

WebCodecs in WebKit

With GStreamer!

Philippe Normand
GStreamer conference 2023

WebCodecs, what is this?

- <https://www.w3.org/TR/webcodecs/>
- “This specification defines interfaces to codecs for encoding and decoding of audio, video, and images.”
- Gives JS direct access to encoders and decoders provided by the WebEngine
- Ability to modify `VideoFrames` and `AudioData` before encoding or after decoding
- No need for 3rd-party WASM-based framework
- Muxing and demuxing handled at JS layer (MP4Box.js, etc)

WebCodecs, use-cases

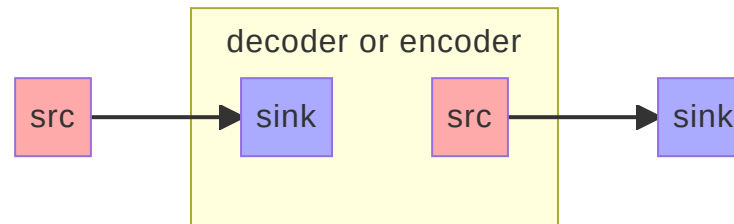
- Video editors
- WebRTC “Funny hats”
- Integration with WebAudio, WebGL, WebTransport
- {en,de}coding can be moved to another thread with WebWorkers

WebCodecs, in WebKit!

- First approach with `appsrc ! {en,de}codebin ! appsink` worked, but was not spec compliant
- We need “direct” access to the codec element, no additional thread (queue)
- Push data to sink pad, pull from src pad
- Something a bit similar to GstHarness

WebKit's GStreamerElementHarness

- A C++ class, no pipeline per-se
- Simple API, `pushBuffer()`, `pushEvent()`, `flush()`,...
- Internal state: output buffers and events stored in Double Ended Queues (deque)
- Simple case: one input pad, one output pad:

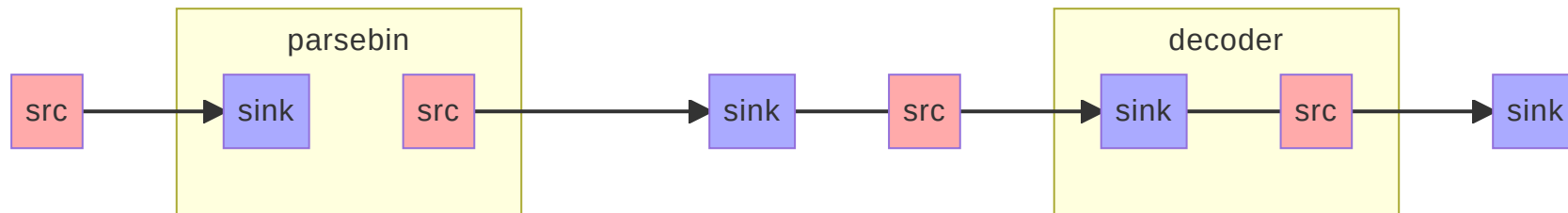


- Push data on src pad
- Processing ends in ... sink pad
- C++ closure-based processing triggered with `::processOutputBuffers`



Chaining

- one to many pads
- callback-based “plugging”
- “sometimes” pad handling, with possibility of sending buffers to an `identity-` based harness



Implementation status in WebKitGTK & WPEWebKit

- (H.264, H.265, AV1, VP8, VP9) video encoding/decoding: shipping in 2.42
- (AAC, mp3, opus, vorbis, fLaC, alaw, ulaw, LPCM) audio encoding/decoding: landed in main, will ship in 2.44
- WebPlatform tests coverage is OK
- Room for improvement regarding interaction with WebAudio and WebGL

Demo

- All usable from the GNOME Web Canary flatpak
- Ask me!

