W3C WebRTC WG Meeting

TPAC
September 22-23, 2016

Chairs: Harald Alvestrand
Stefan Hakansson

W3C WG IPR Policy

- This group abides by the W3C patent policy https://www.w3.org/Consortium/Patent-Policy-20040205
- 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 meeting of the W3C WebRTC WG!
- During this TPAC meeting, we hope to make progress on some outstanding issues before transition to CR
- Editor’s Draft update to follow meeting

About this Meeting

Information on the meeting:

- Meeting info:
    - https://www.w3.org/2011/04/webrtc/wiki/September_22_-_23_2016
- Link to Slides has been published on WG wiki
- Scribe? IRC http://irc.w3.org/ Channel: #webrtc
- The meeting is being recorded.
- WebEx info here

High level agenda Thursday

- Thursday morning
  - PC - queing and order of operations, negotiation needed
  - PC - Errors (not) thrown
  - PC - events not defined
- Thursday afternoon
  - Mediacapture-main
    - Interactions between Permissions and Media: State of play
    - Issues to resolve
    - Test suite status
  - Mediacapture-depth
  - Mediacapture-output(?)
  - Getting docs to CR

High level agenda Friday (all about WebRTC)

- Morning
  - Homework reporting
  - Sender/Receiver/Transceiver (skipping 305)
  - ~~DTMF~~
  - Stats
- Afternoon
  - Certificate + IdP stuff. At 13.00
  - ICE
  - Test suite
  - Summary, conclusions

Thursday morning

- ~~Certificate + IdP stuff~~ (moved to Friday)
- PC - queuing and order of operations, negotiation needed (Adam)
    - #787, #782+#792, #645
- PC - Errors (not) thrown (Taylor, Adam)
    - #746, #727
- PC - events not defined (Adam)
    - #526

Issue 782: pc.createOffer(), pc.addTrack(track) should not include track (Adam)

```
pc.addTrack(trackX, stream);
pc.createOffer().then(offer => { ... })
pc.addTrack(trackY, stream);
```

- Desired effect: trackX is represented in offer while trackY isn't
- PR #792 proposes:
  - In the synchronous part of createOffer/Answer, take a snapshot of:
    - Added media (RTCRtpSenders)
    - If createDataChannel has been called for the first time
  - Use snapshot to create offer/answer

Issue 787: Integrate RTCRtpTransceiver into setlocal/remote steps (Adam)

- Transceivers are pretty well integrated into addTrack and addTransceiver
- Still not mentioned in createOffer/Answer and setLocal/RemoteDescription
- Blocks (would be easier to fix)
  - #788 Clean up remaining uses of 'set of receivers'
  - #578 Need to specify precisely when MID generation happens

- Proposed fix includes
  - Use ‘provisional mid’ for new transceivers to be used in offers
  - ‘provisional mid’ is used when setLocalDescription sets the mid
  - ‘provisional mid’ will be discarded if a transceiver gets its mid from the remote side via setRemoteDescription
  - Other transceiver behavior from JSEP

Issue 645: Public negotiation-needed flag as readonly (Adam)

- Arguments for
  - Detect event already fired case
  - Useful in multi-threaded apps
- Against
  - Always possible to register event (after construction)
  - Internal flag is in the ‘non-stable’ state

- Latest activity in issue
  - Agreement that the negotiatioin-needed flag isn’t suitable to be exposed in its current form

Issue 746: What happens if createDataChannel is called with an invalid id (Taylor)

- **Case A:** createDataChannel is called with ID 65535 (which is always invalid, since the maximum possible ID is 65534).
  - **Proposal**: throw an OperationError (or IndexSizeError) from createDataChannel.
- **Case B:** createDataChannel is called with an unspecified ID, and no more valid IDs are available (all valid IDs that correspond to the DTLS role are taken).
  - Sometimes, we can't know if this is the case at the time of `createDataChannel`, because the DTLS role hasn't been negotiated.
  - **Proposal**: Asynchronously invoke the `onerror` event handler once the DTLS role has been negotiated. If it's already negotiated, still invoke `onerror` asynchronously.
    - TODO: What goes in the error?

Issue 746: What happens if createDataChannel is called with an invalid id (continued)

- **Case C:** createDataChannel is called with an ID that's outside the negotiated range.
    - Sometimes, we can't know if this is the case at the time of \`createDataChannel\`, because the SCTP handshake hasn't yet occurred.
    - **Proposal**: Asynchronously invoke the \`onerror\` event handler once the SCTP handshake has finished. If it's already finished, still invoke \`onerror\` asynchronously.
        - TODO: What goes in the error? Same thing as in Case B?

[Issue 746](): What happens if createDataChannel is called with an invalid id (continued)

Example (extreme corner case which covers both scenarios):

1. App calls createDataChannel with negotiated=true, for ALL even IDs.
2. App calls createDataChannel with negotiated=false, for data channel “foo”.
3. DTLS role is negotiated; this PeerConnection is the client (even IDs).
   a. This is where “foo” would normally be assigned an ID, but there are no more valid IDs available, so its error handler is invoked (Case B).
4. SCTP handshake finishes; only 1024 SCTP streams are negotiated.
   a. The error handler is invoked for every data channel with an ID greater than 1023 (Case C).

Issue 727: removeTrack: throw exception if sender is not in set of senders (Taylor)

- Proposal: Throw an exception if and only if the sender was not created by the PeerConnection.

- ```
  var sender = pc.addTrack(track, stream);
  pc.removeTrack(sender); // fine
  pc.removeTrack(sender); // fine
  ```

- ```
  var sender = pc1.addTrack(track, stream);
  pc2.removeTrack(sender); // exception
  ```

Issue 526: NetworkError event is not defined (Adam)

- When the underlying transport of a DataChannel is closed with an error, the spec says: fire a NetworkError in addition to the simple close event
- Issue: NetworkError is not defined
- WebSocket spec
  - Fires simple error event
  - Has more descriptive close event (e.g. code and reason)

- How do we want to report these errors?
- What are implementations currently doing?

Thursday afternoon

- Mediacapture-main (Jan-Ivar)
  - Interactions between Permissions and Media: State of play
  - Issues to resolve (Jan-Ivar for 350, 380, 387 and 389; Dan for #394)
- Test suite status (Dom)
- Mediacapture-depth (Ningxing)
- Mediacapture-output (? unconfirmed)
- Getting docs to CR (Harald)

Interactions between Permissions & Media: State of play (jan-ivar)

- w3c.github.io/permissions has pivoted to be an algorithm hub + `.query()`
- `.query()` returns ***permission state*** i.e. `"prompt"` (default), `"granted"` or `"denied"`.
- ***"request permission to use"*** algorithm returns `"granted"` or `"denied"` (per-use!)
  - Prompts the user if ***permission state*** == `"prompt"`
  - **MAY** update ***permission state*** per realm or origin based on "user intent" (up to UA; ~~http~~).
- MediaCapture spec before: Per-use vs. Persisted to origin (https).
- After: Per-use, Per-realm (https), or Persisted to origin (https).

Issues to resolve

- **Issues**
  - **Issue 387**: Reinstate strong language on permission ending when tracks stop, lost by editorial mistake (Jan-Ivar)
  - **Issue 389**: Camera light and "disabled" tracks (Jan-Ivar)
  - **Issue 350**: New permission definitions are wrong (Jan-Ivar)
  - **Issue 380**: Remove redundant list-devices permission (Jan-Ivar)
  - **Issue 394**: Browser's ability to dynamically change settings mistakenly removed (Dan)

#387: The spec has a privacy problem.

The permission rewrite refactored out a privacy guarantee that likely was never there. “Revoke permission on last stop(), unless stored” is toothless in https, as UAs may temporarily “store” permission without consequence, to work around it (rendering the stored/unstored distinction moot).

Turning the camera light off on "disabled" tracks (#389) has the same problem: Sites may resume recording at any time without user awareness (google “WebcamGate”).

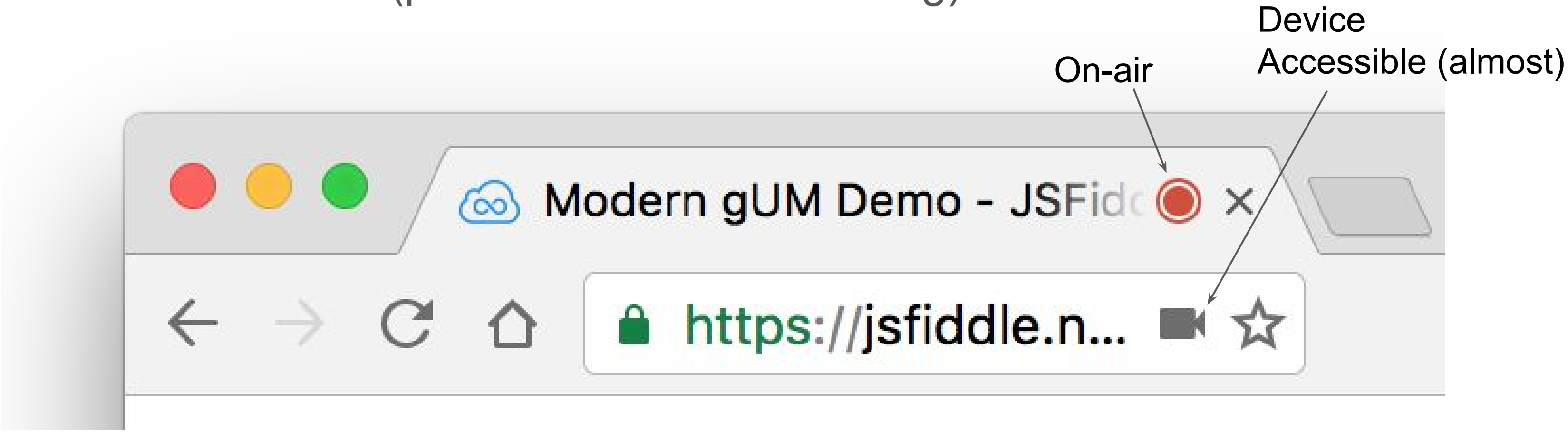
Mandating privacy choices like per-use permission would be one remedy.

We need to revisit our intent. We discussed privacy in Washington DC in 2014...

#387: Compromise: stronger language on indicators.

"Washington agreement" https://www.w3.org/Bugs/Public/show_bug.cgi?id=25784
In 2014 in DC we defined two privacy indicators with strong consensus:

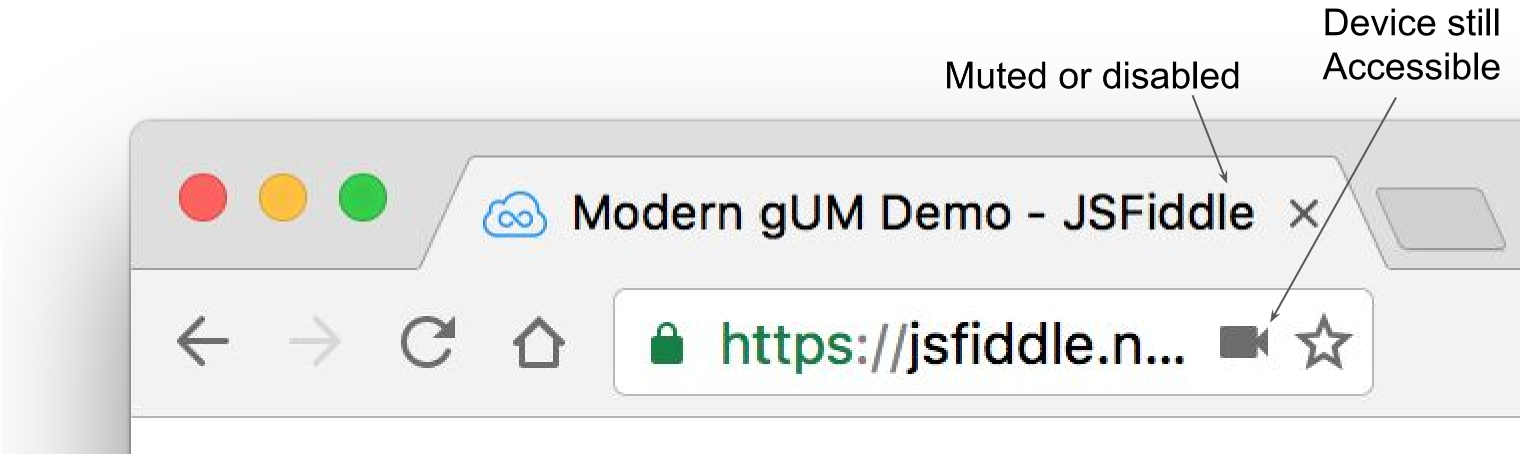
1. "On-air" (actively recording), and
2. "Device accessible" (power to resume recording)



#389: Allow “on-air” + cam light off on mute/disabled.

We get reports users worry when they mute and “on-air” and camera light stay on.

Users need an indicator that the site can unmute at any time (“Device accessible”)
This is the same privacy concern driving per-use permissions.

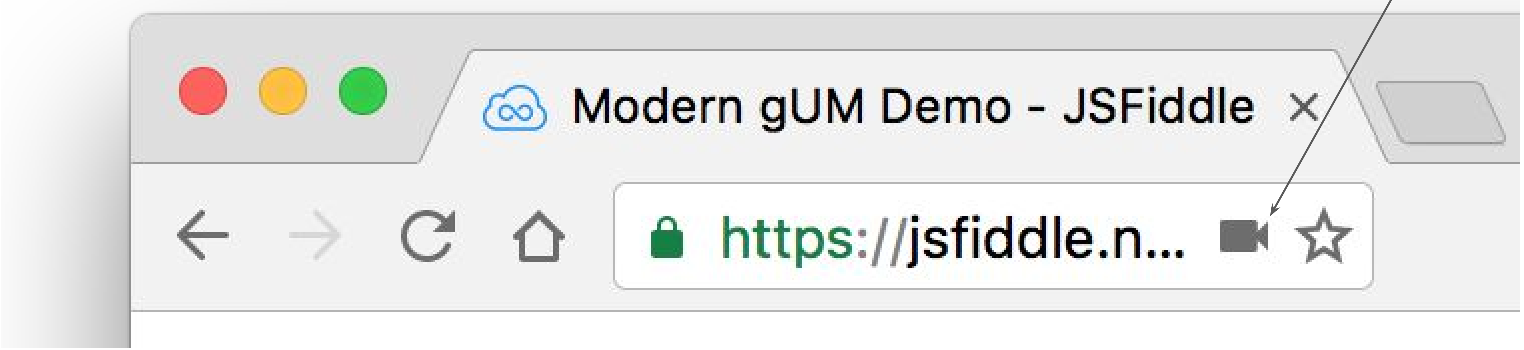


#387/389: Solution: Make indicators normative MUST

All browsers implement “on-air” already.
LIVE recording is more personally sensitive than all other powerful features.

Washington agreement taken to its conclusion solves mute/disabled and stop() in the same way, consistently across browsers (and allows Edge’s https behavior)

On page-load it means persistent permission



#387/389: Definition of indicator states for device (1)

Under 9.2 MediaDevices:

On page load, for each kind of device and each individual device, *deviceOrKind*, on this system that is to interact as described in this specification, execute the following steps:

1. let *permission* be the result of retrieving the permission state of the primary permission associated with *deviceOrKind* (e.g. “camera” or “microphone”), with *deviceOrKind*’s `deviceId` as input when *deviceOrKind* is a device.
2. Create a [[**<*deviceOrKind*>Accessible**]] internal slot on the **relevant global object** and initialize it to either 1 if *permission* is “granted” or 0 otherwise.
3. If *deviceOrKind* is a device, create a [[**<*device*>Live**]] internal slot on the **relevant global object** and initialize it to 0.

For each *deviceOrKind*, whenever the permission state transitions to “granted” from another value, increment the corresponding [[**<*deviceOrKind*>accessible**]] internal slot by 1.

For each *deviceOrKind*, whenever the permission state transitions from “granted” to another value, decrement the corresponding [[**<*deviceOrKind*>accessible**]] internal slot by 1.

#387/389: Definition of indicator states for device (2)

Under 10.1 getUserMedia steps:

If the result of the request is “granted”, then for each device, *device*, in the provided media, increment the corresponding [[**<*device*>Live**]] internal slot by 1.

If the UA wants to be able to temporarily turn off a device while all tracks connected to that device are muted or disabled, then whenever a [[**<*device*>Live**]] internal slot is incremented, the UA MUST also increment the corresponding [[**<*device*>Accessible**]] internal slot by the same amount, and whenever a [[**<*device*>Live**]] internal slot is decremented, the UA MUST also decrement the corresponding [[**<*device*>Accessible**]] internal slot by the same amount, so that a [[**<*device*>Accessible**]] internal slot never falls below 1 while a track is connected to the device.

#387/389: Definition of indicator states for device (3)

Under 4.3 MediaStreamTrack:

When all tracks using a device have been stopped or ended by some other means, the device is **stopped**, and the UA MUST decrement the corresponding [[**<device>Live**]] internal slot by 1.

Under 4.3.1 Life-cycle and Media Flow:

If the UA wants to be able to temporarily turn off a device while all tracks connected to that device are muted or disabled, then whenever a [[**<*device*>Live**]] internal slot was incremented, the UA MUST also have incremented the corresponding [[**<*device*>Accessible**]] internal slot by the same amount, and whenever a [[**<*device*>Live**]] internal slot was decremented, the UA MUST also have decremented the corresponding [[**<*device*>Accessible**]] internal slot by the same amount, so that a [[**<*device*>Accessible**]] internal slot never falls below 1 while a track is connected to the device.

Provided the above has been met and continues to be followed, when all tracks connected to a device are muted or disabled, the UA MAY decrement the corresponding [[**<device>Live**]] internal slot by 1, provided the UA increments it back by 1 when the track is no longer muted or disabled.

#387/389: Use of indicator states for device (1)

- Assume we have *Device*Accessible and *Device*Live booleans for each *Device*.
- For each *DeviceType*,
  - Define *DeviceType*Accessible to be LogicalOr(*Device1*Accessible, *Device2*Accessible, etc.) for all the devices of that *DeviceType*.
  - Define *DeviceType*Live to be LogicalOr(*Device1*Live, *Device*2Live, etc.) for all the devices of that *DeviceType*
- Define Accessible to be LogicalOr(*DeviceType1*Accessible, *DeviceType2*Accessible, etc.) for all values of *DeviceType*.
- Define Live to be LogicalOr(*DeviceType1*Live, *DeviceType*2Live, etc.) for all values of *DeviceType*.

#387/389: Use of indicator states for device (2)

- Requirements
  - The UA MUST indicate to the user when the value of Accessible changes.
  - The UA MUST indicate to the user when the value of Live changes.
  -
  - If the UA provides any indication of Accessible or Live to the user per *DeviceType*, at a minimum it MUST indicate when each such value changes.
  - If the UA provides any indication of Accessible or Live to the user per *Device*, at a minimum it MUST indicate when each such value changes.
  - Any off-to-on transition MUST remain observable for a sufficient time that a reasonably-observant user could become aware of it.
  -
  - Any of the above indications MAY be combined as long as the combined indication cannot transition to off if any of its component indications are still on.

#387/389: Use of indicator states for device (3)

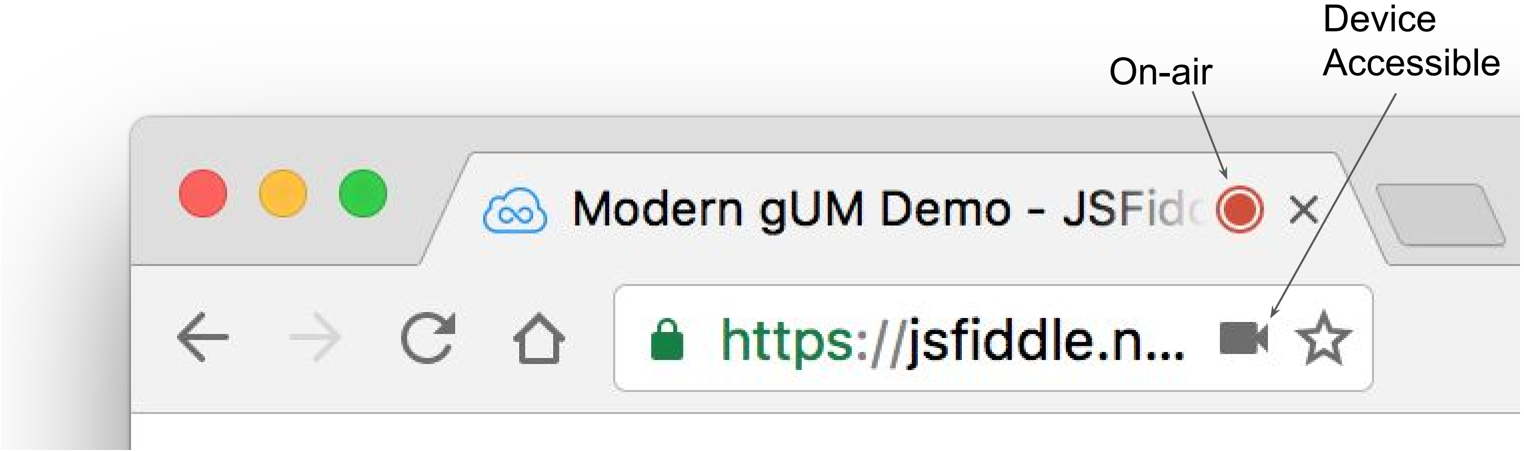
- Suggestions
  - The UA is encouraged to provide ongoing indication of the current state of Accessible.
  - The UA is encouraged to provide ongoing indication of the current state of Live and to make any generic hardware device indicator light match.
  - If the UA provides any indication of Accessible or Live to the user per *DeviceType*, it is encouraged to provide ongoing indication of the current state of each such value. It is encouraged to make any device-type-specific hardware indicator light match the corresponding Live value.
  - If the UA provides any indication of Accessible or Live to the user per *Device*, it is encouraged to provide ongoing indication of the current state of each such value. It is encouraged to make any device-specific hardware indicator light match the corresponding Live value.
  - Any optional ongoing indication MAY be used instead of the corresponding required transition indication provided the off-to-on transition requirement is met.

#387/389: Use of indicator states for device (Example)

- Assume we have Camera1, Camera2, Camera3, Mic1, and Mic2 and want device-type-specific current status indicators
- Then we will have the following status variables:
  - Camera1Accessible, Camera1Live, Camera2Accessible, Camera2Live, Camera3Accessible, Camera3Live, Mic1Accessible, Mic1Live, Mic2Accessible, Mic2Live
    - CameraAccessible, CameraLive, MicAccessible, MicLive
    - Accessible, Live
  - And the following indicators:
    - CameraAccessible - On when any of Camera*N*Accessible is true
    - CameraLive - Turns on whenever all Camera*N*Live are false and one becomes true, then remains on for max(any of Camera*N*Live is true, 2 seconds)
    - MicAccessible - On when any of Mic*N*Accessible is true
    - MicLive - Turns on whenever all Mic*N*Live are false and one becomes true, then remains on for max(any of Mic*N*Live is true, 2 seconds)

#387/389: Possible UIs (non-normative)

- UAs may translate the two states into something other than two discrete indicators, provided all possible transitions remain indicated.
  - E.g. A cellphone browser in fullscreen may use well-worded toast messages to indicate transitions instead.
- Allows a UA to show “device accessible” indicator whenever “live” is on
- Allow a UA to merge the two states into a single 4 (3?) state indicator?



Signs of WG confusion in the spec

Indicators appear optional at first:

Strong MAY

> *"If the result of the request is "granted", User Agents are **encouraged** to include a prominent indicator that the devices are "hot" (i.e. an "on-air" or "recording" indicator), as well as a "device accessible" indicator indicating that the page has been granted access to the source."*

*"When all tracks connected to a source are muted or disabled, the "on-air" or "recording" indicator for that source can be turned off; when the track is no longer muted or disabled, it **MUST** be turned back on."*

Yet elsewhere, privacy-protecting normative language takes "on-air" for granted

#350: Language OK. One behavioral change:

- No reprompts in getUserMedia of sources already live on page (pull #319)
  - Reason: Equivalent to stream.clone().

```
let gUM = constraints => navigator.mediaDevices.getUserMedia(constraints);
let sameDevice = {video: {deviceId: {exact: ”HfcBVSHEjObnXqPar”}}};

let stream1 = await gUM(sameDevice); // Prompts the user
let stream2 = await gUM(sameDevice); // MUST NOT prompt.

[stream1, stream2].forEach(s => s.getTracks()[0].stop());

let stream3 = await gUM(sameDevice); // May prompt again (MUST in http)
```

#380: device labels + devicechange and stop()

- When access to `deviceInfo.label` and `devicechange` events expires:
  - **Before:** ~~MUST be same as gUM permission (e.g. after last `.stop()`)~~
  - **After:** Separate `"device-info"` permission ***weaker than*** `"camera"` and `"microphone"`

    (If *camera* OR *microphone* are `"granted"`, then *device-info* is also `"granted"`.
    If *device-info* is `"denied"`, then camera AND microphone are also `"denied"`.)

#380: All legal states

camera	microphone	device-info
prompt	prompt	prompt
prompt	prompt	granted
prompt	granted	granted
prompt	denied	prompt
prompt	denied	granted
granted	prompt	granted
granted	granted	granted

camera	microphone	device-info
granted	denied	granted
denied	prompt	prompt
denied	prompt	granted
denied	granted	granted
denied	denied	prompt
denied	denied	granted
denied	denied	denied

Maybe add: If *camera* AND *microphone* are `"denied"`, then *device-info* is also `"denied"` ?

Mediacapture-depth

Issue 394: Browser's ability to dynamically change settings mistakenly removed (Dan)

- **SelectSettings algorithm**: determines current settings for a source
- **But**, only executed when either getUserMedia() or applyConstraints() is called. This is a **MISTAKE**.
- **Original intent with constraints:** settings can change at any time as long as those changes would not cause an OverconstrainedErrorEvent on any current tracks.
- **Should work like this** (in a world with only one constrainable property):
  - Track A: constraints allow either 1 or 2
    - UA chooses 1
  - Track B: constraints allow only 2.
    - UA should change source to setting 2, affecting both track A and track B.
  - Track C: constraints allow either 1 or 3
    - Should be rejected (even though track A could have accepted this).
  - Now assume setting 2 becomes impossible due to changes outside the control of the UA (source availability or configurability).
    - Track B should receive an OverconstrainedErrorEvent and the source should change to 1, affecting track A.

- Thoughts?

State Of Test Suite
Media Capture & Stream

Dr. Alex Gouaillard

Cosmo Software Consulting

Satellite view

Specification	TR Working draft	Stable draft (Last Call)	Implementors feedback (CR)	Standard (Rec)	Test Suite
WebRTC 1.0: Real-time Communication Between Browsers	28-Feb-16	N/A	Q2 2016?		Y
Media Capture and Streams (getUserMedia)	12-Feb-15	14-Apr-15	19-May-16		Y
MediaStream Recording	8-Sep-15	N/A			**N**
MediaStream Image Capture	27-Jan-15	N/A			**N**
Media Capture Depth Stream Extensions	26-Feb-16	N/A			**N**
Media Capture from DOM Elements	19-Feb-15	N/A			**N**
Audio output devices API	10-Feb-15	N/A			**N**
Identifiers for WebRTC's Statistics API	6-Feb-15	N/A			**N**
Screen Capture	10-Feb-15	N/A			**N**

APIs View

Media Capture and Streams (getUserMedia)	isTested	Size	Coverage
MediaStream	Y	179	**95%**
MediaStreamTrack	Y	512	**34%**
Media stream in Media element	Y	77	**100%**
Enumerating Local Media Devices	Y	159	**45%**
GUM	Y	281	68%
Constraints	N	136	**0%**
Error Handling	N	100	**0%**

Easy Fixes

- Forgotten methods
  - MediaStream.getTracks
  - MediaStreamTrack.clone
- New Methods and Objects
  - MediaTrack
    - SupportedConstraints,
    - Capabilities,
    - Constraints,
    - ConstraintSet,
    - Settings
  - OverConstrainedError
  - Constrainable Pattern ?
- Errors which changed name or type
- Types and event missed (ondevicechanged, …)

Hard to test

- When “source” is hardware
  - Source disconnected or exhausted
  - All tracks ended => source stopped
  - Tracks ends for any other reasons than calling stop()
- Several UA algorithms
  - MediaDeviceInfo list creation algorithm
  - Constraint resolution and mediastream acquisition
  - Stored permissions
  - Cookies like deviceId
  - Origins (and frames)
- Permission prompt complex behavior

Where do we stand? (march 2016)

OS	browser	OK	FAIL	Time Out
OSX 10.x	Safari 9.0.3	0	32	0
	Safari Nightly	0	32	0
	FF 44	6	26	0
	Cr 48	1	20	22
win 10	edge	47	42	8
Media Capture and Streams				

OS	browser	OK	FAIL	Time Out
OSX 10.x	Safari 9.0.3	0	32	0
	Safari Nightly	2	19	22
	FF 44	14	17	1
	Cr 48	45	48	2
win 10	edge	45	44	8
Media Capture and Streams (adapter.js)				

Mediacapture-output

- Problem area: reserved identifiers
  - #19 Are they the Right Thing?
  - #18 Should they have “id-” in front?
  - #30 Behavior on no ID needs specifying

Mediacapture-image

- According to editors, this is ready to go to CR
- The chairs will issue a final request for reviews right after TPAC

Friday morning

- Homework
- Sender/Receiver/Transceiver (Adam, Bernard, Cullen, Peter)
  - #764 (PRs #758+#786), #763, #698(+#578), #624(PR), ~~#305 Done yesterday~~
- ~~DTMF (Bernard)~~
  - ~~#784 #775, #773, #772, #770, #769, #768 Done yes.~~
- Stats (Harald and Varun)
  - TBD

Homework reports

- Adam + J-I to update PR for webrtc-pc #782
- (J-I to write test for neg-needed for later)
- J-I + Dan to work on mediacapture-main #387/389/350
  - To become a PR later, for tomorrow general idea
- J-I to create specific text for #380 that we can look at to determine if that is the way to go
- ?ERROR reporting (Cullen, Bernard)

Errors needed in IDP Section (1 of 3)

IDP Load Error

When fetching a HTTP need to separate out various networking errors such as can't resolve DNS, can connect, failed TLS, TLS cert pinning failed, HTTP error code, etc. Also need error when the protocol is not supported by the IdP.

IDP Script Error

When loading a IdP proxy, if there is a problem with the IdpScript need an error indicating this.

IDP Login Needed Error

Error needs to provide URL for login

Errors needed in IDP Section (2)

IDP Invalid Token

- Error needed if the token was invalid

IDP Expired Token

- Error needed if the token has expired

IDP TImeOut

- Need an error if IDP took too long and timer expired

Errors needed in IDP Section (3)

Need Fingerprint mismatch error

- Error if fingerprint in SDP does not match IDP Token. Need to know which mid had the failed fingerprint.

Need Custom Error for IdP

- IdP JS code needs a way to to pass IDP defined error to the main application

PR 784: interToneGap and Duration fixes (Bernard)

- Are duration and interToneGap optional arguments or not? Today we have:

```
void insertDTMF (DOMString tones, optional unsigned long duration = 100, optional unsigned long
interToneGap = 70);
```

- Conclusion: WebIDL spec is ok as is.
- Problem: This may not behave as applications expect:
  - insertDTMF(tones, 200, 100)
  - insertDTMF(tones) // duration = 100, interToneGap = 70!

Issue 799: Unclear when a DTMFToneChangeEvent is fired with an empty string (Bernard)

- Section 7.2 insertDTMF() states:

If a *Playout task* is scheduled to be run; abort these steps; otherwise queue a task that runs the following steps (*Playout task*):

1. If `toneBuffer` is an empty string, fire an event named `tonechange` with an empty string at the`RTCDTMFSender` object and abort these steps.
2. Remove the first character from `toneBuffer` and let that character be tone.
3. Start playout of tone for `duration` ms on the associated RTP media stream, using the appropriate codec.
4. Queue a task to be executed in `duration` + `interToneGap` ms from now that runs the steps labelled *Playout task*.
5. Fire an event named `tonechange` with a string consisting of tone at the `RTCDTMFSender` object.

- This seems to imply the event is only generated with an empty string when the toneBuffer is empty, so insertDTMF("AAB") would result in "A" "A" "B" "".”

Issue 799: Unclear when a DTMFToneChangeEvent is fired with an empty string (cont'd)

- On the other hand, Section 7.3 definition of *tone* states:

"The `tone` attribute contains the character for the tone that has just begun playout (see `insertDTMF`). If the value is the empty string, it indicates that the previous tone has completed playback."

- That seems to imply that an empty string is generated after each tone, so that insertDTMF("AAB") would result in "A" "" "A" "" "B" "".
- PR 807 Proposal: Change Section 7.3 text to "the previous tones have completed playback"

Issue 714/PR 740: STUN/TURN OAuth Token Parameter

- Discussion at August WebRTC WG Virtual Interim
- Next steps?

Stats: Changes since TPAC Sapporo - Oct 2015

- Basic concepts: Unchanged
  - Definition of time for “remote” stats: Changed from remote generation to arrival time
- Individual stats: Expanded
  - IETF XR-block stats added
  - ICE counters added
  - RTT observations
  - Codec implementation
  - One outstanding proposal
- Linkage with WebRTC-PC: Not clarified
  - Text proposal in PR #TODO1
- Conformance requirements: Not clarified
  - Text proposal in PR #TODO2
- Registry maintenance: Awaiting feedback
  - Proposal is PR #43

Stats: Implementation status

- Chrome
  - Implements an older version of the spec - quite inconsistent
  - New implementation is in progress - crbug.com/627816
- Firefox
  - Implements the current API
- Edge
  - Implements the current API, but on different objects than PeerConnection

Stats: Work in progress: New stats

- QP statistic - issue #57
  - Valid for encoded (sender) and decoded (receiver) video frames
  - Codec dependent (by definition)
  - Valuable for monitoring quality vs bandwidth tradeoffs
- Circuit Breaker - issue #61
  - Have 2 out of 3 of the information to implement this.
  - Need the
    - latest timestamp of an RTP packet for each SSRC sent and received
    - Current and average RTCP interval.
- ICE pacing configured values:
  - global pacing limit in Kbps?
  - defaultTa vs chosenTa
  - currentTotalNumIceAgents
  - droppedIceChecksBeforeSend
- The process for new stats includes filing a bug!

Stats: Extension mechanism - issue #25, PR #43

- Consensus model
  - Assumes some forum that's interested enough to comment if controversial
- Living document as registry
  - Presupposes that someone continues to take care of it in perpetuity
  - Recommends dated version if stability is crucial
- Alternatives include:
  - First occupier/implementor owns (no central repository)
  - Using IANA or an IANA-like function with FCFS policy (central repository)
  - IANA, with a more restrictive registration policy
  - Invent something new
- The editors believe the proposed model is the simplest given circumstances.

Stats: WebRTC-PC / WebRTC-Stats split

Issue #561 in webrtc-pc, [#23 in webrtc-stats]

- Having stuff normative in 2 places is unequivocally bad
  - Having copies in 2 places can easily confuse too
- My opinion
  - API belongs in webrtc-pc
  - Object definitions belong in webrtc-stats
  - Conformance belongs in webrtc-pc
  - See PR #TODO1 for a proposal
- This implies defining a language for conformance (#23)
  - Not necessarily formal!
  - See PR #59 for a proposal

Stats: Conformance

Issue #23

- Claim: Conformance is a property of the implementation
  - Therefore, it belongs in webrtc-pc, not webrtc-stats
- Proposal - PR #59:
  - Stats defines that conformance stays with referencing body
  - Stats describes how conformance is described
  - Suggestion: “MUST define object X with properties A, B, C, object Y with properties D, E F….”
  - That is - English with a reasonably strict format, not a formal language

Open issues (other, non-editorial)

- #20 What to do when a function is not supported should be clear
    - “” for strings, 0 and -1 for integers are all legal values under some circumstances.
    - Proposal:
        - Unsupported stats are not present in the dictionary
        - Stats that haven’t started counting are present with a suitable starting value
            - Special considerations when 0 is a suitable value (such as RTT)!
- #26 Behavior of stats on closed, stopped and never-started objects must be clear
    - Proposal: State is frozen on stop. Including timestamp.
    - The last state lasts until the PC it belongs to is gone.
- #49 Should there be an API version flag?
    - Proposal: NO

Issue 295: Guidance for extending objects versus extending Stats (Harald)

- We have only one object model
  - Senders, receivers, transceivers, transports, tracks, and PCs
- Stats reflect one aspect of the object model
  - Focused on recording state-at-a-specific-time
- APIs reflect another view on the same object model
  - Focused on controlling behavior in the future
- When objects overlap, names need to be the same

For Future Work: Define getStats() APIs on objects

- Issue: Recording relationships between objects
  - PC getStats gets “everything”, so consistency not a problem
  - Object getStats may be harder to coordinate

Other issues?

Watch for a new version coming your way shortly!

Friday afternoon

- Certificate & IdP issues (Cullen)
  - #720/PR#738, #678, #555
- Sender/Receiver/Transceiver (Adam, Bernard, Cullen, Peter)
  - #764 (PRs #758+#786), #763, #698(+#578), #624(PR), ~~#305 Done yesterday~~
- ICE (Bernard and Peter)
  - #760, #757/#726
- Stats (Harald and Varun)
- Test suite (Alexandre)
- Summary, conclusions (chairs)

Issue 720/PR 738: Getting the fingerprint of an RTCCertificate

Had a PR 738 to add:

```
readonly attribute RTCDtlsFingerprint fingerprint;
```

Returns fingerprint and hash algorithm using the hash that the cert is signed with. This PR is based on last virtual interim where we thought the cert only had one fingerprint (which matched the one used in the cert signature).

It looks like IETF is moving towards having multiple fingerprint per cert in 4572-update (vs single fingerprint per cert in RFC 4572). JSEP now references 4572-update.

Revised proposal in PR 738:

```
readonly attribute FrozenArray<RTCDtlsFingerprint>
fingerprints;
```

Issue 720/PR 738: Getting the fingerprint (cont'd)

Alternative:

```
partial interface RTCCertificate {
    RTCDtlsFingerprint getFingerprint(DOMString algorithm);

    //algorithm from the hash algorithm registry established in RFC 4572 Section 8
};
```

Problems:

1. Discovering which hash functions are supported.
2. Application uses `Promise<RTCCertificate> generateCertificate(AlgorithmIdentifier keygenAlgorithm)` to generate certificate but needs algorithm to generate the fingerprint.

Support assertions that identify the recipient #678

Waiting for PR from Martin but agreed on path forward a few meetings ago

Sort out requirements around IdpLoginError #555

Need to add IdpLoginError to WebIDL

Need to have way to return loginUrl when user needs to login

Option 1) extend fail case of promise to include loginUrl

Option 2) consider a login failure a “success” of promise and return loginUrl as a success

Proposal:

- Go with option 1.

Issue 764/PR 758/PR 786: How an RtpSender should handle an ended track (Adam)

- PR #758: Updated RTCRtpSender.replaceTrack to allow the script to swap out an ended track
- PR #786: How an RTCRtpSender should treat an ended (or muted) track
- Raised question
  - How should any track consumer see an ended track
- Suggestion
  - An ended track looks, from the consumer point of view, like a muted track
  - Ended means that the track is muted forever
  - Peter: we should say "an ended or muted track is treated the same as a null track" (addTransceiver that has never had a track, for example).

Issue 763: Simulcast Errors (Bernard)

- Today there is no way to determine how many simulcast streams an implementation can send.
  - Do we need RTCRtpCodecCapability.maxSimulcastStreams?
- Also, how can the developer determine what went wrong in sender.setParameters(params)?
  - Currently, only errors are `InvalidModificationError` (for modifying a read-only parameter), `RangeError` (for scaleResolutionDownBy < 1.0)
  - .catch(*error*): Is there any information in *error* on what was wrong with params? (e.g. encodings, headerExtensions, rtcp, codecs, degradationPreference)?
  - May not have called createOffer/Answer() or setLocal/Remote so not necessarily any SDP to reference.

Issue 698: JSEP/WebRTC mismatch on empty remote MID (Adam)

- Description of RTCRtpTransceiver mid attribute talks about a generated value when not provided by remote side
- Reference points to JSEP section 3.5.2.1 (ICE Candidate Format)
  - Lacks any information about mid generation
- Bug filed on JSEP (Issue #325)
  - Fixed by Justin (PR #327)
- We should soon be able to reference JSEP section 5.9 (Applying a Remote Description)

Issue 624 – Upscale Policy

- To recap, some applications don't want the browser to create fake data because
  - A vision algorithm, like QR codes, may have a minimum image size it can use. The browser doubling the size of raw video won't work
  - A audio recognition algorithm may have minimum requirements
  - Current spec does allow upscale but like to add policy to relax that
- Proposal
  - Add a member to RTCRtpParameters dictionary to control policy
  - Current PR adds:
    - `boolean upscaleAllowed = false;`
  - video temporally or spatially, audio sampling rate, number of bits in the audio sample
  - Points out IETF stuff specifies upscale is forbidden, unless policy set to override
  - Like all similar items, we get an error if the negotiated session can't be met

Issues

  - Default to false – Could invert to upscaleNotAllowed
  - One policy or multiple policies (video size, audio rate, audio sample size)
  - Error to fire if negotiated session can't be met
  - Enum or boolean

[Issue 305](): Describe what happens when media changes (Peter)

- RtpSender can already reduce resolution, framerate, and bitrates. And we have consensus on allowing resolution upscale.
- What's left? **aspect ratios.**
- The problem, specifically: The RtpSender configured to send a different aspect ratio than the track provides. Now what?
- Options:
  - don't send
  - crop
  - add black

Recommendation: Allow the RtpSender to crop to fix aspect ratios. If the app doesn't like it, fix the RtpSender (or track). How to crop (center, scale first, etc) is left undefined.

Issue 760: Figure out backward-compatible way to indicate ufrag+mid on end-of-candidates (Peter)

```
dictionary RTCPeerConnectionIceEventInit : EventInit {
    DOMString       ufrag;
    RTCIceCandidate candidate;
    DOMString       url;
};
partial interface PeerConnection {
    Promise<void> addIceCandidate(... candidate, optional DOMString ufrag);
}
```

- PR #819
- Solves #760, #757, #726
- Backwards compatible and simple addition.
- Issue: The app must opt-in. If it doesn't, things won't work perfectly during ICE restarts. But hopefully anyone that knows how to do ICE restarts will know how to pass in the ufrag.

Issue 812: RTCIceGatheringState Definition (Bernard)

- RTCIceGathering state used in both RTCPeerConnection and RTCIceTransport:
  - Section 4.3.2:

```
partial interface RTCPeerConnection : EventTarget {
    readonly        attribute RTCIceGatheringState      iceGatheringState;
                    attribute EventHandler              onicegatheringstatechange;
};
```

  - Section 5.5:

```
partial interface RTCIceTransport {
    readonly        attribute RTCIceGatheringState gatheringState;
};
```

- RTCIceGatheringState definition in Section 4.4.2:

`New:` The object was just created, and no networking has occurred yet.

`Gathering:` The ICE agent is in the process of gathering candidates for this RTCPeerConnection.

`Complete:` The ICE agent has completed gathering. Events such as adding a new interface or a new TURN server will cause the state to go back to gathering.

[PR 818](): RTCIceGatheringState Cleanup (Peter)

Proposal:

- Section 5.5:

```
partial interface RTCIceTransport {
    readonly        attribute RTCIceGathererState gatheringState;
};
```

- Definition of RTCIceGathererState:

`New:` The RTCIceTransport was just created, and has not started gathering candidates yet.

`Gathering:` The RTCIceTransport is in the process of gathering candidates.

`Complete:` The RTCIceTransport has completed gathering and the end-of-candidates indication for this transport has been sent. It will not gather candidates again until an ICE restart causes it to restart.

- Updated definition of RTCIceGatheringState:

`New:` Any of the RTCIceTransports are in the new gathering state and none of the transports are in the gathering state, or there are no transports.

`Gathering:` Any of the RTCIceTransports are in the gathering state.

`Complete:` The RTCIceTransports have completed gathering and at least one of them is in the completed gathering state.

Issue 808: Can the ICE gathering state go from completed to gathering without an ICE restart? (Peter)

- Definition of "complete" in Section 4.4.2 (RTCIceGatheringState):

Events such as adding a new interface or a new TURN server will cause the state to go back to gathering.

- Huh??
- PR 818: iceTransport.iceGatheringState and pc.iceGatheringState can only transition from "complete" to "gathering" via an ICE restart.

State Of Test Suite webrtc 1.0

Dr. Alex Gouaillard

Cosmo Software Consulting

Satellite view

Specification	TR Working draft	Stable draft (Last Call)	Implementors feedback (CR)	Standard (Rec)	Test Suite
WebRTC 1.0: Real-time Communication Between Browsers	28-Feb-16	N/A	Q2 2016?		Y
Media Capture and Streams (getUserMedia)	12-Feb-15	14-Apr-15	19-May-16		Y
MediaStream Recording	8-Sep-15	N/A			**N**
MediaStream Image Capture	27-Jan-15	N/A			**N**
Media Capture Depth Stream Extensions	26-Feb-16	N/A			**N**
Media Capture from DOM Elements	19-Feb-15	N/A			**N**
Audio output devices API	10-Feb-15	N/A			**N**
Identifiers for WebRTC's Statistics API	6-Feb-15	N/A			**N**
Screen Capture	10-Feb-15	N/A			**N**

APIs View

webrtc spec	IsTested	Only IDL?	comments
PeerConnection	Y	Y	
RTP Media API (send/recr/trans)	Not Really	Y	only PC interface ext
P2P Data	Not Really	Y	only PC interface ext
DTMF	Not Really	Y	only PC interface ext
Stats	Not Really	Y	only PC interface ext
Identity	Not Really	Y	only PC interface ext
MS ext. for Network Use	Not Really	Y	only PC interface ext

WebRTC Testing SOA

Dr Alex Gouaillard

Cosmo Software Consulting

Manual Single page tests

web-platform, adapter.js, whatever

https://github.com/w3c/web-platform-tests

https://github.com/webrtc/adapter

Canary	Chrome	Opera	Dev	Firefox	Bowser	Edge	Safari	Nightly
50	50	50	49	49	-	47	0	4

Automated Single page tests

web-platform, adapter.js, whatever

https://github.com/webrtc/adapter/tree/master/test

Automated Single page tests

how easy would it be to extend the automation?

Easy		Med.	Easy		Hard	Medium		
?	50	?	?	49	?	?	?	?
Canary	Chrome	Opera	Dev	Firefox	Bowser	Edge	Safari	Nightly

Easy	Just add the browser binary on the (virtual) machine
Med.	Need a new web-driver, and the browser binary.
Hard	Need everything, and it's on a mobile OS !

Automated Single page tests

WIP (AFAIK)

			Nils			IMTC	Alex	
?	50	?	?	49	?	?	?	?
Canary	Chrome	Opera	Dev	Firefox	Bowser	Edge	Safari	Nightly

Automated Single page tests

web-platform, adapter.js, whatever

Canary	Chrome	Opera	Dev	Firefox	Bowser	Edge	Safari	Nightly
?	50	?	?	49	?	?	?	?

Wait, automated …

on Debian ONLY !!

Manual Single page tests

web-platform, adapter.js, whatever

	Canary	Chrome	Opera	Dev	Firefox	Bowser	Edge	Safari	Nightly
	?	?	?	?	?	-	-	-	-
iOS	?	?	?	?	?	?	-	?	?
	?	?	?	?	?	-	-	0	4
	?	?	?	?	?	-	47	-	-
debian	50	50	50	49	49	-	-	-	-

Automated Single page tests

WIP (AFAIK)

	Canary	Chrome	Opera	Dev	Firefox	Bowser	Edge	Safari	Nightly
	?	?	?	?	?	-	-	-	-
iOS	?	?	?	?	?	?	-	?	?
	?	?	?	?	?	-	-	Alex	
	?	IMTC	?	?	IMTC	-	IMTC	-	-
debian	?	50	?	Nils	49	-	-	-	-

Intermediate Conclusion (1)

- With webrtc, single-page show their limit
  - 2 PC objects in one page
- One browser testing is not enough
  - 2 PC objects in two tabs

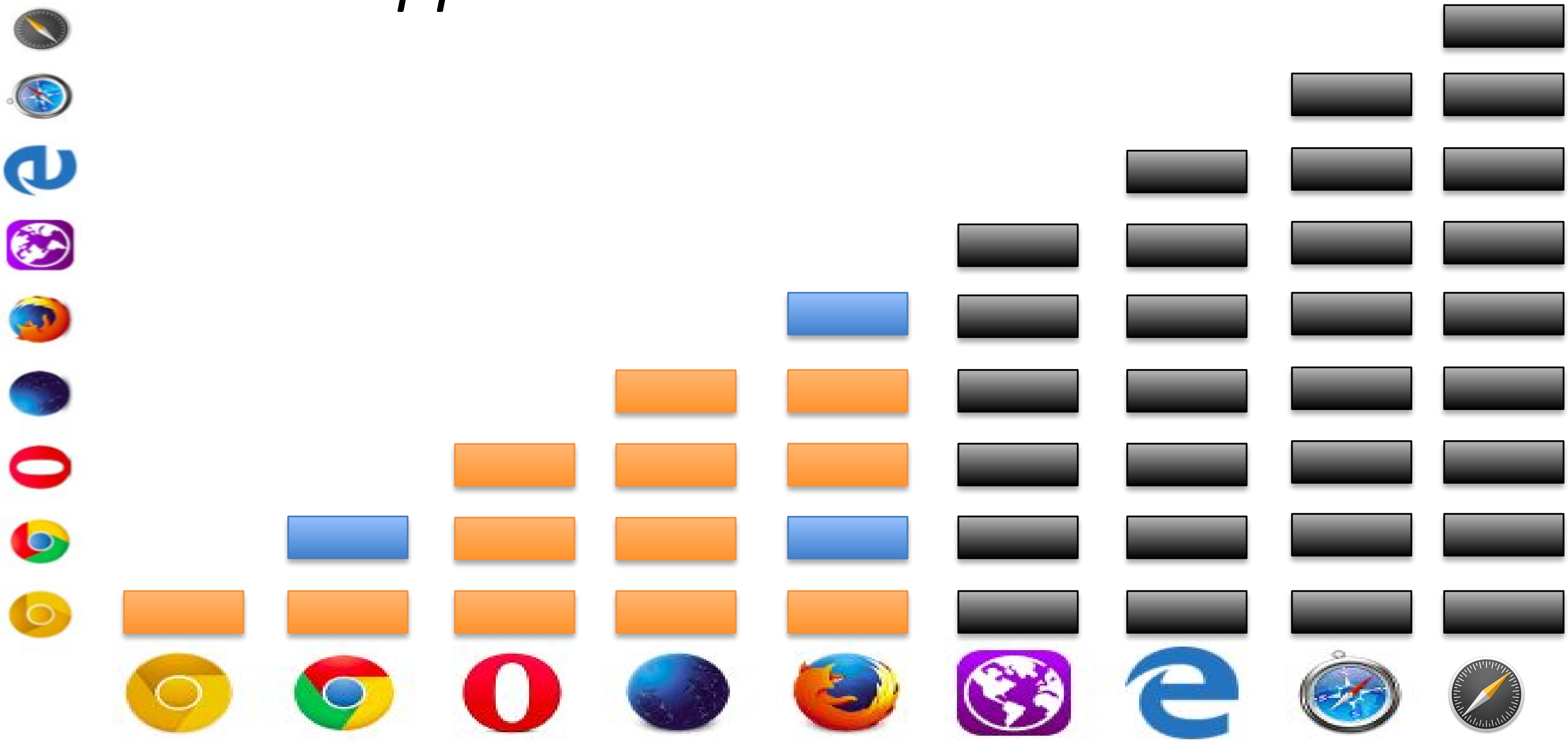
- Next step is to test interoperability between two separate browsers
  - Possibly with proxy, NAT, in-between

- Original, single-machine, single OS, interop code part of adapter.js test suite is a good start, but need support for more browsers.

- IMTC sponsors the work to add edge browser support, and full

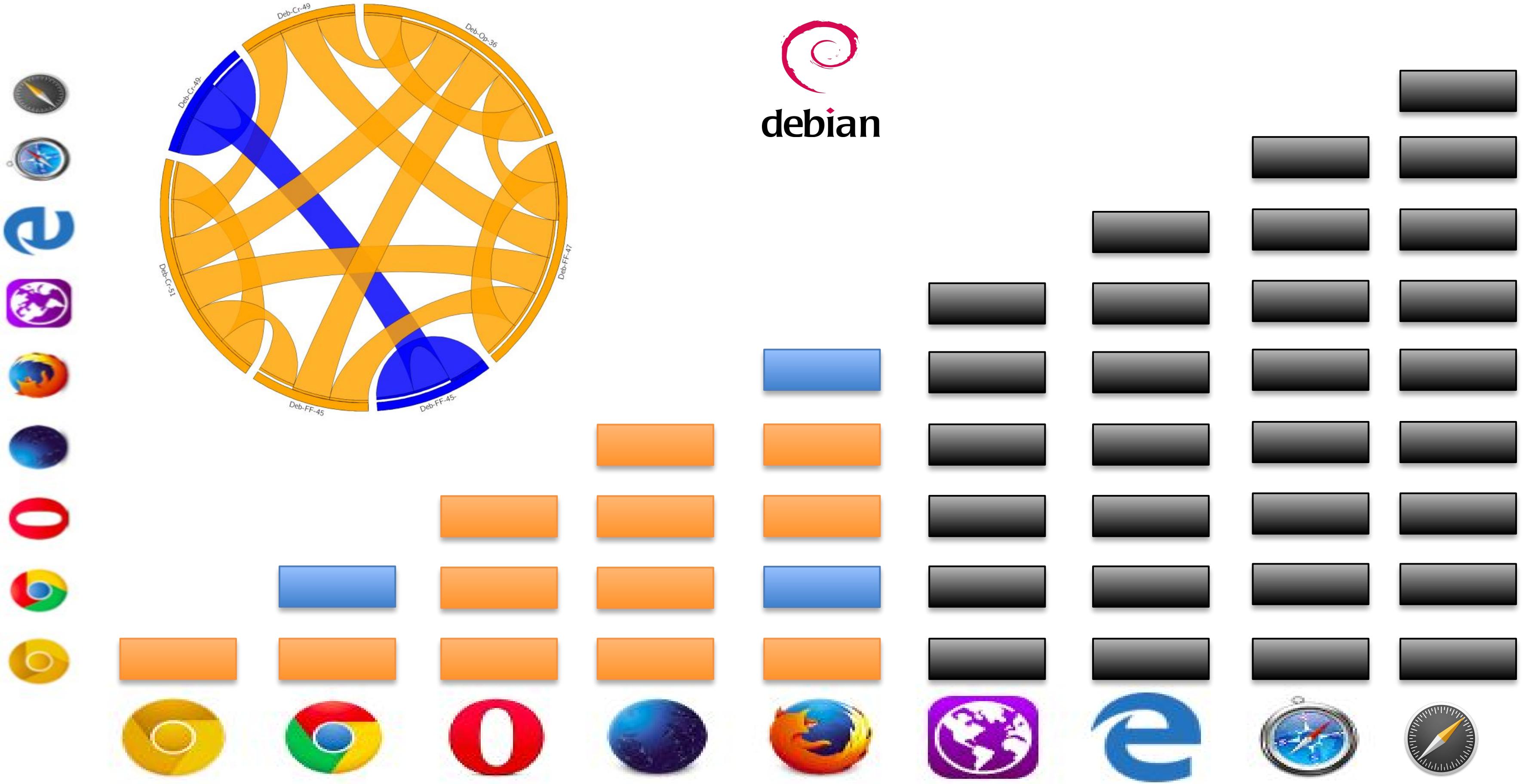
Interop Tests (Debian)

appRTC Hosted + raw local



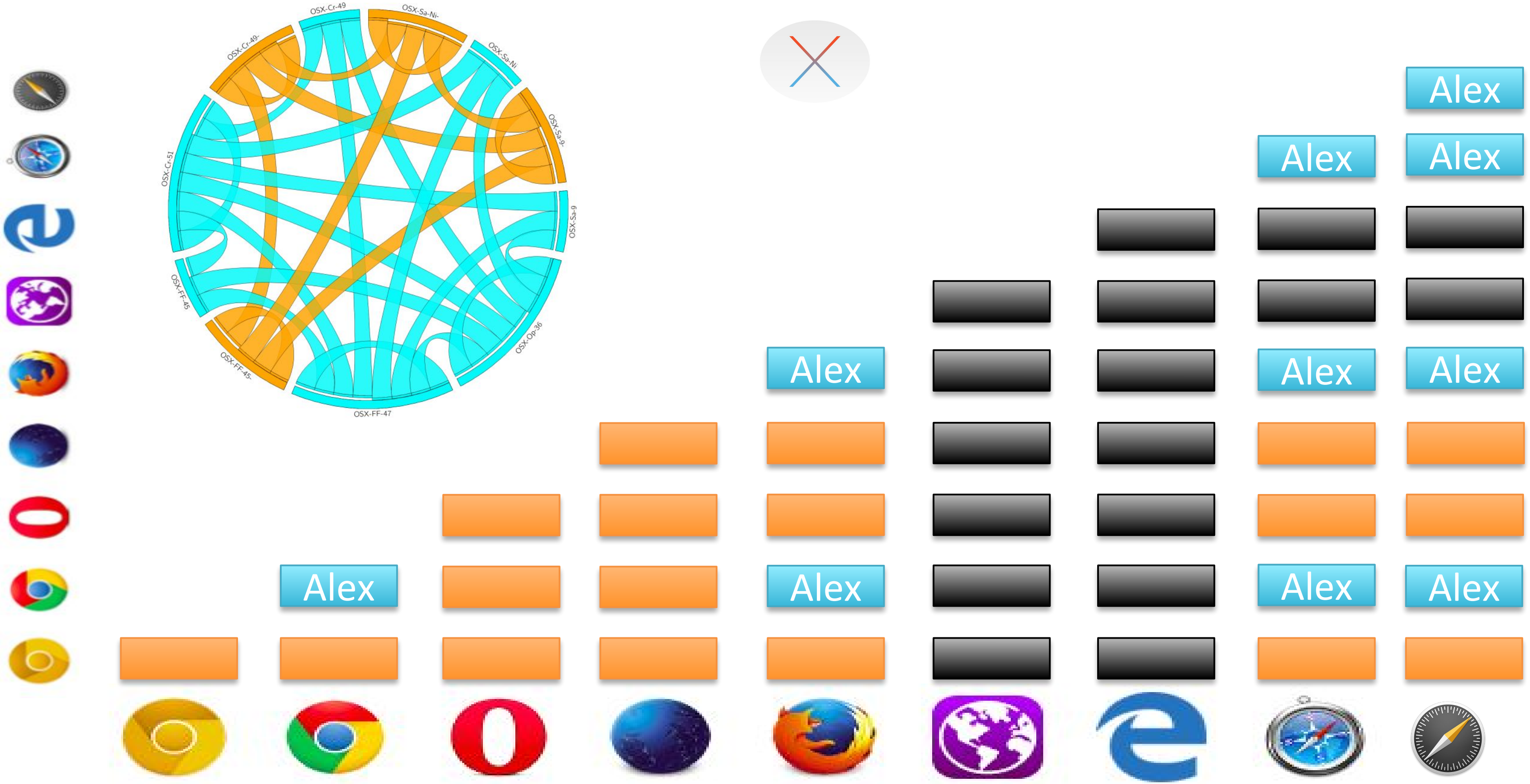
Interop Tests (Debian)

Interop Tests (Debian)

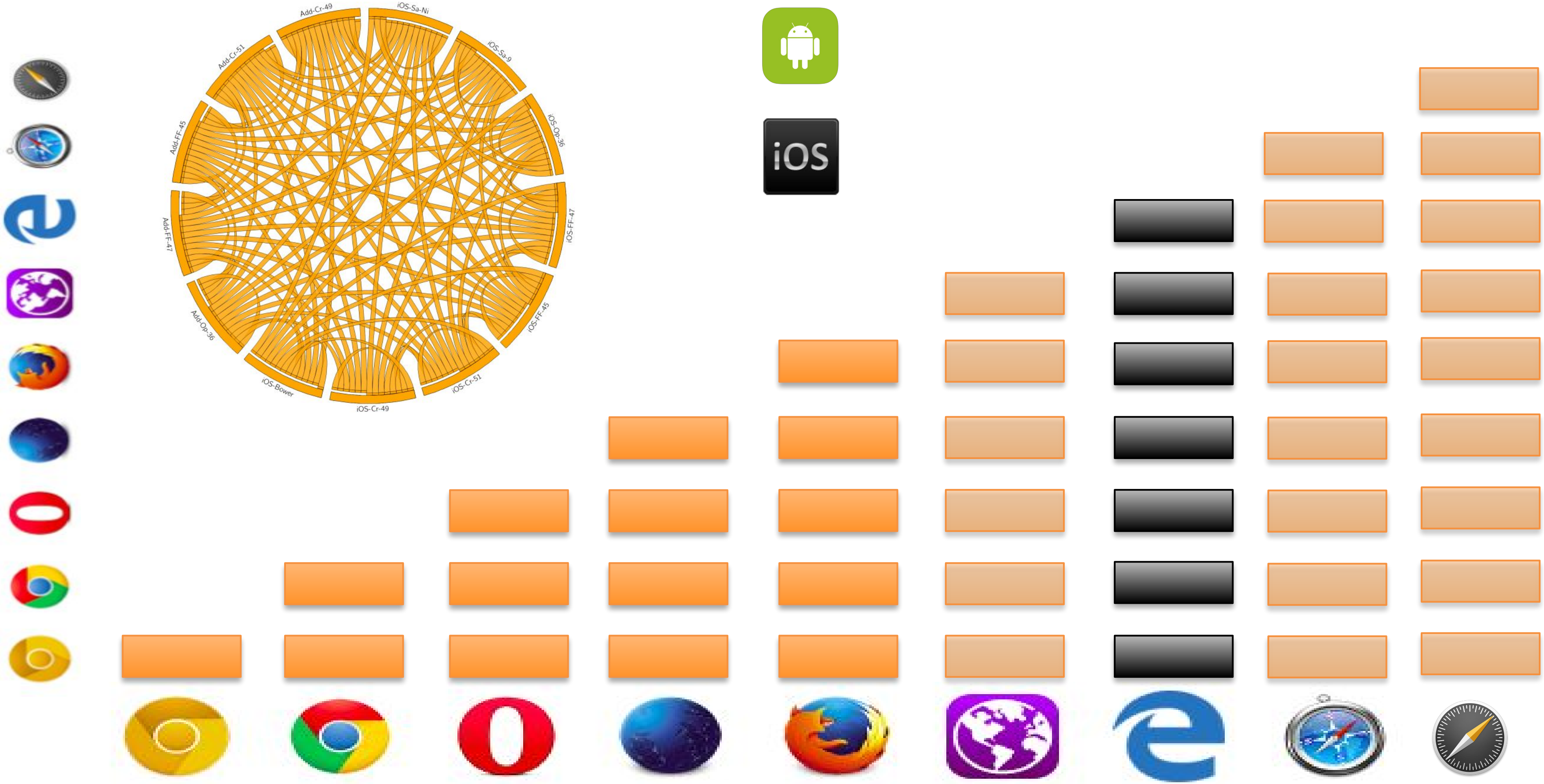


Interop Tests (Win10)

Interop Tests (MacOSX)



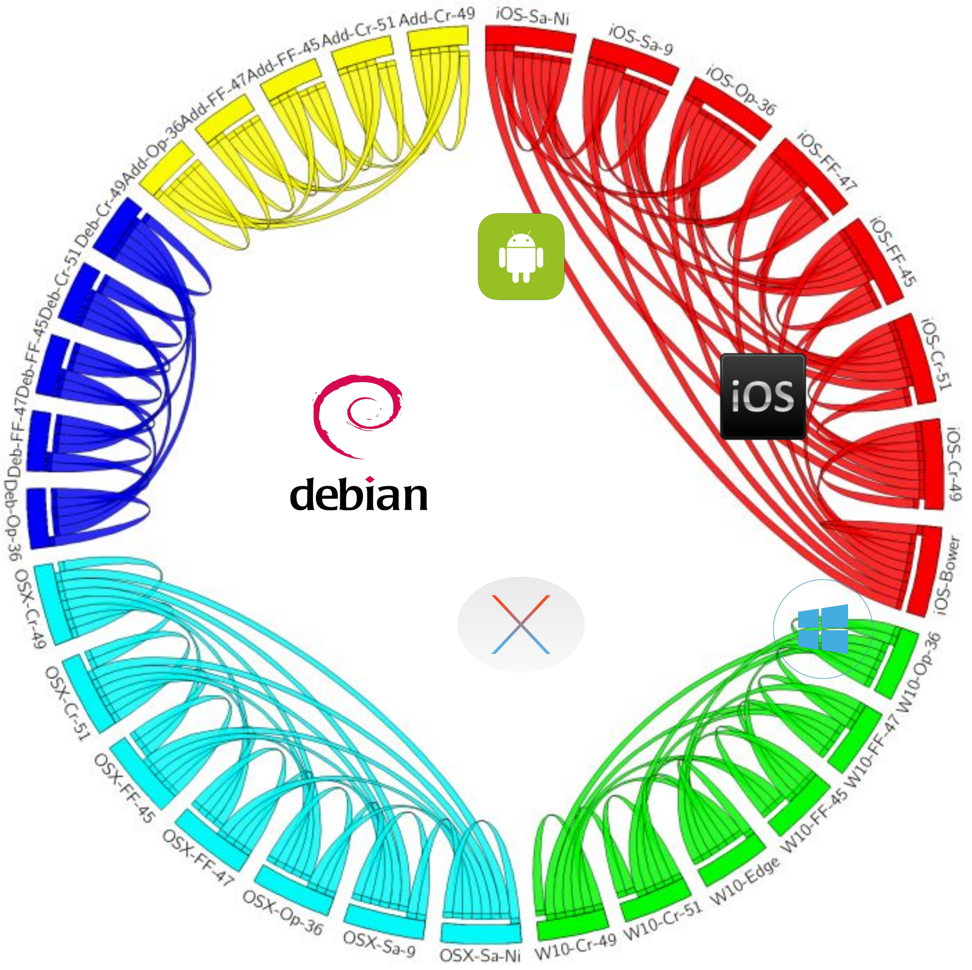
Interop Tests (Android and iOS)



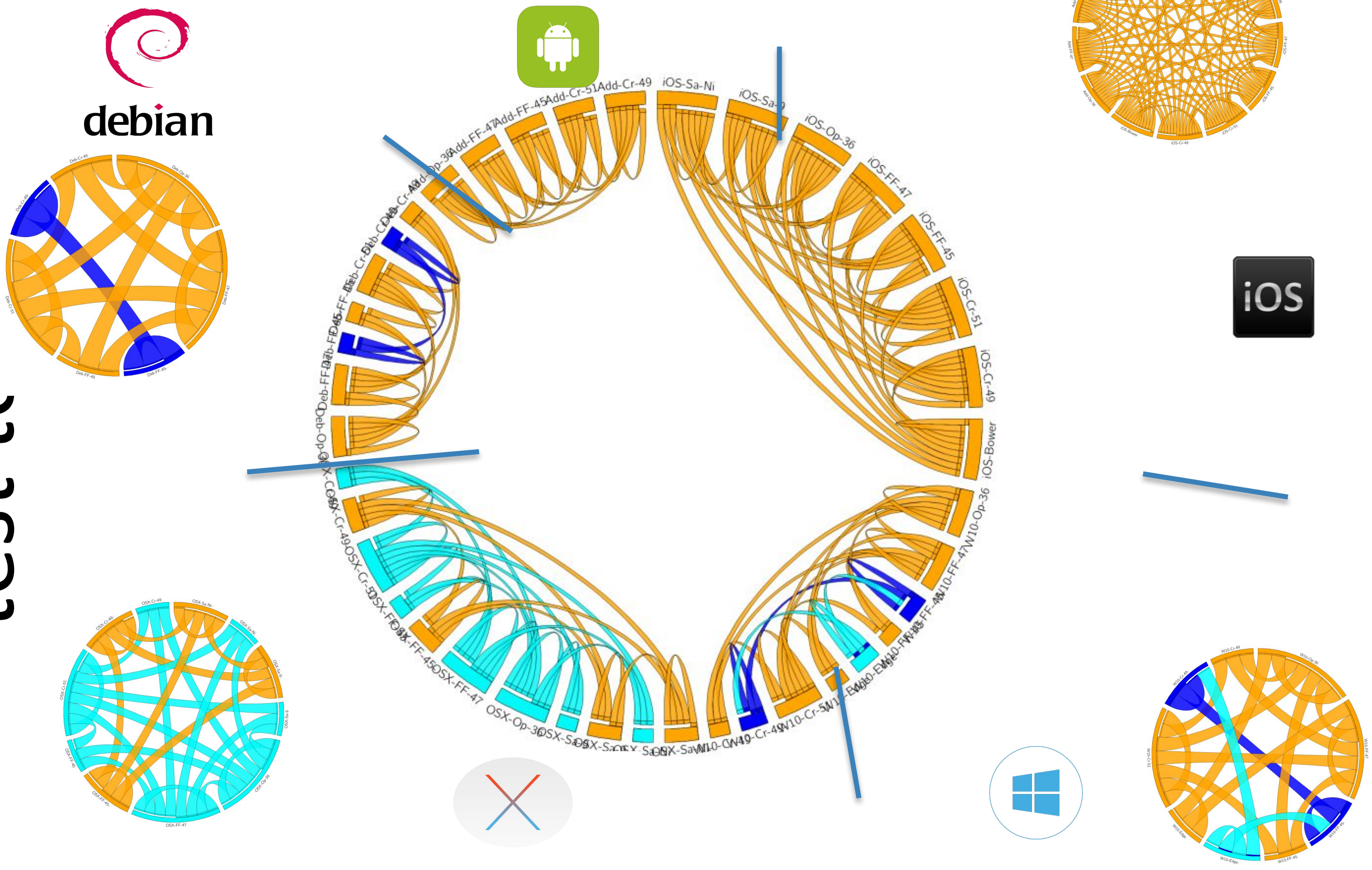
Intermediate Conclusion (2)

- Powerpoint is not the good tool for multidimensional data visualization ☺
- The real interesting cases are not covered today:
  - Desktop to mobile
  - Cross OS: Chrome on Mac against Edge on Windows
- How to quantify what is done, and what is left TBD?

What we **COULD** test today



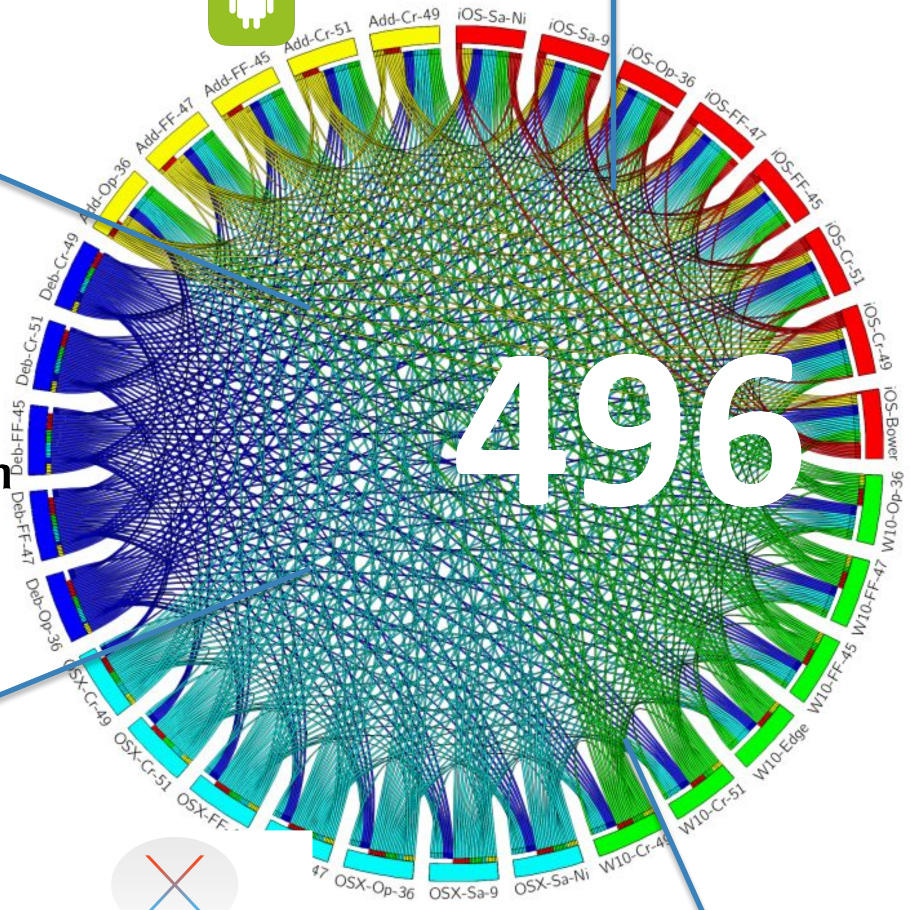
What we **ACTUALLY** test today



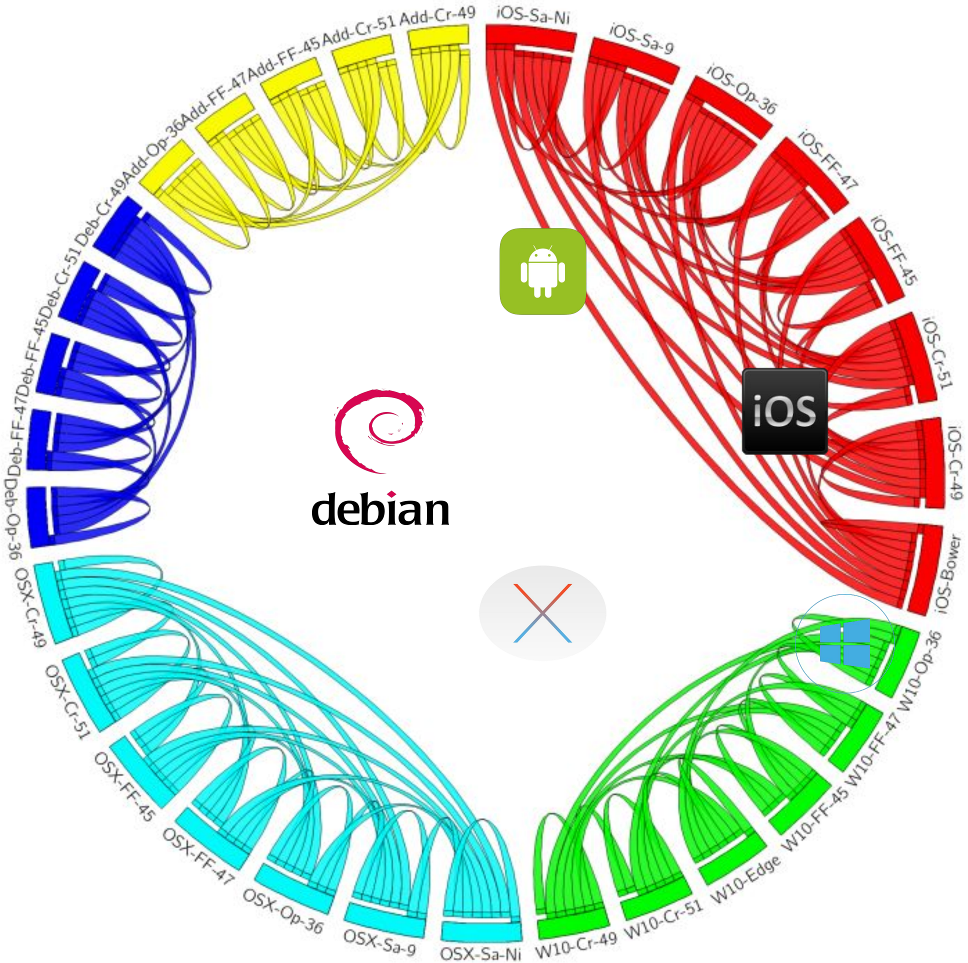
Intermediate Conclusion (3)

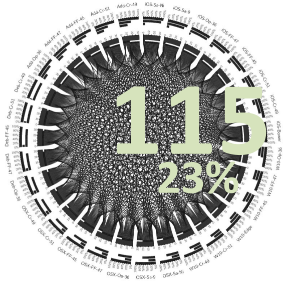
- State of the (open source) Art is two browsers running the same tests on the same OS.
- How to quantify what is done, and what is left TBD?

What we WANT to test.

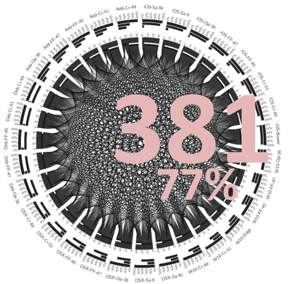


What we could test today





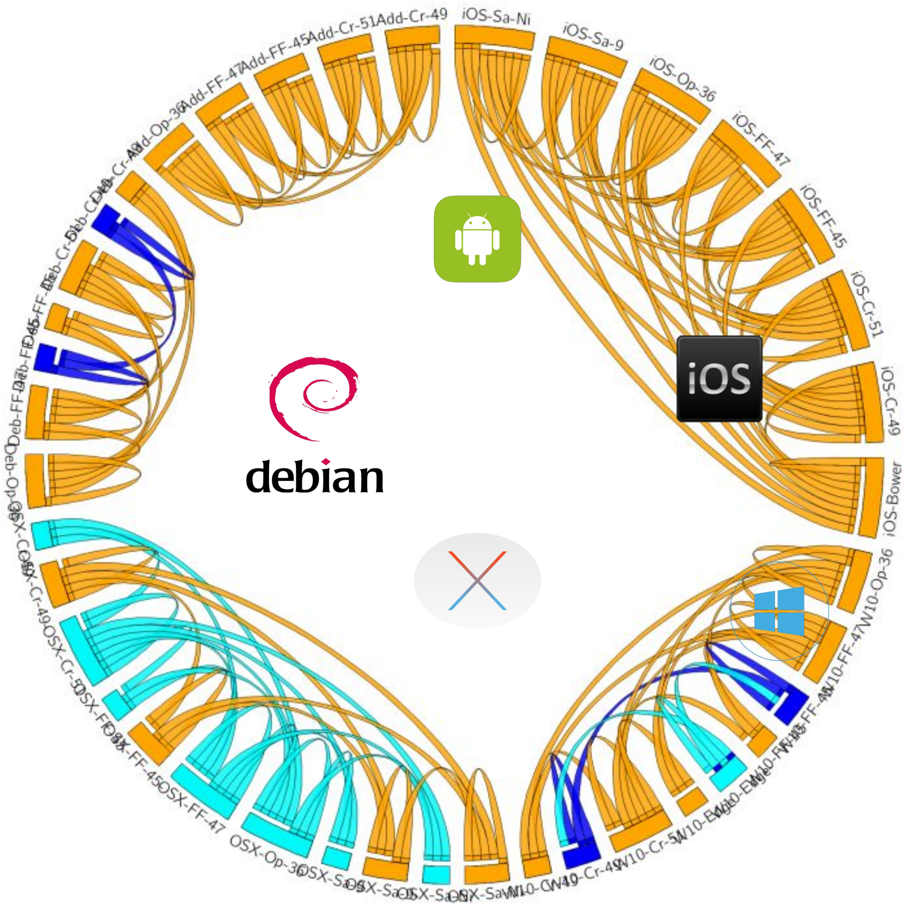
1 OS

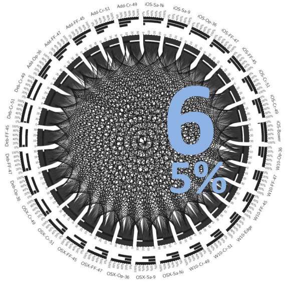


2 Oses

(Not tested)

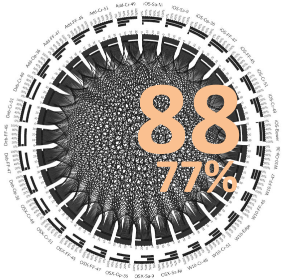
What we ACTUALLY test today





Tested

WIP



Not tested

2016 old RoadMap (march 2016)

- Improve W3C tests
- Additional browser support in Adapter.js
- new web drivers to the interop test suite.
  - New desktop Browsers, and mobile browsers.
- Improve webdrivers
  - security prompt support
  - Safari / GTK+ / webkit support
- Add a Conductor for cross-browser interop tests
- Add appRTC standalone support
- Add a SIP interop app to the suite (IMTC scope)

2016 old RoadMap (march 2016)

- Improve W3C tests
- Additional browser support in Adapter.js
- new web drivers to the interop test suite.
  - New desktop Browsers, and mobile browsers.
- Improve webdrivers
  - security prompt support
  - Safari / GTK+ / webkit support
- Add a Conductor for cross-browser interop tests
- Add appRTC standalone support (not Edge ☹)
- Add a SIP interop app to the suite (IMTC scope)

- Test suite (Alexandre)

Content goes here

trickle ICE w/ufrag attempt #2

```
dictionary RTCIceCandidatesEnded {
    DOMString?      sdpMid;
    unsigned short? sdpMLineIndex;
    DOMString       ufrag;
}
dictionary RTCIceCandidate candidate {
    DOMString ufrag;
    ...
};
partial interface PeerConnection {
    Promise<void> endIceCandidates(RTCIceCandidatesEnded end);
    attribute EventHandler onicecandidatesended;
}
```

- Doesn't break anything
- Clean; nothing hacky
- Apps opt-in

trickle ICE w/ufrag examples

Regular candidate

```
{
  candidate: {
    ufrag: "ufrag",
    sdpMid: "audio",
    candidate: "candidate:..."
  },
}
```

```
.addIceCandidate(
  {sdpMid: "audio", ufrag: "ufrag",
   candidate: "..."});
```

End of candidates event:

```
{
  sdpMid: "audio",
  ufrag: "ufrag"
}
```

```
.endIceCandidates(
  {sdpMid: "audio", ufrag: "ufrag"});
```

or this?

```
Regular candidate:
{
  sdpMid: "audio"
  candidate: "candidate:... ufrag 1111"
}

End of candidates:
{
  sdpMid: "audio"
  // The last candidate or bogus candidate?
  candidate: "candidate:... ufrag 1111 trickle done"
}
```

- Happens automatically
- Hacky
- Might break stuff?

Summary and conclusions

Content goes here

Thank you

Special thanks to:

W3C/MIT for WebEx

WG Participants, Editors & Chairs

BACKUP SLIDES

These slides need to be moved to their appropriate agenda section.

WebRTC

- **Pull Requests**
  - Issue 720/PR 738: Getting the fingerprint of an RTCCertificate (Bernard)
  - Issue 726/PR 757: Add ufrag attribute to RTCIceCandidate (Cullen Jennings)
  - Issue 764/PR 758: How an RtpSender should handle an ended track (Adam)
- **Issues**
  - Issue 295: Guidance for extending objects versus extending Stats (Harald)
  - Issue 305: Describe what happens when media changes (Cullen Jennings)
  - Issue 526: NetworkError event is not defined (Adam)
  - Issue 555: Sort out requirements around IdpLoginError (Martin Thomson)
  - Issue 561: Normatively site webrtc-stats for Section 8.X (Harald)
  - Issue 624: Upscale Policy (Stefan?)
  - Issue 727: removeTrack: throw exception if sender is not in set of senders (Taylor)
  - Issue 746: What happens if createDataChannel is called with an invalid id (Taylor)
  - Issue 763: Simulcast Errors (Bernard)
  - Issue 782: pc.createOffer(), pc.addTrack(track) should not include track (Adam)
  - Issue 787: Integrate RTCRtpTransceiver into setlocal/remote steps (Adam)