



Chromium Embedded Framework Integration

Presented by

Zoltan Kuscsik, PhD

Date

July 2015

Contact: zoltan.kuscsik@linaro.org

- Linaro is #3 company contributor

- Kernel maintainers from Linaro

- Coresight framework
- Backlight Class
- Clocksource and Clocksource Core
- CPU Frequency drivers
- CPU Idle drivers
- ARM big.LITTLE
- DMA Buffer sharing framework
- Generic PM domain
- GPIO Subsystem
- KVM for ARM and ARM64
- Multifunction devices (MFD)
- MMC/SD/SDIO Subsystem
- Open firmware and flattened device tree
- Pin control subsystem

Top Linux Contributors by Company: Kernels 3.11-3.18		
1	None	11,968 (12.4%)
2	Intel	10,108 (10.5%)
3	Red Hat	8,078 (8.4%)
4	Linaro	5,415 (5.6%)
5	Samsung	4,290 (4.4%)
6	Unknown	3,842 (4.0%)
7	IBM	3,081 (3.2%)
8	SUSE	2,980 (3.0%)
9	Consultants	2,451 (2.5%)
10	TI	2,269 (2.4%)

Source: <https://www.kernel.org/doc/linux/MAINTAINERS>



Linaro Home Group

- Linux Kernel Upstreaming
- LSK - Linaro Stable Kernel
- Energy Aware Scheduling
- Open Portable TEE
- LHG - Digital Home Group
- LMG - Mobile Group
- LCG - Community Board Group
- LAVA - **L**inaro **A**utomated
Validation **A**rchitecture
- Landing Team and Premium
Services

Linaro Home Group

Our Members



HTML5 on Embedded Linux

Hot topics:

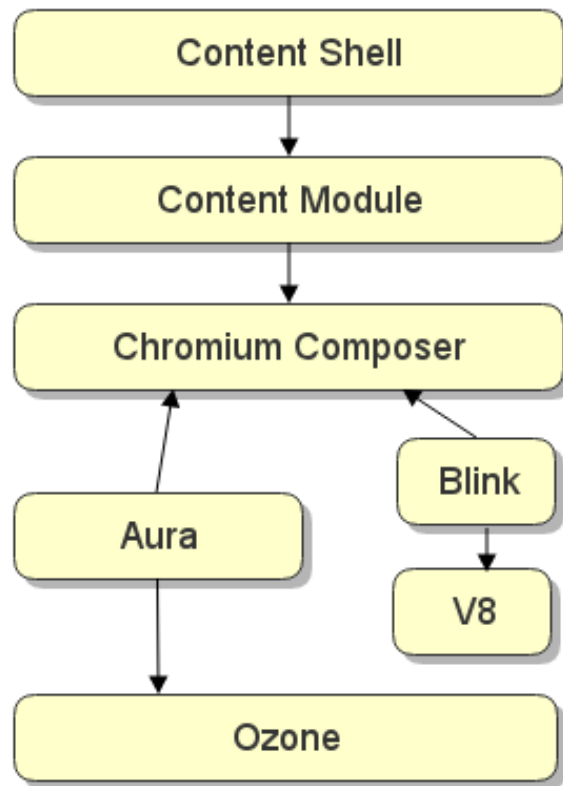
- **How to embed a browser into native applications?**
 - CEF, WebKit, QtWebengine, Opera, Android WebView (Chromium).
- **HW accelerated video playback.**
 - Everyone has a hack for this.
- **Encrypted Media Extensions**
 - Adding DRM to the video playback hack

Why Chromium?

- It has GPU accelerated rendering.
- Uncertain future of WebKit.
- Various abstraction layers, plugin support: Ozone, Wayland support, PPAPI and NaCL.
- Sandboxing support

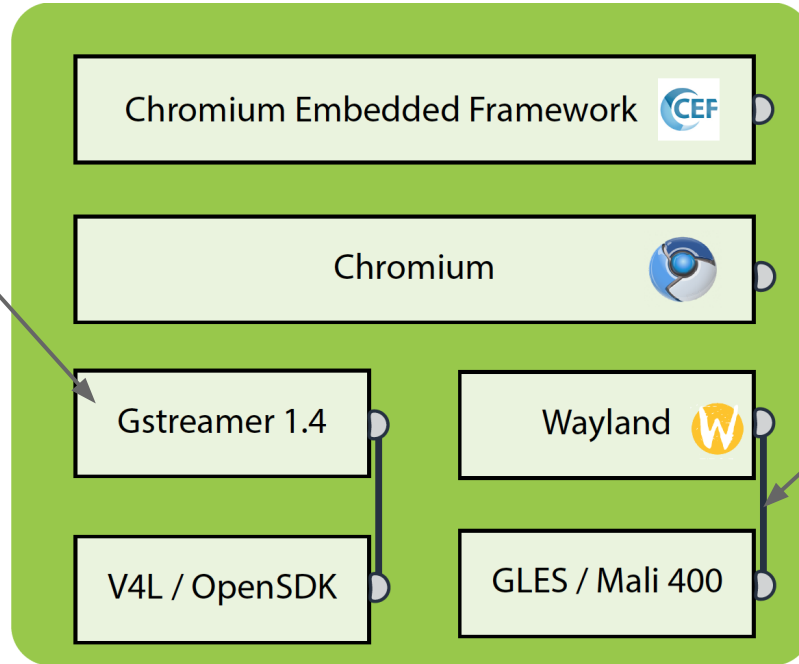
Chromium internals

- **Blink** - is the layout engine of Chromium based on webkit.
- Content Module** - is the code base that enables to render a page using Blink. It includes all the HTML5 platform features, GPU acceleration.
- Content Shell** - simple browser implementation on the top of the Content API
- Aura** - Is the newly introduced window manager in Chromium. Historically, the Content Module was directly depending on the GTK2 and Windows HWND.
- Ozone** - is an abstraction layer for input events and graphics below the Aura window manager. Its purpose is to abstract away the OS specific windowing systems..



Chromium/CEF on ARM/Linux based Set-Top-Boxes

Gstreamer interface using a PPAPI plugin



Adding DMABuf / DRM support without the need of interfacing GBM/Mesa

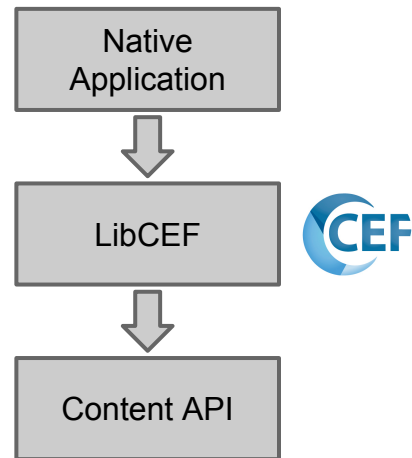
Chromium Embedded Framework

- Small library wrapping the chromium into a library **libcef**.
 - Allows embedding browser window into third party applications. The goal is to provide a stable API on the changing Content API.
 - Enables native JavaScript CEF extensions.

Upstream CEF supports only X86 builds with GTK/X11 backends. We added support for ARM, Aura based window system and Ozone interfaces:

<https://github.com/OSSystems/meta-browser>

<https://github.com/kusc sik/chromiumembedded>



Graphics/Input system hardware abstraction: Ozone

- Better support for out-of-tree Chromium sub-projects.
- **Support for hardware overlay based view composition!!!**
- Provides abstract interfaces Platform Window, Cursor and Surfaces.

Running chromium with a selected Ozone implementation:

```
# ./chromium --ozone-platform=egl
```

Ozone implementations: *x11-egl*, *DRM* (Mesa/GBM), *Wayland*, *Fullscreen EGL* (Linaro), *test* (png output), *caca* (text based).

GPU acceleration

- Requires EGL extensions for Fence sync
- Various workarounds in the Chromium tree for different embedded GPUs.
- Fallback to CPU based rendering if startup GPU test fails.
- Improves the speed of CSS2 animations - able to get around 40FPS for full screen animations at 1080p on a set-top-box.

No dedicated benchmark to measure improvements. Benchmarks are focusing are mostly exercising CPU based rendering and arithmetic operations.

Building Chromium using OE/Yocto

- OE recipes are provided by **meta-browser**:
<https://github.com/OSSystems/meta-browser>

The layer OE layer depends on openembedded-core and meta-openembedded available from <http://git.openembedded.org/>.

meta-browser supports X11 or Wayland enabled OE builds and x86, x86-64, ARMv6, ARMv7 targets.

Video playback

FFMPEG - works on all platforms, but SW decoding is not an option on Linux embedded systems.

VAAPI - GPU accelerated video playback. Compile time option, not enabled in Chrome builds.

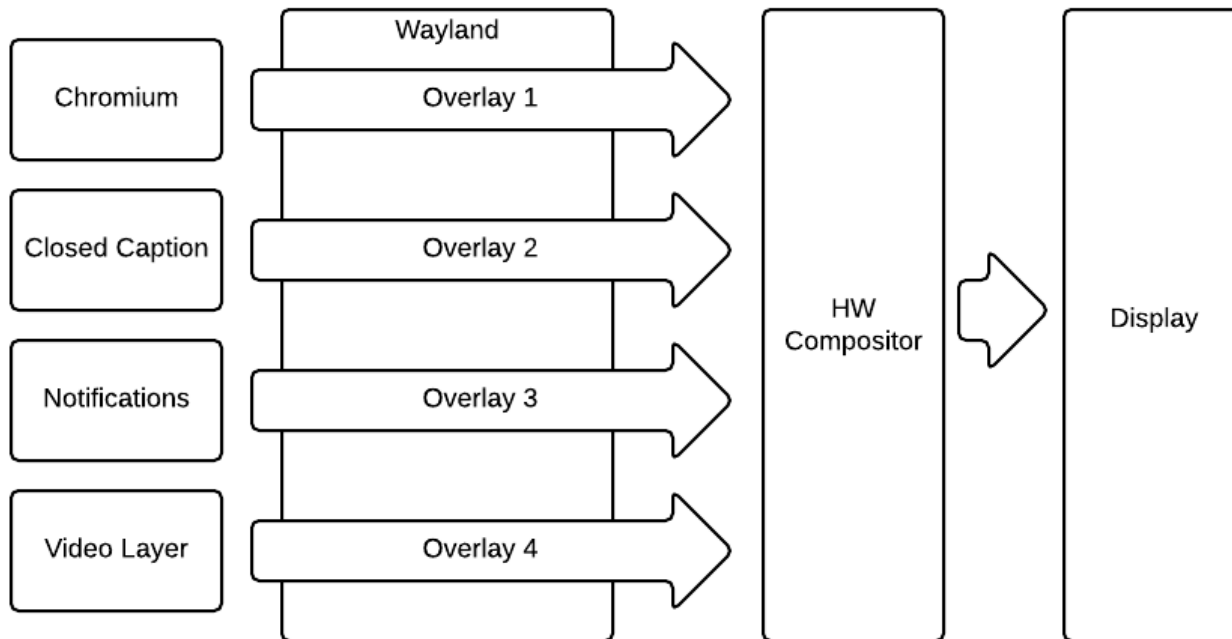
PPAPI - Video playback using a PPAPI plugin

Linaro PPAPI/Gstreamer

- Rationale - concerns about the performance of non-tunneled video playback and lack of secure video data path.
- Using hole punching - chromium MUST be launched on fullscreen.
- The approach works only with sandboxing disabled.

```
<html>
<body>
<br/>
<br/>
<div style="text-align:center">
  <button onclick="stop()">Stop</button>
  <button onclick="playPause()">Play/Pause</button>
  <button onclick="makeBig()">Big</button>
  <button onclick="makeSmall()">Small</button>
  <button onclick="makeNormal()">Normal</button>
  <br>
  <embed id="video1" type="application/x-ppapi-gst"
    width="420" height="200" src="http://192.168.10.2:8080/data.ts">
</div>
...
-
```

Wayland - Ozone Composition



Demo time

Chromium - overlay support

