W3C WebRTC/MediaCapture WG Meeting

February 21, 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 media capture, screen capture, webrtc-stats and webrtc-PC
- Editor’s Draft updates to follow meeting

About this Virtual Meeting

Information on the meeting:

- Meeting info:
  - https://www.w3.org/2011/04/webrtc/wiki/February_21_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
- Content hints
  - Issue 478: Content hints (Thomas Daede and Mo Zanaty)
- 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)

For Discussion Today (cont'd)

- **WebRTC-Stats**
  - **Report on dead statistics**
- **WebRTC-PC**
  - **Issue 1694/1739: RTCCertificate backup/Private Key Access (Bernard)**
  - **Issue 1706: Should rollback fire addtrack/removetrack events? (Jan-Ivar)**
  - **Issue 1756: Need to clear [[AssociatedMediaStreams]] in RemoveTrack? (Harald)**

WebRTC WG Charter (Chairs)

- **Current charter expires in March 2018.**
- **Draft Charter: http://w3c.github.io/webrtc-charter/webrtc-charter.html**
  - **Extends charter to March 31, 2020.**
- **Changes based on feedback from the last virtual interim:**
  - **API functions added:**
    - API functions for accessing the data in these media streams
    - API functions for transferring data between peers
  - **Normative specifications**
    - **Added:** Data Transfer Functions: API functions to provide interfaces that enable the transfer of data between peers, Included in this category are API functions for message-based as well as stream-based communications. The WG will consider any necessary API changes or extensions to enable use of more than one data transfer protocol to support the data transfer functions.
    - **Removed:** Links to WebRTC-ICE and WebRTC-QUIC specifications

WebRTC WG Charter (cont'd)

- **Updated External Organizations:**
  - IETF Applications and Real-Time Area (ART)
    - The RTC APIs developed by this group will build upon the protocols and formats developed in the IETF RTCWeb Working Group. Subsequent to the termination of that WG, this WG will liaise with other groups of the ART area and elsewhere in the IETF as appropriate; of particular interest are the MMUSIC, AVTEXT and QUIC working groups.
  - IETF Transport Area Working Group (TSVWG)
    - The TSVWG develops SCTP on which WebRTC data channels relies.
  - Web Hypertext Application Technology Working Group (WHATWG)
    - The RTC APIs developed by this group will potentially reference the Fetch, Streams and other API specifications maintained by the WHATWG

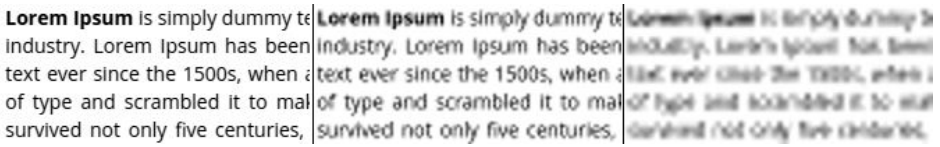
WebRTC WG Charter (cont'd)

- **Added Timeline:**
  - Media Capture and Streams: updated Candidate Recommendation in Q2 2018, Proposed Recommendation in Q4 2018, Recommendation in Q1 2019
  - WebRTC 1.0: updated Candidate Recommendation in Q2 2018, Proposed Recommendation in Q4 2018, Recommendation in Q1 2019
  - WebRTC Object-oriented APIs: First Public Working Draft in Q3 2018, Candidate Recommendation in Q2 2019

For Discussion Today

- **Content hints**
  - **Issue 478: Content hints (Thomas Daede and Mo Zanaty)**

[Issue 478](): What are Content hints?



- WebRTC 1.0 provides degradationPreference enum for video encoders: “maintain-resolution”, “maintain-framerate” or “balanced”.
- But what about audio and APIs other than WebRTC?
  - Providing a “content hint” in an MST property enables usage by MediaStreamRecorder and other APIs with less flexible controls, without having to modify their specifications (scales better).
  - WebRTC: content hint can inform “balanced”.
  - Demo: https://webrtc.github.io/samples/src/content/capture/video-contenthint/
- Example behavior:

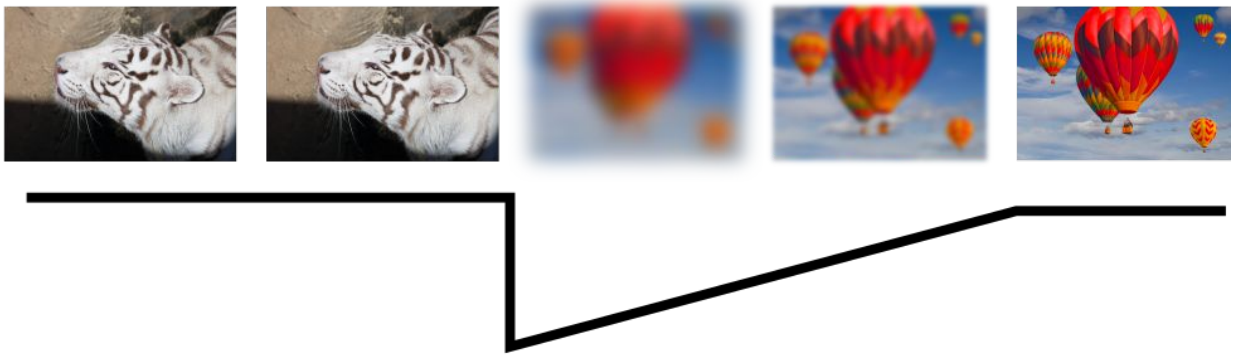
Motion video: Downscale / use higher max QP to preserve motion.
Detail video: Drop frames / use lower max QPs to preserve individual frame quality.
Speech: Use noise suppression and echo cancellation by default. Maybe enhance intelligibility?
Music: Turn off noise suppression (preserve snares), tune echo cancellation differently / turn it off.

Issue 478: Content hints (Thomas Daede and Mo Zanaty)

- Video hints
  - Rate control hints (provided by content hints and degradationPreference)
  - Encoder/format feature hints (NOT in proposed solution)
    - Intrabc
    - Perceptual hints

Video Rate Control Hints

"Motion" hint



"Detail" hint

Video Rate Control Hints

- Controls whether to drop frame and use bits towards next frame vs code the frame at a low quality
- Libvpx/webrtc.org implementation
  - “Detail” turns off downscaling which makes dropping frames more likely
- OpenH264 implementation
  - iUsageType = SCREEN_CONTENT_REAL_TIME or CAMERA_VIDEO_REAL_TIME
  - Uses different settings for frame skipping, quantization thresholds, long term frames, etc.

Encoder/format feature hints

- Intrabc - AV1 Only
  - Effective on text, degrades other content
  - Very expensive to choose automatically, but a wrong hint is even worse
- QM, AQ - AV1 / H.264 High Only
  - Dependent on content type and viewer distance
  - Hard to select automatically
  - Getting it wrong isn’t so bad, but benefit is also small

Desired decision from the WG

Possible decisions:

- Out of scope for the WG - ignore.
- Proposed solution is good - incorporate.
- Please develop another solution.

Presenter’s Recommendations

- Mo Zanaty: <insert recommendation here>
- Thomas Daede - Rate control hint only, but resolve duplication with degradationPreference

Current Status of Screen Capture

- 11 open issues:
  - 6 for discussion today
  - 2 enhancements

For Discussion Today

- 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 29: Powerpoint is special (Suhas)

- Applications that go full-screen might need special treatment

Modes for consideration:

> “application” DisplayCaptureSurfaceType selected
>
> “window” DisplayCaptureSurfaceType selected

In either case, if the UA is able to make unambiguous 1:1 determination that the invisible window displayed in full screen mode maps to the window (initially selected) from the same application, Screen Capture should be allowed.

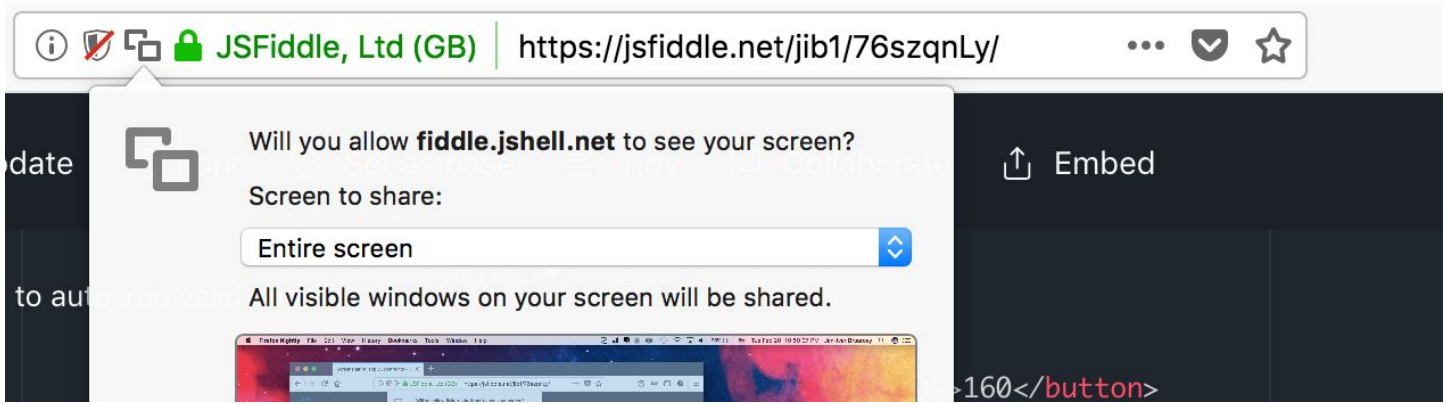
Proposal: Write informal note to explain the above and consequences of mapping failure.

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:



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`, only afterwards:
  - ```
    let stream = await navigator.getDisplayMedia({video: true});
    let track = stream.getTracks()[0];
    await track.applyConstraints({width: 640, frameRate: 5});
    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."
- More consistent ergonomics encourages downscaling and avoids temporarily unwieldy tracks (e.g. 2880x1800x60):
  - ```
    video.srcObject = await navigator.getDisplayMedia({video: {
        width: 640, frameRate: 5}});
    ```

Issue 51: Browser tab sharing (Suhas)

- Proposal
  - 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.

For Discussion Today

- **WebRTC-Stats: report on dead stats (Harald)**
- **WebRTC-PC**
  - **Issue 1694/1739: RTCCertificate backup/Private Key Access (Bernard)**
  - **Issue 1706: Should rollback fire addtrack/removetrack events? (Jan-Ivar)**
  - **Issue 1756: Need to clear [[AssociatedMediaStreams]] in RemoveTrack? (Harald)**

Webrtc-Stats: Report on dead stats

- Old version: They accumulate, with a marker
  - Not so nice for long-running apps
  - Mainly because of clutter, not because of memory
- New version: They arrive to the app in an event
  - PR on webrtc-pc and webrtc-stats ready
  - We seem to have consensus
  - Ready to merge?

Issue 1694/1739: RTCCertificate backup/Private Key Access (Bernard)

- Section 4.10:
  - The `generateCertificate` function causes the user agent to create and store an X.509 certificate [X509V3] and corresponding private key. A handle to information is provided in the form of the `RTCCertificate` interface.… The resulting certificate must not include information that can be linked to a user or user agent. Randomized values for distinguished name and serial number should be used.
- Section 4.10.2:
  - The `RTCCertificate` interface represents a certificate used to authenticate WebRTC communications. In addition to the visible properties, internal slots contain a handle to the generated private keying materal ([[KeyingMaterial]]) and a certificate ([[Certificate]]]]) that `RTCPeerConnection` uses to authenticate with a peer...For the purposes of this API, the [[Certificate]] slot contains unstructured binary data. Note that an `RTCCertificate` might not directly hold private keying material, this might be stored in a secure module. The `RTCCertificate` object can be stored and retrieved from persistent storage by an application.

Issue 1694/1739: RTCCertificate backup/Private Key Access (cont'd)

- Question
  - There is no mechanism for an application to access the [[KeyingMaterial]] internal slot. Yet the text talks about an `RTCCertificate` object being stored and retrieved. Is it expected that the private keys can be backed up and restored?
- Proposed resolution:
  - The `RTCCertificate` object represents a handle to both the certificate as well as the private key, so that it is expected that both can be stored and retrieved from persistent storage. The format in which the object is stored is not specified (e.g. PEM, DER PKCS #12, etc.)

Issue 1706: Should rollback fire events? (jib)

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

Issue 1756: Need to clear [[AssociatedMediaStreams]] in RemoveTrack? (Harald)

- Version 1: pc.addStream(s); pc.removeStream(s)
- Version 2: sender = pc.addTrack(s.getAudioTracks()[0], s); sender.removeTrack()
- Version 3: sender = pc.addTransceiver(s.getAudioTracks()[0], s); sender.removeTrack()

In (1) it's clear that nothing's attached any more.
In (2) we have 2 alternatives being proposed:

1. Leave the "s" reference attached to this sender forever
2. Remove the "s" reference.

Why make a difference?

For extra credit

Name that bird!

Thank you

Special thanks to:

W3C/MIT for WebEx

WG Participants, Editors & Chairs

The bird