Add the power of the Web to your embedded devices with WPE WebKit

Mario Sánchez Prada

2023 Embedded Recipes

5th edition
28 to 29 september

igalia

About me

- **CS Engineer**, partner of **Igalia**
- **Involvement** in some **Open Source communities**
  - e.g. Chromium, WebKit, GNOME
- Other **work** done in the past:
  - Linux-based OS's (i.e. Endless OS, Litl OS)
  - Maemo (Hildon Application Manager)
  - Samsung SmartTV platform

Currently **coordinating** Igalia's **WebKit team** igalia

Open Source Consultancy

About Igalia

- Specialized **Open Source consultancy**, founded in 2001
- **Fully remote**, headquartered in **A Coruña, Galicia** (Spain)
- **Worker-owned, employee-run, flat structure** (140+ igalians)
- **Top contributors** to the main **Web Rendering Engines**:
  - WebKit, Chromium, Gecko and Servo
- **Active contributor to other areas and OSS projects**
  - V8, SpiderMonkey, JSC, LLVM, Node.js, GStreamer, Mesa, Linux Kernel...
- **Members of several working groups**:
  - W3C, WHATWG, WPT, TC39, OpenJS, Test262, Khronos...

https://www.igalia.com

Web Rendering Engines

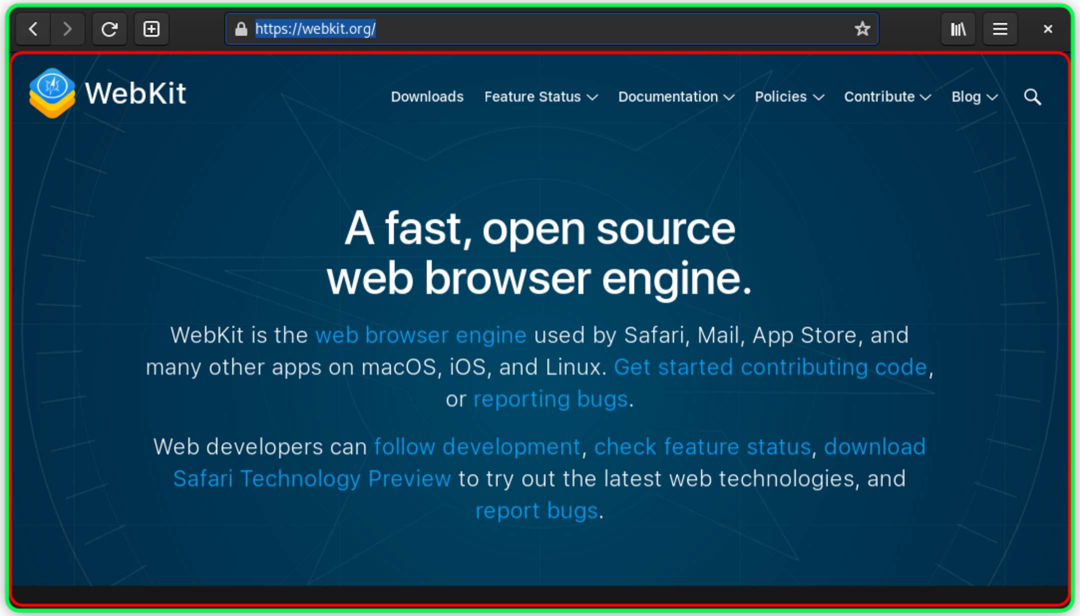
Gecko
servo

What is WebKit?

What is WebKit?

- An Open Source **Web rendering engine** (mostly BSD)
  - Started by Apple as a fork of KHTML and KJS in 2001.
  - Forked again by Google to become *Blink* in 2013.
- **Goals**: performance, portability, stability, compatibility, standards compliance, security and *"hackability"*. ***Embedded-ability***.
- Available for **different platforms** and operating systems:
  - **Desktop & Mobile**: Mac, iOS and Linux ((e.g. Safari, GNOME Web...)
  - **Embedded**: set-top-boxes, video game consoles, smart home appliances, in-vehicle/inflight entertainment, GPS devices, digital signage...

What is WebKit?



A fast, open source web browser engine.

WebKit is the web browser engine used by Safari, Mail, App Store, and many other apps on macOS, iOS, and Linux. Get started contributing code, or reporting bugs.

Web developers can follow development, check feature status, download Safari Technology Preview to try out the latest web technologies, and report bugs.

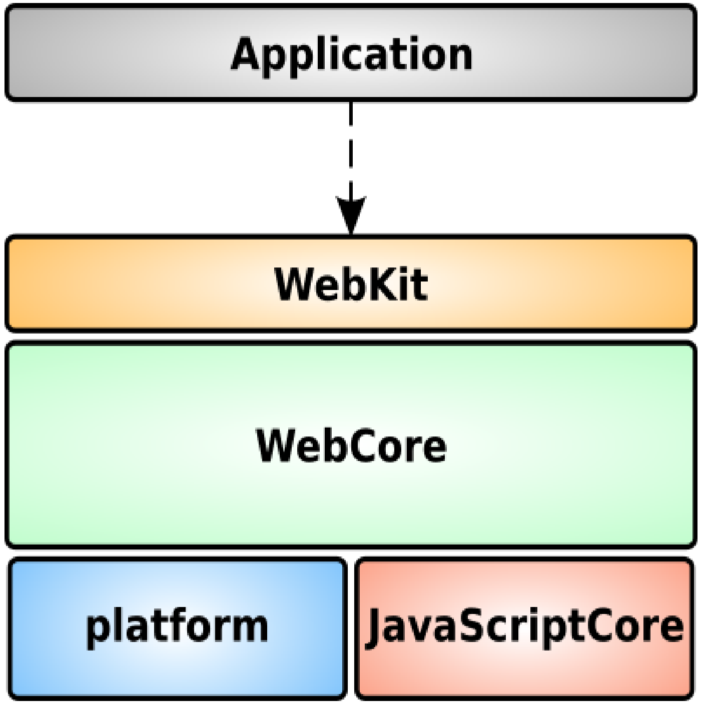
Web Engine

Web Browser

Web Browser **Engine** ≠ Web **Browser**

WebKit Architecture

- **Application**:
  - What end-users interact with.
- **WebKit**:
  - Exposes an API to applications and implements the split-process model.
- **WebCore**:
  - Layout, rendering, network, multimedia, accessibility...
- **JavaScriptCore**:
  - The JavaScript engine.
- **Platform**:
  - Platform-specific hooks.



WebKit Ports

- **WebKit port**: adaptation of WebKit to a specific platform.

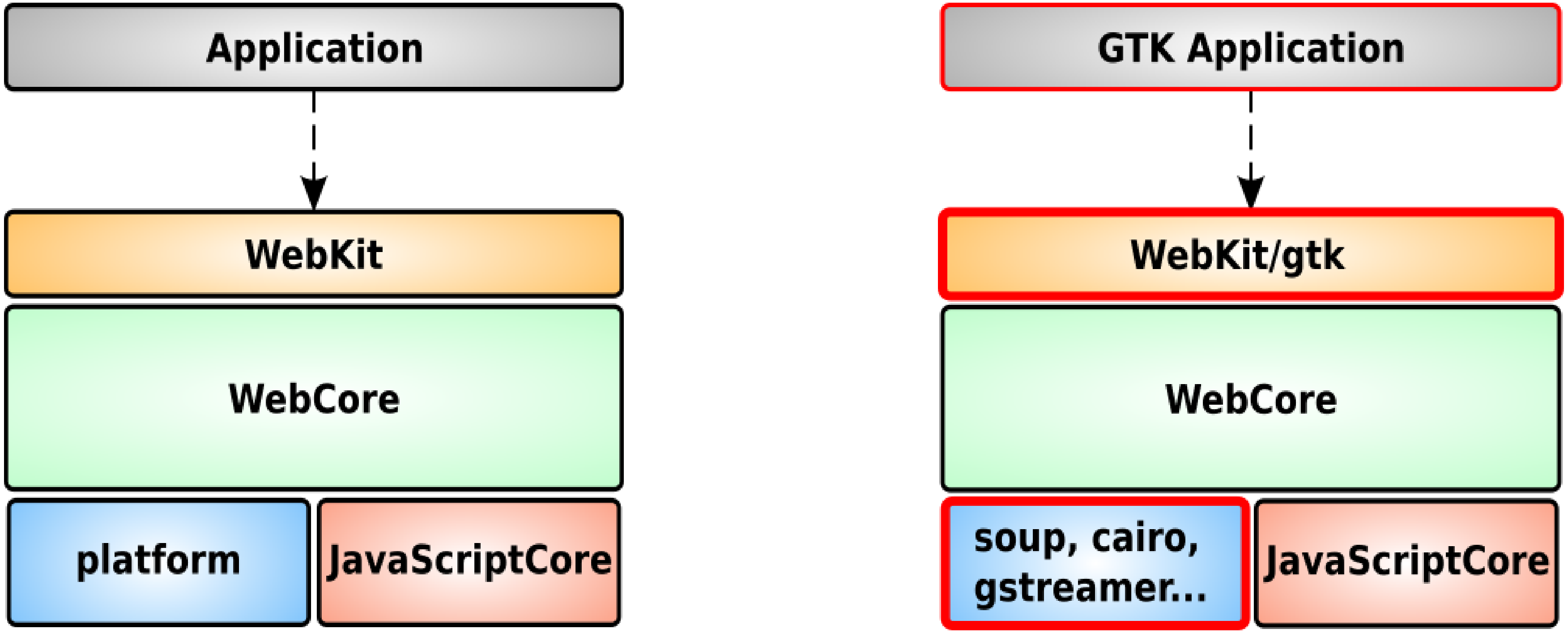
- **Official WebKit Ports** (*upstream* ports)
  - **Mac**: Safari, Apple Mail, iTunes, App Store...
  - **iOS**: every browser on iOS devices (including Chrome).
  - **AppleWin** (deprecated): iTunes, iCloud on Windows
  - **WinCairo** (current): Microsoft Playwright, Playstation SDK
  - **Playstation**: Playstation s4 & Playstation 5
  - **WebKitGTK**: GNOME Web, Evolution, Shotwell...
  - **WPE**: Cog and other custom-made "browsers" for embedded devices.

https://docs.webkit.org/Ports/Introduction.html

WebKitGTK and WPE

- **WebKit ports** targetting **Linux-based** systems
  - **Common parts**: GLib, libsoup (networking), GStreamer (multimedia)...
  - **Key differences**: graphics stack, input handling. Different use cases.
- **WebKitGTK**:
  - Go-to solution to embed Web content in GTK applications.
  - Integration with GNOME components. Supports GTK3 and GTK4.
- **WPE**:
  - Lower level, aimed at embedded devices.
  - Requires graphics and input backend to work.

WebKit Ports: WebKitGTK



What is WPE?

What is WPE?

*WPE is a **WebKit port optimized for Linux embedded devices***

- Modern and comprehensive implementation of the **Web Platform**.
- Focus on **flexibility**, **security** and **performance**.
- **Minimal set of dependencies**, **backends-based architecture**.
- **Low memory and storage footprint**.
- Great support for **HW-accelerated graphics** and **multimedia**.

ℹ️ https://wpewebkit.org/

What is not WPE?

***WPE is NOT** a general purpose Web Browser*

- Provides just the building blocks for Web-based applications.
- Doesn't implement all the APIs found on other WebKit ports.
- Does not rely on any particular UI Toolkit (i.e. backends).
- Can also be useful for less conventional use cases

  e.g. server-side rendering, headless mode...

Upstream & Downstream WPE

- **Upstream WPE**:
  - Generic, device-agnostic, free of customizations.
  - Doesn't assume a particular chipset or platform.
  - Lives upstream at **https://github.com/WebKit/WebKit**.

- **Downstream WPE** (aka *WebPlatformForEmbedded*):
  - Optimized for set-top boxes on specific HW platforms.
  - Customizations for Broadcom SoCs and other types of devices.
  - Better integration with the Reference Design Kit (*RDK*).
  - Lives in **https://github.com/WebPlatformForEmbedded/WPEWebKit**

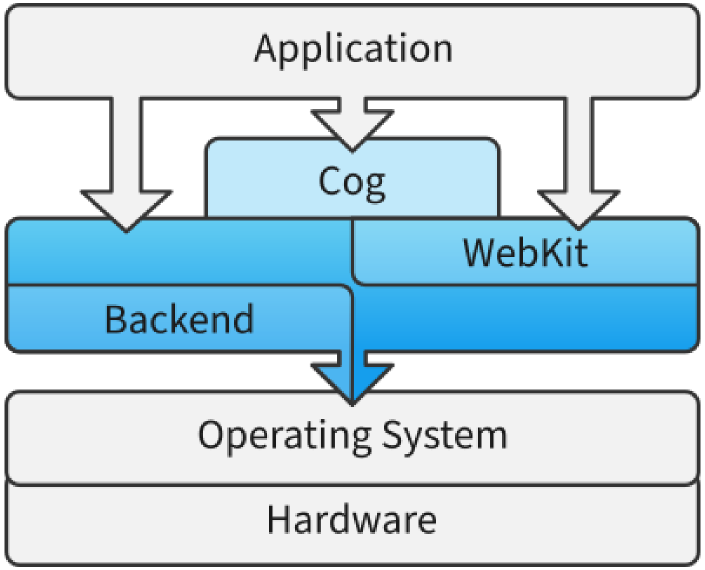
ℹ️ Check **https://rdkcentral.com** for more info on *RDK*

WPE-based products

- Some **examples of use cases** we are aware about:
  - Set-Top-Boxes (both *RDK* and non *RDK* based)
  - Smart Home Appliances
  - HiFi audio/sound systems & music streaming
  - Digital Signage
  - GPS navigation devices
  - Video/Audio conference
  - Headless server-side rendering
  - QA and testing
  - ...

WPE Architecture

- **Application**:
  - The end application, which can use WPE directly or via the provided Cog launcher.
- **WebKit**:
  - The actual WebKit port, including the API layer to link against from applications.
- **Backend**:
  - Platform-specific implementation of the graphics and/or the input layers.



WPE components

- **WPEWebKit**:
  - The actual WebKit port.
  - Relies on the backends for page display and input.
- **libwpe**:
  - Provides rendering-related callbacks implemented by the graphical backend.
  - Allows the input backend to rely events from the application to WebKit.
- **WPEBackend-FDO**
  - The reference FreekDesktop.Org-based backend (i.e. Wayland).
  - Supports several architectures plus regular PC architectures.
  - Can be replaced by a device-specific backend
- **Cog**:
  - Small single “window” launcher for WPE, with no user interface.

WPE: Graphics & Multimedia

WPE: HW-accelerated graphics

- **ANGLE** Support (*Almost Native Graphics Layer Engine*).
  - i.e. better WebGL conformance & WebGL2.
- Supports **DMABuf** for efficient buffer-sharing (+fallback impls).
  - Fallback implementation for DMABuf/GBM-less systems.
- **New SVG engine**
  - Unify HTML/CSS and SVG + enable HW acceleration.
- Experimental **GPUProcess** support (WIP).

WPE: Multimedia

- **GStreamer-based back-ends** for different use cases along with **new GStreamer features** developed upstream (core & plugins).
  - e.g. Media Capture, Media Stream, Media Recorder, WebAudio, MSE, EME...
- **Improved performance** by providing **HW acceleration solutions**.
  - Supports **DMABuf** for GStreamer decoders.
- **Experimental** support (*WIP*): WebCodecs, WebRTC.

Demos

Future plans

Future plans (1/3)

- Release a **new and simplified design** of WPEWebKit:
  - One library with minimum dependencies. Less layers for better IPC.
  - API/ABI backwards compatibility.
- Improved **graphics pipeline**:
  - Efficient zero-copy buffer sharing when possible (e.g. DMABuf).
  - HW-accelerated 2D rendering, multiple-buffer support.
  - GPUProcess, WebGPU.
- Improved **multimedia** stack:
  - Improve WebCodecs integration with WebGL and WebAudio.
  - GStreamer-based WebRTC implementation.

Future plans (2/3)

- Improved **tooling** for developers and automated testing:
  - A new container-based SDK is in the works
  - Improves workflow both on WebKit and its dependencies (e.g. GStreamer).
- Improved **QA processes**:
  - Better maintenance of WPE's continuous integration system.
  - Leverage the improvements from having better tooling.
- Improved **documentation**:
  - Automatically generated API documentation (introspection).
  - Other documentation.

Align the development of **WPE with WebKitGTK**.

Future plans (3/3)

- Support **WPE running on Android**:
  - Provide a WebKit-based alternative to the Android WebView widget.
  - Support for multiple architectures: arm64, armv7, x86-64, x86.
  - Integration with Android's main loop and Android's Process Management.
  - HW-accelerated media playback and WebGL support.
  - Remote Web Inspector.
  - WebDriver support.
  - ...

No new port needed: **uses WPE's public API**

Wrapping up

Wrapping up

- Open Source **port of WebKit** for Linux embedded devices.
- **Modular** and **flexible** architecture, **low resources footprint**.
- Fits and can adapt to a **wide range of use cases**.
- **Hardware acceleration** for graphics and multimedia.
- Two flavours: **upstream** and **downstream** (*RDK*-centric).
- **Deployed on millions** of all sorts of **devices**.
- Big **improvements coming soon** around key areas.
- Experimenting with **WPE Android**.

Questions?

Thanks!