

Pairing WebKit and Wayland for Linux-based Embedded Web Content Presentation Systems

Žan Doberšek
Igalia

Düsseldorf, October 13th, 2014

twitter @falconsigh
www blogs.igalia.com/zdobersek



Topics

- WebKit
- Wayland
- Web Content Presentation Systems
- Raspberry Pi

WebKit

Portable Web engine.

- PCs, phones, TVs, IVI, smartwatches
- Mac, iOS, Windows, Enlightenment, GNOME, Haiku

Wayland

Successor to the X windowing system.

Deployed on PCs, phones, TVs,IVI.

Web Content Presentation Systems

Embedded systems presenting Web interface or content.

- Smart TV interfaces
- IVI
- Infoboards

YouTube



Popular on YouTube



Alfonso Ribeiro Carlton Dance 2014 DWTS

by AMOGG Productions

508,900 views 1 day ago



SPOILERS The Drama Continues This Autumn -

by BBC

110,845 views 1 day ago



Jimmy Fallon feat. will.i.am Ew! (Official Music Video)

by The Tonight Show Starring Jimmy

3,414,420 views 1 day ago

S

Search

youtube.com/tv

Raspberry Pi

The popular single-board mini-computer.

- Single-core 700MHz ARMv6 CPU
- 512MB RAM (Model B)
- VideoCore IV GPU

Raspberry Pi – Wayland support

Drivers partially open.

Openly available implementation of the Wayland EGL platform.

Support in Mesa – in progress.

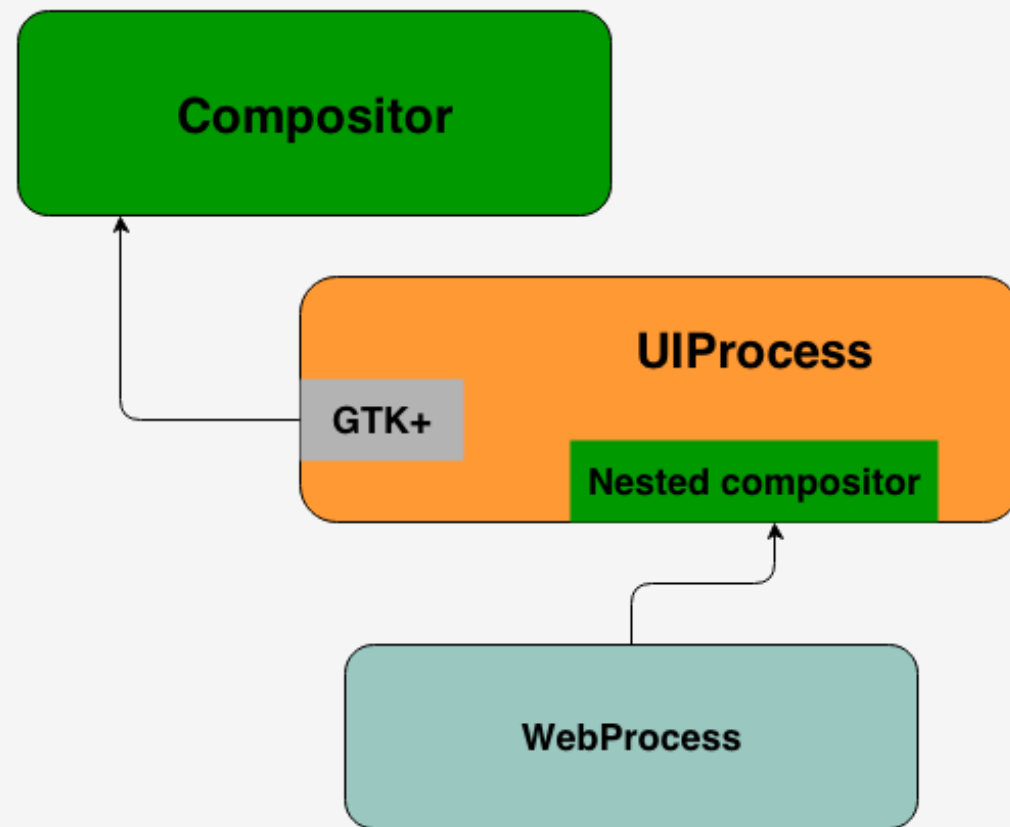
WebKit and Wayland

Wayland support provided by the toolkit.

Must be run under the parent compositor.

Complicated hardware-accelerated compositing of Web content.

- A nested Wayland compositor needed in the UIProcess.



... but why the toolkit?

Adds unnecessary complexity in the architecture.

No need for various widgets, theming support.

Not run under the traditional desktop environment.

The idea

Merge the UIProcess and the compositor.

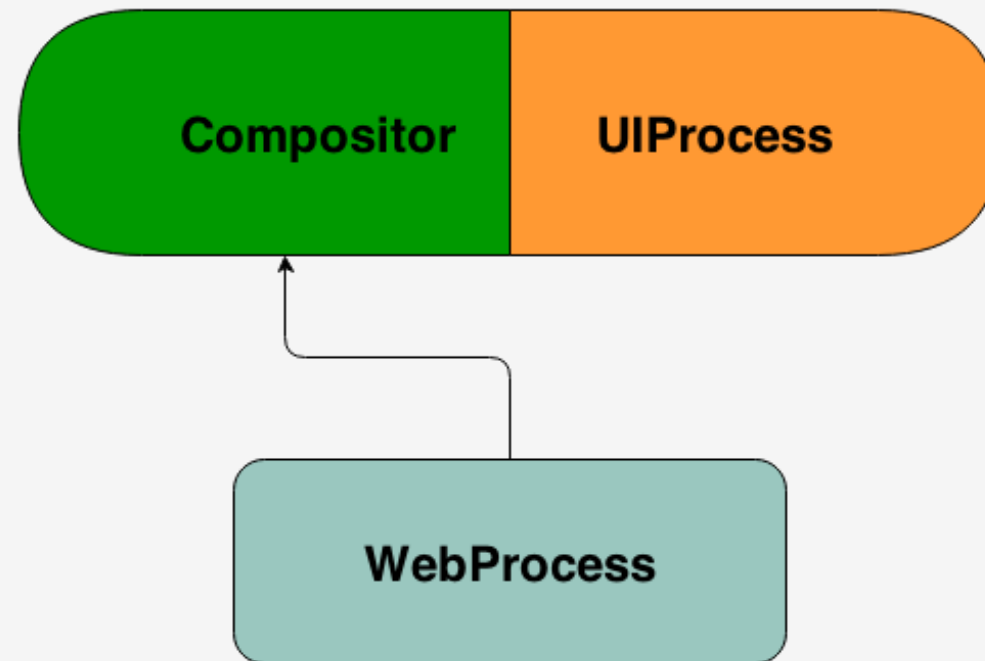
Have the WebProcess interact directly with the compositor.

Benefits

Removes the intermediate purpose of the UIProcess.

Reduces the architectural complexity.

Enables simplifying the compositor.



Implementation

UIProcess now a shared library that the compositor loads.

Compositor shows the single surface designated to the WebProcess.

Input events originate from the compositor, handled by the UIProcess.

Testing it out on the Raspberry Pi

Targeting 720p, 1080p a sweet dream.

60FPS on simple CSS 3D animations, WebGL content.

Fluent 720p video playback (hardware decoder).

Reducing memory consumption.

Still, weak and limiting hardware.

Simplified Wayland compositor

Initially stuck with Weston.

DE-oriented compositors include window management.

Target use cases give leeway to simpler implementations.

Simplicity yields better performance.

The future

Eventually address more complex use cases.

Some sort of window management might be required.

Support for additional forms of input.

Web standards support.

Other (better!) hardware

#1 dependency: Wayland EGL platform implementation.

- NVIDIA Tegra (Jetson TK1)
- Intel HD graphics (Minnowboard MAX)
- ARM Mali (Odroid U3)

Questions

Thank you.