W3C WebRTC/MediaCapture WG Meeting

April 12, 2018
8 AM Pacific Time

Chairs: Stefan Hakansson
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 interim meeting of the W3C WebRTC WG!
- During this meeting, we hope to:
    - Discuss updates to the WG Charter
    - Make progress on open issues in webrtc-pc and screen capture
- Editor’s Draft updates to follow meeting

June f2f (Stockholm, Sweden)

Doodle: https://doodle.com/poll/xmu3fqb3mqtkcnd2

Proposal: go with June 19-20. Any objection?

TPAC (Lyon, France)

Proposal: we meet Monday October 22 and Tuesday the 23rd of TPAC week (has been on list)

About this Virtual Meeting

Information on the meeting:

- Meeting info:
  - https://www.w3.org/2011/04/webrtc/wiki/April_12_2018
- Link to latest drafts:
  - https://w3c.github.io/mediacapture-main/
  - https://w3c.github.io/webrtc-pc/
  - https://w3c.github.io/mediacapture-screen-share/
  - https://w3c.github.io/webrtc-stats/
- 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

For Discussion Today

- **WebRTC WG Charter**
- **WebRTC-PC**
  - **Issue 1813: Simulcast makes no sense for Audio (Bernard)**
  - **Issue 1174: ssrc in RTCRtpEncodingParameters (Bernard)**
  - **Issue 1706: Should rollback fire addtrack/removetrack events? (Jan-Ivar)**
  - **Issue 1775: RTCPeerConnection lacks identity marker beyond current process (Harald)**
  - **Issue 1826: No way to add stream associations to a transceiver (Jan-Ivar)**

For Discussion Today (cont’d)

- mediacapture-screen-share Issues
  - Issue 29: Powerpoint is special (Suhas)
  - Issue 31: Define behavior of existing constraints in screen sharing (Jan-Ivar)
  - Issue 39: Non-top-level browsing contexts (Jan-Ivar)
  - Issue 43: Disable local playback during audio sharing (Martin)
  - Issue 49: Bring back constraints for downscaling (Jan-Ivar)
  - Issue 51: Browser tab sharing (Suhas)
  - Issue 53: Clarify ‘origin’ concept (Stefhak)
  - Issue 55: Why does getDisplayMedia live on navigator and not navigator.mediaDevices? (Suhas)

WebRTC WG Charter (Chairs)

- Current charter has been extended until May 2018.
- Draft Charter: https://w3c.github.io/webrtc-charter/webrtc-charter.html
  - Extends charter to March 31, 2020.
- Changes based on feedback from the last virtual interim:
  - Liaisons
    - Added IETF ICE WG
  - Success criteria
    - Added data transfer
  - Feature removal policy
    - Features not implemented by at least two browsers can be separated out into an extension document, or dropped.
  - Timeline:
    - WebRTC-PC: Updated CR (Q4 2018), PR (Q3 2019), Recommendation (Q4 2019)
    - WebRTC-Stats: CR (Q2 2018), PR (Q3 2019), Recommendation (Q4 2019)

For Discussion Today

- **WebRTC-PC**
  - **Issue 1813: Simulcast makes no sense for Audio (Bernard)**
  - **Issue 1174: ssrc in RTCRtpEncodingParameters (Bernard)**
  - **Issue 1706: Should rollback fire addtrack/removetrack events? (Jan-Ivar)**
  - **Issue 1775: RTCPeerConnection lacks identity marker beyond current process (Harald)**

Issue 1813: Simulcast makes no sense for Audio (Bernard)

- Currently, no browser supports simulcast audio.
  - Question 1: Is support for simulcast audio required?
    - Redundant Audio Coding (RED) is superior to simulcast for audio:
      - Allows for multiple encodings (e.g. Opus at multiple bit rates) with less overhead than simulcast.
      - Can also protect against burst loss (not possible with Opus FEC)
  - Question 2: How should addTransceiver behave when asked to support (too many) simulcast encodings? Suggestion:
    - TypeError in `addTransceiver` on `sendEncodings.length >` `MAX_SUPPORTED` (0 or 1 for audio)

Issue 1174: ssrc in RTCRtpEncodingParameters (Bernard)

- Problem: WebRTC 1.0 compatibility with SFUs
    - “Plan B” widely used with SFUs supporting SSRC-multiplexing.
        - SFUs supporting signaling of SSRCs currently ignore RID header extensions.
        - SFUs need a way to obtain SSRCs in order to transition to addTransceiver/Unified Plan.
- How to enable WebRTC 1.0 applications to signal SSRCs to an SFU?
    - PR 1531 removed SSRC support from RTCRtpEncodingParameters.
    - Alternative: Assume that SSRCs are surfaced in encodings[].ssrc, encodings[].rtx.ssrc and encodings[].fec.ssrc.

Issue [1706](#): Should rollback fire events? (jib)

- Rollback = “set things back” / undo SDP.
- `SRD(offer_with_new_track)` causes `stream.addTrack(receiver.track)` to happen and fires `addtrack` and `track` events.
- Should `SRD(rollback)` of that cause `stream.removeTrack(receiver.track)` and fire `removetrack` and `mute` events?
- `SRD(offer_sans_track)` causes `stream.removeTrack(receiver.track)` to happen and fires `removetrack` and `mute` events.
- Should `SRD(rollback)` of that cause `stream.addTrack(receiver.track)` and fire `addtrack` and `track` events?
- Yes, if intent is to have things function normally after rollback only.

Issue 1775: RTCPeerConnection lacks identity marker beyond current process (Harald)

- Problem identified in practice: Talking about a PC outside its JS context
  - JS objects have no intrinsic “identity”
  - Mapping tables were not feasible (two APIs in use)
- Discussed a dozen ways to “hide” an identifier
  - Extra stats objects, magic tokens in APIs, non-standard APIs
  - Decided to try the simplest thing and standardize it
- Alternatives (seen from implementation team)
  - Have the standard adopt an “ID” attribute
  - Use a non-standard way to label PeerConnections
- Benefits
  - Solves the problem cleanly and fully
- Costs?

Issue 1826: No way to add stream associations to a transceiver (jib)

- Transceivers only let us answer with stream-less tracks (see blog).
- addTrack works but locks in streams in getTransceivers() order.
- Need general way to add a track w/stream to any given transceiver.
- Proposal:

```
await pc.setRemoteDescription(description);
let transceiver = foo(pc);
transceiver.direction = "sendrecv";
await transceiver.sender.replaceTrack(stream.getVideoTracks()[0]);
transceiver.sender.setStreams([stream]);
await pc.setLocalDescription(await pc.createAnswer());
```

- Would trigger negotiationneeded in “stable” signalingState.

For Discussion Today (cont’d)

- mediacapture-screen-share Issues
    - Issue 29: Powerpoint is special (Suhas)
    - Issue 31: Define behavior of existing constraints in screen sharing (Jan-Ivar)
    - Issue 39: Non-top-level browsing contexts (Jan-Ivar)
    - Issue 43: Disable local playback during audio sharing (Martin)
    - Issue 49: Bring back constraints for downscaling (Jan-Ivar)
    - Issue 51: Browser tab sharing (Suhas)
    - Issue 53: Clarify ‘origin’ concept (Stefhak)
    - Issue 55: Why does getDisplayMedia live on navigator and not navigator.mediaDevices? (Suhas)

Status of Screen Capture

- **12 open issues**
- **8 for discussion today**

Issue 29: Powerpoint is special (Suhas)

- Applications that go full-screen might need special treatment

PR has been created:

https://github.com/w3c/mediacapture-screen-share/pull/57

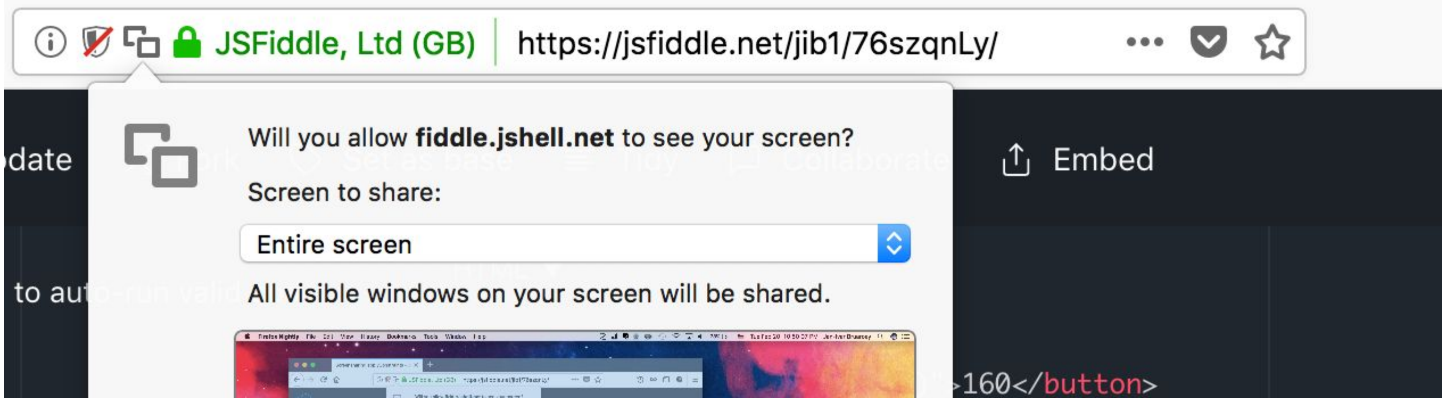
Next Steps: Address review comments from mt.

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

- Down-scaling is critical. 2880 x 1800 x 60 fps = too rich for uplink.
- Usage: `track.getSettings(); track.applyConstraints();`
- Questions:
  - What to support? All vs. explicit list: `width`, `height`, `frameRate`
  - Crop vs. no crop (all settings dictionaries have same aspect)?
- No cropping: `aspectRatio` and `resizeMode` become redundant.
  - Informative value? `let {aspectRatio} = track.getSettings();`
- Cropping gets complicated. Low value.
- Proposal: explicit list, no cropping.

Issue 39: Screen-sharing from iframes (Jan-Ivar)

- Feature-policy: <iframe allow="camera; microphone; *screen*"> ?
- Use-case: Outsourced “customer support” sandboxed service.
  - Disallow by default? (consistent with getUserMedia)
  - Disallow always? (because of getDisplayMedia attack risk)
    - Difficulty communicating about iframe origin:



JSFiddle, Ltd (GB) https://jsfiddle.net/jib1/76szqnLy/

Will you allow **fiddle.jshell.net** to see your screen?

Screen to share:

Entire screen

All visible windows on your screen will be shared.

Issue 43: Disable local playback during audio sharing (Martin)

- PR: https://github.com/w3c/mediacapture-screen-share/pull/44
- Use case is to be able to capture from a device and play audio remotely (think big screen projection)
- Recommendation: We shouldn’t do this. This overrides both user and origin preferences about audio playback in a non-transparent way. The use case would be more easily handled with audio playback devices, or a tab-level mute.

Issue 49: Bring back constraints on getDisplayMedia (jib)

- Spec says no constraints on `getDisplayMedia`, just afterwards:

```
let stream = await navigator.getDisplayMedia({video: true});
let track = stream.getTracks()[0];
await track.applyConstraints({width: 640, frameRate: 5}); // OK
video.srcObject = stream;
```

- Prevents websites from influencing user’s source selection.
- But we could have done that with prose:
  - “UAs are restricted from using constraints to influence the end-user choice of what to share.”
- Encourage downscaling & avoid temporary 2880x1800x60 tracks:

```
video.srcObject = await navigator.getDisplayMedia({video: {
    width: 640, frameRate: 5
}});
```

Issue 49: Bring back constraints on getDisplayMedia (jib)

- But, a problem, since UAs can't limit selection:

```
try {
  await navigator.getDisplayMedia({video: {width: {min: 640}}});
} catch (e) {
  // Bad UX: User selects 320x200 window followed by OverconstrainedError!
}
```

- Solution: Ban or ignore min, max and exact in getDisplayMedia()

  a. Ban (always reject with OverconstrainedError without prompt)

  b. Ignore (min, treat exact and max like ideal, and succeed)

- Keep track.applyConstraints() as is?

Issue 51: Browser tab sharing (Suhas)

Option - 1

- Define a new DisplayCaptureSurfaceType: “active-window”
- “active-window” is the current active display area (say the current browser tab) that triggered the getDisplayMedia()
- List “active-window” information in the UX selection list for sharing.
  - This avoids listing all the tabs otherwise and bad UX.

Option - 2 (mt)

- This is user design issue. Probably having UX list all the tabs with current tab listed first is all we need.

Proposal - Pick option 2, no change to specification is needed. OK ?

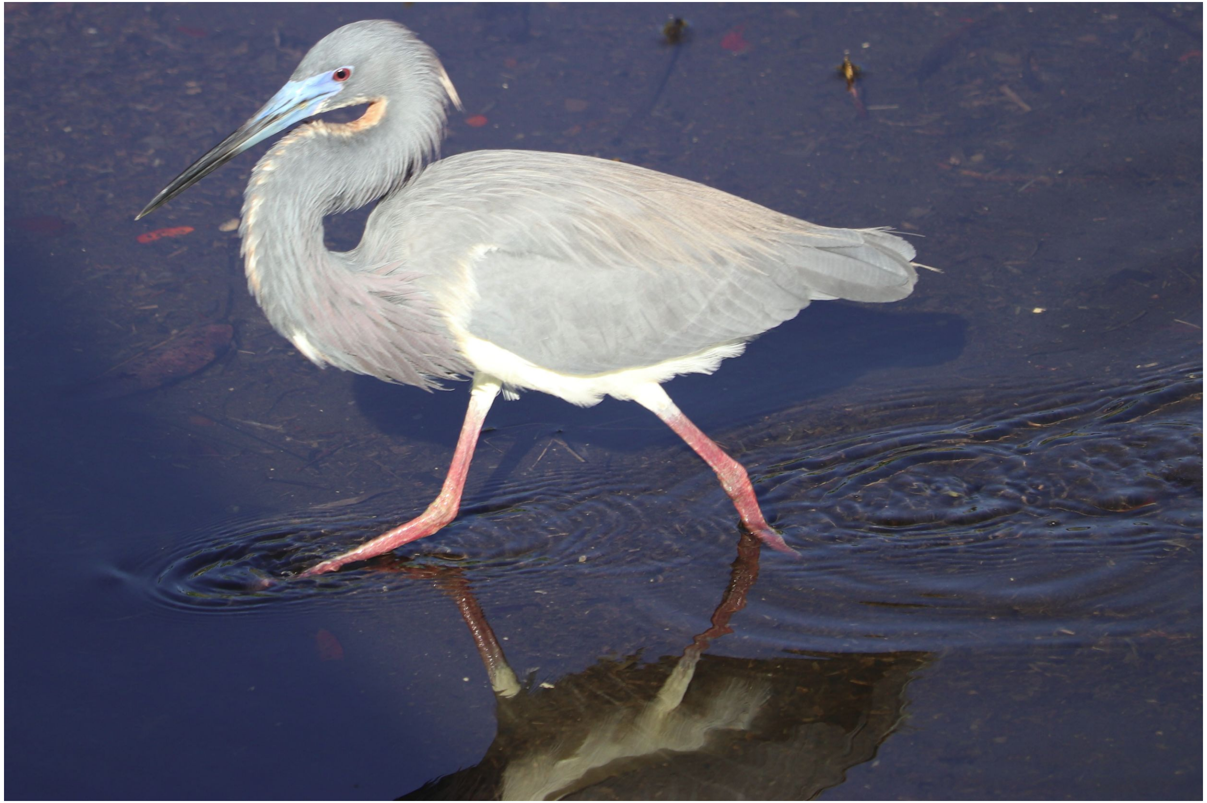
Issue 53: Clarify ‘origin’ concept (Stefhak)

- From PING review:
  - "From Element has its own developed privacy considerations section, particularly focused on origin separation. Is it expected that handling origin separation is to be handled by implementors or is there best practices or implementation guidelines that will be provided? If the latter, is Feature Policy an applicable solution here?"
- Proposal from Martin T: ‘We should add: "Media that is acquired by a given origin is only available to that origin."’
- OK?

Issue 55: Why does getDisplayMedia live on navigator and not navigator.mediaDevices? (Suhas)

- Discussed with mt , proposal is to move it (back) to navigator.mediaDevices.
- OK?

For extra credit



Name that bird!

Thank you

Special thanks to:

W3C/MIT for WebEx

WG Participants, Editors & Chairs

The bird