WPE, The WebKit port for Embedded platforms

FOSDEM 2020, Brussels

Philippe Normand

Talk outline

- Intro
- WPE architecture overview
- Using WPE in Qt applications
- Using WPE in GStreamer Multimedia applications
- Outro / Demo

igalia

Who am I

- Fiddling with WebKit and GStreamer since 2009
- WebKit committer and reviewer
- GStreamer committer
- Partner at Igalia:
  - Worker-owned coop, currently around 90 happy Igalians around the world
  - Provides consulting services for various Free Software projects

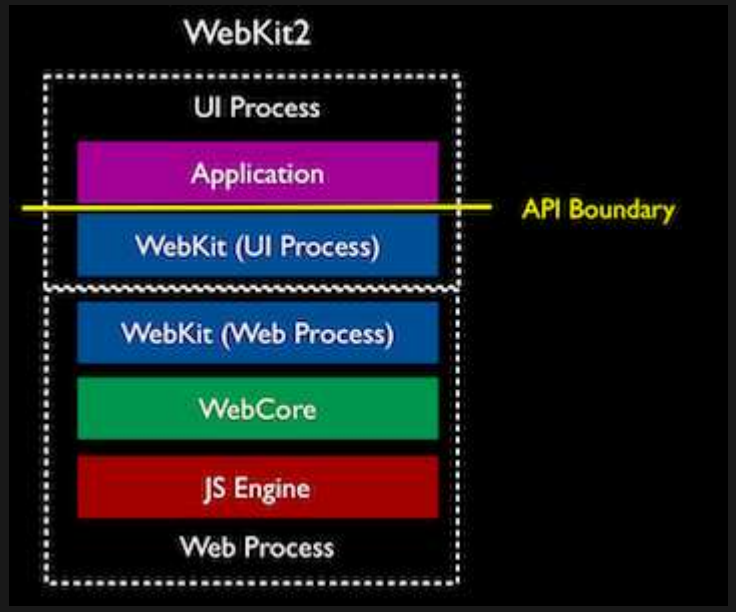
WPE, Web Platform for Embedded

igalia

What is WebKit

- WebEngine aimed for embedding HTML/CSS/JS in native applications
- Forked from KHTML by Apple in 2004
- Powers Safari, but also dozens of applications on various platforms
- APIs provided by WebKit ports

Multi-process!



igalia

WPE, a decoupled WebKit port

- Upstream in webkit.org
- 6 months release cycle, security updates
- Not tied to any widget toolkit
- Rendering and input events handling via loadable backends

Rendering backends

- Wayland-EGL extension usually required
- Various implementations, most notably:
  - FDO backend
  - RDK backend (various devices supported under the RDK/Comcast umbrella)

WPEBackend-FDO

- Depends on EGL, usually provided by Mesa or binary drivers
- High level API provided for applications (browsers, but also other apps!)
- Recommended by upstream WPE community

Cog, the minimal WPE-based browser

- Optional (but recommended) dependency on WPEBackend-FDO
- No window Chrome (optional GTK Chrome could be considered)
- Single web view
- Can be operated through D-Bus

Running Cog without Wayland compositor

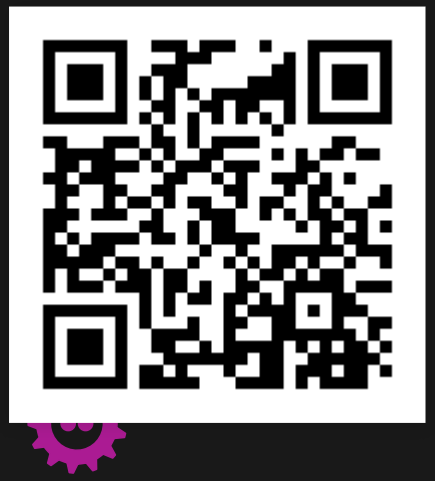
- Through the DRM backend!
- Wayland buffers (or DMABufs) imported as GBM Buffer objects
- Rendering through DRM
- Input events handled with libinput
- App use-cases: Kiosks, Set-Top-Box UIs, ... any fullscreen display!

Cog-based app example (1/2)

- DLNA MediaServer ContentDirectory client embedded in a WebKit WebExtension
- WebExtension dynamically modifies the DOM to display MediaServer contents
- Video playback done with an injected `<video>` element

Cog-based app example (2/2)

Demo: https://www.youtube.com/watch?v=VEQRBVKnN8o



igalia

Using WPE in Qt applications

WPEQt

- Dependency on WPEBackend-FDO
- Provides a QML module
- API compatibility with the unmaintained QWebView QtWebKit-based module

Pros & Cons

- ✔ Well maintained & security issues support
- ✔ Works on Wayland-EGL and EGLFS QPA environments
- ✕ Linux only
- ✕ Dependency on Wayland-EGL

igalia

Enabling it

- `ENABLE_WPE_QT_API` CMake option in upstream WPEWebKit
- At runtime, ensure libqtwpe.so is in `$QML2_IMPORT_PATH`

How-To

```
import org.wpewebkit.qtwpe 1.0

Window {
  id: main_window
  WPEView {
    url: qsTr("https://fosdem.org")
    onTitleChanged: {
      main_window.title = title;
    }
  }
}
```

igalia

Using WPE in GStreamer applications

GStreamer

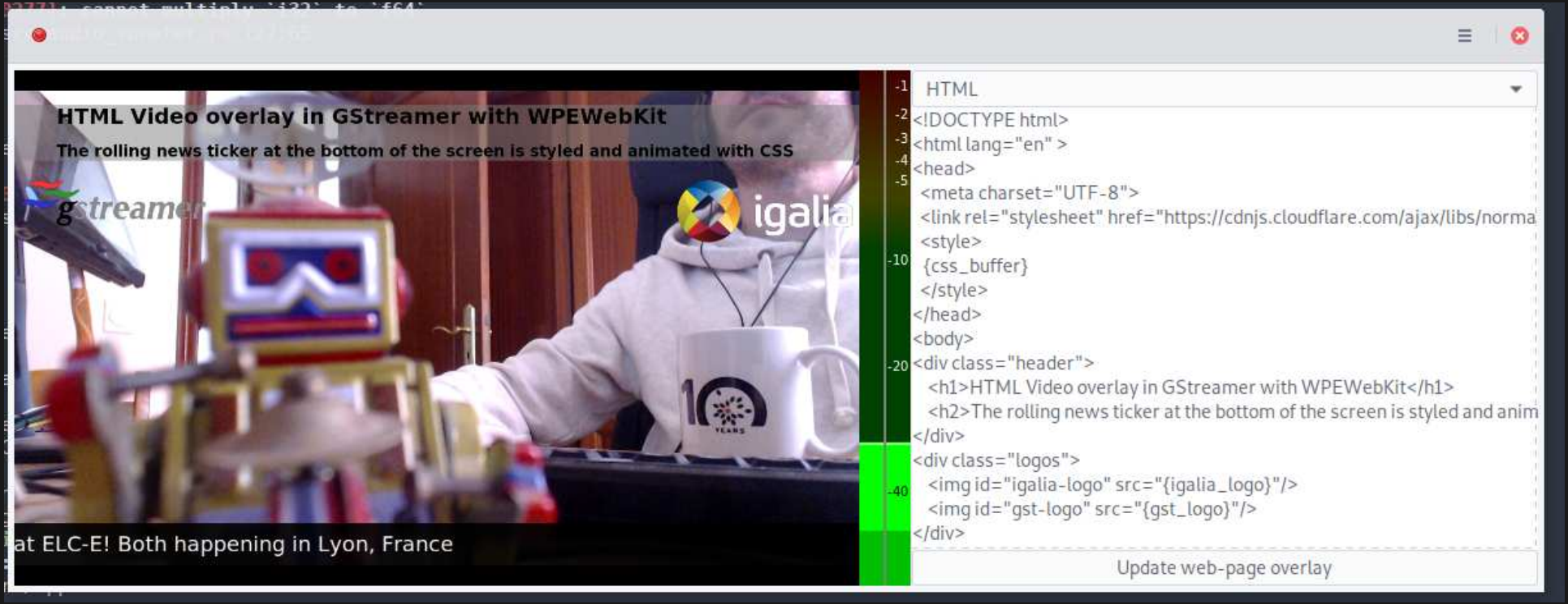
igalia

GstWPE

- GStreamer source element producing a video stream of a WPE WebView
- Zero-copy from WPEBackend-FDO EGLImages to GStreamer GL
- Use-cases: HTML overlays, streaming/cloud browsers

igalia

HTML overlays



Ongoing work

- Audio support: one src pad per audio stream (From `<audio>` and `<video>` HTML elements + WebAudio + WebRTC)
- Improved navigation handling (touch, mouse scroll, keyboard)

Outro

OK! Show me the code

- WPEWebKit (+WPEQt), WPEBackend-FDO, Cog: https://wpewebkit.org
- GstWPE: upstream GStreamer
- Yocto overlay: https://github.com/Igalia/meta-webkit
- Buildroot: recipes already upstream

Questions?