W3C WebRTC/MediaCapture WG Meeting

December 12, 2016 1 PM PDT

Chairs: Harald Alvestrand

Stefan Hakansson

W3C WG IPR Policy

- This group abides by the W3C patent policy https://www.w3.org/Consortium/Patent-Policy-20040205
- 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 outstanding issues within both the mediacapture -main and webrtc-pc specifications
- Editor’s Draft updates to follow meeting

Limited editor resources for a period

- During December and January the webrtc-pc (and mediacapture-main) editor availability will be lower than normal
- To help out we ask everyone to, when possible, file not only Issues but also proposed solutions (in the form of PRs)

About this Virtual Meeting

Information on the meeting:

- Meeting info:
  - https://www.w3.org/2011/04/webrtc/wiki/December_12_2016
- Link to latest drafts:
  - https://rawgit.com/w3c/mediacapture-main/master/getusermedia.html
  - https://rawgit.com/w3c/webrtc-pc/master/webrtc.html
- 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

- **Media Capture Issues**
  - Issue 350: New permission definitions are wrong (Jan-Ivar)
  - Issue 380: Define restrictions on device-info permission (Harald)
  - Issue 387: Reinstate strong language on permission ending when tracks stop (Stefhak)
  - Issue 403: Polling enumerateDevices potentially being a fingerprint (Bernard)
  - Issue 414: Devicechange events when not focus - permitted or forbidden? (Shijun)
  - Issue 417: result of enumerateDevices when there is no origin (Shijun)

For Discussion Today (cont'd)

- **WebRTC-PC**
  - **Pull Requests**
    - Issue 952/PR 953: Hold Examples (Bernard)
    - Issue 714/PR 776: STUN/TURN OAuth token parameter (misi)
    - Issue 760/Issue 726/PR 968: Adding ufrag to candidates, and ufrag+mid to end-of-candidates (Taylor)
  - **Issues**
    - Issue 849: AllowUnverifiedMedia RTCConfiguration Property (Fluffy)
    - Issue 921: currentRemoteDescription.sdp - does it need to match the last SDP set via setRemoteDescription? (Bernard)
    - Issue 924: Remove legacy getStats API? (Harald)
    - Issue 945: setParameters changing simulcast parameters (Bernard)
    - Issue 941: STUN/TURN auto discovery handling (misi)

Media Capture Issues

- Issue 350: New permission definitions are wrong (Jan-Ivar)
- Issue 380: Define restrictions on device-info permission (Harald)
- Issue 387: Reinstate strong language on permission ending when tracks stop (Stefhak)
- Issue 403: Polling enumerateDevices potentially being a fingerprint (Bernard)
- Issue 414: Devicechange events when not focus - permitted or forbidden? (Shijun)
- Issue 417: result of enumerateDevices when there is no origin (Shijun)

Issue 350: New permission definitions are wrong (Jan-Ivar, Harald)

- Recent context changes
  - The “Feature policy” proposal seems to be the new hotness in delegating permissions to iframes
  - The “request” method is back in permissions API (is it?)
- New proposed language (as agreed at TPAC)
  - Only text remaining here is PR 421 [preview] I (jib) think

Issue 380: Define restrictions on device-info permission (Harald)

- Relevant PR: permissions/131
  - Explicitly sets device-info permission for realm w

    a device is granted access
  - Always revoke when info on a realm is cleared
  - Note: A realm is a browsing context, not an origin
- (Merged) language in mediacapture-main
  - Change device-info on foreground tabs (but see #414)
  - Event fires in sync with device-info changing

Issue 387: Reinstate strong language on permission ending when tracks stop (Stefhak)

- Spec used to say "`<all tracks stopped =>> the source is stopped. Unless there is a stored permission for the source in question, the given permission is revoked and the User Agent SHOULD also remove the "permission granted" indicator for the source.`"
- Issue basically says "revoke" is gone from spec at that time (August)
- Now, the spec refers to "Permission state" in permission spec, and if that is not "granted" then "`set [[devicesAccessibleMap]] [deviceId] to false.`"
- Seems to fix Issue 387 to me, comments?
- (separate: add use of `[[devicesAccessibleMap]]`, see PR#421 [preview])

Issue 403: Polling enumerateDevices potentially being a fingerprint (Bernard)

- Concern raised by @npdoty (in privacy review) and in Issue 333:
  - Particularly if this event [DeviceChange] will be fired before any permission is granted, it is important that it not be fired simultaneously in all browsing contexts. Sites can use simultaneous firing to correlate browsing activity in different tabs, different windows (including private windows), different browsers, in a way that may be unexpected to the user and undermine other protections they're attempting to implement.
  - Spec encourages fuzzing the timing on firing the event, but does not require it.
- How to address use of enumerateDevices for the same purpose?
  - Harald: This can be closed if we specify that the devicechange event is always fired before enumerateDevices() returns new information, right?
  - PR 412: Mandates that enumerateDevices() return old data until the event has fired.

Issue 414: Devicechange events when not focus - permitted or forbidden? (Shijun)

- The current spec requires that when adding or removing input/output devices, the devicechange event MUST be fired, when the following is true:
  - Permission is granted, or
  - A local device is attached to an active mediaStream, or
  - Document is fully active and has focus
- Question - Whether user agent MUST NOT or MAY fire devicechange event when:
  - Permission not granted, and
  - No local device attached to active mediaStream, and
  - Document fully active but not in focus
- Options
  - **MUST NOT** - enforces security protection against fingerprinting (Issue 403)
  - **MAY** - permits current behavior and allows more time for browser vendors to catch up with the specific security protection.

Issue 417: result of enumerateDevices when there is no origin (Shijun)

- The current security model requires deviceId's be unique per origin. The behavior is not defined when there is no origin, e.g., a JavaScript console for a "about:blank" page.
- Github discussions
  - Since no cookies or other data are stored, the deviceID should be unique per session and will not be persistent across sessions.
  - Whether making sense to reject the promise? It'd be nice to keep the JavaScript console as a valid test option for web developers (for example, as in Firefox and Edge).
- Proposal
  - Allow enumerateDevices() to return successfully with unique deviceId's, but do not persist any deviceId across sessions.

WebRTC PC Pull Requests

- Issue 952/PR 953: Hold Examples (Bernard)
- Issue 714/PR 776: STUN/TURN OAuth token parameter (Misi)
- Issue 760/Issue 726/PR 968: Adding ufrag to candidates, and ufrag+mid to end-of-candidates (Taylor)

Issue 952/PR 953: Hold Examples (Bernard)

EXAMPLE 3: To send music to a peer and cease rendering received audio (Music on Hold)

```
To send music to a peer and cease rendering received audio:
// Assume we have an audio transceiver and a music track named musicTrack
audio.sender.replaceTrack(musicTrack);
// Set the direction to send-only (requires negotiation)
audio.setDirection("sendonly");
```

Issue: starts playing music immediately, but only stops rendering the received track after negotiation. Proposed fix:

```
// Assume we have an audio transceiver and a music track named musicTrack
audio.sender.replaceTrack(musicTrack);
// Mute received audio
audio.receiver.track.enabled = "false";
// Set the direction to send-only (requires negotiation)
audio.setDirection("sendonly");
```

Issue 952/PR 953: Hold Examples (cont'd)

EXAMPLE 4: To stop sending audio to a peer

```
var params = audio.sender.getParameters();
params.encodings[0].active = false;
audio.sender.setParameters(params);
```

Issue: Doesn't include the context (response to Example 3's "sendonly" offer), stops a single audio stream. Proposed fix:

```
// In response to a remote peer's "sendonly" offer:
// Mute the outgoing audio (sends silence)
audio.sender.track.enabled = "false";
// Set the direction to recv-only (requires negotiation)
audio.setDirection("recvonly");
```

Issue 952/PR 953: Hold Examples (cont'd)

- EXAMPLE 5: To re-enable sending audio captured from a microphone as well as rendering of received audio

```
//assume we have an audio transceiver and a microphone track named micTrack
audio.sender.replaceTrack(micTrack);
// Set the direction to sendrecv (requires negotiation)
audio.setDirection("sendrecv");
```

Issue: Explanation doesn't provide context (to remove Music on Hold).
Proposed fix: add context to the text.

Issue 952/PR 953: Hold Examples (cont'd)

- EXAMPLE 6: To re-enable sending audio to a peer

```
var params = audio.sender.getParameters();
params.encodings[0].active = true;
audio.sender.setParameters(params);
```

Issue: Explanation doesn't provide context (response to Example 5's being taken off hold), only re-enables a single stream, doesn't negotiate the change in direction. Proposed fix:

```
To respond to being taken off hold
// Stop sending silence
audio.sender.track.enabled = "true";
// Set the direction sendrecv (requires negotiation)
audio.setDirection("sendrecv");
```

Issue 714/PR 776: STUN/TURN OAuth Token Parameter

- PR update soon
- Filed by Misi: How is RFC 7635 (STUN Extension for OAuth 2.0) supported within RTCIceServer? Currently, we have:

```
dictionary RTCIceServer {
    required (DOMString or sequence<DOMString>) urls;
             DOMString                          username;
             DOMString                          credential;
             RTCIceCredentialType               credentialType = "password";
};

enum RTCIceCredentialType {
    "password",
    "token"
};
```

Issue 714/PR 776 STUN/TURN OAuth Token (cont'd)

Options (see issue 714 for details):

1. **Add another attribute.**
   https://github.com/misi/webrtc-pc/tree/issue-714-patch
2. **Fully separate password, oauth/token auth, (future).**
   https://github.com/misi/webrtc-pc/tree/issue-714-patch2
3. **Hybrid.**

Preferences:

- Juberti: 3, 1, 2
- Misi: 2, 1, 3
- Others?
- Chairs?

Consensus, Decision?

Issue 714/PR 776 STUN/TURN OAuth Token (cont'd)

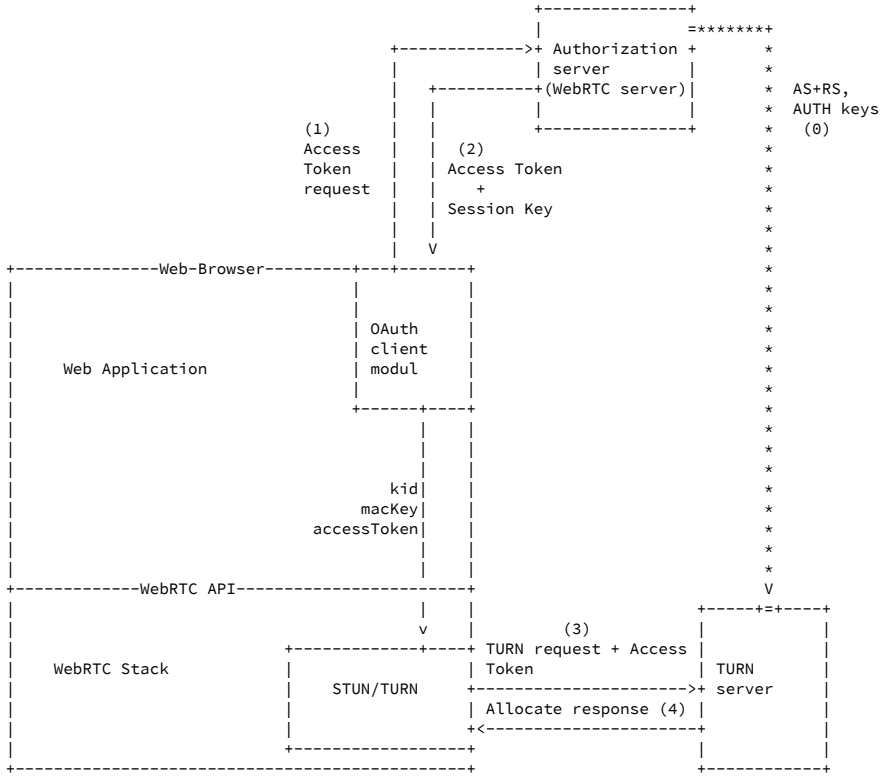
- RFC 7635 Appendix B example of a token credential:

```
{
   "access_token":
"U2FsdGVkX18qJK/kkWmRcnfHglrVTJSpS6yU32kmHmOrfGyI3m1gQj1jRPsr0uBb
HctuycAgsfRX7nJW2BdukGyKMXSiNGNnBzigkAofP6+Z3vkJ1Q5pWbfSRroOkWBn",
   "token_type":"pop",
   "expires_in":1800,
   "kid":"22BIjxU93h/IgwEb",
   "key":"v51N62OM65kyMvfTI08O"
   "alg":HMAC-SHA-256-128
}
```

Issue 714/PR 776 STUN/TURN OAuth Token (cont'd)

- According last meeting decision, removed the STUN client supported HMAC alg(s) list
  - TODO: document static key length: 256 bit
  - OAuth Client param "alg":"HS256" ?
- PoP OAuth key distribution draft
  "key" param is JWK or JWE encrypted JWK
  - So Web app need to extract from "key" OAuth param the base64(mac_key) that is in "k" param in JWK
  - Example JWK

```
{
 "kty":"oct",
 "kid":"id123",
 "alg":"HS256",
 "k":"ZoRSOrFzN_FzUA5XKMYoVHyzff5oRJxl-IXRtztJ6uE"
}
```



Issue 760/Issue 726/PR 968: Adding ufrag to candidates, and ufrag+mid to end-of-candidates (Taylor)

- **Background:** We should have the ufrag included with ICE candidates so you could disambiguate candidates and end-of-candidates from different ICE generations (restarts). We need to:
  - Add ufrag to ICE candidate event and addIceCandidate
  - Also add ufrag and mid to the “done gathering” indication

Issue 760/Issue 726/PR 968: Adding ufrag to candidates, and ufrag+mid to end-of-candidates (cont'd)

- Simplest solution is in PR 968 (updated version of PR 757):
  - Normal candidates are still:

    `{"sdpMid":"foo","ufrag":"frag", "candidate":<candidate SDP blob>}`
  - end-of-candidates is:

    `{"sdpMid":"foo","ufrag":"frag", "candidate":null}`
  - The global "done gathering" null candidate will still be emitted for backwards compatibility.
- Is this acceptable?

Issue 760/Issue 726/PR 968: Adding ufrag to candidates, and ufrag+mid to end-of-candidates (cont'd)

If changing the API, the current best idea is to replace "ICE candidate" with "ICE action":

```
partial interface RTCPeerConnection {
    attribute EventHandler oniceaction;
    Promise<void> handleIceAction(RTCIceAction action);
}

dictionary RTCIceAction {
    // not all members are defined for all types of actions
    required RTCIceActionType type;
    DOMString sdpMid;
    DOMString sdpMLineIndex;
    DOMString ufrag;
    DOMString candidate;  // not defined if type is "end-of-candidates"
}
```

Issue 760/Issue 726/PR 968: Adding ufrag to candidates, and ufrag+mid to end-of-candidates (cont'd)

```
// Would be easy to extend if new action types are added to ICE.
enum RTCIceActionType { "add-candidate", "end-of-candidates" };

interface RTCPeerConnectionIceEvent : Event {
    RTCIceAction getAction();
    RTCIceCandidate? getCandidate();  // To inspect candidate attributes.
    readonly attribute DOMString? url;
};
```

Issue 760/Issue 726/PR 968: Adding ufrag to candidates, and ufrag+mid to end-of-candidates (cont'd)

Usage:

```
pc.oniceaction = evt => {
    signalingChannel.send(JSON.stringify({ iceAction: evt.getAction() }));
};

signalingChannel.onmessage = evt => {
    // ...
    if (message.iceAction)
        pc.handleIceAction(message.iceAction).catch(logError);
};
```

Issue 760/Issue 726/PR 968: Adding ufrag to candidates, and ufrag+mid to end-of-candidates (cont'd)

If we do this, should the new API go on RTCPeerConnection or RTCIceTransport?

Pros of putting it on RTCPeerConnection:

- Just as simple to use as before. No need to wait for transports to be created and hook up events at the right point in time.

Cons of putting it on RTCPeerConnection:

- Requires an extra field (sdpMid) that wouldn't be necessary if it was on RTCIceTransport.
- Doesn't match the object model. Unless you view it as handing an event to the per-PeerConnection "ICE agent", in which case we're fine.

WebRTC PC Issues

- Issue 849: AllowUnverifiedMedia RTCConfiguration Property (Fluffy)
- Issue 921: currentRemoteDescription.sdp - does it need to match the last SDP set via setRemoteDescription? (Bernard)
- Issue 924: Remove legacy getStats API? (Harald)
- Issue 945: setParameters changing simulcast parameters (Bernard)
- Issue 941: STUN/TURN auto discovery handling (misi)

Issue 849: AllowUnverifiedMedia RTCConfiguration Property 1/2 (Fluffy)

- RFC 4572 Section 6.2:
  - [the server endpoint] MUST NOT assume that the data transmitted over the TLS connection is valid until it has received a matching fingerprint in an SDP answer. If the fingerprint, once it arrives, does not match the client's certificate, the server endpoint MUST terminate the media connection with a bad_certificate error, as stated in the previous paragraph.
- The default behavior needs to be not to render this data

(more on next slide)

Issue 849: AllowUnverifiedMedia RTCConfiguration Property (cont'd)

- Alice calls Bob. Bob's browser does ICE while ringing. Once Bob answers, Bob does have alice's fingerprint and can immediately say "hello". Alice will likely receive the RTP before the fingerprint. For applications that display to alice that the speaker is not known and the connection is not secure, the risk of Alice hearing hello is not a big deal. Once Alice's application receives Bob's fingerprint, the application can enable outbound media from Alice and display the call as secure to Bob.
- To support this:
    - transceiver.receiver.track returned by addTransceiver can immediately be hooked up to an audio or video tag
    - A DtlsTransport can provide a limited buffer for unverified media (so as to prevent loss of packets in a key frame)

Issue 921: currentRemoteDescription.sdp - does it need to match the last SDP set via setRemoteDescription? (Bernard)

- Filed by Philipp Hancke:
  - After calling pc.setRemoteDescription(description) and examining pc.remoteDescription.sdp, there are changes (e.g. addition of ICE candidates, lines in different order, etc.).
  - Should remoteDescription.sdp === description.sdp?
- Proposed resolution:
  - It is **not** required that current or pending local or remote description.sdp match the description.sdp provided as an argument to setLocal/setRemoteDescription.
  - Reasoning:
    - Implementations parse and validate description.sdp and then create their own internal representation. In the process additions (ICE candidates), subtractions (a=lines that are not understood) and edits (changing line order) can occur.
    - As a result, even requiring remote/localDescription.sdp to be “equivalent” to description.sdp would be difficult (as would defining “equivalent”).

Issue 924: Remove legacy getStats API? (Harald)

- Legacy getStats takes a function argument
  - Returns a stats object as per current spec
- Chrome implements an older version
  - Different format of stats
  - Different stats names
  - New, conformant API is in progress
- Firefox implements a different name set (dashes changed)
- **Proposal: Delete the legacy API - nobody's going to conform to it anyway**

Issue 945: setParameters changing simulcast parameters (Bernard)

- Section 5.2 states:

“setParameters does not cause SDP renegotiation and can only be used to change what the media stack is sending or receiving within the envelope negotiated by Offer/Answer. The attributes in the RTCRtpParameters dictionary are designed to not enable this, so attributes like ssrc that cannot be changed are read only.”

- Question 1: If a transceiver is constructed with sendEncodings specifying N simulcast encodings, can setParameters be used to increase or decrease the number of simulcast encodings sent?
- Proposed resolution:
  - setParameters must operate within the envelope negotiated by Offer/Answer. Once setLocalDescription has been called, the envelope is set and the number of simulcast encodings sent cannot be changed by setParameters.
  - However, if a transceiver is constructed with sendEncodings specifying N simulcast encodings, and setLocalDescription has not yet been called, setParameters can be used to change the number of encodings and the parameters set will be reflected in the SDP produced by createOffer/createAnswer.

Issue 945: setParameters changing simulcast parameters (Bernard)

- Question 2: Can setParameters be used to activate or inactivate a simulcast encoding being sent?
- Answer: Yes.

sender.setParameters can be used to activate or inactivate one or more simulcast encodings. To stop sending simulcast encoding *i*, set encodings[*i*].active to "false". To start sending simulcast encoding *i*, set encodings[*i*].active to "true".

Issue 941: STUN/TURN auto discovery handling (misi)

- STUN/TURN Discovery
  - Auto discovery needs input: domain name
    - From DHCP domain attribute
    - From Identity domain part (IdP domain?)
  - DNS-SD (NAPTR/SRV based domain)
    - mDNS (_turn._udp.local.)
  - Anycast IP
    - STUN
      - Allocate/Bind?

Issue 941: STUN/TURN auto discovery handling (misi)

- WebRTC engine is in position to provide such discovery service.
  - Works in low level, and so
    - Has access to DHCP domain Attribute
    - IdP domain (identity?)
    - Informed about network changes
      - Could re-Run discovery process on any topology change
    - ICE agent implements STUN/TURN client already
      - Sending out Allocate/Bind? request to anycast address

Issue 941: STUN/TURN auto discovery handling (misi)

- A list of discovered servers passed up to WebApp
  - App decides which is appropriate to use (Trusted, Contracted, etc.)
  - Request credential, and Pass back to PC on usual way.
- Some benefits of integration in PC
  - Faster adoption of TRAM TURN discovery (default discovery)
    - Separated API adoption will take longer
  - Utilize and exploit that WebRTC stack works in low level
    - Reuse STUN/TURN client functionalities
  - Optimization benefits
    - Possible TURN connection pre-establishment after discovery, during credential claiming.

Thank you

Special thanks to:

W3C/MIT for WebEx

WG Participants, Editors & Chairs