

Waltham working session

October 11, 2018

Harsha Mallikarjun

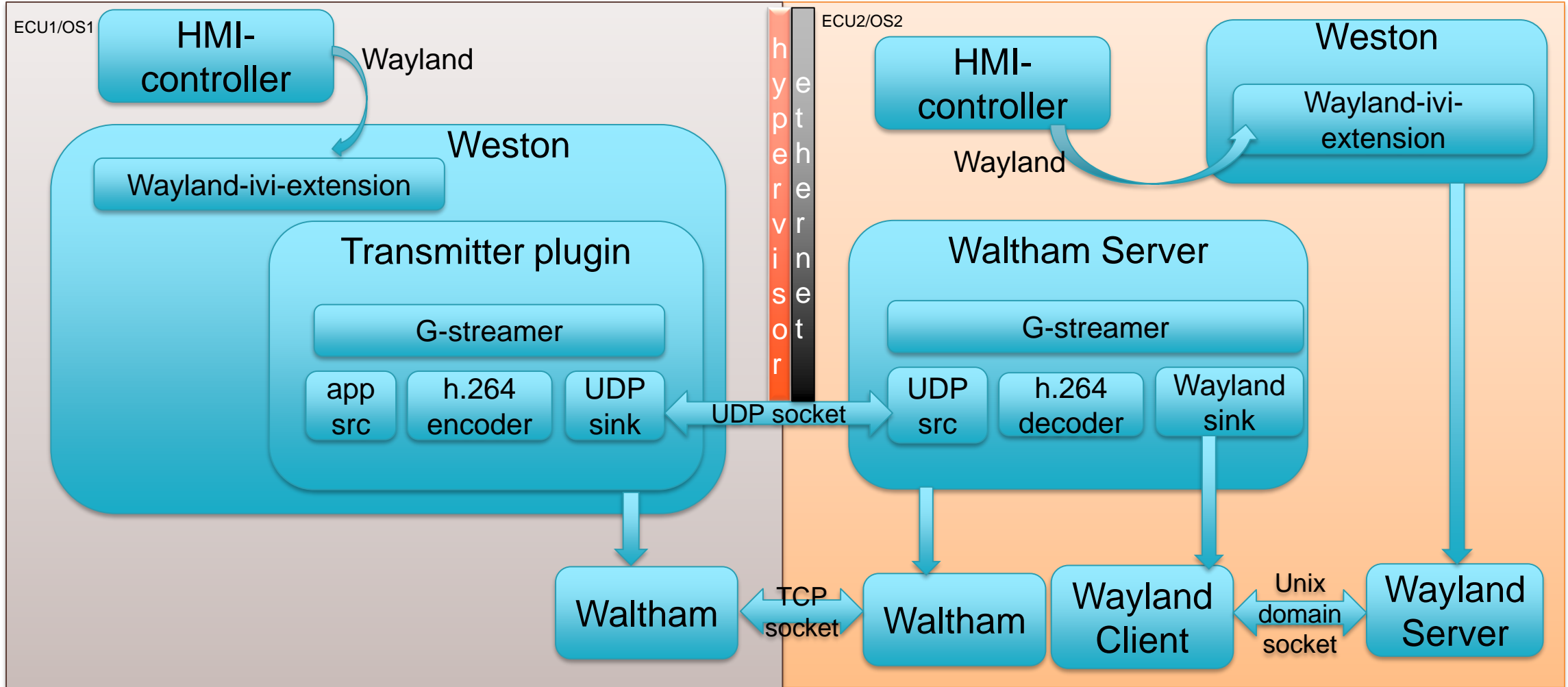
ADIT, GENIVI Alliance

This work is licensed under a Creative Commons Attribution-Share Alike 4.0 (CC BY-SA 4.0)
GENIVI is a registered trademark of the GENIVI Alliance in the USA and other countries.
Copyright © GENIVI Alliance 2018.

Introduction

- Advanced Driver Information Technology GmbH Advanced Driver
Information Technology
 - Joint venture between BOSCH and DENSO.
 - Platform development for IVI systems.
 - Maintainer of wayland-ivi-extension, DLT.
- About Me:
 - Graphics architect at Robert Bosch Engineering and Business Solutions Private Ltd.
 - Working for ADIT since 8 years in the Graphics domain.

Waltham in practice at ADIT



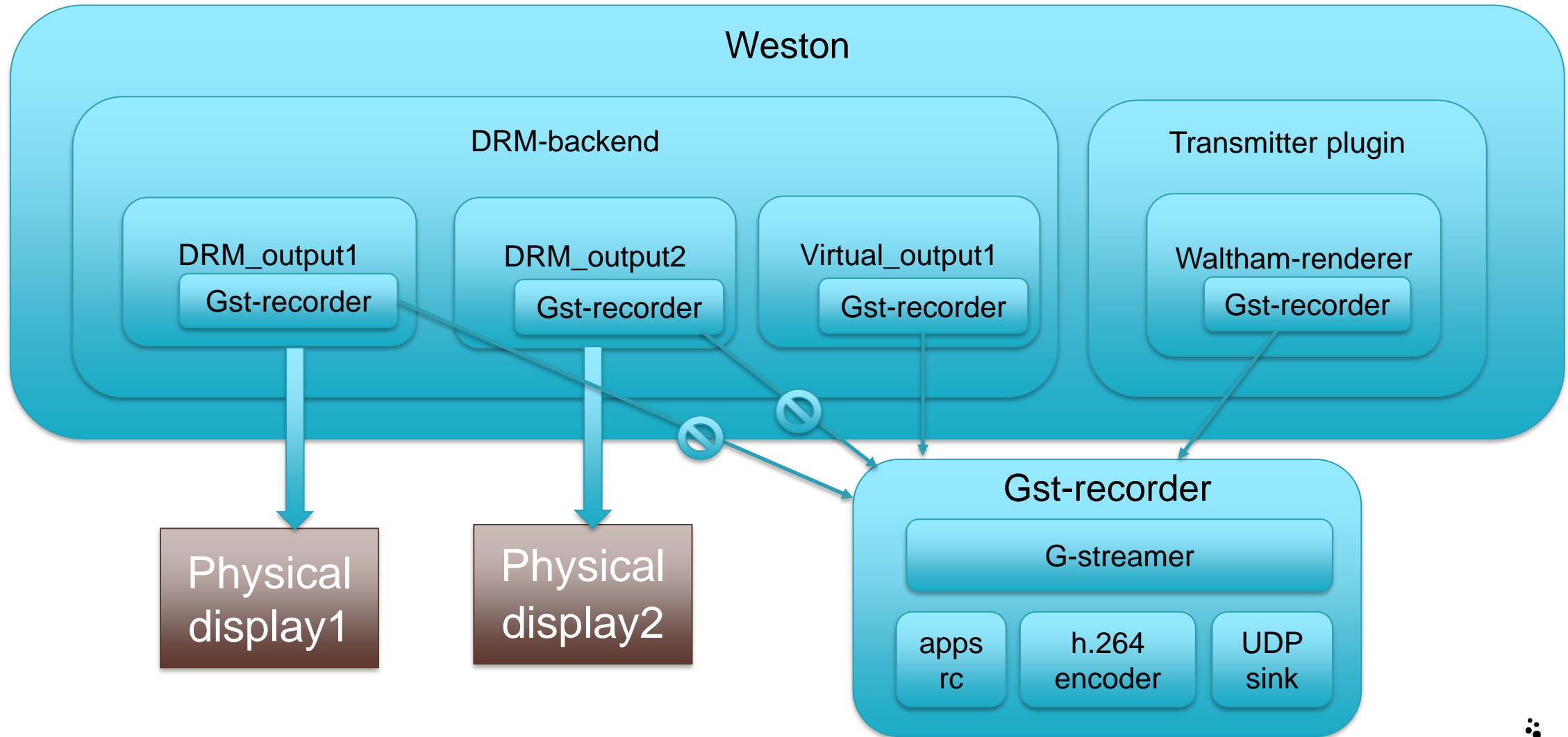
Waltham in practice at ADIT

- Weston is used as the wayland compositor.
- Transmitter plugin is implemented for weston which acts as a Waltham based virtual display.
- Application surface which should be sent to other ECU/OS is assigned to Waltham virtual display by HMI-controller. HMI-controller uses wayland-ivi-extensions to achieve this.
- Pixel data of the surface is sent via g-streamer to another ECU/OS as the raw pixel data transfer over Waltham is inefficient.
- Control data and input events (pointer, keyboard, touch) for the surface, are handled via Waltham.

Waltham working session

- What to look forward for?
 - Gst-record and transmitter plugin.
 - Waltham on android.
 - Additional use cases for transmitter plugin.
 - Challenges.

Gst-record and transmitter plugin

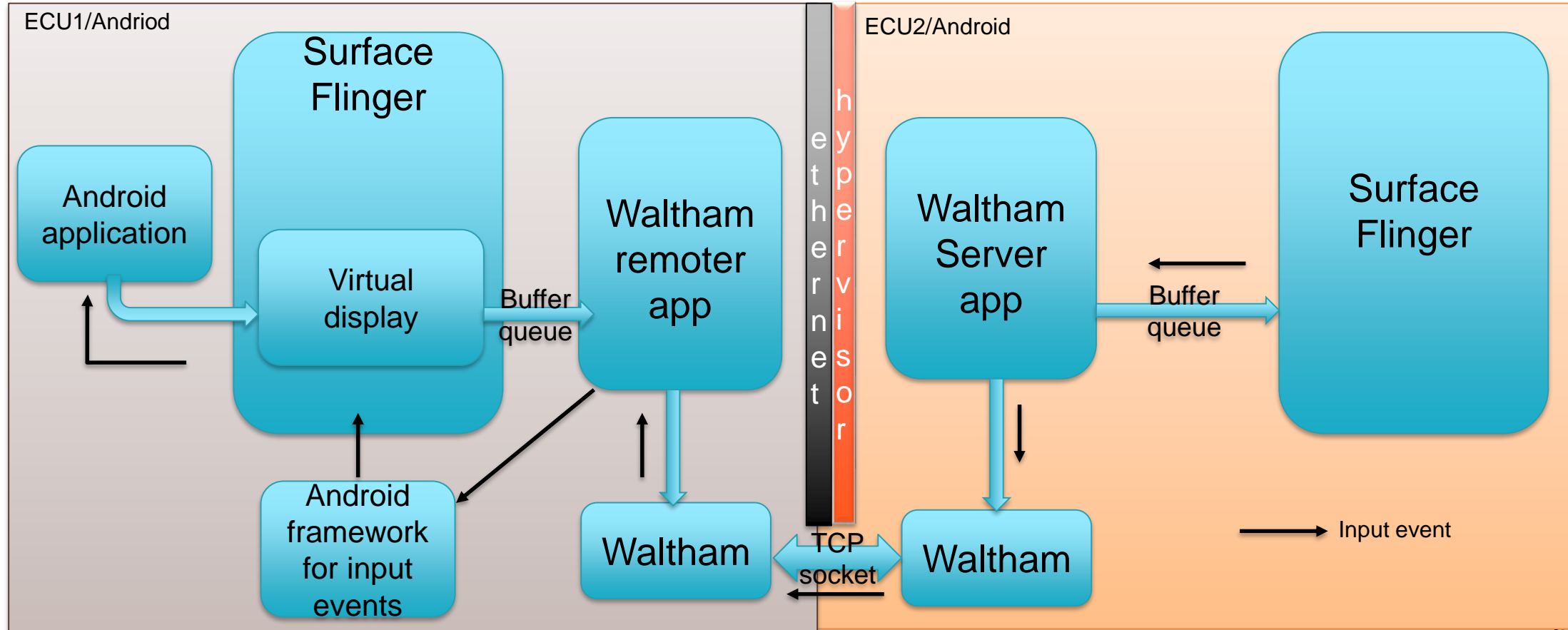


Gst-record and transmitter plugin

- Gst-recorder implementation from Renesas allows streaming out outputs (physical, virtual displays) over network.
- Does not handle input events over virtual displays.
- Can composite surfaces using gl-renderer
- <https://gerrit.automotivelinux.org/gerrit/#/c/13921/>
- https://gitlab.freedesktop.org/wayland/weston/merge_requests/7
- Transmitter-plugin implements makes use of a part of gst-recorder implementation to stream out the Waltham output.
- handles input events over Waltham.
- Lacks composition capability.
- <https://gerrit.automotivelinux.org/gerrit/#/q/project:AGL/meta-agl>

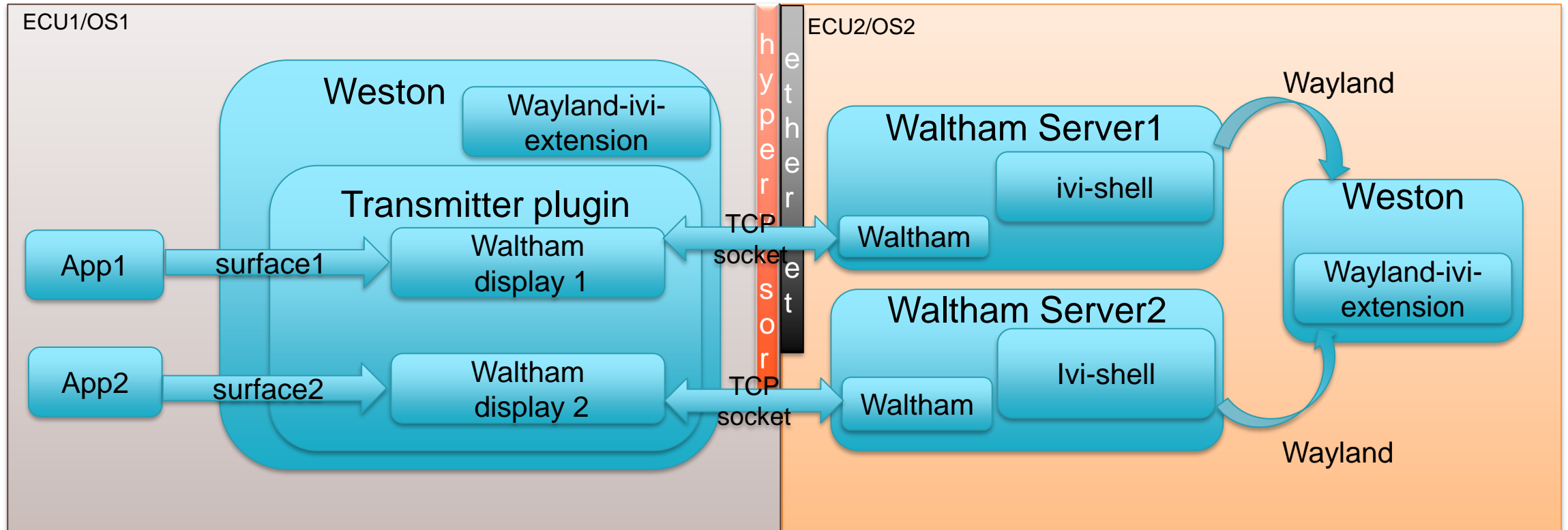
Waltham on Android

- An initial idea



Multiple surfaces over Waltham

- One virtual display for each surface.

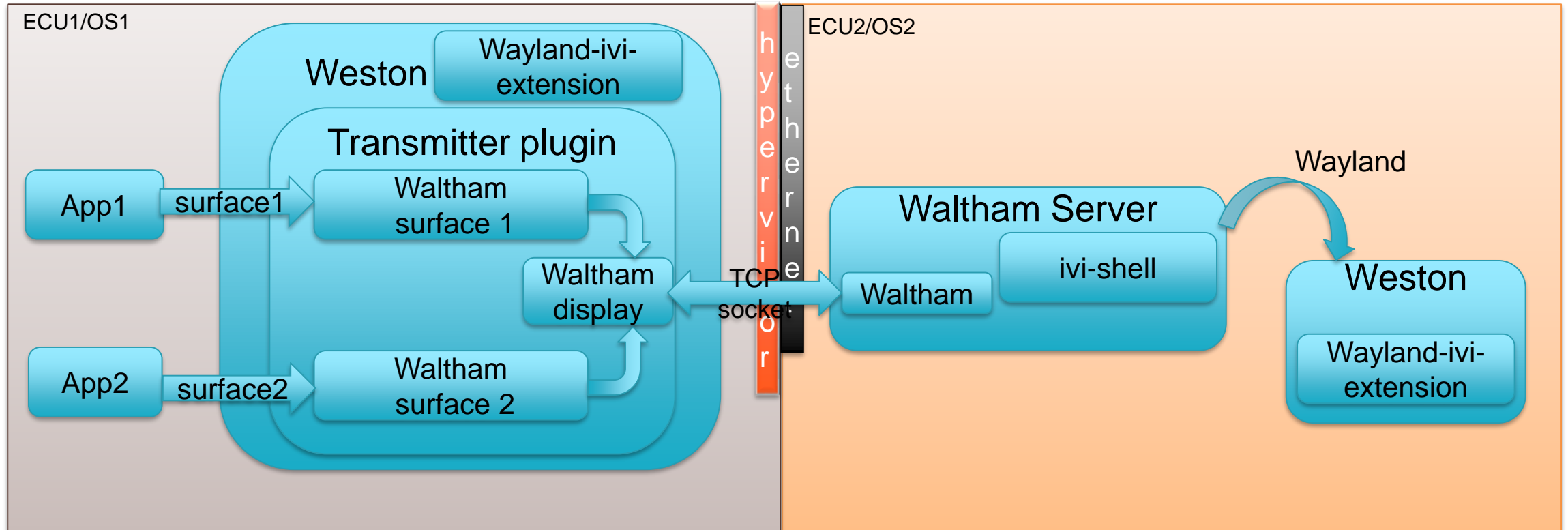


Multiple surfaces over Waltham

- One virtual display for each surface.
 - Is this an optimal solution?
 - In the case of two different ECUs.
 - High usage of resources due to one g-streamer pipeline for each Waltham display. Solution does not look scalable.
 - In the case of two OS running on an hypervisor.
 - Here surface sharing technology (e.g. Hyperdmabuf OR any other) can be used to share pixel data and no g-streamer needed. This solution does not look optimal due to multiple TCP connections and Waltham server processes.

Multiple surfaces over Waltham

- One Waltham surface for each application surface

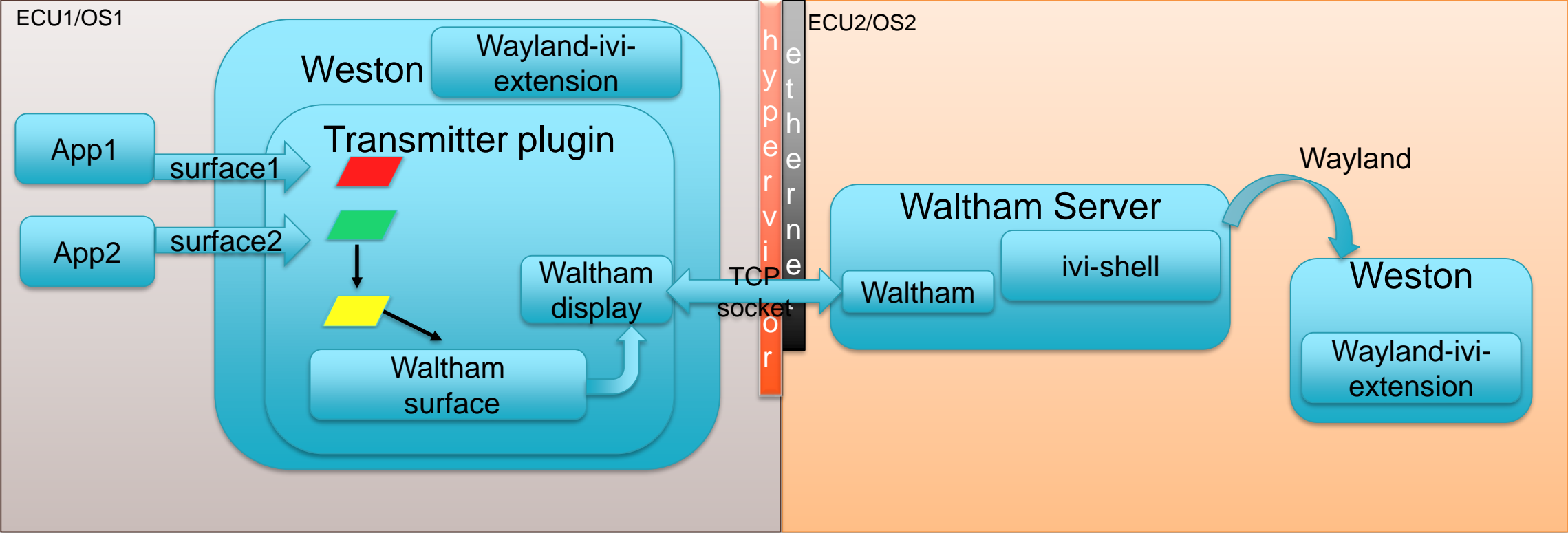


Multiple surfaces over Waltham

- One Waltham surface for each application surface.
 - Is this an optimal solution?
 - In the case of two different ECUs.
 - Optimal usage of resources due only one g-streamer pipeline for each Waltham display.
 - Since only one g-streamer pipeline is used, frames of each of the applications need to be demultiplexed at Waltham sever. Control data which comes over Waltham is specific to each surface. How to de-multiplex frames on the receiver end?
 - In the case of two OS running on an hypervisor.
 - One communication channel for surface handle and control data. Looks optimal.
 - How to handle atomic updates when surfaces are dependent?

Multiple surfaces over Waltham

- Composite application surfaces.



Multiple surfaces over Waltham

- Composite application surfaces.
 - Is this an optimal solution?
 - In the case of two different ECUs.
 - Optimal usage of resources due only one g-streamer pipeline for each Waltham display.
 - In case if the surfaces are independent then additional meta-data for locating and segregating surfaces is needed.
 - In the case of two OS running on an hypervisor.
 - One communication channel for surface handle and control data. Looks optimal.
 - In case if the surfaces are independent then additional meta-data for locating and segregating surfaces is needed.
 - Atomic update can be handled during composition.

Multiple-surface challenge

How to enhance transmitter plugin to meet all the use cases described in previous slides?

Design and upstream challenges

- Transmitter plugin has features of a weston-backend but it is implemented as a plugin. Reason for this being, weston does not support running multiple backend at one point of time. Transmitter plugin is not a perfect fit to weston architecture.
 - Small changes to other backend OR weston-core needed for transmitter plugin to work faithfully.
 - Can weston support multiple backends simultaneously?
- Transmitter plugin is still not upstreamed to and maintained by wayland community. Update of weston calls for changes in this plugin.
 - There are efforts in this direction in AGL

Security challenge

- Waltham does not provide any features to encrypt communication between client and server.
 - How to secure the communication?
 - Use ssh tunneling to forward a local port to remote system port. What is your view?
 - Other options?

References

- <https://people.collabora.com/~pq/Adit/Weston-IVI-remoting.pdf>
- <https://github.com/waltham/waltham>
- <https://gerrit.automotivelinux.org/gerrit/#/q/project:AGL/meta-agl>
- https://wiki.automotivelinux.org/_media/eg-ui-graphics/20170209_ui_and_graphics_eg_waltham.pdf

Thank you!

Visit GENIVI at <http://www.genivi.org> or <http://projects.genivi.org>

Contact us: help@genivi.org

harsha.manjulamallikarjun@in.bosch.com

This work is licensed under a Creative Commons Attribution-Share Alike 4.0 (CC BY-SA 4.0)
GENIVI is a registered trademark of the GENIVI Alliance in the USA and other countries.
Copyright © GENIVI Alliance 2018.

