🤖

Jumping over the walled garden wall

WPE WebKit on Android

Adrián Pérez de Castro

aperez@igalia.com

Web Engines Hackfest

About Me

- CS Engineer, partner of Igalia.
- Systems person turned web engine developer.
- WebKit jack of all trades since ~2012.
- Current focus: platform layer, hardware bringup, release engineering.
- I like old computers, too!

About Igalia

- Specialized **Open Source consultancy**, founded in 2001.
- **Fully remote**, HQ in A Coruña (Spain). **Flat structure**.
- **Top contributors** to all 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…

Agenda

1. WPE Android 101
2. Project History
3. What Now?

WPE ~~Android~~ WebKit 101

Adrián Pérez de Castro
2025-06-02

101: WebKit

- Open Source **Web Engine**:
  - Ingests HTML/CSS/JavaScript/etc.
  - Produces rendered content.
- **Port**-able:
  - Port = Adaptation for a specific platform.
  - Runs in more places than one may imagine.
  - The WPE port focuses on embeded Linux systems.

101: WPE ~~Android~~ WebKit

Platformless WebKit port. Designed to be **embeddable**.

Adrián Pérez de Castro
2025-06-02

101: WPE ~~Android~~ WebKit

Platformless WebKit port. Designed to be **embeddable**.

Extensible.

Adaptable.

Minimal* dependencies.

101: WPE ~~WebKit~~ Android

Android WPE **backend** and custom **widget**.

Adrián Pérez de Castro
2025-06-02

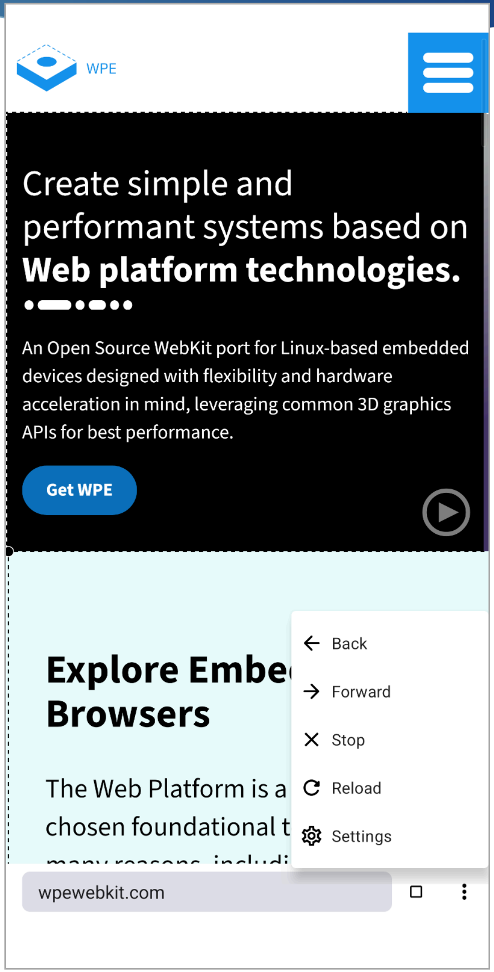
101: WPE ~~WebKit~~ Android

Android WPE **backend** and custom **widget**.

Cerbero for native cross-compilation (Thanks GStreamer!)

WPEBackend-android.

WPE Android proper: glue, widget, tools.

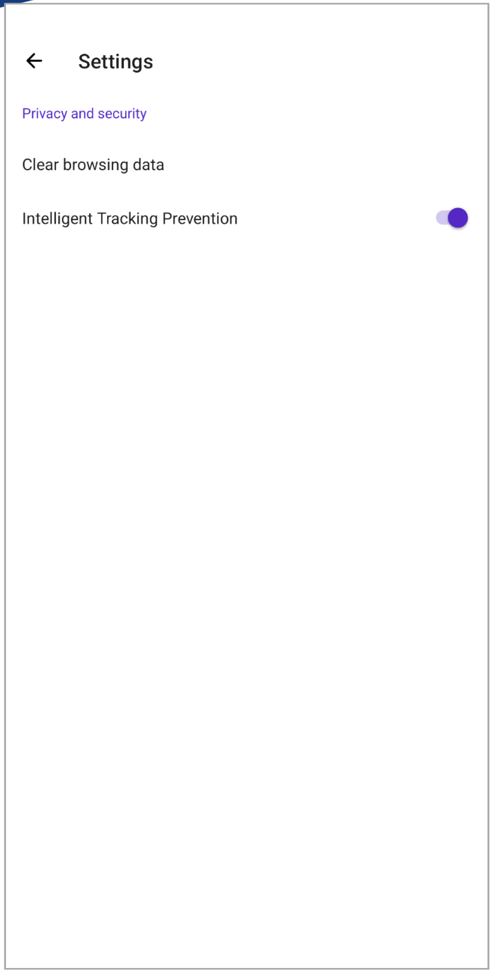


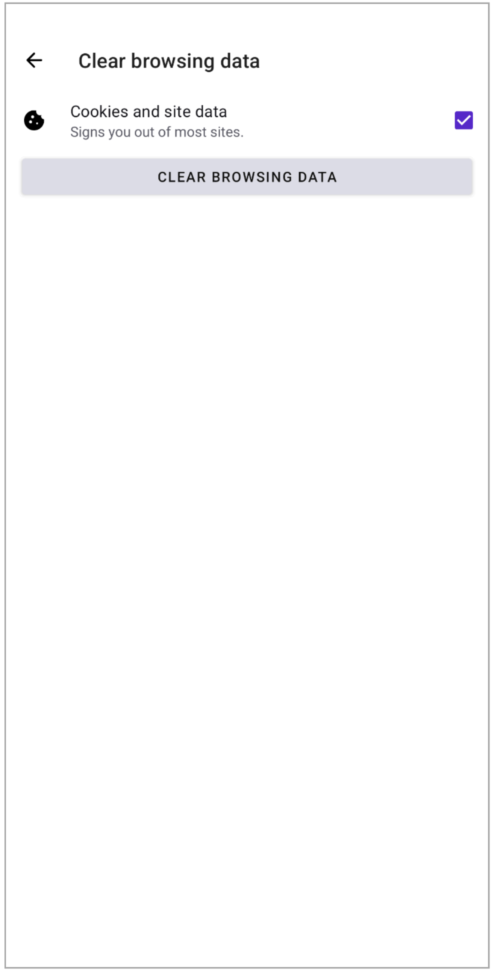
Create simple and performant systems based on **Web platform technologies.**

An Open Source WebKit port for Linux-based embedded devices designed with flexibility and hardware acceleration in mind, leveraging common 3D graphics APIs for best performance.

Explore Embe
Browsers

The Web Platform is a
chosen foundational t
many reasons, includi





Cookies and site data ☑
Signs you out of most sites.

Project History

Adrián Pérez de Castro
2025-06-02

Aside: Android WebView

2008-2013 System, WebKit-based, up to Jellybean.

2013-2014 System, Chromium-based, only KitKat.

2014-now Unbundled, Chromium-based, since Lollypop.

Motivation

Android as **base system**:

- Support by hardware manufacturers (BSPs).
- Optionally without ART/JVM and frameworks.
- Stable* target platform.

WPE WebKit exists:

- Bring back WebKit to Android systems.
- Reuse it, no need for a full-fledged WebKit port.
- Prove its adaptability.

More Motivation

Provide a Web engine option that is more customizable.

Avoid reliance on Chromium / Google.

More Motivation

Provide a Web engine option that is more customizable.

Avoid reliance on Chromium / Google.

Contribute to a more **diverse Web engine ecosystem**.

Project History

2017 Project starts, basic demo.

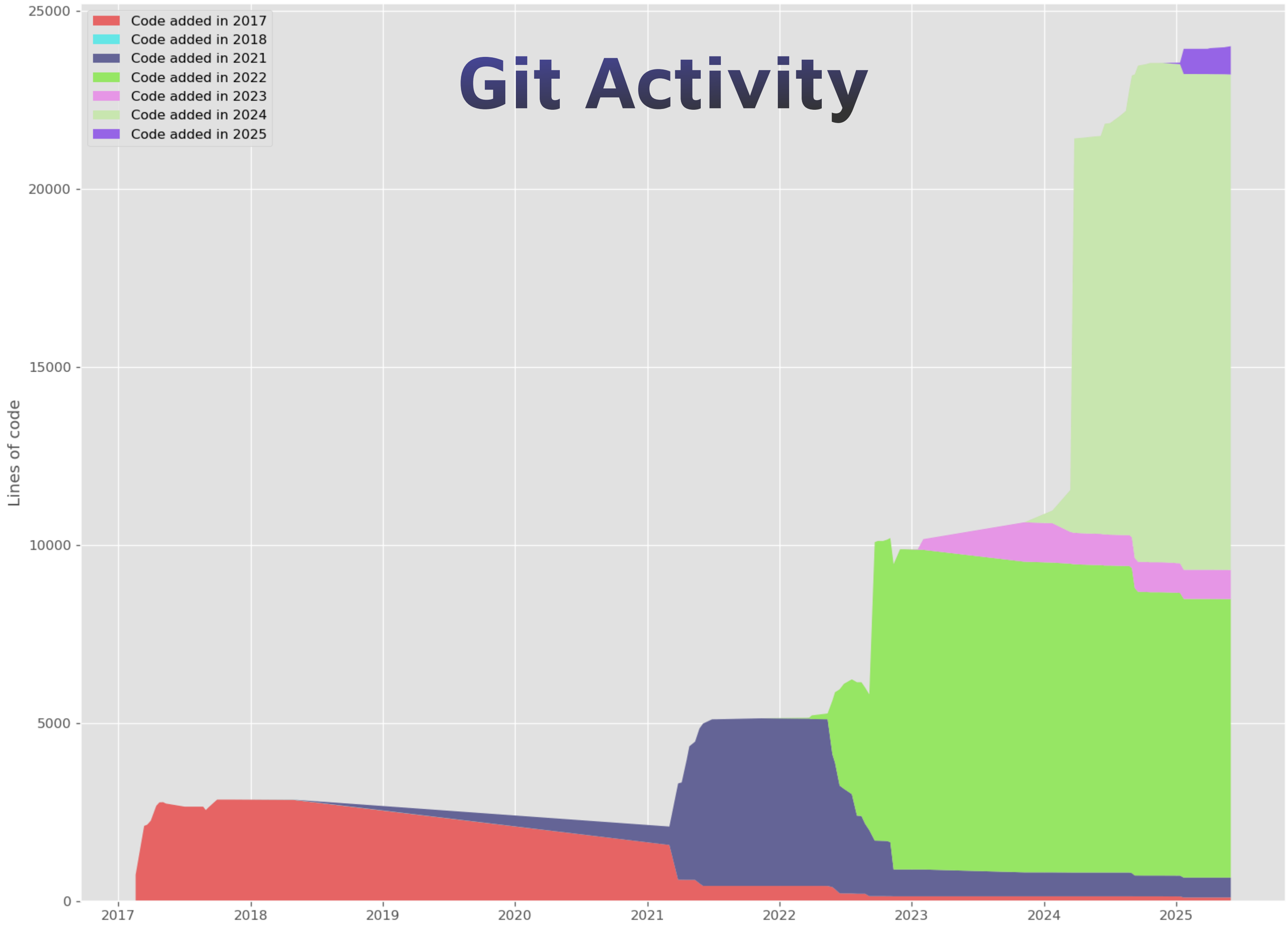
2021 WebView widget API based on `GLSurfaceView`.

2023 Development picks up again, NLnet grant.

2024, March Drop 32-bit, mainloop integration, PSON, multimedia hardware decoding, WebGL, WebDriver.

2024, August Many patches upstreamed, HTTP/2, output scaling, JS dialogs, remote Web Inspector, more WebView API.

2025 Skia, WebKit builds OOTB, Android 15, WebKit 2.48, Maven repository.



Git Activity

What Now?

Behold: The Future

- Track the WebKit `main` branch.
- Switch to the WPEPlatform API.
- Add more WebView APIs.
- Quality-of-Life improvements.
- Continue regular maintenance.
- More system integration: `PowerManager`, long-tap gesture, geolocalization, etc.
- Reduce binary sizes.
- WebXR.

Problem: API Verboseness

Cog Launcher

~1.4K LoC

~3.5K (Wayland)

~4.7K (backend)

MiniBrowser

~600 LoC ~1K (backend)

WPEPlatform

32 LoC.

WPEPlatform: Example

```c
#include <wpe/webkit.h>

int main(int argc, const char *argv[]) {
    g_autoptr(GMainLoop) loop = g_main_loop_new(NULL, FALSE);
    g_autoptr(WebKitWebView) view = webkit_web_view_new(NULL);

    webkit_web_view_load_uri(view,
        (argc > 1) ? argv[1] : "https://wpewebkit.org");
    g_main_loop_run(loop);
    return EXIT_SUCCESS;
}
```

Problem: API Loose-ness

Classic

- Ad-hoc, plain C.
- Very flexible.
- Naming is hard.
- Confusing what does what.
- Does not nudge in the right direction.

WPEPlatform

- Based on `GObject`.
- UIProcess-only.
- Better documentation.
- Encourages good practices.

WPEPlatform

Bonuses

- More opportunities for graphics buffers zero-copy.
- Better render pacing, `DisplayLink`.

Cons

- Needs adding `AHardwareBuffer` sharing implementation inside WebKit (WIP).
- No process launching extensibility (yet).
- API still in flux.

Q&A

Adrián Pérez de Castro
2025-06-02

igalia