W3C WebRTC WG Meeting

March 25, 2021
9:30 - 11 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 March 2021 interim meeting of the W3C WebRTC WG!
    - During this meeting, we will discuss Testing, Insertable Streams, relationship between APIs, getViewportmedia, DataChannel in Workers and transferring MSTs.

About this Virtual Meeting

- Meeting info:
  - https://www.w3.org/2011/04/webrtc/wiki/March_25_2021
- 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-encoded-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 being recorded. The recording will be public.

Issues for Discussion Today

- Testing (Bernard)
- An Insertable Streams Use Case (Harald)
- Relationships between APIs (Bernard)
- getViewportMedia (Jan-Ivar)
- Issue 777: Defaults for getUserMedia (Youenn)
- Issue 64: Transfer RTCDataChannel (Youenn)
- Issue 16: Transfer MediaStreamTrack (Youenn)
- MediaCapture Transform feedback (Youenn)

Test Proposal from Last Meeting

- Build a content reflector that speaks the WebRTC stack - as minimal as possible
- https://github.com/jlaine/aiortc-wpt-demo/
  - only 60 lines of code
  - terminates STUN + DTLS, decrypts SRTP, echoes RTP/RTCP packets via WebSockets
  - uses RTCPeerConnection from aiortc
  - WPT already uses aioquic for quic tests
- Simple tests!
  - create a peerconnection, connect WebSocket
  - get raw packets
    - parsing RTP/RTCP/SCTP...

Status Update

- Some tests written:
  - NACK (potential FF bug),
  - RTCP BYE (webrtc.org bug),
  - VP8-simulcast
- Lessons learned
  - Get more minimal in the server by moving to ORTC
  - Could validate/send STUN or DTLS too
- Next steps
  - Ask WPT folks to give us such a server?
  - Write more tests!
  - Scope of tests? What is “web”, what is IETF
  - See also https://github.com/sipsorcery/webrtc-echoes

Issues for Discussion Today

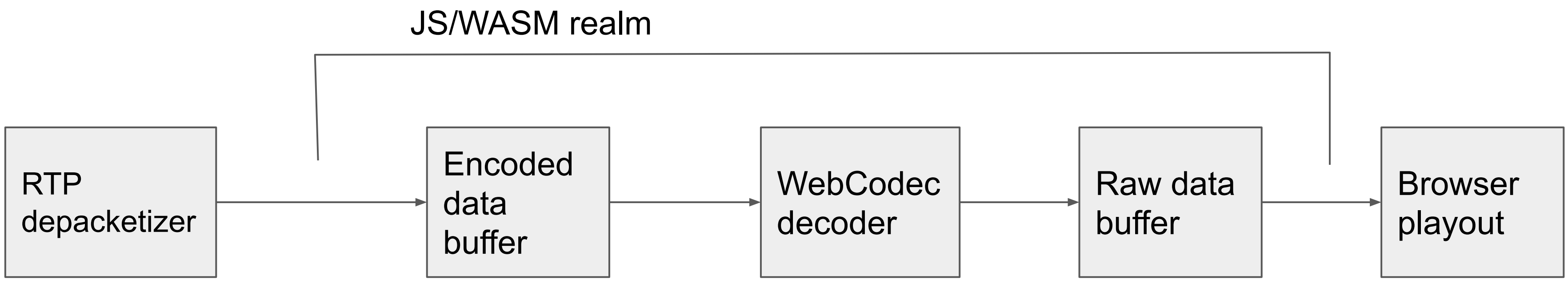
- An Insertable Streams Use Case (Harald)
- Relationships between APIs (Bernard)
- getViewportMedia (Jan-Ivar)
- Issue 777: Defaults for getUserMedia (Youenn)
- Issue 64: Transfer RTCDataChannel (Youenn)
- Issue 16: Transfer MediaStreamTrack (Youenn)
- MediaCapture Transform feedback (Youenn)

Insertable Streams
A Use Case Study

The problem statement

- “NetEQ in WASM”
- Adjust the timing of incoming audio to ensure smooth playback in the face of jitter and packet loss
- Requires adaptation to both jitter and to decode speed
  - Desire to use WebCodec’s Opus wrapper rather than the builtin WebRTC one
- Desire to iterate on implementation independent of browser engine

The desired pipeline



Today’s Insertable Streams API

When an incoming track is created, the browser will:

- Set up depacketizing
- Create a decoder
- Create a track
- Fire “ontrack”
- Allow streams to be fetched using CreateBreakoutStreams(), OR
- fires an event at a relevant Worker containing input & output streams

In this scenario, we only want the first of these 4 steps.

On the raw side, we already have MediaStreamTrackGenerator, which is OK.

An API that would be nice

```
pc = new RTCPeerConnection(... stuff that tells what to do with incoming media ....);

output = new MediaStreamTrackGenerator();

createaworkerforthis(pc, output);

// the following should happen in a worker, not on the main thread
decoder = new AudioDecoder(“audio/opus”);
… set up feed from the encoded-buffer into the decoder ….
… set up feed from decoder into decoded-buffer ….
… add processing where appropriate to cover lost samples,
   adjust clock rate and so on ...
someobject.onsomeeventliketrackbutgivingastream = (inputstream, outputstream => {
   … setup code to feed the encoded samples into a buffer …
   … setup code to feed the decoded-buffer into the outputstream ...
}
```

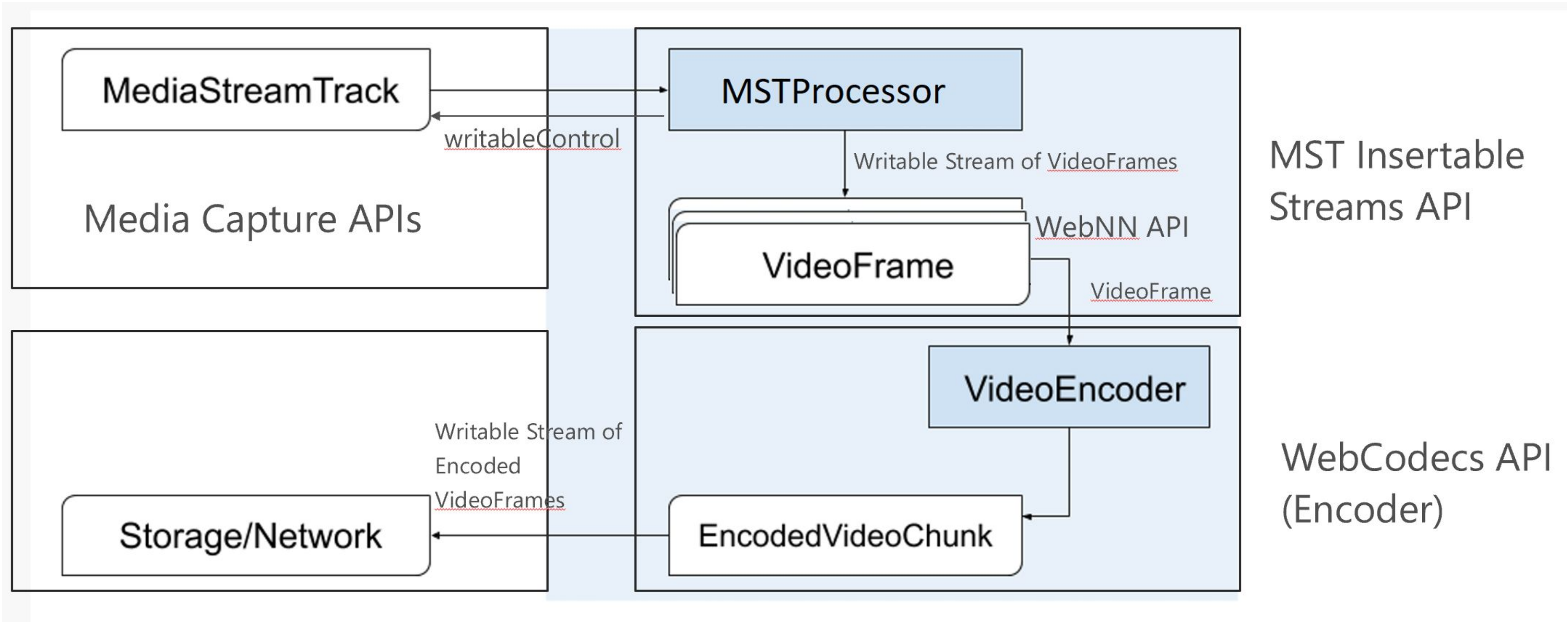
Missing Pieces

- Tell the PeerConnection what to do on incoming media streams (do NOT create a track, give a stream of encoded frames to a worker instead)
- Specific API for getting the streams to the worker
  - should we make this specific, or just extract the streams and pass them?
- Guarantees that once audio frames are emitted from JS, they are presented in a timely manner (“no more jitter”)

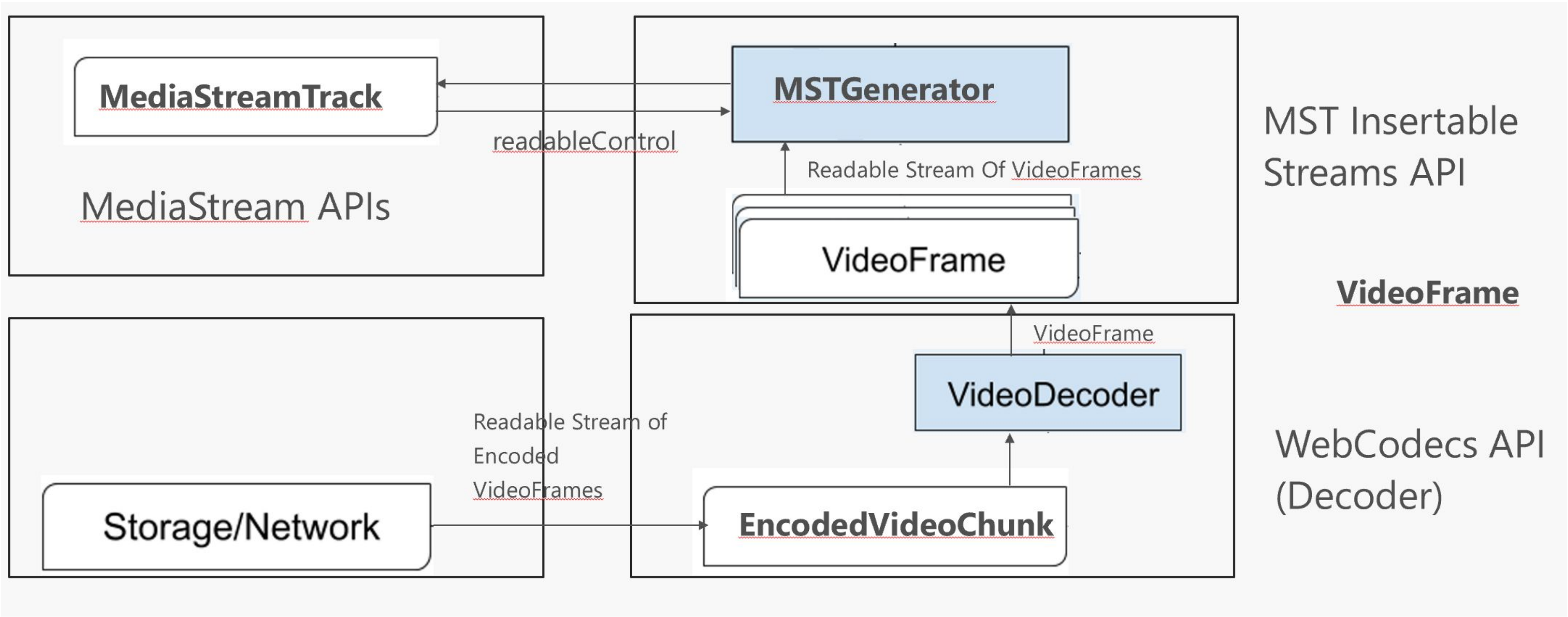
Not needed in this use case (as far as we know):

- Feedback from JS modules to the transport

API Relationships (Sending)



API Relationships (Receiving)



Some Questions

- 4+ WGs involved in design of these APIs. How do we coordinate?
- Does the pipeline avoid unnecessary copies? Examples:
  - Can an encodedVideoChunk produced by WebCodecs be sent by WebTransport (or WebRTC Insertable Streams) without an additional copy?
  - Can WebTransport (or WebRTC Insertable Streams) write encodedVideoChunks to a GPU buffer, so WebCodecs can decode without a copy?
  - One copy seems unavoidable within each of the above (for process separation).
- Do we understand:
  - Which structures can reside in GPU buffers?
  - Whether we need additional WebIDL support (e.g. for “read-only buffers”)?
  - How to integrate with WASM efficiently?
  - When copies do (and do not occur)?
    - Copies within main memory.
    - Copies between GPU buffers and main memory.
    - Example: Cloning a videoTrack generates a reference, but does not copy.

[Issue 155](): getViewportMedia (Jan-Ivar)

Update: Good news!

- Agreement on “Site isolation” + some opt-in document header. Clear up:
  - Full 👻-[`window.crossOriginIsolated`]() (COOP+COEP) or just COEP?
  - Shape of HTML header?
  - Failure mode: Block loading of non-opt-in iframes, or kill capture?
- Agreement:
  - Resources are on their own. Vulnerable to 👻 anyway.
- Another rename! **getViewportMedia()**
  - [MDN]() on “viewport”: *“refers to the part of the document you're viewing which is currently visible in its window (or the screen, if the document is being viewed in full screen mode). Content outside the viewport is not visible onscreen until scrolled into view.”*
  - Precedent for term viewport exposed to web: [`window.visualViewport`]()

Issue 777: Defaults for getUserMedia (Youenn)

Summary of the issue

- User Agent has to select device settings values when starting capture
- There are a lot of settings values with the same fitness distance
- Specification does not provide guidelines on how to select amongst them

Question

- Should the specification say more? Guidelines? Normative text?

Two different opinions

1. Yes: this will help interop
2. No: this will make future evolutions more difficult
    - Assumption: Web pages should provide the necessary constraints

Issue 777: Defaults for getUserMedia (Youenn)

Is it a real problem?

- Spec says that constraints should be provided by web page for all important properties

GetUserMedia examples

- No constraint on echoCancellation: webex, whereby
- No constraint on frameRate: jitsi, whereby
- Maybe they use applyConstraints to refine settings if needed
  - Suboptimal as require recalibration of the camera and/or microphone

Web sites do not follow specification guidance

- This is a real problem

Issue 777: Defaults for getUserMedia (Youenn)

Impact

- User Agents heuristics are very important!
- User Agents with small number of users NEED to apply popular heuristics
- And the specification does not provide any guidance

Conclusion

- Current state is not good enough
- Specification has to provide some guidance

Issue 777: Defaults for getUserMedia (Youenn)

What do Chrome, Firefox and Safari?

- echoCancellation=true by default
- 640x480x30 by default

Proposal

1. Properties with OS default values: use OS default values
   - sampleRate, sampleSize
2. Properties without OS defaults: provide implementation advice
   - Recognize that default values should be selected carefully
   - Document what known implementations currently do
3. Discuss tightening the settings selection algorithm
   - Algorithm to unbreak fitness distance ties given a default value set

Issue 64: Transfer RTCDataChannel (1/2)

Web sites use data channel to transmit data but process the data in workers

- Conferencing: Zoom
- Game streaming/Remote desktop: Parsec
- Audio/video low latency transmission: receiving and sending

Potential solution: **make data channels transferable**

Safari prototype

- In-process: fairly straightforward and no additional cost
- Out-of-process: requires more code and has a limited processing cost

Issue 64: Transfer RTCDataChannel (2/2)

Safari prototype rules to ensure delivering of all events

- Transfer the data channel as soon as created
    - Just after createDataChannel / In ondatachannel event handler

Alternative

- Transfer the data channel if no buffered write
    - Optionally with no event handler registered
- No guarantee to deliver all events (messages or state updates)

Proposal

- Choose between the two alternatives
- Write a PR, probably in webrtc-extensions

Issue 16: Transfer MediaStreamTrack (Youenn)

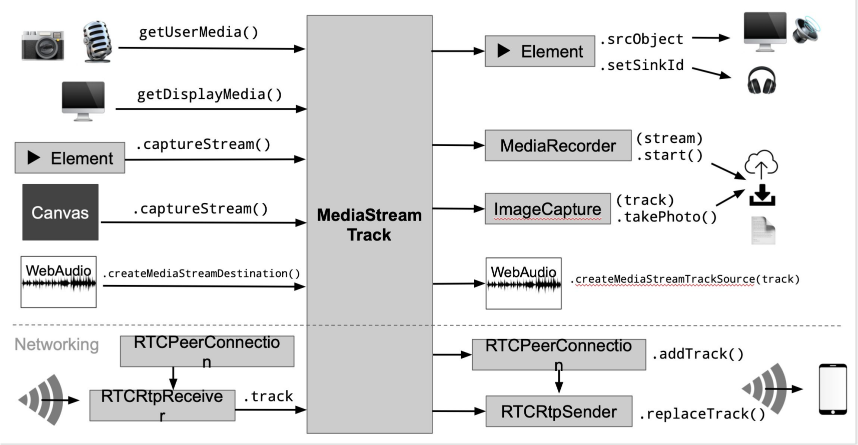
Why do it?

- Create a track in the iframe that makes most sense
  - Transfer the track in a different iframe that will do the networking
- Do processing in a worker
  - MediaStreamTrack -> WebCodec -> RTCDataChannel
  - RTCDataChannel -> WebCodec -> MediaStreamTrack

Issue 16: Transfer MediaStreamTrack (Youenn)

Can we do it? Cross process?

- MediaStreamTrack is already flowing out of process
  - Content is very often produced & processed out of process
- Browsers already do this efficiently



Issue 16: Transfer MediaStreamTrack (Youenn)

How can it be done?

What is needed is very similar to RTCDataChannel

- Transfer algorithm
- 'Neutered' behavior
- Lifetime of transferred MediaStreamTrack tied to creation context
    - Like data channel

Issue 16: Transfer MediaStreamTrack (Youenn)

Alternative

- Use mediacapture transforms as a shim

Potential downsides

- More difficult to optimize this code path than transferring a MediaStreamTrack
- Not the same support as transferable streams out of the box
    - Especially for capture tracks: getSettings, applyConstraints…

Should we work on transferring MediaStreamTrack support?

Media Capture Transform Thoughts (Youenn)

Media Capture Transform vs. AudioWorklet

Processing should be done in Worker by default

Consuming a MediaStreamTrack is more efficient than piping a ReadableStream

Controlling channel is unclear

Controlling channel is potentially incomplete (muted/ended events)

No visible benefit between Media Capture Transform and VideoTrackReader

For extra credit

Name the bird!

Thank you

Special thanks to:

WG Participants, Editors & Chairs

The bird