

A series of vertical lines in red, orange, yellow, and blue on the left side of the slide.

# GStreamer ♥

# WPEWebKit

# Inject the Web into the pipeline

Philippe Normand

[philn@igalia.com](mailto:philn@igalia.com)

FOSDEM conference

Bruxelles

2<sup>nd</sup> February 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



# WPE, the basics

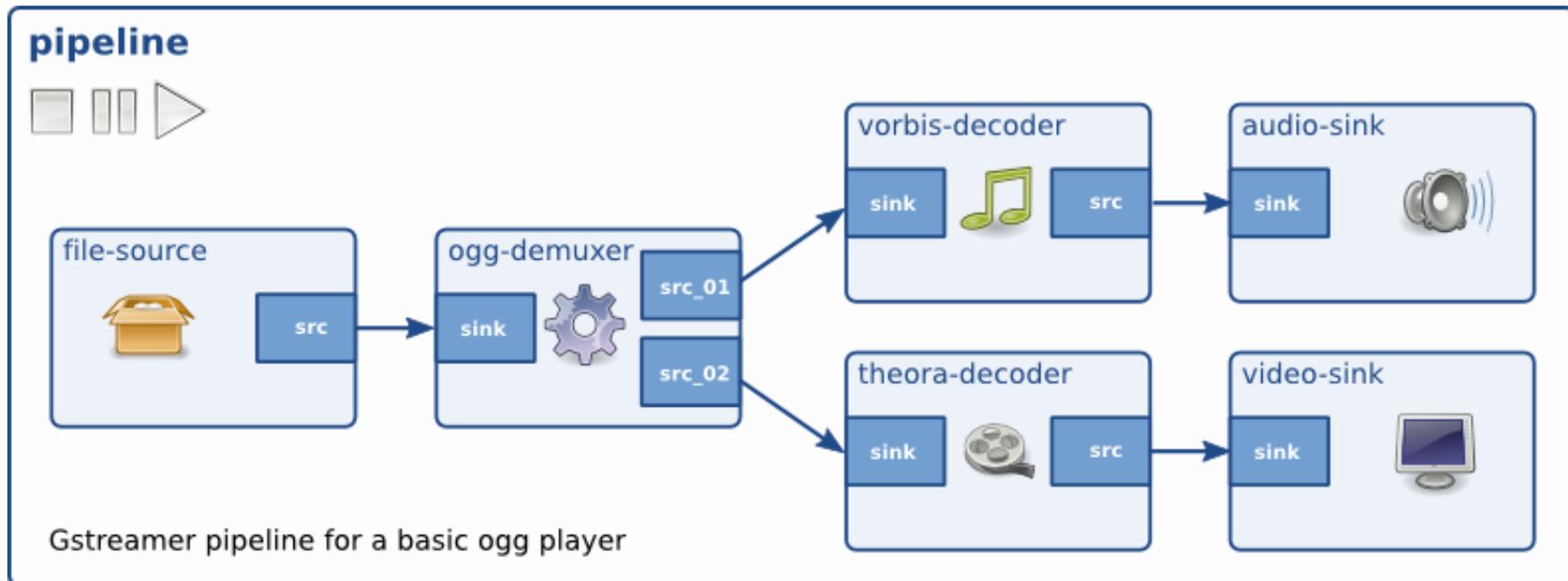
- Web-engine based on WebKit, tailored for a wide range embedded devices
- 6 months release cycle synchronized with WebKitGTK, including security updates
- No dependency on any UI toolkit library
- Pluggable view backends in charge of final rendering
  - Wayland
  - Android experiment
  - Device-specific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

# GStreamer

- Were you at the previous 3 talks in this room?
- A cross-platform framework for creating Multimedia applications
- Graph-based processing, example:



# Some HTML overlay use-cases

- Notification display for live TV broadcasting
- Live video stream “banners”
- Content composition for conference live streaming, for instance:
  - background picture
  - conference logo
  - animations

# GstWPE

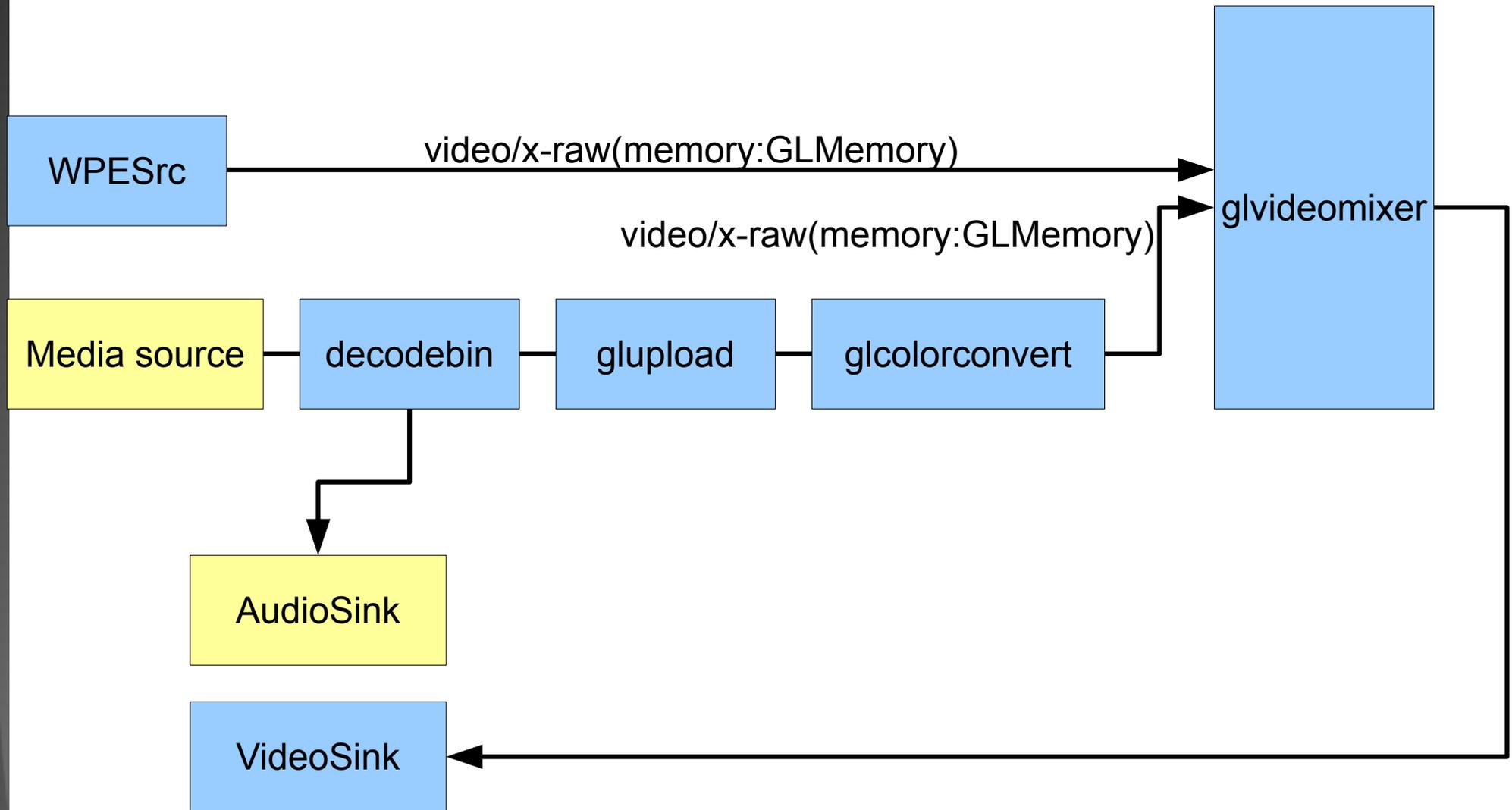
- EGLDisplay and EGLContext shared from downstream element(s)
- WPEBackend-FDO exportable object created for shared EGLDisplay
- `wpesrc` creates a WPE WebView
- EGLImages provided by WPEBackend-FDO wrapped into GstEGLImages for shared EGLContext
- GstEGLImages inserted into GstGLMemory and pushed to source pad towards downstream element

# The most basic GstWPE example

- Input events relayed from the video sink to the source element
- Known GstWPE limitation: Audio stream not exposed



# Video mixing with GstWPE



# Demo

- <https://www.youtube.com/watch?v=J2eb-KRk9gw>



# GstWPE, pros/cons

- Pros:
  - WPE's API designed for application embedding
  - Renders web pages to GPU
  - Rather small memory footprint (known to work on 256MB devices)
- Cons:
  - Audio buffers rendered in WPE not relayed to GstWPE **yet**
  - Limited input events support
  - Currently works only on wayland-egl capable Linux platforms

# Future work

- [Almost finished!] WPE WebView transparent background support
- Audio support
  - Unified audio renderer for WebKit
  - New Audio capture API for WPEBackend-fdo
  - Audio source pad for the wpesrc element
- Improved input events support
- WPE ↔ GstWPE communication channel?
- Support more platforms?

# Wrapping up

- <https://is.gd/sODd5b> : GstWPE upstream in gst-plugins-bad
- <https://wpewebkit.org/>
- One last demo:  
<https://www.youtube.com/watch?v=1JUa3Ldb3vE>





igalia