

W3C WebRTC WG Meeting

September 15, 2020
8AM Pacific Time

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 September interim meeting of the W3C WebRTC WG!
 - During this meeting, we will talk about WebRTC Extensions as well as media capture issues.

About this Virtual Meeting

- Meeting info:
 - https://www.w3.org/2011/04/webrtc/wiki/September_15_2020
- Link to latest drafts:
 - <https://w3c.github.io/mediacapture-main/>
 - <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://w3c.github.io/webrtc-dscp-exp/>
 - <https://github.com/w3c/webrtc-insertable-streams>
 - <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 being recorded. The recording will be public.

Issues for Discussion Today

- WebRTC Extensions
 - [Issue 47](#): RTP Header Extension Encryption (Henrik)
- Media Capture
 - [Issue 646](#): Should enumerateDevices by default return an empty list? (Youenn)
 - [Issue 640](#): Only reveal labels of devices user has given permission to (Youenn)
 - [Issue 645](#): enumerateDevices should only provide device info permission for granted device types (Youenn)

Issues for Discussion Today (cont'd)

- Media Capture Image
 - [Issue 229/PR 707](#): Document OverConstrainedError for PTZ in explainer (Youenn)
 - [Issue 243](#): Determine how PTZ permission prompts are supposed to work (Youenn)
 - [Issue 245](#): Sanitize PTZ capabilities/settings based on PTZ permission (Youenn)

Issues for Discussion Today

- WebRTC Extensions
 - [Issue 47](#): RTP Header Extension Encryption (Henrik)

Issue 47: RTP Header Extension Encryption (Henrik) - 1/2

Today's implementations encrypt RTP data. But not RTP headers.

RFC 6904 to encrypt RTP header extension data has not been (correctly) implemented.

- RTP header extension data has been leaking for years. E.g. audio levels.
- Complexity has hindered implementation...
 - SDP bloat per m= section, encrypted/unencrypted forms, bugs in libsrtp/libwebrtc... See AVTCore slides.
- AVTCore to resolve RTP layer issues.

This Working Group: API needed to control negotiating encrypting header extensions.

Issue 47: RTP Header Extension Encryption (Henrik) - 2/2

Proposal:

- All-or-nothing encryption.
 - Encrypt RTP headers on all m= sections or none of them.
- Always attempt to negotiate it.
 - E.g. "a=extmap-encrypted" in offer SDP (todo: avtcore).
- Add `RTCCConfiguration.rtpHeaderEncryptionPolicy`
 - "negotiate": If other endpoint doesn't support it, send unencrypted.
 - "require": If the other endpoint doesn't support it, fail to negotiate (throw an exception on `setRemoteDescription`).
- Add `RTCRtpTransceiver.encryptionNegotiated` flag.

Rollout: We can default to "negotiate" and once this is widely supported we can change the default to "require". (Phase 1: Opt-in. Phase 2: Opt-out.)

Issues for Discussion Today

- Media Capture
 - [Issue 646](#): Should enumerateDevices by default return an empty list? (Youenn)
 - [Issue 640](#): Only reveal labels of devices user has given permission to (Youenn)
 - [Issue 645](#): enumerateDevices should only provide device info permission for granted device types (Youenn)

Issue 646: Should enumerateDevices by default return an empty list?

- Current consensus: expose presence of mics and cameras
 - Camera + Mic: [{ kind: "audioinput"}, { kind: "videoinput" }]
 - No Camera, Mic: [{ kind: "audioinput"}]
 - No Camera, No Mic: []
 - Expose more information past getUserMedia resolution
- Use-cases
 - Allow web pages to expose UI in case devices are available
- Proposal: keep current consensus
 - High risk of breakage, small fingerprinting risk
 - User Agents can fake presence of devices if no real devices

[Issue 640](#): Only reveal labels of devices user has given permission to

- Current consensus
 - Expose all devices past getUserMedia resolution
- Use-cases
 - Device picker built by web pages
- Proposal: keep current consensus
 - Reconsider once in-chrome picker is defined

[Issue 645](#): enumerateDevices should only provide device info permission for granted device types

- Current consensus
 - Expose all devices past getUserMedia resolution
 - Even if only camera access is granted, all microphone information is granted (plus potentially speakers)
- Use-cases
 - Simpler implementation for device pickers built by web pages
 - Workaround to enumerateDevices recent restrictions
- Proposal
 - Restrict to granted device types only
 - And extend exposure to devices with matching 'groupId'

Issues for Discussion Today

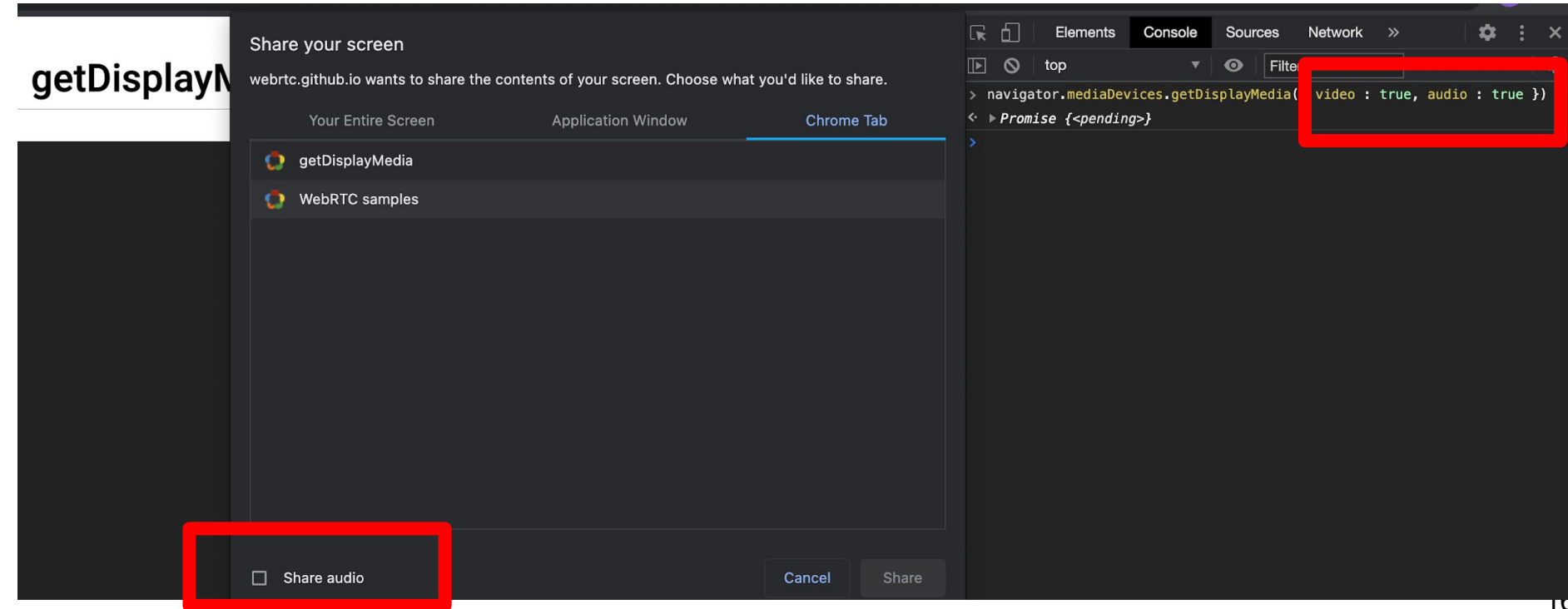
- Media Capture Image
 - [Issue 243](#): Determine how PTZ permission prompts are supposed to work (Youenn)
 - [Issue 245](#): Sanitize PTZ capabilities/settings based on PTZ permission (Youenn)
 - [Issue 229](#)/[PR 707](#): Document OverConstrainedError for PTZ in explainer (Youenn)
 - [Issue 256](#):/ [PR 257](#) Clarify meaning of PTZ constraints presence Option 3 (Jan-Ivar cannot make it... A volunteer?)

[Issue 243](#): Determine how PTZ permission prompts are supposed to work (Youenn)

- Give user the ability to choose between various permissions
 - No Camera
 - Camera-PTZ
 - Camera+PTZ
 - Allow User Agent to build prompts that allow all those choices

Issue 243: Determine how PTZ permission prompts are supposed to work (Youenn)

- `getDisplayMedia` as an example



[Issue 243](#): Determine how PTZ permission prompts are supposed to work (Youenn)

- Page does not have the ability to require system audio capture
 - No interest so far in allowing web pages to require it
 - Page has to communicate its needs to the user before the `getDisplayMedia` call
- User Agent prompts should be able to offer 'Camera-PTZ' choice
 - Page should not have a way to force user in either 'Camera+PTZ' or 'No Camera' within `getUserMedia`

[Issue 243](#): Determine how PTZ permission prompts are supposed to work (Youenn)

- What happens with `getUserMedia({ video : { pan : { exact : '10' } } })`?
- System has no matching PTZ device: `OverConstrained` error
 - Increase the scope of an existing fingerprinting issue
- System has matching PTZ device: prompt to the user
 - User chooses 'Camera-PTZ'
 - `OverConstrained` error
- Need to skip PTZ constraints in case of 'Camera-PTZ'
 - If no matching PTZ device, we should consider being in 'Camera-PTZ' and no longer reject either
- Conclusion: skip all PTZ required constraints in `getUserMedia`

[Issue 243](#): Determine how PTZ permission prompts are supposed to work (Youenn)

- If user grants PTZ permission, selected camera SHOULD be PTZ
 - Especially important for User Agent UI without device picker
 - Exception: hardware error when opening PTZ device
- Current spec status
 - Request PTZ permission whenever a PTZ constraint is added
 - Plan: request PTZ permission only if there is a PTZ device
 - Refinement: request PTZ permission only if there is at least one selectable PTZ device
- Make sure PTZ devices have the lowest fitness distances whenever PTZ permission is granted so that they are selected first

[Issue 243](#): Determine how PTZ permission prompts are supposed to work (Youenn)

- Proposed solution: `getUserMedia({ video : { ... }, ptz : true })`
 - If 'ptz' is true, move all PTZ devices first in the device list
 - If 'ptz' is true and the first device in the list is PTZ, request PTZ permission
- Keep the definition of pan, tilt and zoom as numerical constraints
 - For use in `applyConstraints`
- Skip any pan, tilt and zoom required constraint in `getUserMedia`
 - Or skip all PTZ constraints in `getUserMedia` to simplify API
 - `applyConstraints` can be used once `getUserMedia` resolves

[Issue 243](#): Determine how PTZ permission prompts are supposed to work (Youenn)

- Advantages
 - No fingerprinting increase
 - Fixes [Issue 229](#)
 - Ability for User Agents to sanitize PTZ position if needed
 - As it could be used as side channel to communicate
 - Limited added complexity
 - Future proof for device-picker based getUserMedia
 - Future proof as 'ptz' field can be extended to a dictionary
 - To fine tune the order of PTZ devices

[Issue 243](#): Determine how PTZ permission prompts are supposed to work (Youenn)

- Potential changes to mediacapture-main
 - Use a list instead of a set to store the list of selectable devices
 - Add a hook to sort the list of selectable devices based on getUserMedia parameters
 - Add a hook to compute the requested permissions based on the list of selectable devices

Issue 245: Sanitize PTZ capabilities/settings based on PTZ permission (Youenn)

- A page can use a PTZ device with or without PTZ permission
- If PTZ was requested and granted, expose everything
- Otherwise, it is still useful for the page to know that device is PTZ
- Proposal if PTZ permission is not requested or not granted:
 - Do not expose anything in settings
 - Expose { pan : { }, tilt : { } zoom : { } } in capabilities
 - No min/max values
 - Do not allow dynamic changes of capabilities
 - Even if PTZ permission changes on the fly

[Issue 256](#):/ [PR 257](#) Clarify meaning of PTZ constraints presence

Option 3 (Jan-Ivar cannot make it today... A volunteer to present?)

- Proposal: Make true a first-class value for pan, tilt, zoom constraints
- Gives us expected behaviors naturally out of constraints algorithm
- `getUserMedia({ video : { pan: true, tilt: true, zoom: true } })`
 - Same syntax, but true is a first class value with fitness distance (actual == ideal) ? 0 : 1 when used in place of numerical value
- Same semantics across all constraints.
- Easy on the user: Use a boolean to leave a value alone.
- (boolean or double) avoids need for separate ptz constraint
- All JS-input corner cases become well-defined (yes you can constrain on false, yes you can do `{exact: true}` or even `{max: 10, ideal: false}`). No harm. Predictability trumps usefulness at the edges.

For extra credit



Name these Birds!

Thank you

Special thanks to:

WG Participants, Editors & Chairs

The birds