Wayland in IVI systems

weston, wayland ivi extension, waltham

FOSDEM 2019 Eugen Friedrich
Agenda

- Who I am and what is my daily business?
- High level overview of the Linux graphics stack
- Why we need something different in ivi as on desktop
- How weston is used for ivi?
  - Weston overview
  - ivi-shell
    - ivi-application protocol
    - ivi-wm protocol
- What's next?
  - ID agent
  - Distributed HMI
Who I am and what is my daily business?

- Eugen Friedrich

1984 born in Ukraine, since 1999 in Germany

school, army, master degree in applied computer science, married

since 2010 till today employee at ADIT

Responsible for graphics stack in ivi platform including display pipeline, GPU, video-in, input handling.

Graphics domain lead

Private favourite activities: mountains, marathons, mathematics
Who I am and what is my daily business?

DENSO Products

Requirements

DENSO Crafting the Core

BOSCH Products

Requirements

BOSCH Invented for life

ADIT

HMI

Application A
Application B
Application C

Middleware
OS (BSP)
SoC

BOSCH Products

ADIT Company Overview Jan 2018
© Advanced Driver Information Technology Corp reserves all rights, including rights of disposal such as copying and passing on to third parties.
High level overview of the Linux graphics stack
wayland case

- Tuner app
- Media app
- OpenGLES application
- Wayland application
- Weston (Wayland compositor)
  - renderer
  - backend
  - shell
- EGL/OpenGLES driver
- GPU driver
- KMS
- Linux kernel

Software
- Application framework (Qt, GTK+, Kanzi)
- OpenGLES driver
- Tuner app
- Media app

Hardware
- GPU
- Display controller
- GPU driver
- KMS
- Linux kernel
- Weston (Wayland compositor)
- renderer
- backend
- shell
- OpenGLES application
- Wayland application
Why we need something different in ivi as on desktop?

Window management: user controlled vs system controlled

- Wayland architecture suggests to implement the display server, compositor and window manager in single component wayland server -> Weston follows this suggestion
  - Weston provides well defined plugin infrastructure

Main difference in ivi projects is the window management (shell plugin) and the applications

- Window management is project specific and typically user has less explicit control in ivi systems
- Every OEM* has a set of applications or develops new applications for the particular project

Particular IVI project could implement own shell with own logic and implement applications accordingly, but this has far-reaching consequences:

- Applications are less portable
- Harder to update Weston core since window management is implemented as plugin of weston

*Original Equipment Manufacturer: here some car manufacturer
How weston is used for ivi?

Graphical architecture with ivi-shell

1. Tuner app
2. Media app
3. Application framework (Qt, GTK+, Kanzi)
4. OpenGLES application
5. Wayland application
6. Weston (Wayland compositor)
7. ivi-application, ...
8. EGL/OpenGLES driver
9. GPU driver
10. GPU
11. KMS
12. Display controller
13. Linux kernel
14. Hardware

Software

ivi-shell (ivi-controller)
ivi-shell (ivi-input)
ivi-wm, ...

HMI controller (window manager) (project specific)
How weston is used for ivi?

wayland-ivi-extension

weston repository

weston

<table>
<thead>
<tr>
<th>gl renderer</th>
<th>drm backend</th>
<th>ivi shell</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>hmi controller</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ivi controller</td>
</tr>
<tr>
<td></td>
<td></td>
<td>id agent</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ivi input</td>
</tr>
</tbody>
</table>

ivi-wm

ivi-input

wayland ivi extension repository

examples, wrapper over the wayland protocol
minimalistic hmi controller

client impl. ivi controller

client impl. ivi input
How weston is used for ivi?

ivi-wm protocol

- ivi-wm protocol allow to build scene graph from 3 types of objects

ivi screens

ivi layers

ivi surfaces

- ivi-wm protocol provides events for object creation and deletion and also changes of the object properties
**What's next?**

**ID agent**

- **Motivation**
  - avoid requirement to use ivi-application protocol in applications and frameworks
  - add support for xdg protocol in the ivi-shell
  - xdg protocol is designed to support desktop use cases and is supported also by many frameworks and even other wayland compositors (mir, gnome)

- **Solution: ID agent**
  - additional plugin for Weston (for ivi-shell) to generate and assign the ivi-id inside of the ivi-shell
    - Policy for id generation is decided by the concrete ID agent implementation, e.g. it can use some configuration which will map window-name attribute from xdg protocol to some ivi-id
  - application can still use the xdg protocol and their visibility and e.g. position will be controlled by the ivi-wm protocol
  - not all of the xdg protocol request can be full field as expected, e.g. dimensions of the client will be always limited by the ivi-layer
What's next?

ID agent

Tuner app → Media app

Application framework (Qt, GTK+, Kanzi)

OpenGLES application → Wayland application

xdg-protocol

Weston (Wayland compositor)

ivi shell (xdg support) (ID-agent)

gl renderer → drm backend

EGL/OpenGLES driver

HMI controller (window manager) (project specific)

xdg-protocol or ivi-application

ivi-wm, ...

EGL/OpenGLES driver → GPU driver

GPU driver → KMS

KMS → Linux kernel

Hardware

software

GPU

Display controller

05.01.2019 · Wayland in IVI systems
© Advanced Driver Information Technology Corp reserves all rights, including rights of disposal such as copying and passing on to third parties.
What's next?
distributed hmi, waltham

- Distributed HMI use-case
  - Graphical content is generated on one device and displayed on another
  - Systems can run on same or on different hardware
  - Systems can be very different
    - mix of Android, Linux, QNX, Integrity

- Idea: implement additional Weston plugin 😊
  - Waltham plugin should use Waltham communication protocol for…
    - Negotiation of the sharing mechanism
    - Sharing meta data
    - Forward input events
What's next?
distributed hmi, waltham

- Tuner app
- Media app
- OpenGLES application
- Wayland application
- Weston (Wayland compositor)
- xdg.ivi-application
- gl renderer
- drm backend
- ivi shell
- ivi-wm, ...
- HMI controller (window manager) (project specific)
- Streaming over ethernet
- waltham transmitter
- waltham receiver
- EGL/OpenGLES driver
- GPU driver
- KMS
- Linux kernel
- GPU
- Display controller
- Application framework (Qt, GTK+, Kanzi)
- Different system
- Tuner app
- Media app
- OpenGLES application
- Wayland application
- Weston (Wayland compositor)
- xdg.ivi-application
- gl renderer
- drm backend
- ivi shell
- ivi-wm, ...
- HMI controller (window manager) (project specific)
- Streaming over ethernet
- waltham transmitter
- waltham receiver
- EGL/OpenGLES driver
- GPU driver
- KMS
- Linux kernel
- GPU
- Display controller
- Application framework (Qt, GTK+, Kanzi)
- Different system

05.01.2019 · Wayland in IVI systems
© Advanced Driver Information Technology Corp reserves all rights, including rights of disposal such as copying and passing on to third parties.
A joint venture company of
Robert Bosch GmbH / Robert Bosch Car Multimedia GmbH and DENSO Corporation
What's next?

- Write set of Waltham plugins for OS’s and sharing mechanism

- Virtualization
  - Needs performant implementation for memory sharing and standard protocol for communication about the changes and new content

- Animation support in the compositor?

- Gesture support in the compositor?
Do you have some?
Additional reading

- **Wayland**
  - [https://wayland.freedesktop.org/](https://wayland.freedesktop.org/)

- **Wayland ivi extension**
  - [https://github.com/GENIVI/wayland-ivi-extension](https://github.com/GENIVI/wayland-ivi-extension)

- **Waltham**
  - [https://github.com/waltham/waltham](https://github.com/waltham/waltham)

- **Genivi**
  - [https://www.genivi.org/](https://www.genivi.org/)

- **AGL**
  - [https://www.automotivelinux.org/](https://www.automotivelinux.org/)

Slides are provided under the CC-BY licence