W3C WebRTC WG Meeting

January 14, 2020
8 AM 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 interim meeting of the W3C WebRTC WG!
  - During this meeting, we hope to make progress on a few WebRTC issues as well as privacy concerns, and discuss a potential extension to WebRTC-PC.

About this Virtual Meeting

Information on the meeting:

- Meeting info:
  - https://www.w3.org/2011/04/webrtc/wiki/January_14_2020
- Link to latest drafts:
  - https://w3c.github.io/mediacapture-main/
  - 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-stats/
  - https://www.w3.org/TR/mst-content-hint/
  - https://w3c.github.io/webrtc-nv-use-cases/
  - https://w3c.github.io/webrtc-dscp-exp/
  - 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.

Issues for Discussion Today

- WebRTC-PC
  - Issue 2412: Is the testing policy being applied/working? (Harald)
- Media Capture and Streams
  - PING's Privacy-By-Default flow (Jan-Ivar)
    - Issue 640: Only reveal labels of devices granted permission
    - Issue 656: No way to choose correct camera & mic upfront
    - Issue 649: #632 broke ability for sites to override Firefox picker
    - Issue 652: In-content device selection a mistake. Leaks; Complex
  - Issue 642: Only Firefox turns off device on disabled track. Stronger language needed? (Jan-Ivar)
  - Issue 655: How to avoid wide-lens back-facing cam on new phones (Jan-Ivar)
  - Issue 639: Enforcing user gesture for getUserMedia (Youennf)
- WebRTC-extensions
  - API for controlling RTP header extensions

Issues for Discussion Today

- WebRTC-PC
  - Issue 2412: Is the testing policy being applied/working? (Harald)

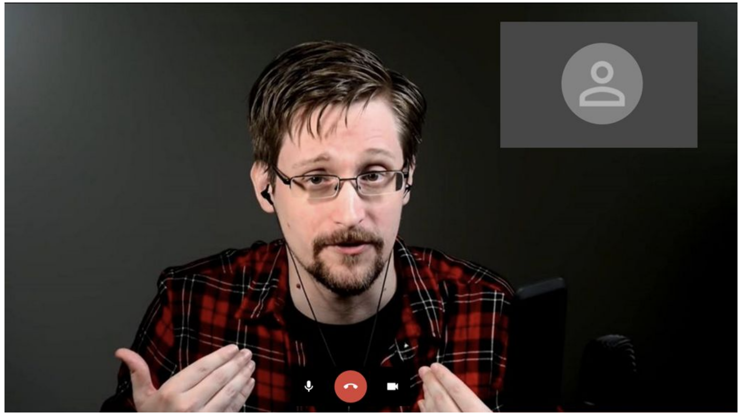
Issue 2412: Is the testing policy being applied/working? (Harald)

- Intent of testing policy
  - Whoever suggests a protocol change also supplies a test
  - Test suite is always in sync with spec, implementations trail
- Practice
  - Whoever implements a protocol change also makes a test
  - Spec leads, tests roughly follow fastest implementor
  - Not all aspects of protocol changes get tested
  - Some features have no WPT test coverage (e.g. simulcast), others lack meaningful tests (e.g. degradationPreference)
  - Policy more focused on WPT than KITE tests.
  - Irregular review of WPT Issues and PRs:
    - 22 open WPT Issues, 10 more than a year old
    - 13 unmerged PRs, 5 more than 6 months old
- What can we do better?
  - Abandon testing policy?
  - Ask for resources for testing policy?
  - Adopt “no test - no merge”?
  - Once specification “final CR” issues, shift focus to review of test Issues and PRs?

Issues for Discussion Today

- Media Capture and Streams
  - PING's Privacy-By-Default flow (Jan-Ivar)
    - Issue 640: Only reveal labels of devices granted permission
    - Issue 656: No way to choose correct camera & mic upfront
    - Issue 649: #632 broke ability for sites to override Firefox picker
    - Issue 652: In-content device selection a mistake. Leaks; Complex
  - Issue 642: Only Firefox turns off device on disabled track. Stronger language needed? (Jan-Ivar)
  - Issue 655: How to avoid wide-lens back-facing cam on new phones (Jan-Ivar)
  - Issue 639: Enforcing user gesture for getUserMedia (Youennf)

Issues 640 / 656 / 649 / 652: PING's Privacy-By-Default flow (Jan-Ivar)



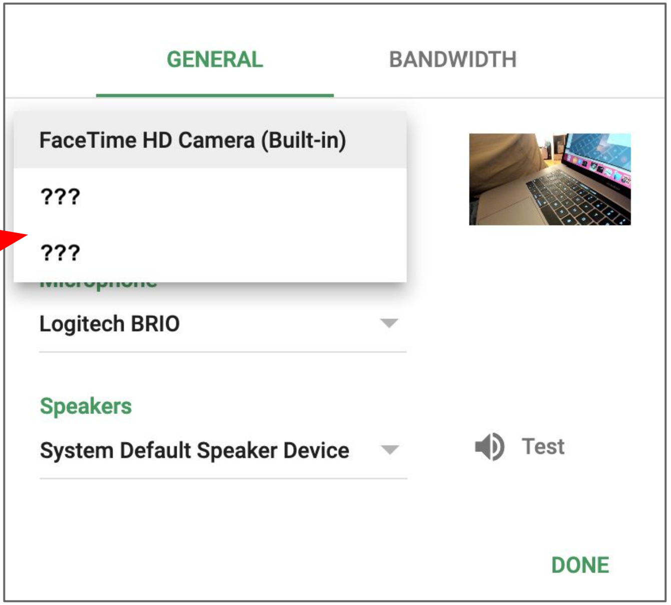
Why now?

- **PING 2019 review raised privacy concerns**
- Privacy climate has changed for legit reasons
- Web sites are tracking users bigtime:
  - enumerateDevices() 2017 usage 0.07%
  - enumerateDevices() 2019 usage > 4.5% of all page loads
  - getUserMedia() remains at 0.06%, the rest are trackers
- In 2020, exposing all the user's devices beyond the one they're using, is not POLA. It goes beyond fingerprinting, revealing actual private information users did not intend to share about what they own and have plugged in (such as prototype devices, "adult" devices, etc.)
- It's not **"the minimal information needed to achieve user goals"** (PING quote).

Issue 640: Only reveal labels of devices user has given permission to

“Solution”: Put `label` behind gum permission

But Firefox shares all labels *instead of* all devices, so taken alone, this would break device selection, forcing Firefox to grant all devices to make it work, Which is worse for privacy. Not **PING**’s intention.



PING wants a 2020 privacy-by-default flow:

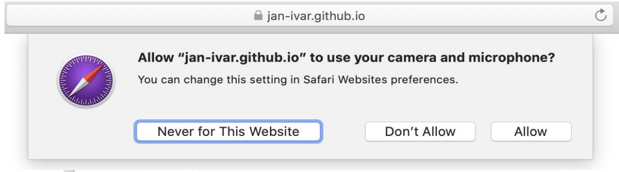
1. site asks for category (or categories) of device
2. *browser* prompts user for one, many or all devices
3. site gains access to only the device + label, of hardware the *user selects*.

This is *in-browser AKA in-chrome selection*. Can the spec accommodate this? We need an API that works well in all browsers (even more private ones).

Issue 656: No way to choose correct camera & mic upfront

Most browsers say *“use your camera & microphone”* but don’t tell you which ones if you have several.

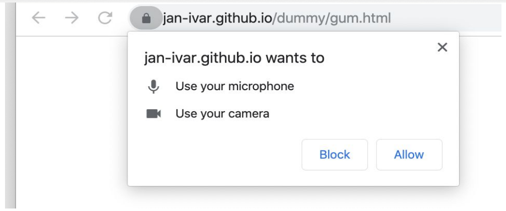
- If it's wrong, users need to correct it after the fact
- Web compat issue: browsers may not choose same (E.g. if Chrome has headset detection, sites may 🤷‍♂️)



jan-ivar.github.io

Allow “jan-ivar.github.io” to use your camera and microphone?

You can change this setting in Safari Websites preferences.



jan-ivar.github.io/dummy/gum.html

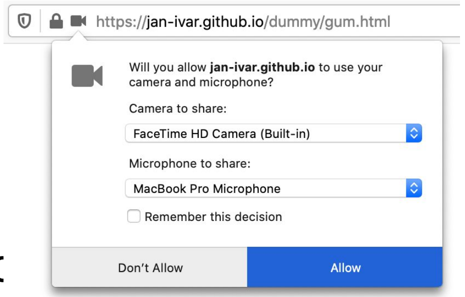
jan-ivar.github.io wants to

Use your microphone

Use your camera

Firefox shows cam & mic used + lets user change it (within app constraints)

- User may override `ideal` (default), but not `exact`.
- But not everyone w/multiple devices use Firefox



https://jan-ivar.github.io/dummy/gum.html

Will you allow **jan-ivar.github.io** to use your camera and microphone?

Camera to share: FaceTime HD Camera (Built-in)

Microphone to share: MacBook Pro Microphone

☐ Remember this decision

Better: User gets to choose based on how many devices they have, not what browser they use.
(maybe regardless of permission if app is indecisive on subsec

It feels like this should be an app decision, not a trait of one browser.

Issue 649: #632 broke ability for sites to override Firefox’s picker

Today in Firefox, you can override its picker by forcing the default device:

```
const devices = await navigator.mediaDevices.enumerateDevices();

const exact = devices.find(({kind}) => kind == "videoinput").deviceId; // use 1st cam
await navigator.mediaDevices.getUserMedia({video: {deviceId: {exact}}});
```

...ensuring the same behavior in all browsers.

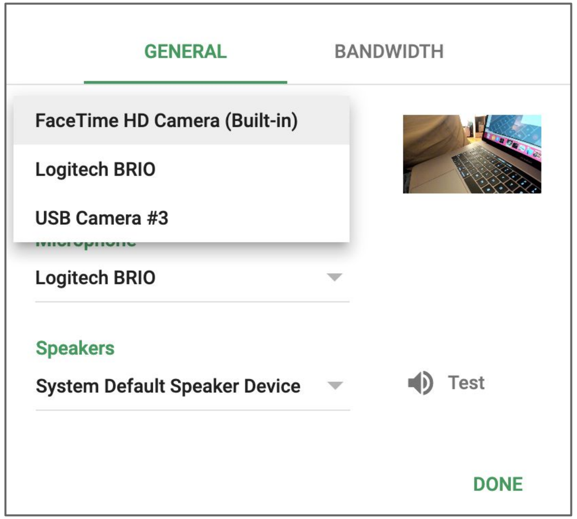
But #632 removed deviceId from enumerateDevices initially, so this no longer works.

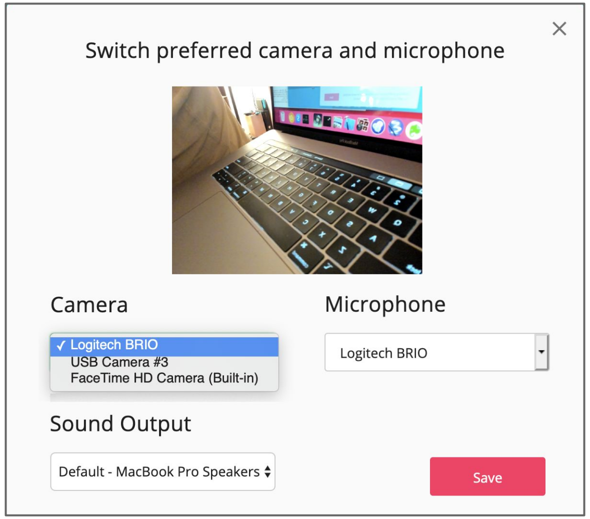
Alternatives being discussed in issue involve standardizing a new `“default”` deviceId.

But hold that thought.

Issue 652: In-content device selection is too complicated, leaky (jib)

After 7 years, the state of the art of in-content device selection is still decidedly basic:



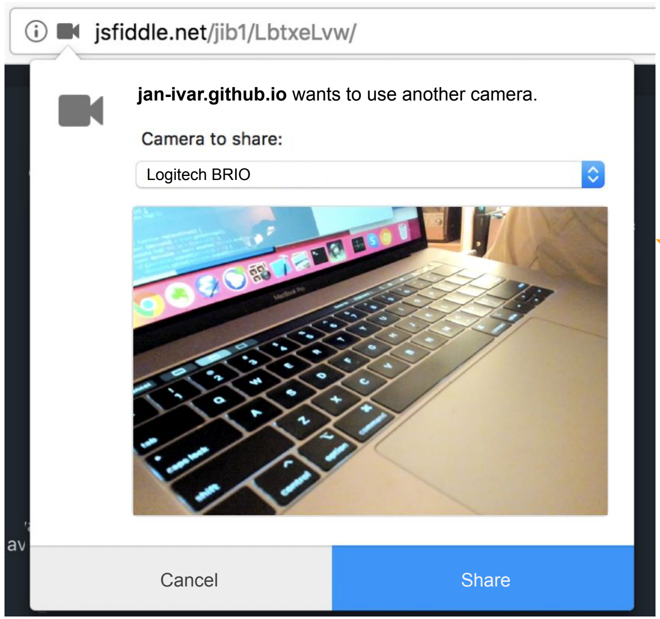

Switch preferred camera and microphone
Camera
✓ Logitech BRIO
USB Camera #3
FaceTime HD Camera (Built-in)
Microphone
Logitech BRIO
Sound Output
Default - MacBook Pro Speakers
Save

Problems:

- **Leaks private info**—it's the sole reason we expose labels of all the user's devices. Fails PING review #640
- **Limited by permission envelope**—no way to ensure correct camera & microphone upfront #656
- **Clunky without persistent permission**—UX assumes all devices granted up-front to work effectively ("**double-prompts**" in Firefox)
- **Poor web compat (few sites get everything right)**—having every site write a decent picker compatibily has been an abject failure:
    - Exhibit A: webrtc samples still re-prompts *both tracks* & flickers. whereby.com too + mixes front vs. back on Firefox mobile.
    - Mobile devices usually can't open more than one device at a time, requiring "stop-then-pick" approach (inferior on desktop).
    - Limited previews. Designs are impeded by A) different browser permission models, B) mass previews expose site power = creepy.
- **Inconsistent user experience**—every site is on its own configuring this user preference. Customization potential hasn't materialized.
- **Model is inherently limiting**—no path to privacy (relies on leaking labels. Can't avoid redundant re-prompts *after* user selection)

Issue 652: In-content device selection too complicated, leaky (jib)

In hindsight, we could have done everything in-chrome (modeled on success of getDisplayMedia)

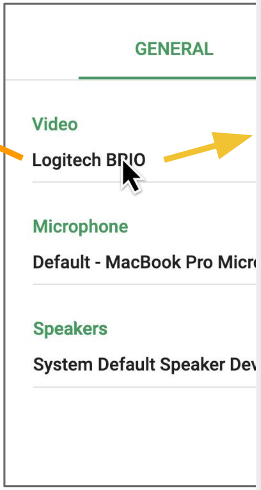


jsfiddle.net/jib1/LbtxeLvw/

jan-ivar.github.io wants to use another camera.

Camera to share:

Logitech BRIO



Video

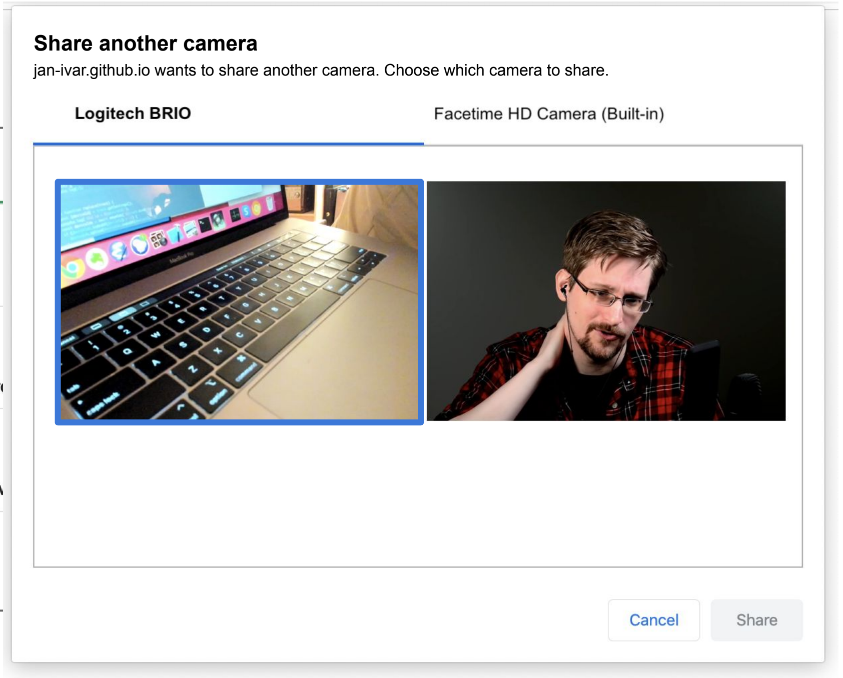
Logitech BRIO

Microphone

Default - MacBook Pro Micr

Speakers

System Default Speaker Dev



Share another camera

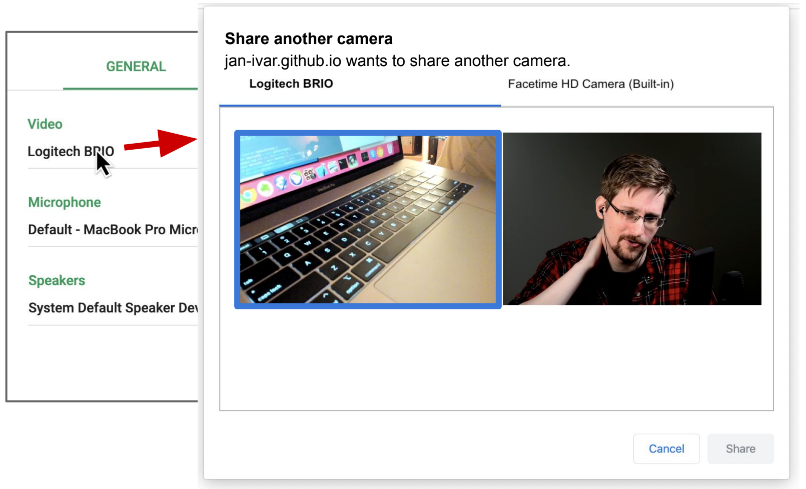
jan-ivar.github.io wants to share another camera. Choose which camera to share.

- In-chrome selectors remove the need to grant permission to all devices + works upfront! ❤️
- Desktop: UAs can implement previews safely for all (even non-chosen) cameras (in a grid)
- Mobile: UAs can handle tricky platforms that can’t open multiple devices (e.g. temporary mute)

Issue 652: In-content device selection too complicated, leaky (jib)

Transition: In-chrome selection would let us:

- **Stage 1:** Put label behind gum permission. Might compete on merit w/in-content selection
- **Stage 2:** Remove label in `enumerateDevices` (Deprecate in-content selection)
- (keep `deviceId` & number of devices/kind)
- `track.label` still gives current device label



GENERAL

Video
Logitech BRIO

Microphone
Default - MacBook Pro Micr

Speakers
System Default Speaker Dev

Share another camera
jan-ivar.github.io wants to share another camera.

Logitech BRIO | Facetime HD Camera (Built-in)

Semantic difference between initial permission vs. adding/changing to another device:

(Without discussing APIs yet) User agents *may* detect this solely from context (already live track):

1. Wording: s/Share/Share another/ or perhaps “Change”, based on UA permission model.
2. Less about permission, more about choosing another device.
3. Benign “Cancel” option instead of the all-permission-revoking “Get lost!” Deny option.
4. Mass grid previews that might have been surprising or creepy upfront may be OK now.

Default choice may be passed in by app (Firefox today uses `{deviceId}` without `exact` for this).

PR 644: Prompt user to choose unless constraints reduce to 1 (jib)

Proposal A: Mandate in-chrome gUM selector (a'la Firefox) ***when*** a user has multiple devices ***and*** constraints don't reduce to 1 (removes user agent as a decision maker). May over-prompt code calling gUM too often, though most users won't notice. *Changes*:

User with previously ***persisted*** permission for camera:	Single camera	Front/back camera	Multiple cameras

`getUserMedia({video: true})`
`getUserMedia({video: {facingMode: "user"}})`
`getUserMedia({video: {deviceId: cameraId}})`
	Granted!	*Selector!*	*Selector!*

`getUserMedia({video: {facingMode: {exact: "user"}}})`	Granted!	Granted!	*Selector!*
`getUserMedia({video: {deviceId: {exact: cameraId}}})`	Granted!	Granted!	Granted!

User *without* persisted permission: replace "Granted!" with "Prompt!" (or single-choice "Selector")

PR 644: Prompt user to choose unless constraints reduce to 1 (jib)

Proposal A (continued): Enshrine new behavior in the spec:

```
-  1. Request permission to use a
+ 1. Let descriptor be a
       PermissionDescriptor with its name member set to the permission name associated with kind
       (e.g. "camera" for "video", "microphone" for "audio"),
-      and, optionally, consider its deviceId member set to any appropriate device's deviceId,
       while considering all devices attached to a live and same-permission MediaStreamTrack in the
       current browsing context to have permission status "granted",
+ 2. If the number of unique devices sourcing tracks of media type kind in candidateSet is 1,
+      then set descriptor's deviceId member to the deviceId of the sole device, and
+      request permission to use the sole device with descriptor, resulting in provided media.
+      Otherwise, prompt the user to choose a device with descriptor,
       resulting in provided media.
```

It's always been a website's job to use constraints and deviceId to manage a user's device(s). When done correctly using `exact` there's no change in behavior.

BUT! Backward compat issue with sites using ideal `{deviceId: id}` on revisit: Multi-device users (only) would get picker every visit, until site updates to use `exact`.

PR 644+: Optional prompt to choose unless constraints reduce to 1

Proposal B: Add a new getUserMedia boolean for this new **A** behavior:

```
await navigator.mediaDevices.getUserMedia({video: true, chosen: true});
```

"Chosen" means tracks must be chosen by the user (or app), *not* the user agent.

Applies to both audio and video if present. Happy to bikeshed name! `{PING: true}`

Proposal C: Same as **B** but as a new method:

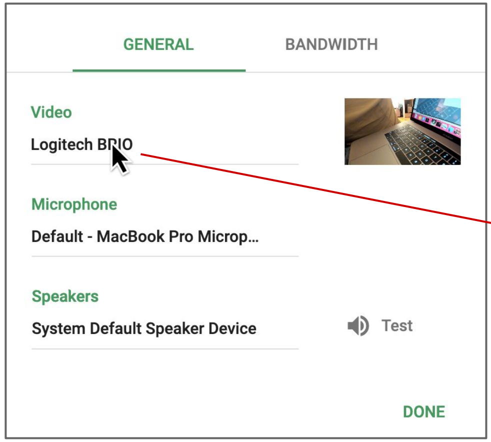
```
await navigator.mediaDevices.chooseUserMedia({video: true});
```

Avoids boolean arg, but may be overkill when everything else is the same / falsely suggesting more differences / might lead to more refinement / stall?

PR 644+: A: Optional prompt to choose unless constraints reduce to 1

Example: settings button launches in-chrome picker to change camera:

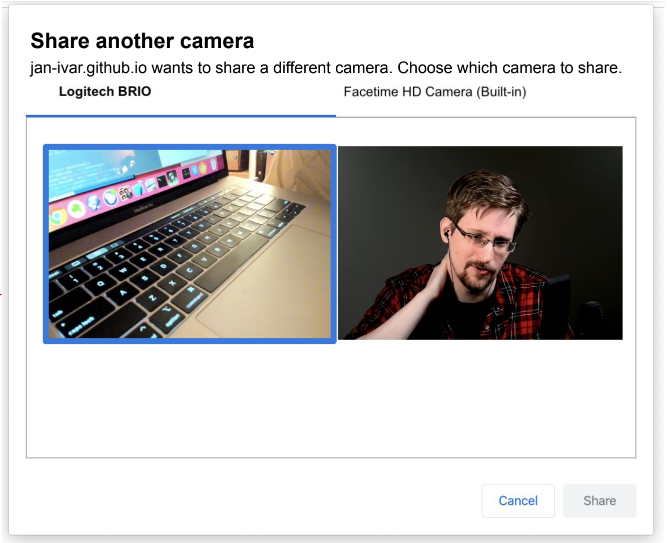
```
button.onclick = async () => {
  if (numberOfVideoInputDevices < 2) return;
  const constraints = {
    video: {deviceId: cameraTrack.getSettings().deviceId}, chosen: true // pass in existing id
  };
  cameraTrack = (await navigator.mediaDevices.getUserMedia(constraints)).getVideoTracks()[0];
  button.innerText = cameraTrack.label;
}
```



Video
Logitech BRIO

Microphone
Default - MacBook Pro Microp...

Speakers
System Default Speaker Device



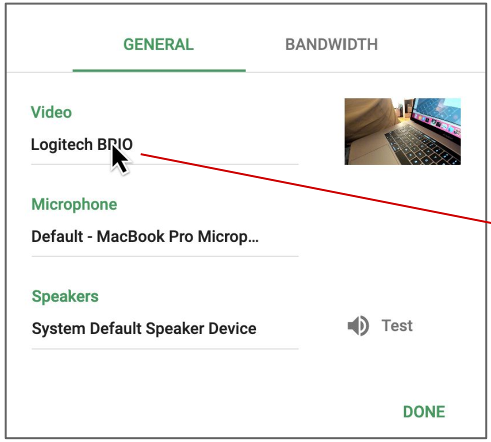
Share another camera

jan-ivar.github.io wants to share a different camera. Choose which camera to share.

B: New choose method prompts unless constraints reduce to 1

Example: settings button launches in-chrome picker to change camera:

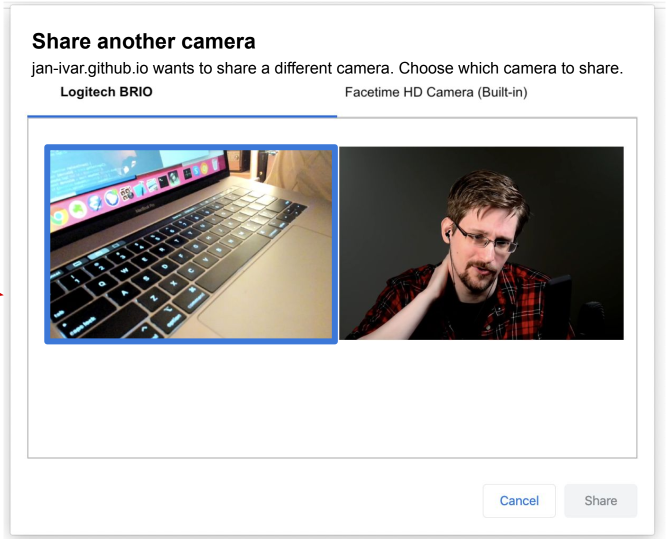
```
button.onclick = async () => {
  if (numberOfVideoInputDevices < 2) return;
  const constraints = {
    video: {deviceId: cameraTrack.getSettings().deviceId} // use existing id as default
  };
  camTrack = (await navigator.mediaDevices.chooseUserMedia(constraints)).getVideoTracks()[0];
  button.innerText = cameraTrack.label;
}
```



Video
Logitech BRIO

Microphone
Default - MacBook Pro Microp...

Speakers
System Default Speaker Device



Share another camera

jan-ivar.github.io wants to share a different camera. Choose which camera to share.

PR 644/+: (Optional) prompt to choose unless constraints reduce to 1

Solves 640: Only reveal labels of devices user has given permission to

Lets us put deviceInfo.label behind gum permission + eventually kill it.

Solves 656: No way to choose correct camera & mic upfront

Apps can decide whether they prefer a picker upfront in all browsers:

```
await navigator.mediaDevices.getUserMedia({video: true, chosen: true}) // User picks
await navigator.mediaDevices.getUserMedia({video: true}) // User agent picks
```

Solves 649: Broke ability for site to say "give me first device" even in Firefox

No need for `“default”` deviceId to always pick default device upfront (provided Firefox gives up its picker when chosen is false)

Solves 652: In-content device selection too complicated, leaky

Model no longer requires mass permission. UAs handle previews & tricky platforms. Better overall cross-browser web experience & compat.

Issue 640: Only reveal labels of devices user has given permis.. (Youenn)

If difficult to enforce per-device exposure rule, enforce per-device-type exposure

- Expose all microphones if one microphone is granted
- Expose all cameras if one camera is granted
- Do not expose speakers once output speaker picker API is available

Still possible to use `groupId` to get microphone corresponding to camera

Issue 642: Only Firefox turns off device on disabled track. Stronger language needed? (Jan-Ivar)

Live cam/mic tracks can be turned on/off with `track.enabled`. Sole use case:

```
    videoMute.onclick = () => cameraTrack.enabled = !muted.checked;
```

Semantically, this is the spec-agreed abstraction for web sites, regardless of UA permission model. But users expect the privacy of not staring at the hot camera (or microphone) hardware light.

Spec agrees[1]: ”when a track becomes either muted or disabled, and this brings all tracks connected to the device to be either muted, disabled, or stopped, then the UA *MAY*, using the device's deviceId, *deviceId*, set [[devicesLiveMap]][*deviceId*] to false”

Firefox agrees, and implements the spec advice. See blog Better privacy on camera mute

Meet/Hangouts agrees, but instead hacks it (camera only) to work in Chrome by stopping track and calling getUserMedia() again, which doesn't work with one-time permission. This backfires with needless re-prompt on unmute in Firefox.   A web compat problem.

1) The spec doesn't actually mention hardware, only “privacy indicators”. However, this omission seems consistent with the overall design of this spec (e.g. constraints, mute/unmute events etc.) where user agents are left to manage hardware however they best see fit within the constraints outlined by the API. Having physical and logical indicators align seems a reasonable end-user expectation; the spec notably does not say this intuitive interpretation is forbidden.

Issue 642: Only Firefox turns off device on disabled track. Stronger language needed? (Jan-Ivar)

How Firefox works (fiddle):

1. Relinquishes device when all its tracks are disabled.
2. Reacquires device when any of its tracks are re-enabled (~1 second to reacquire hardware device)
3. Failure to reacquire fires ended event on track(s).
4. *Live* indicator always on for 3 seconds minimum (*"MUST remain observable for a sufficient time"*)
5. *Accessible* mandated Privacy Indicator remains in URL bar while muted.
6. Privacy mitigation plan is to have Firefox fire muted event if re-enabled without focus (Bug 1598374)

Ask other vendors to do what Firefox is doing, to give web devs a consistent API that works in all browsers, even ones with one-shot permissions.

Option A: Stronger language to enforce web compat around this behavior in all browsers
Option B: Drop track.enabled?
Option C: Do nothing (leaves websites having to feature detect to avoid both HW light & prompt)
Option D: We think existing language is strong enough. File bugs on vendors.

Issue 655: How to avoid wide-lens back-facing cam on new phones (jib)

Flagship phones have multiple back cameras now. How can apps distinguish them?
This SO question asks how to avoid (or pick) the often unsuitable wide-lens camera.

Wide-lens often (always?) means fixed-focus, represented by 0 in android, so this

```
{video: {focusDistance: {exact: 0}}}    // pick wide-lens video camera
{video: {focusDistance: {min: 0.0001}}} // avoid wide-lens video camera
```

...might work. Defined in ImageCapture (no implementation). What if you have several?

Proposal: A new *focalLength* constraint.

```
{video: {focalLength: {min: 0.0026}}} // avoid all wide-lenses < 26mm
```

This would be the distance between *sensor* and lens (not lens and object)
This seems to often be an inherent property of the lens, e.g. on the Samsung S10.

Q: Where to specify new MediaStreamTrack constraints? Here or ImageCapture?

Issue 639: Enforcing user gesture for getUserMedia (Youenn)

- Problem: getUserMedia should only be callable on user gesture
  - Most modern APIs add such restrictions
  - This is not web compatible
- Can we start shipping such restrictions in getUserMedia?
- Potential ideas
  - Require a user gesture past initial page load
  - Require user gesture once a previous call to getUserMedia was denied for the given page
    - Implemented in Safari

RTP Header Extensions (Harald)

This section:

```
a=extmap:14 urn:ietf:params:rtp-hdrext:toffset
a=extmap:2 http://www.webrtc.org/experiments/rtp-hdrext/abs-send-time
a=extmap:13 urn:3gpp:video-orientation
a=extmap:3 http://www.ietf.org/id/draft-holmer-rmcat-transport-wide-cc-extensions-01
a=extmap:12 http://www.webrtc.org/experiments/rtp-hdrext/playout-delay
a=extmap:11 http://www.webrtc.org/experiments/rtp-hdrext/video-content-type
a=extmap:7 http://www.webrtc.org/experiments/rtp-hdrext/video-timing
a=extmap:8 http://tools.ietf.org/html/draft-ietf-avtext-framemarking-07
a=extmap:9 http://www.webrtc.org/experiments/rtp-hdrext/color-space
a=extmap:4 urn:ietf:params:rtp-hdrext:sdes:mid
a=extmap:5 urn:ietf:params:rtp-hdrext:sdes:rtp-stream-id
a=extmap:6 urn:ietf:params:rtp-hdrext:sdes:repaired-rtp-stream-id
```

The need:

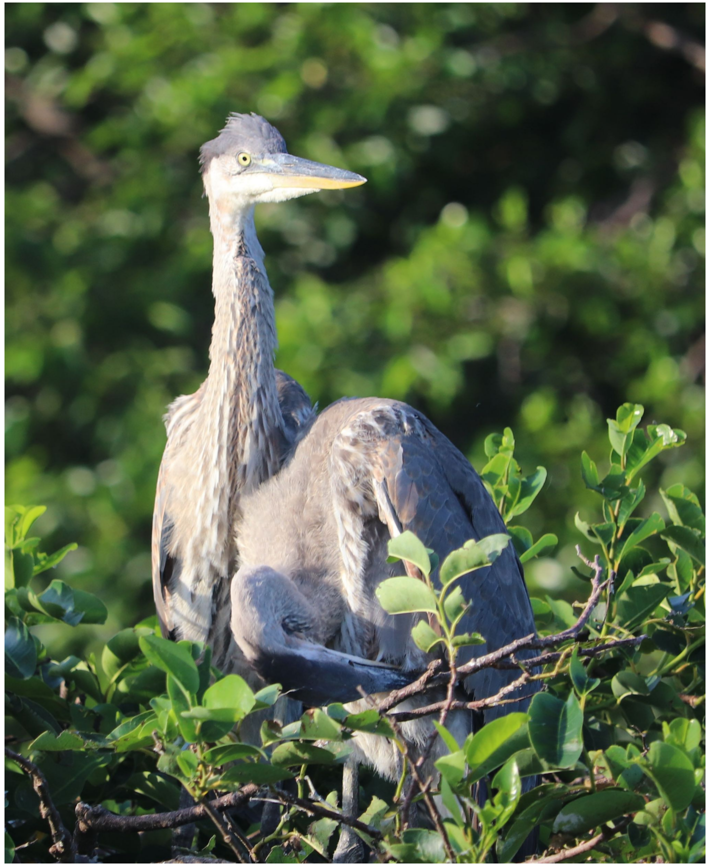
- Controlling what RTP header extensions get negotiated in SDP
  - SDP munging is not a long term viable method
- Controlling what RTP header extensions get sent in RTP

RTP header extensions (Harald)

The proposal

- Add to RTCRtpTransceiver a control for negotiation
  - Modeled on setCodecPreferences()
  - Affects negotiation only
- Add to RTCRtpSender a control for usage
  - Field in RTCRtpHeaderExtensionsParameters
  - Used with getParameters/setParameters
  - Can only enable/disable negotiated header extensions
  - Need to validate header extension information provided to setParameters.
- Detailed proposal
- Next steps:
  - Prepare a PR on webrtc-extensions
  - Ask for approval to merge

For extra credit



Name that bird!

Thank you

Special thanks to:

WG Participants, Editors & Chairs

The bird