W3C WebRTC WG Meeting

January 16, 2024
8 AM - 10 AM

Chairs: Bernard Aboba
Harald Alvestrand
Jan-Ivar Bruaroey

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 January 2024 interim meeting of the W3C WebRTC WG, at which we will cover:
    - WG document status, blocking issues, webrtc-extensions
- Future meetings:
    - February 20
    - March 26
    - April 23

About this Virtual Meeting

- Meeting info:
  - https://www.w3.org/2011/04/webrtc/wiki/January_16_2024
- Link to latest drafts:
  - https://w3c.github.io/mediacapture-main/
  - https://w3c.github.io/mediacapture-extensions/
  - https://w3c.github.io/mediacapture-image/
  - https://w3c.github.io/mediacapture-output/
  - https://w3c.github.io/mediacapture-screen-share/
  - https://w3c.github.io/mediacapture-record/
  - https://w3c.github.io/webrtc-pc/
  - https://w3c.github.io/webrtc-extensions/
  - https://w3c.github.io/webrtc-stats/
  - https://w3c.github.io/mst-content-hint/
  - https://w3c.github.io/webrtc-priority/
  - https://w3c.github.io/webrtc-nv-use-cases/
  - https://github.com/w3c/webrtc-encoded-transform
  - https://github.com/w3c/mediacapture-transform
  - https://github.com/w3c/webrtc-svc
  - https://github.com/w3c/webrtc-ice
- Link to Slides has been published on WG wiki
- Scribe? IRC http://irc.w3.org/ Channel: #webrtc
- The meeting is (still) being recorded. The recording will be public.
- Volunteers for note taking?

W3C Code of Conduct

- This meeting operates under W3C Code of Ethics and Professional Conduct

- We're all passionate about improving WebRTC and the Web, but let's all keep the conversations cordial and professional

Virtual Interim Meeting Tips

This session is (still) being recorded

- Click Raise hand to get into the speaker queue.
- Click Lower hand to get out of the speaker queue.
- Please wait for microphone access to be granted before speaking.
- If you jump the speaker queue, you will be muted.
- Please use headphones when speaking to avoid echo.
- Please state your full name before speaking.
- Poll mechanism may be used to gauge the “sense of the room”.

Understanding Document Status

- Hosting within the W3C repo does ***not*** imply adoption by the WG.
  - WG adoption requires a Call for Adoption (CfA) on the mailing list.
- Editor's drafts do ***not*** represent WG consensus.
  - WG drafts ***do*** imply consensus, once they're confirmed by a Call for Consensus (CfC) on the mailing list.
  - Possible to merge PRs that may lack consensus, if a note is attached indicating controversy.

Issues for Discussion Today

- 08:10 - 08:30 AM WG Document Status (Bernard)
- 08:30 - 09:30 AM Issues blocking Spec Advancement
- 09:30 - 09:50 AM WebRTC-Extensions (Florent and Sameer)
- 09:50 - 10:00 AM Wrapup and Next Steps (Chairs)

Time control:

- A warning will be given 2 minutes before time is up.
- Once time has elapsed we will move on to the next item.

WG Document Status

Start Time: 08:10 AM

End Time: 08:30 AM

WEBRTC WG Document Status

- For this status update, we will focus on:
  - WebRTC-PC
  - MediaCapture-Main
  - MST-Content-Hint
  - WebRTC-SVC
  - Encoded-transform
  - Mediacapture-transform
- Potential metrics
  - Spec status
  - Implementation
  - Issue status
  - Test status

WEBRTC-PC Spec Status

https://www.w3.org/standards/history/webrtc/

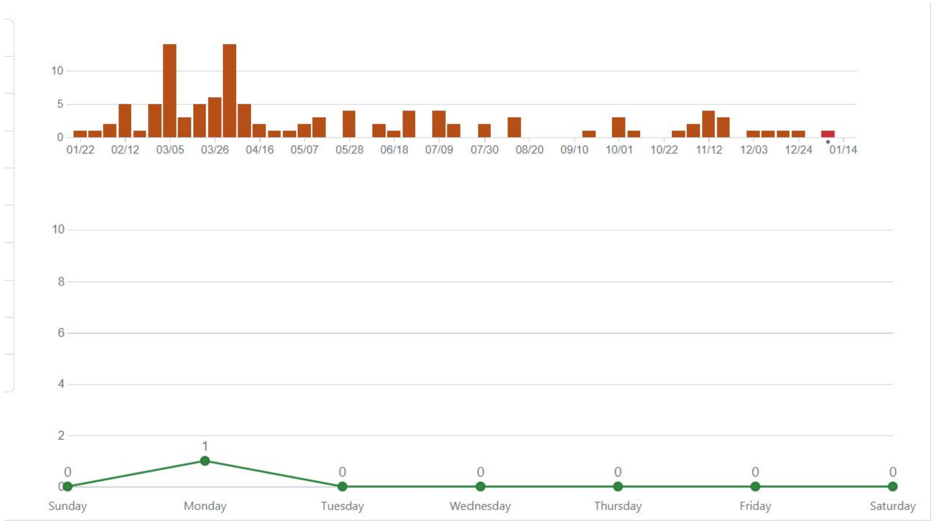
WebRTC: Real-Time Communication in Browsers publication history

Date	Status
6 March 2023	Recommendation
1 March 2023	Recommendation
26 January 2021	Recommendation
15 December 2020	Proposed Recommendation
3 December 2020	Candidate Recommendation Draft
1 December 2020	Candidate Recommendation Draft
27 November 2020	Candidate Recommendation Draft
25 November 2020	Candidate Recommendation Draft
19 November 2020	Candidate Recommendation Draft

WebRTC-PC Status

https://github.com/w3c/webrtc-pc/graphs/commit-activity

- Implementation
  - All browsers
- Issues
  - 50 open issues, 20 open > 1 year
- Commits



Mediacapture-Streams Spec Status

https://www.w3.org/standards/history/mediacapture-streams/

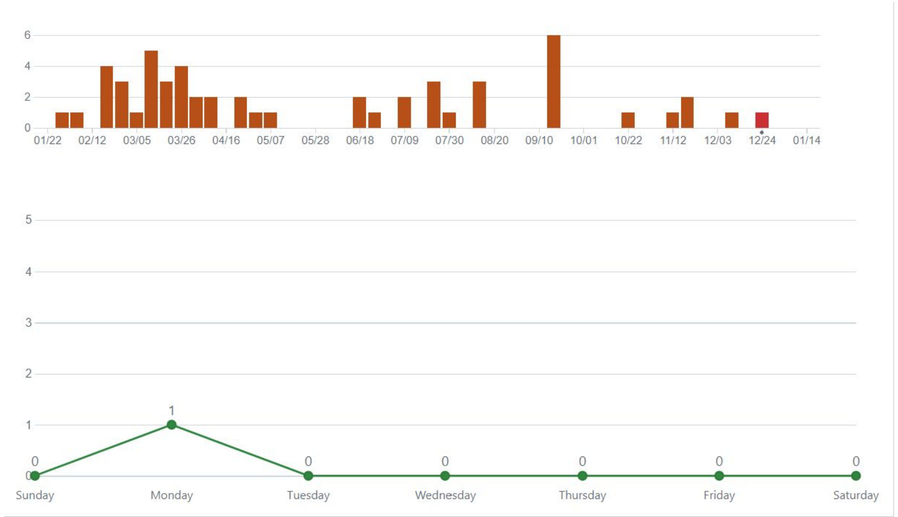
Media Capture and Streams publication history

Date	Status
20 November 2023	Candidate Recommendation Draft
21 September 2023	Candidate Recommendation Draft
17 August 2023	Candidate Recommendation Draft
19 June 2023	Candidate Recommendation Draft
4 May 2023	Candidate Recommendation Draft
27 April 2023	Candidate Recommendation Draft
13 April 2023	Candidate Recommendation Draft
6 April 2023	Candidate Recommendation Draft
23 March 2023	Candidate Recommendation Draft

MediaCapture-Streams Status

https://github.com/w3c/mediacapture-main/graphs/commit-activity

- Implementation
  - All browsers
- Issues
  - 31 open issues, 9 open > 1 year
- Commits



MediaCapture-Streams WPT Status

https://wpt.fyi/results/mediacapture-streams

Path	Chrome 122 Linux 20.04 53d5a05 Jan 16, 2024	Edge 122 Windows 10.0 53d5a05 Jan 16, 2024	Firefox 123 Linux 20.04 53d5a05 Jan 16, 2024	Safari 186 preview macOS 13.6 53d5a05 Jan 16, 2024
MediaStream-supported-by-feature-policy.html	2 / 2	2 / 2	0 / 2	0 / 2
MediaStream-video-only.https.html	1 / 1	1 / 1	1 / 1	1 / 1
MediaStreamTrack-applyConstraints.https.html	8 / 8	8 / 8	6 / 8	6 / 8
MediaStreamTrack-getCapabilities.https.html	112 / 112	108 / 112	0 / 36	80 / 112
MediaStreamTrack-getSettings.https.html	22 / 22	21 / 22	15 / 22	15 / 22
MediaStreamTrack-id.https.html	1 / 1	1 / 1	1 / 1	1 / 1
MediaStreamTrack-iframe-audio-transfer.https.html	0 / 1	0 / 1	0 / 1	0 / 1
MediaStreamTrack-iframe-transfer.https.html	0 / 1	0 / 1	0 / 1	0 / 1
MediaStreamTrack-init.https.html	1 / 1	1 / 1	1 / 1	1 / 1
MediaStreamTrack-MediaElement-disabled-audio-is-silence.https.html	1 / 1	1 / 1	1 / 1	1 / 1
MediaStreamTrack-MediaElement-disabled-video-is-black.https.html	2 / 3	3 / 3	3 / 3	1 / 3
MediaStreamTrack-transfer-video.https.html	0 / 1	0 / 1	0 / 1	0 / 1
MediaStreamTrack-transfer.https.html	0 / 1	0 / 1	0 / 1	0 / 1
MediaStreamTrackEvent-constructor.https.html	3 / 3	3 / 3	3 / 3	3 / 3
overconstrained_error.https.html	2 / 2	2 / 2	0 / 2	0 / 2
parallel-capture-requests.https.html	2 / 2	2 / 2	2 / 2	2 / 2
Subtest Total	418 / 456	413 / 456	269 / 372	351 / 448

MST-ContentHint Spec Status

https://www.w3.org/standards/history/mst-content-hint/

MediaStreamTrack Content Hints publication history

Date	Status
22 July 2021	Working Draft
1 July 2021	Working Draft
19 January 2021	Working Draft
2 December 2020	Working Draft
30 November 2020	Working Draft
28 July 2020	Working Draft
14 February 2020	Working Draft
3 July 2018	First Public Working Draft

MST-ContentHint WPT Status

https://wpt.fyi/results/mst-content-hint

Path	Chrome 122 Linux 20.04 53d5a05 Jan 16, 2024	Edge 122 Windows 10.0 53d5a05 Jan 16, 2024	Firefox 123 Linux 20.04 53d5a05 Jan 16, 2024	Safari 186 preview macOS 13.6 53d5a05 Jan 16, 2024
idlharness.window.html	9 / 9	9 / 9	6 / 9	9 / 9
MediaStreamTrack-contentHint.html	7 / 7	7 / 7	2 / 7	7 / 7
RTCRtpSendParameters-degradationEffect.html	0 / 1	1 / 1	0 / 1	0 / 1
RTCRtpSendParameters-degradationPreference.html	6 / 6	6 / 6	2 / 6	6 / 6
Subtest Total	22 / 23	23 / 23	10 / 23	22 / 23

Relevant links for */mst-content-hint* results

/mst-content-hint/RTCRtpSendParameters-degradationPreference.html > : https://bugs.webkit.org/show_bug.cgi?id=262609

WebRTC-SVC Spec Status

https://www.w3.org/standards/history/webrtc-svc/

Scalable Video Coding (SVC) Extension for WebRTC publication history

Date	Status
16 December 2023	Working Draft
12 December 2023	Working Draft
15 November 2023	Working Draft
2 November 2023	Working Draft
17 April 2023	Working Draft
15 April 2023	Working Draft
14 April 2023	Working Draft

MediaCapture-Streams Status

https://github.com/w3c/mediacapture-main/graphs/commit-activity

- Implementation
  - Chromium, Media Capabilities shows support in Safari Tech Preview?
- Issues
  - 1 open issue (not an extension), > 1 year

WebRTC-SVC WPT Status

https://wpt.fyi/results/webrtc-svc

Path	Chrome 122 Linux 20.04 53d5a05 Jan 16, 2024	Edge 122 Windows 10.0 53d5a05 Jan 16, 2024	Firefox 123 Linux 20.04 53d5a05 Jan 16, 2024	Safari 186 preview macOS 13.6 53d5a05 Jan 16, 2024
RTCRtpParameters-scalability-av1.html	28 / 28	28 / 28	0 / 28 ⚠	0 / 28 ⚠
RTCRtpParameters-scalability-h264.html	3 / 3	3 / 3	0 / 3 ⚠	0 / 3 ⚠
RTCRtpParameters-scalability-vp8.html	3 / 3	3 / 3	0 / 3 ⚠	0 / 3 ⚠
RTCRtpParameters-scalability-vp9.html	28 / 28	28 / 28	0 / 28 ⚠	0 / 28 ⚠
RTCRtpParameters-scalability.html	5 / 5	5 / 5	2 / 5	2 / 5
Subtest Total	67 / 67	67 / 67	2 / 67	2 / 67

Encoded Transform Spec Status

https://www.w3.org/standards/history/webrtc-encoded-transform/

WebRTC Encoded Transform publication history

Date	Status
12 December 2023	Working Draft
7 December 2023	Working Draft
12 October 2023	Working Draft
9 October 2023	Working Draft
4 October 2023	Working Draft
22 September 2023	Working Draft
27 July 2023	Working Draft
13 July 2023	Working Draft
29 June 2023	Working Draft
2 June 2023	Working Draft

Encoded-Transform WPT Status

https://wpt.fyi/results/webrtc-encoded-transform

Path	Chrome 122 Linux 20.04 53d5a05 Jan 16, 2024	Edge 122 Windows 10.0 53d5a05 Jan 16, 2024	Firefox 123 Linux 20.04 53d5a05 Jan 16, 2024	Safari 186 preview macOS 13.6 53d5a05 Jan 16, 2024
idlharness.https.window.html	34 / 64	34 / 64	44 / 64	27 / 64
script-audio-transform.https.html	0 / 1	0 / 1	1 / 1	1 / 1
script-change-transform.https.html	0 / 1	0 / 1	1 / 1	1 / 1
script-late-transform.https.html	0 / 1	0 / 1	0 / 1	0 / 1
script-metadata-transform.https.html	0 / 13	0 / 13	11 / 13	7 / 13
script-transform-generateKeyFrame-simulcast.https.html	0 / 3	0 / 3	3 / 3	1 / 3
script-transform-generateKeyFrame.https.html	0 / 10	0 / 10	10 / 10	4 / 10
script-transform-sendKeyFrameRequest.https.html	0 / 6	0 / 6	6 / 6	5 / 6
script-transform.https.html	0 / 4	0 / 4	4 / 4	4 / 4
script-write-twice-transform.https.html	0 / 1	0 / 1	1 / 1	1 / 1
set-metadata.https.html	1 / 1	1 / 1	0 / 1	0 / 1
sframe-keys.https.html	0 / 2 ⚠	0 / 2 ⚠	0 / 2 ⚠	0 / 2 ⚠
sframe-transform-buffer-source.html	0 / 1	0 / 1	0 / 1	0 / 1
sframe-transform-in-worker.https.html	0 / 1	0 / 1	0 / 1	0 / 1
sframe-transform-readable.html	0 / 1	0 / 1	0 / 1	0 / 1
sframe-transform.html	0 / 6	0 / 6	0 / 6	0 / 6
tentative/	19 / 27	19 / 27	0 / 27	0 / 27
Subtest Total	54 / 143	54 / 143	81 / 143	51 / 143

MediaCapture Transform Spec Status

https://www.w3.org/standards/history/mediacapture-transform/

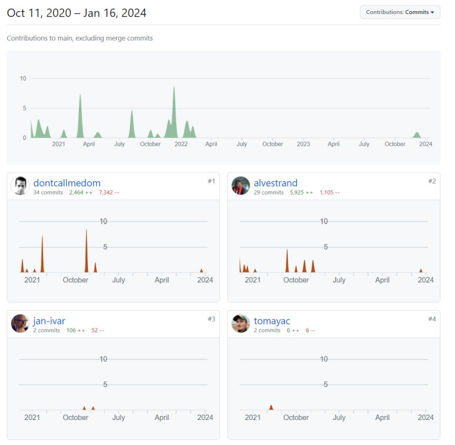
MediaStreamTrack Insertable Media Processing using Streams publication history

Date	Status
20 October 2022	Working Draft
10 February 2022	First Public Working Draft

MediaCapture Transform Status

https://github.com/w3c/mediacapture-transform/issues/

- Implementation
    - Chromium, Safari Test Preview
- Issues
    - 18 open issues, 17 open > 1 year
- Activity



MediaCapture Transform WPT Status

https://wpt.fyi/results/mediacapture-insertable-streams

Path	Chrome 122 Linux 20.04 53d5a05 Jan 16, 2024	Edge 122 Windows 10.0 53d5a05 Jan 16, 2024	Firefox 123 Linux 20.04 53d5a05 Jan 16, 2024	Safari 186 preview macOS 13.6 53d5a05 Jan 16, 2024
idlharness.any.worker.html	2 / 23	2 / 23	2 / 23	2 / 23
MediaStreamTrackGenerator-audio.https.html	5 / 5	5 / 5	0 / 5	0 / 5
MediaStreamTrackGenerator-in-service-worker.https.html	0 / 1	0 / 1	0 / 1	0 / 1
MediaStreamTrackGenerator-in-shared-worker.https.html	0 / 1	0 / 1	0 / 1	0 / 1
MediaStreamTrackGenerator-in-worker.https.html	0 / 3	0 / 3	0 / 3	0 / 3
MediaStreamTrackGenerator-pipes-data-in-worker.https.html	0 / 1	0 / 1	0 / 1	0 / 1
MediaStreamTrackGenerator-video.https.html	7 / 7	7 / 7	0 / 7	0 / 7
MediaStreamTrackProcessor-audio.https.html	2 / 2	2 / 2	0 / 2	0 / 2
MediaStreamTrackProcessor-backpressure.https.html	1 / 1	1 / 1	0 / 1	0 / 1
MediaStreamTrackProcessor-video.https.html	3 / 3	3 / 3	0 / 3	0 / 3
VideoTrackGenerator.https.html	0 / 8	0 / 8	0 / 8	0 / 8
Subtest Total	20 / 55	20 / 55	2 / 55	2 / 55

Discussion (End Time: 08:30)

-

Blocking Issues

Start Time: 08:30 AM

End Time: 09:30 AM

For Discussion Today

- WebRTC-PC
  - Issue 2888: setCodecPreferences vs unidirectional codecs (Fippo)
  - Issue 2915: WebRTC spec should explicitly specify all causes of a PeerConnection-sourced track being muted (Jan-Ivar)
- WebRTC-SVC
  - Issue 176/PR 212: General approach to capabilities negotiation
  - Issue 92: Align exposing `scalabilityMode` with WebRTC “hardware capabilities” check (Bernard)
- Mediacapture-transform
  - Issue 81: How does generator.mute change track states? (Harald)
- Encoded-transform
  - Issue 220: Is RTCEncodedVideoFrameMetadata.frame_id actually an unsigned long long or does it wrap at 16 bits? (Tony Herre)

For Discussion Today (Cont'd)

- MediaCapture-Main
  - Issue 958: Mark resizeMode, sampleRate, sampleSize, latency as features at risk (Jan-Ivar)
- MST-Content-Hint
  - Issue 35/PR 56: Highly detailed text in video content (Harald)
  - Issue 55: Comments and request from APA review (Harald)

#2888 setCodecPreferences vs unidirectional codecs

- JSEP: “setCodecPreferences does not directly affect which codec the implementation decides to send. It only affects which codecs the implementation indicates that it prefers to receive”
  - *sCP* is **not** something you use to pick the send codec works because send codec is dictated by receive preference.
  - *sCP* should not take send codecs into account which webrtc-pc does
  - Typically done in ontrack, see updated samples PR

1. Fix webrtc-pc by removing mentions of send codecs in setCodecPreferences (PR preview)
2. Clarify “codecs match” algorithm
   a. missing consideration for profile-level-asymmetry-allowed and default value inference and CN

Issue 2915: WebRTC spec should explicitly specify all causes of a PeerConnection-sourced track being muted (Jan-Ivar)

- Content goes here

Issue 176: General approach to capabilities negotiation

- Background information
  - MC supports file, media-source, media recorder, WebRTC, indicating whether a encoder/decoder config is “supported”, “powerEfficient” or “smooth”, with no “hardware check”.
  - For decoding, intent is to replace `isTypeSupported()` or `canPlayType()` which indicate if something cannot be decoded but not how well it should perform.
  - For SVC, MC provides info on encoders (supported `scalabilityMode` values) as well as decoders (support for `spatialScalability`).
- PING review (March 2021)
  - Excellent fingerprinting analysis.
  - “Why are we exposing device capabilities to the app for purposes of negotiation? Couldn't we instead have sites expose available media formats and have browsers (perhaps in a way not exposed the application) pick the one they like best?”
- MEDIA WG Meeting Minutes (January 9, 2024)
  - Bernard to submit PR explaining the RTC media negotiation model.
  - Separate PRs to be submitted for other MC use cases (e.g. streaming)

PR 212: RTC Capabilities Negotiation Model (MEDIA WG)

```
+       <p>
+         This specification supports {{MediaDecodingType}} values of {{file}},
+         {{media-source}} or {{webrtc}} as well as {{MediaEncodingType}}
+         values of {{record}} and {{webrtc}}.
+       </p>
+       <p>
+         In realtime communications as supported in [[webrtc]], media is
+         transported between peers. Although web sites can play a role in
+         relaying of signaling between peers, they are typically not involved
+         in media transport, encoding or decoding. For 1-1 calls, peers
+         typically directly negotiate media to be sent and received.
+      </p>
+      <p>
+         In a conferencing scenario, a WebRTC peer can send media for
+         reception by dozens or even hundreds of receivers.  To improve
+         scalability, applications make use of external servers, such as
+         selective forwarding units or conferencing bridges. These servers
+         negotiate media parameters with participants, ensuring consistency
+         across senders and receivers. This is vastly preferrable to
+         allowing browsers to negotiate with each potential peer, which
+         would require N (N -1) negotiations, while ensuring interoperability
+         between peers, which would not be supported if browsers were allowed
+         to "pick the one they like best".
+      </p>
+   </section>
```

[Issue 92](#)/[PR 97](#): Align exposing `scalabilityMode` with WebRTC "hardware capabilities" check (Bernard)

- Filed as part of Issue 117: PING review of WebRTC-SVC (May 4, 2023).
  - "This proposal would expose additional fingerprinting surface. It needs to have some protection to prevent that fingerprinting surface from being misused. I'm suggesting the hardware check being discussed in other specs could be a place for this group to work to solve the privacy risk this proposal adds, but either way, the problem is the additional fingerprinting surface *this* proposal ads to the platform"
- Privacy analysis
  - Through trial and error, `setParameters()` can be used to determine what `scalabilityMode` values are supported for each codec. However:
    - `setParameters()` does not indicate whether codec is supported in hardware or software. Can perhaps be determined after media is flowing.
    - `setParameters()` applies only to RTCRtpSender, not RTCRtpReceiver.
      - Does not indicate whether decoder supports spatial scalability.
      - Lack of support for spatial references can be hidden by software failover
  - "Hardware checks" not compatible with gaming, streaming use cases

PR 97: Rewrite of Privacy Considerations Section

```diff
         <p>
+          The [[?Media-Capabilities]] API provides information on
+          encoder and decoder capabilities, indicating whether a
+          proposed configuration is "supported", "smooth" and
+          "power efficient". [[?Media-Capabilities]] API
+          also indicates whether the decoder supports spatial
+          prediction.
         </p>
         <p>
+          In WebRTC, the use of scalable coding tools is not
+          negotiated between peers, so neither supported
+          {{RTCRtpEncodingParameters/scalabilityMode}} values nor
+          values nor decoder support for spatial prediction is
+          exposed in SDP. By attempting to set
+          {{RTCRtpEncodingParameters/scalabilityMode}} values for
+          each codec using the {{RTCRtpSender/setParameters()}} API,
+          an application can determine supported values by noting
+          which configuration attempts succeed and which ones fail.
```

Issue 81: How does generator.mute change track states? (Harald)

- Mediacapture-transform “consensus” API: VideoTrackGenerator
- VideoTrackGenerator takes a stream of frames and produces a track.
- There’s an attribute called “muted”
- Two possible models:
  a. Setting “muted” fires “muted” on the output track and all its clones
  b. Setting “muted” queues a task that fires “muted” on the output track and all its clones if the “muted” state on the VTG is different from the “muted” state on the track
- Proposal: Go with model b. It’s simpler for the user to reason about.

Issue 220: Is RTCEncodedVideoFrameMetadata.frame_id actually an unsigned long long or does it wrap at 16 bits? (Tony Herre)

- What's in the specification
  - frame_id is defined as a `unsigned long long`, but currently doesn't have any textual definition in the members section beneath.
  - Same for dependencies.
- Current Implementations
  - Taken from Dependency Descriptor Header Extension, frame_number: a 16 bit codec-agnostic id
  - Chromium handles unwrapping into a 64 bit unsigned on the receiver side
  - Other browsers?
- Proposal
  - Keep `unsigned long long`
  - frameId: A monotonically increasing frame counter. Its lower 16 bits will match the frame_number of the Dependency Descriptor Header Extension
  - dependencies: List of the frameIds the RTCEncodedVideoFrame references
  - Both only present on the receiver if DD header extension is sent

- Related: Issue 619/PR 756 in WebCodecs

Issue 958: Mark resizeMode, sampleRate, sampleSize, latency as features at risk (Jan-Ivar)

- Content goes here

Issue 7: Add localPriority as orthogonal value to networkPriority (Harald)

- History: There was only “priority”, controlling both queueing priority and DSCP codepoints
- Experience showed that DSCP codepoints had hard-to-predict effects (and problems with definition in TCP cases), not deployed
- Local priority can be implemented locally, no dependency on network
- Hotfix: Define “networkPriority” to override “priority”, set local priority by setting “priority” to non-default and “networkPriority” to default value
- Not deployed either
- dcSCTP supports local priority scheduling in code, but not in JS API (and not DSCP priority).
- Proposal: Deprecate (and remove?) Priority - add localPriority

MST-Content-Hint

Issue 35/PR 56: Highly detailed text in video content (Harald)

- Original PR was to acknowledge that some scripts (देवनागरी, كتابة عربية) may become unreadable at resolutions that work fine for ASCII
- Bernard suggested that we also note that red on yellow will work worse than black on white if 4:2:0 coding is used, and proposed to recommend (force?) 4:4:4
- Support of 4:4:4 is profile-specific. We may not want to mandate it here.
- Suggestion (1): Reword addition to note that encoding of colored text may cause readability issues, but don't mandate anything
- Suggestion (2): Recommend use of 4:4:4 if colored text dominates when content-hint: "text" is applied, and it's permitted by profile
- Suggestion (3): Mandate use of 4:4:4 for this case
- Pick one?

Issue 55: Comments and request from APA review (Harald)

- Many issues are not addressable within WebRTC
  - Need to reply with “not our problem, needs to be addressed in apps”
- Some issues may need addressing.
  - “Use cases with support files” - captions, audio descriptions
    - These are outside the WebRTC model. Should we address them?
  - Suggested hint “audio-description” goes with link-between-tracks
    - Suggestion: Reject. This is MST-content-hint, not MOQ.
  - Suggested hint “motion-with-transcription” - for video with embedded subtitles, giving “text” treatment to subtitles, “motion” for rest
    - Suggestion: Reject. Not a model that we want to encourage - subtitles should be additional tracks.
  - Suggested treatment by region to allow differences in parts of a video
    - Suggestion: Hold for later extension. At the moment we have no region model to conform to.
  - Asking about transporting hints.
    - Suggestion: Clarify that hints are *not* encoded in the media.
- Discuss!

Discussion (End Time: 09:30)

-

WebRTC-Extensions (Florent)

Start Time: 09:30 AM

End Time: 09:50 AM

For Discussion Today

- WebRTC-Extensions
  - Issue 191: Add API to control encode complexity

Issue 191: Add API to control encode complexity

- Goal: allow applications to optimize the trade-off between device resource usage and compression efficiency for their use cases.
- A higher encode complexity mode can be used to achieve better video quality and/or to reduce video bitrate.
- Modeled after similar settings in other applications and APIs:
    - Android Media: integer 0-9
    - Azure Media Service: speed, balanced, quality
    - x264: presets ultrafast, superfast, veryfast, faster, fast, medium, slow, slower, veryslow
- Results could vary depending on the codec or specific encoder used and are not meant to be fixed by the specification.
- Expected
    - $\text{Avg}(\text{EncodeTime}_{\text{Low}}) \le \text{Avg}(\text{EncodeTime}_{\text{Normal}}) \le \text{Avg}(\text{EncodeTime}_{\text{High}})$
    - $\text{Avg}(\text{Qp}_{\text{Low}}) \ge \text{Avg}(\text{Qp}_{\text{Normal}}) \ge \text{Avg}(\text{Qp}_{\text{High}})$

Issue 191: Add API to control encode complexity

- Suggested WebIDL:

```
enum RTCEncodeComplexityMode {
  "low", // lower device resource usage, worse compression efficiency
  "normal", // default browser settings
  "high" // higher device resource usage, better compression efficiency
};

partial dictionary RTCRtpEncodingParameters {
  RTCEncodeComplexityMode encodeComplexityMode = "normal";
};
```

Discussion (End Time: 09:50)

-

Wrapup and Next Steps

Start Time: 09:50 AM

End Time: 10:00 AM

Next Steps

- Content goes here

Thank you

Special thanks to:

WG Participants, Editors & Chairs