Browser Hosted Video Editing

James Westland Cain, Ph.D
Chief Software Architect
james.cain@grassvalley.com

Michael Weaver
Engineering Manager
michael.weaver@grassvalley.com

James Pearce
Principal Software Engineer
james.pearce@grassvalley.com

Vincent Trussart
Director, Software Development
vincent.trussart@grassvalley.com

version 0.3

Introduction

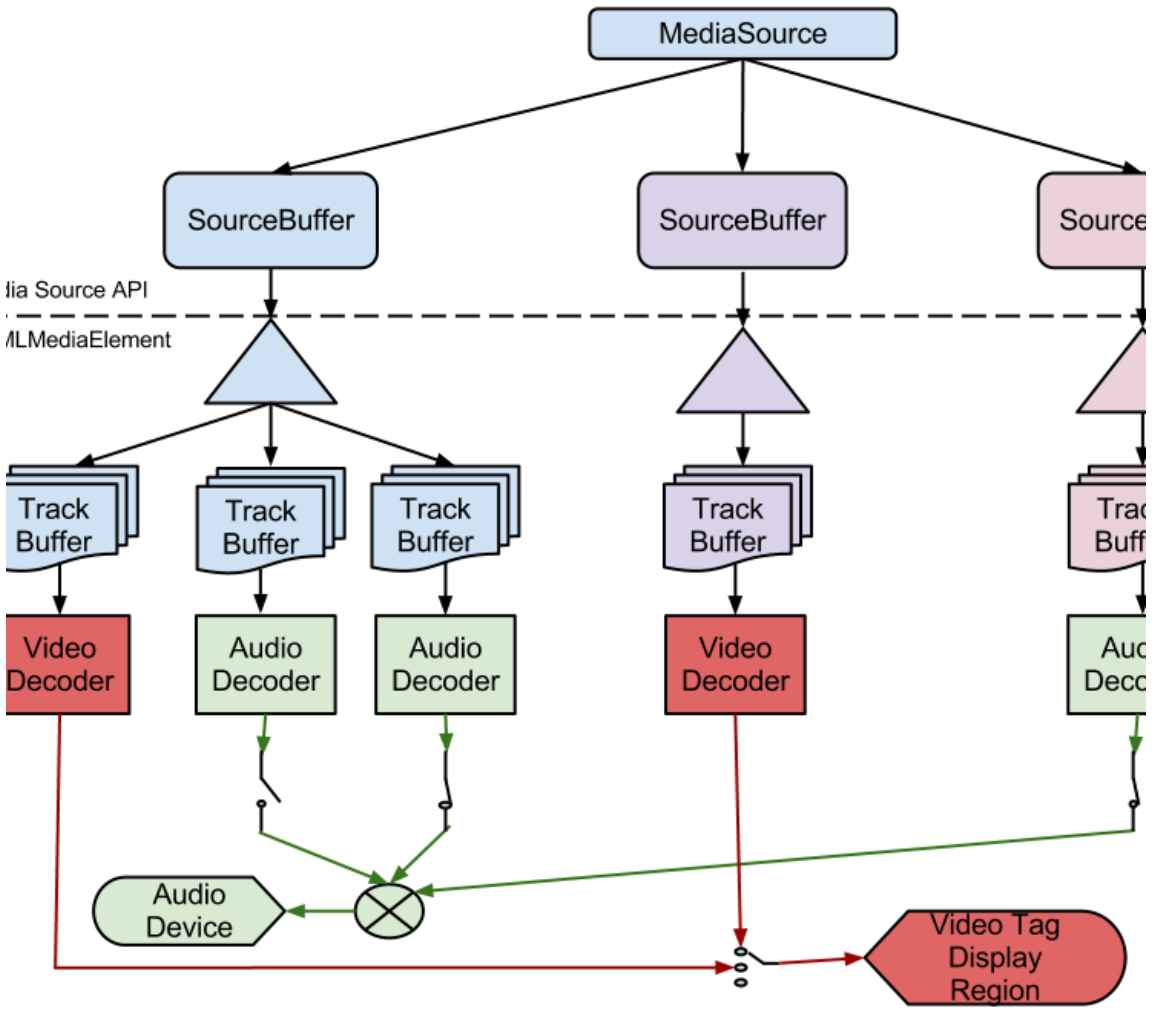
- Between us (a team formed from the merger of Grass Valley and SAM/Quantel/Snell) we have been building video editors for many decades.
- Over the last decade we have been trying to bring this knowhow to browser hosted video editing - mainly using proxies - but including low latency near live streams (think sports / news etc).
- This an experience report that hopefully will show:
  - what can be done today (which is a lot).
  - what we'd like to be able to do.
- We’d like to start a discussion on what we’d like the browser platform of tomorrow to look like.

History

- As separate companies, we both built video editors hosted in Browsers.
- Grass Valley used a server backend to render edits on the fly, and streamed bespoke flat video to the browser using WebRTC.
- Quantel started using Silverlight and Smooth Streaming.
- Quantel then investigated using Media Source Extensions when Silverlight got deprecated by Microsoft.

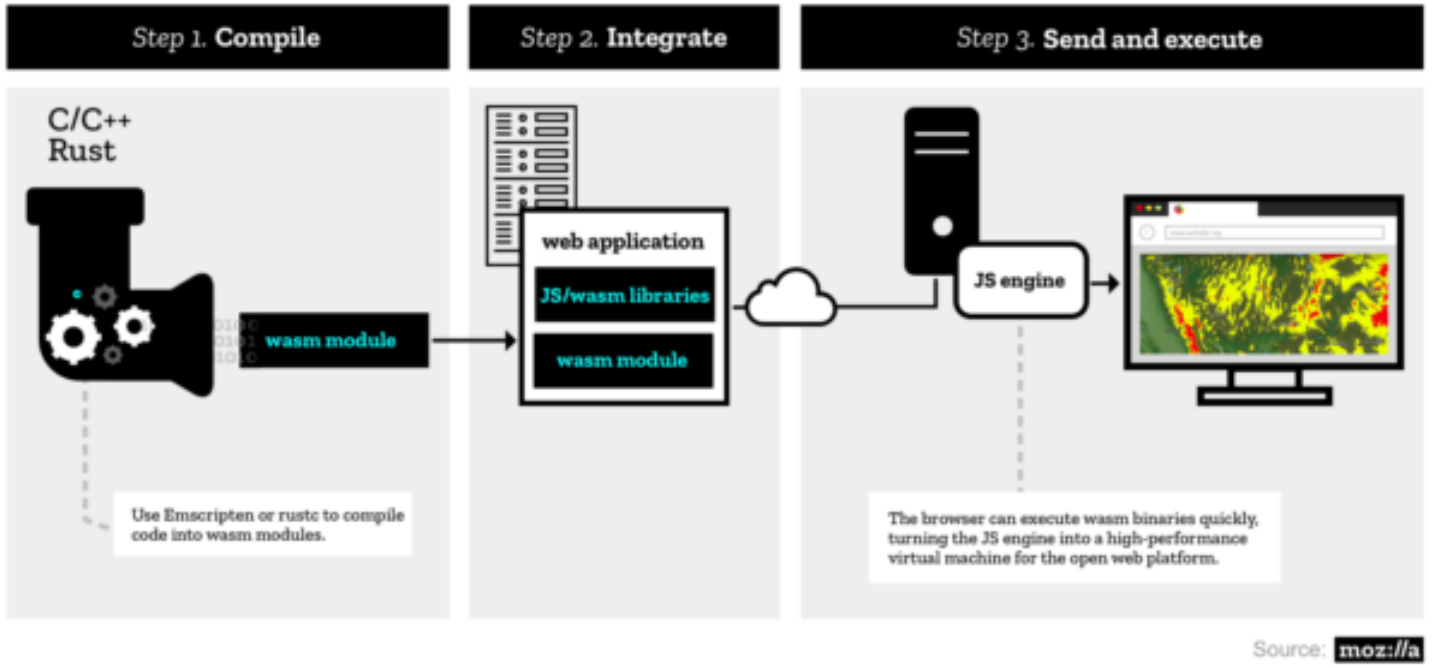
Using MSE to play edited video

- Many problems
  - Splicing
    - Multiple video track buffers not supported
  - Frame accuracy
    - Required overlay of jpegs stills when stopped to know which frame was displayed
  - Aligning Audio to Video
    - Using PTS to align tracks meant edited video / audio track alignment was non-trivial.



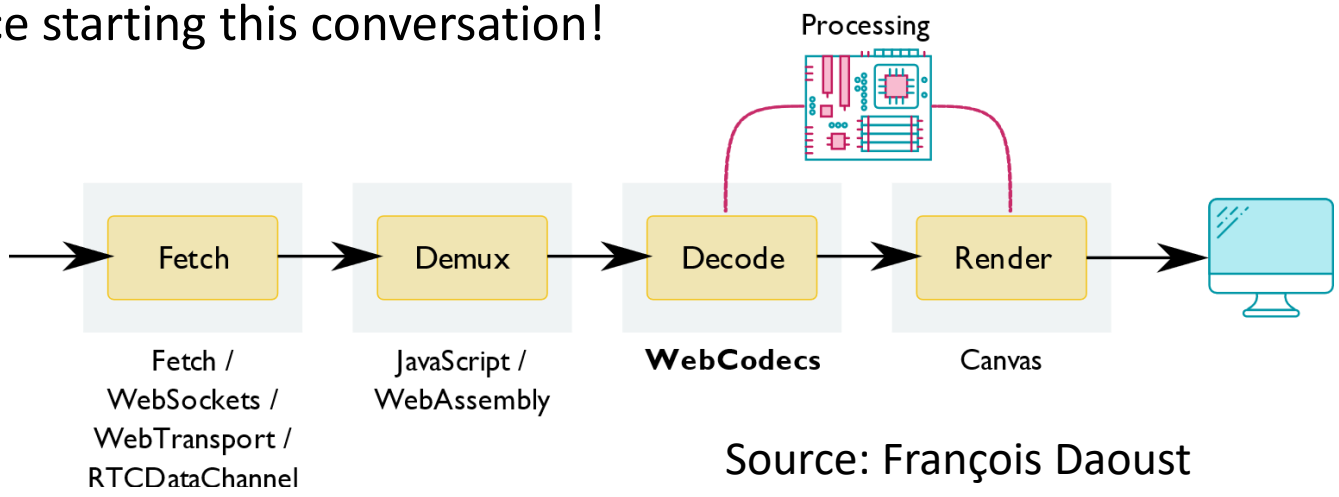
Emscripten and WebAssembly

- For the next generation of product we started using WebAssembly codecs to decompress fragmented moofs (h.264 & AAC).
- This has many benefits over MSE:
  - Frame accuracy – no jpeg overlays.
  - Running precue to get payloads decompressed into cache before they are needed. Enables splicing.
  - Video / audio alignment is much easier to achieve
- This relies on users running over https, being authenticated and trusted.
  - No use of Encrypted Media Extensions (EME) in this use case.
- It runs ***HOT.***
  - We can only do this using constrained proxies.



Using WebCodecs

- We found replacing our use of Emscripten and WebAssembly for decompressing media was easy.
- Huge benefit – it runs much ***cooler***!
- Enables much better picture quality.
- Obviously its not yet standardised – hence starting this conversation!



Source: François Daoust

Comments on WebCodecs

- We need determinism.
  - We need to be able to tell if a codec is available and what formats it can decode using an API rather than try and cope if it fails.
- We'd like easier buffer control
  - We have aggressive buffer management requirements, so would like to be able to keep control of uncompressed frames – even after they have been displayed – as they may be required again.
  - Hardware decoders are likely to have tight buffering constraints. Additional software buffering is required for our use case.
  - Even using a software decoder in place of hardware if required buffering too small.
- Access to SEI messages
  - Timecode, AFDs, Closed Captions etc.

Other APIs

- WebGL and WebGPU enable shader language rendering of effects.
- Can WebCodecs take advantage of Hardware Codecs on GPUs?
  - For example, we use WebRTC to enable remote contribution – so would like encoding to be hardware accelerated if available.
- How would a DRM protected stream work with these techniques?
  - It appears that the decoded buffers could be manipulated by shaders and displayed securely?

Other use cases

- Electronic news gathering (ENG) uses case, where journalists are in the field with video footage they have just shot, and want to get some of the media into the cloud quickly. Requires editing before you upload, hence local media editing.
- Quality camera codecs rather than proxies.
  - AVCi, 10bit YUV Main/High Profile etc.
- Quality render use case.
  - Rendered results stream directly using WebRTC etc.
- Moving to higher quality enables us to consider High Colour Gamut and hence High Dynamic Range video.
  - Colour spaces: REC.709 vs BT.2020.

Looking ahead

- Many broadcasters use cloud hosted distribution for news and sports – as well as more common VoD use cases.
- Taking a strategic view, hosting the production of video in the cloud implies that these backends can start to merge.
- We already have customers that produce bespoke streams for many market segments –rendering each in the datacentre.
- Our customers would like to be able to tailor streams on a per customer basis.
- So tailored TV – rendering recipes at the edge – is a very real opportunity as the browser becomes the next generation operating system.

Browser Hosted Video Editing

James Westland Cain, Ph.D
Chief Software Architect
james.cain@grassvalley.com

Michael Weaver
Engineering Manager
michael.weaver@grassvalley.com

James Pearce
Principal Software Engineer
james.pearce@grassvalley.com

Vincent Trussart
Director, Software Development
vincent.trussart@grassvalley.com

version 0.3