WPEWebKit
The WebKit port for Embedded

Philippe Normand

philn@igalia.com

Linaro Connect

San Diego, California

23rd-27th September 2019

Who am I?

- Fiddling with WebKit and GStreamer since 2009
- WebKit committer and reviewer
- GStreamer committer
- Partner at Igalia
  - Worker-owned coop, currently around 80 happy Igalians around the world
  - Provides consulting services for various Free Software projects

Talk Outline

- What is WPE
- Basic infrastructure for Media playback
- Adaptive streaming in HTML5: Media Source Extensions
- Media-capabilities
- Enabling WPE in your apps
- Hands-on: WPE integration with Yocto on I.MX6 & I.MX8M

WPE, Web Platform for Embedded

WPE, the basics

- Web-engine based on WebKit, tailored for a wide range embedded devices
- 6 months release cycle synchronized with WebKitGTK, including security updates
- No dependency on any UI toolkit library
- Pluggable view backends in charge of final rendering
  - Wayland
  - Android experiment
  - Device-specific graphics drivers

WPE backend for FDO

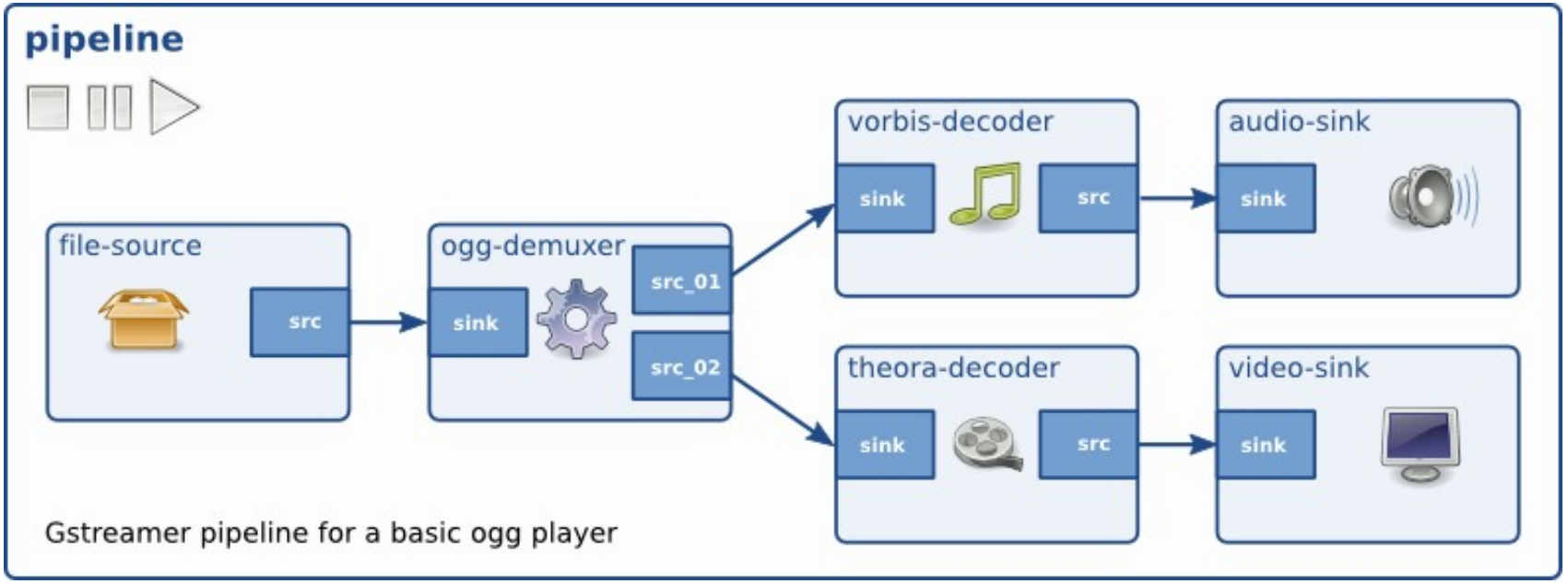
- Relies on wayland-egl
- Cross-process buffer sharing
- API for:
  - EGLImages
  - Or wl_resource objects
  - Or Linux dma-buf information (already used internally)
- Combined with Mesa
- Works on desktop (packaged in Fedora 30, soon Debian) & embedded (Yocto & Buildroot)

igalia

Platform support for media playback in WPE

GStreamer

- A cross-platform framework for creating Multimedia applications
- Graph-based processing, example of a basic ogg/vorbis player:



igalia

<audio> & <video> in WPE

- Playbin-based MediaPlayerPrivate implementation
- GL Video rendering with a custom appsink
- Custom GstAllocator using WebKit’s FastMalloc
- White list of supported containers and codecs
  - Codec installer support effectively useless

Adaptive streaming: MSE

By Alicia Boya & Enrique Ocaña

The MSE backend, TL;DR

- Chunks queued from JavaScript world to a SourceBuffer
- One GStreamer WebKit Append pipeline per SourceBuffer
  - Demuxing and parsing of samples
  - Samples stored at WebCore's MSE layer
- Playback pipeline using a dedicated MediaPlayerPrivate implementation
  - Playbin-based
  - Custom source element (one source pad per SourceBuffer)

igalia

MSE-related improvements in GStreamer

- Quite a few improvements in qtdemux for:
  - Samples demuxing in push-mode
  - Edit list support for push-mode
  - Segment event handling
  - Duration-related bug fixes
  - => Around 15 patches so far!
- Matroskademux improvements
  - Emit no-more-pads earlier (after parsing Tracks) (used to be sent while processing the first Cluster)
  - Multi-Tracks parsing
  - Fixes for WebM byte-stream format handling

Current status & plans

- MSE enabled in GNOME-Web!
- MSE backend widely tested on embedded platforms (RPi, i-MX6, …)
- Infrastructure available for combination with EME
- Youtube (“desktop” and */tv) relying on MSE*
  - *VP9 & opus*
  - *H.264 & AAC also supported*
- *Playbin3 / Stream-collections support: DONE in trunk*
- *Multi-track SourceBuffer support: planned*

igalia

Media-Capabilities

The (draft) spec

- https://wicg.github.io/media-capabilities/
- Goal: provide hints to WebApps regarding the most optimal media encoders & decoders
  - Input: description of the media format (contentType, width, height, framerate, …)
  - Output: 3 booleans:
    - supported
    - smooth
    - powerEfficient

igalia

GStreamer “probing”

- New “Hardware” element metadata Classifier (=> 1.16) 
- Elements may implement probing for their NULL→READY state transition 
- Possibly refine Caps templates to reflect what the hardware supports
  - V4L2 decoders (AVC1 profile/level) (>= 1.18) 
  - More decoders 
- Basic WebKit GStreamer MediaCapabilities backend working for decoders probing

Enabling WPE in your apps

Three examples

Cog, the WPE launcher

- Native application
- Can work in fullscreen or window mode (no window manager decorations yet)
- Provides an API wrapping some of its features
- D-Bus client API for remoting
- Two rendering backends:
  - One based on WPEBackend-FDO (mature)
  - One based on libdrm, gbm, libinput (experimental)

Qt5 QML

- WPEQt QML plugin upstream in WebKit since version 2.24
- API mimicks QwebView => easy migration path!
- Depends on WPEBackend-FDO
- Works on Linux (Wayland and EGLFS)
- More details: https://base-art.net/Articles/introducing-wpeqt-a-wpe-api-for-qt5/

igalia

GstWPE: HTML overlay in GStreamer pipelines

- GStreamer source element acting as a web-browser
- Outputs GStreamer buffers containing EGLImages wrapped in GstGLMemory's
- Allows for zero-copy composition with other GL backed video streams
- More details: https://bit.ly/2kmee2p

igalia

WPE/GStreamer on i.MX6 QuadPlus & i.MX8M with Yocto

Yocto layers

- https://github.com/Igalia/meta-webkit/
  - WPEBackends
  - Cog browser!
- https://github.com/OSSystems/meta-gstreamer1.0
  - Updated GStreamer 1.14.x recipes
- Poky reference distro
- (meta-freescale)

Open-source etnaviv driver

- Requires recent kernel (4.19), Mesa (18.2.2), Wayland (1.16), GStreamer (1.14.4)
- Usable WPEBackends, only working in Weston:
  - WPEBackend-RDK/wayland
  - WPEBackend-fdo (recommended)
- Upstream v4l2 plugin from gst-plugins-good for hardware decoding (and encoding) support

i.MX6 video decoding

- CODA960 kernel driver
- Mature integration with gst-plugins-good v4l2 plugin
- On QuadPlus: H.264 1080P@30 handled but sometimes drops frames: 720P recommended:

```
$ export WEBKIT_GST_MAX_AVC1_RESOLUTION=720P
```

igalia

i.MX8M video decoding

- Experimental Hantro G1 kernel driver and integration with gst-plugins-good
- Development by Collabora and Pengutronix
- Scheduled for the kernel 5.4
- Status:
  - 1080P@30 H.264 smooth playback
  - Should allow for up to 4K@30 with VP8 and H.264 and 4K@60 with VP9 and HEVC
  - HEVC and VP9 support: Work in progress

YUV video rendering

- Current video sink in WebKit only handles RGBA
- GStreamer video decoders often output YUV:
  - Semi-planar NV12 or planar I420 (gst-vaapi, vpxdec)
  - Interleaved YUYV (gst-v4l2)
- One extra colorspace conversion (YUV → RGBA) required due to WebKit
- On-going work in WebKit to directly handle interleaved YUV formats, lowering the cost of the colorspace conversions in the 3D GPU.


igalia