



Questions

1. Can we change IceCandidatePair into an interface instead of dictionary?
2. When the IceCandidatePair.select() is called, does that disable *all* automatic candidate pair reselection? Even if the candidate pair is removed?
3. If nominate() is called when aggressive nomination is disabled and there's no renomination, what do we do?
4. Should we use NetworkInformation or just ConnectionType?
5. Who will implement FlexICE (especially forking)?

For extra credit

Name that bird!

Thank you

Special thanks to:

Google for Hangouts

WG Participants, Editors & Chairs

The bird