Standards Update & Directions – The W3C Part

Daniel C. Burnett, Ph.D.
Aspect WebRTC Expert
StandardsPlay.com

JavaScript APIs – W3C standards

- Status summary
- Recent highlights
  - Output Device Enumeration
  - Promises
  - Authenticated Origins
  - AddStream -> AddTrack
  - RTCRtpSender/Receiver
  - Screen sharing
  - Other tidbits

Status Summary

- Boring! (This is good)
  – A few big topics, then . . .
  – Many issue and pull requests
- Targeting Last Call WD for Media Capture this year
- Trying for Last Call WD for WebRTC 1Q15

Output Device Enumeration/Selection

- Most requested WebRTC feature for Chrome
- Issue: gUM lets you select input but not output
- Proposal:
  - Include output devices in enumeration of devices with new sinkId (just like sourceId for inputs)
  - Permission grant for device actually grants permission for all devices with same group id
  - See https://www.w3.org/wiki/images/d/d6/Output_Device_Selection%2C_TPAC_2014.pdf
- Decision: use as foundation for new output spec, needs coordination with many other groups in W3C

Promises

- W3C wants all async APIs to return Promises rather than using callbacks
- Issue: Promises becoming popular for APIs, e.g.

```
Navigator.mediaDevices.getUserMedia({audio:true, video:true})
.then(gotStream)
.catch(logError);
```

- Decision:
  - Navigator.getUserMedia() will accept callbacks only
  - Navigator.mediaDevices.getUserMedia() will return a Promise only
  - All async RTCPeerConnection methods will accept callbacks and return a Promise

Authenticated Origins

- W3C wants to require authenticated origins, e.g. HTTPS
- Issue: Unauthenticated origins are insecure
- Proposal:
  - Forbid use of HTTP or other unauthenticated origins
- Decision: Specifications will recommend, but not require, that WebRTC content origins be authenticated

AddStream -> AddTrack

- PCs now operate on tracks rather than streams
- Issue: Need better track-oriented connection info and/or controls
- Proposal:
  - RTCRtpSender addTrack(MST track, MediaStream… streams)
  - void removeTrack(RTCRtpSender sender)
  - onaddstream -> ontrack
- Decision: Agreed, done. Some details still TBD. Existing stream commands will move to polyfill library.

RTCRtpSender/Receiver

- New extension objects (originally) from ORTC
- Issue: Need better track-oriented connection info and/or controls
- Proposal:
  - Several layered proposals from Google including info on
    - ICE transports, remote Certs used, selected candidate pair, encoding parameters (get and set for, e.g. pause/resume, maxBitrate)
    - See https://www.w3.org/2011/04/webrtc/wiki/images/6/6c/WebRTC_RTCSender-Receiver%2C_TPAC_2014.pdf
- Decision: Objects added already, ICE info will be added, but other info and controls are under discussion

Screen Sharing

- Second highest request for Google Chrome
- Discussion:
  - Security is tricky, since web sandboxing model assumes one site can't see another's code
  - Proposal is to identify gUM source as display, window, application, e.g., `Navigator.MediaDevices.getUserMedia({audio:true, video:true, source: "display"})`
- Decision: Needs some work, but everyone wants this ☺

Other Tidbits

- Constraints syntax now finalized – see the Media Capture and Streams specification
- Control over DTLS certificate renewal being considered – maybe using WebCrypto?
- Stats API moving into separate document, many more statistics being defined.