

The AGL Wayland compositor

Marius Vlad

marius.vlad@collabora.com



COLLABORA

Open First

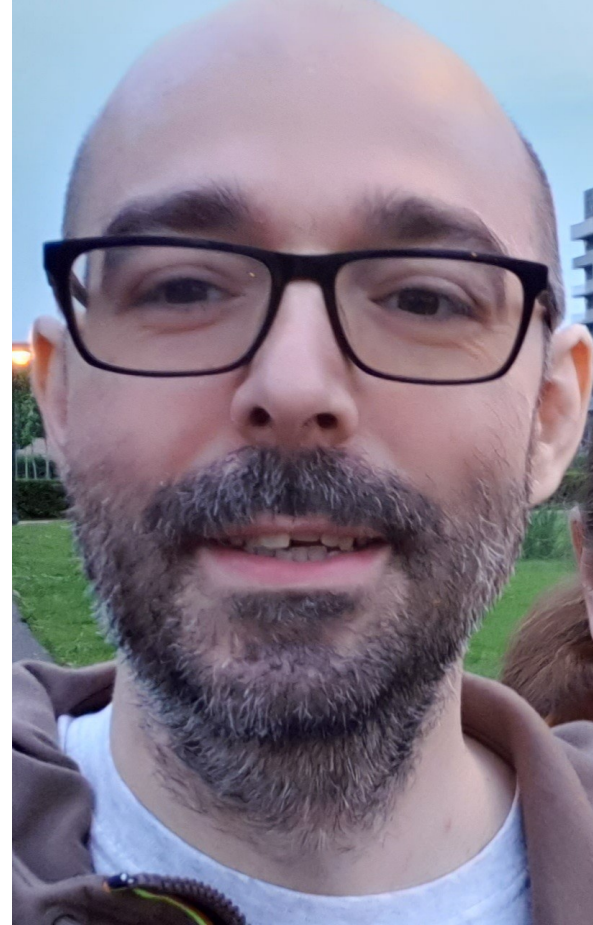


COLLABORA

Hi, I'm Marius

-/-

Open First





COLLABORA

Outline and agenda

Outline and agenda

- Wayland ecosystem, compositor, shells and WMs
- In-Vehicle Infotainment – IVI shell
- AGL Wayland compositor





Wayland ecosystem, compositors, shells and WMs

Wayland, compositors, shells and WMs

- **Wayland** → a protocol; a specification; only local → vs X core protocol, network transparent
- **Wayland Compositor** → an implementation of the (Wayland) protocol; servers → vs X server with a built-in compositing manager



Wayland, compositors, shells and WMs

- **Shell** → how an user and applications interact vs X
Desktop environments: KDE, Gnome, XFCE →
different types of the **same** desktop-shell
- **Window managers** → multiple implementations of
a same interface, wl_shell or xdg-shell

Wayland, compositors, shells and WMs

- ***wayland-protocols*** → other protocol specifications that can standardize different operations:
 - Xdg-shell (suited for traditional DE), linux-dmabuf (dmabuf-based buffers) and many, many more
- Compositor ***private extensions***:
 - Screen shooting, bypass GPU imports, debugging
 - adds **additional functionality** to compositors

Wayland, compositors, shells and WMs

From a system where parts of it are scattered in different components **under** different projects

→ a system where **all** components are under the same project; apart from clients





COLLABORA

In-Vehicle Infotainment shell

In-Vehicle Infotainment shell != Desktop shell

- IVI-shell → different use-cases than on desktop
 - no user interaction for window positioning, spanning or dragging
 - similar to a tiling WM; with a customizable window placement
 - policy: don't show/show certain windows in certain events & conditions



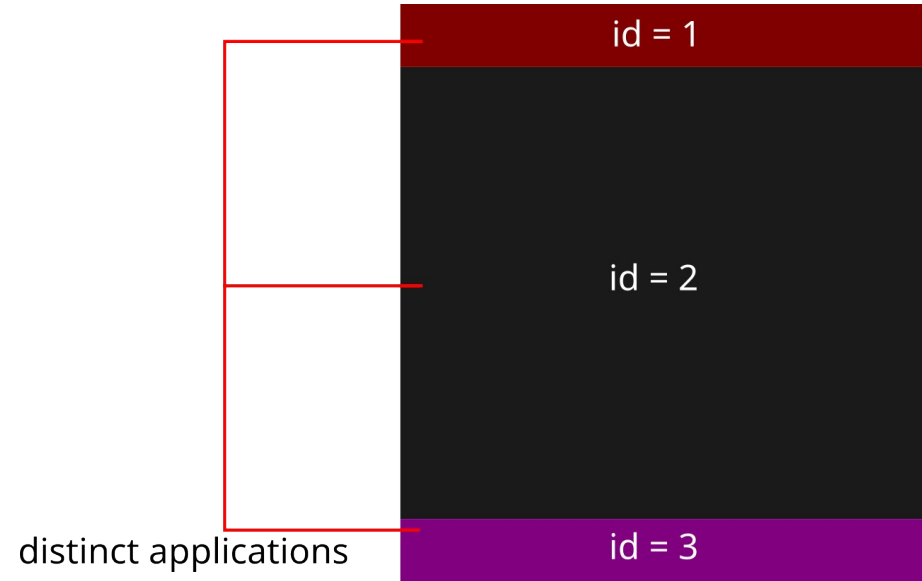
In-Vehicle Infotainment shell

- IVI-shell → seeks to implement the IVI requirements; effort lead by former GENEVI Alliance (now COVESA)
 - private extension in weston compositor
 - based & made up of multiple components
 - retains the idea of splitting components under different projects & processes
 - all clients need to have an implementation



In-Vehicle Infotainment shell

- applications identified as a number: example 3 different applications processes
- requires a controller, acts a window manager that manages layouts and window positioning
- bring your own controller



In-Vehicle Infotainment shell

- Not all doom & gloom
 - it can display desktop clients (xdg-shell)
 - seen an uptick in changes and maintenance effort
- But....
 - maintenance with multiple components & dependencies associated
 - Still a departure from the Wayland compositor paradigm





COLLABORA

The AGL Wayland compositor

The AGL Wayland compositor

- An entire compositor wouldn't be even a bigger issue than writing your own controller?
- Simplicity → fairly similar to Weston; start-up code almost identical
- A fitness process → less components → less maintenance
- Customizable to an different degree → owning the entire compositor

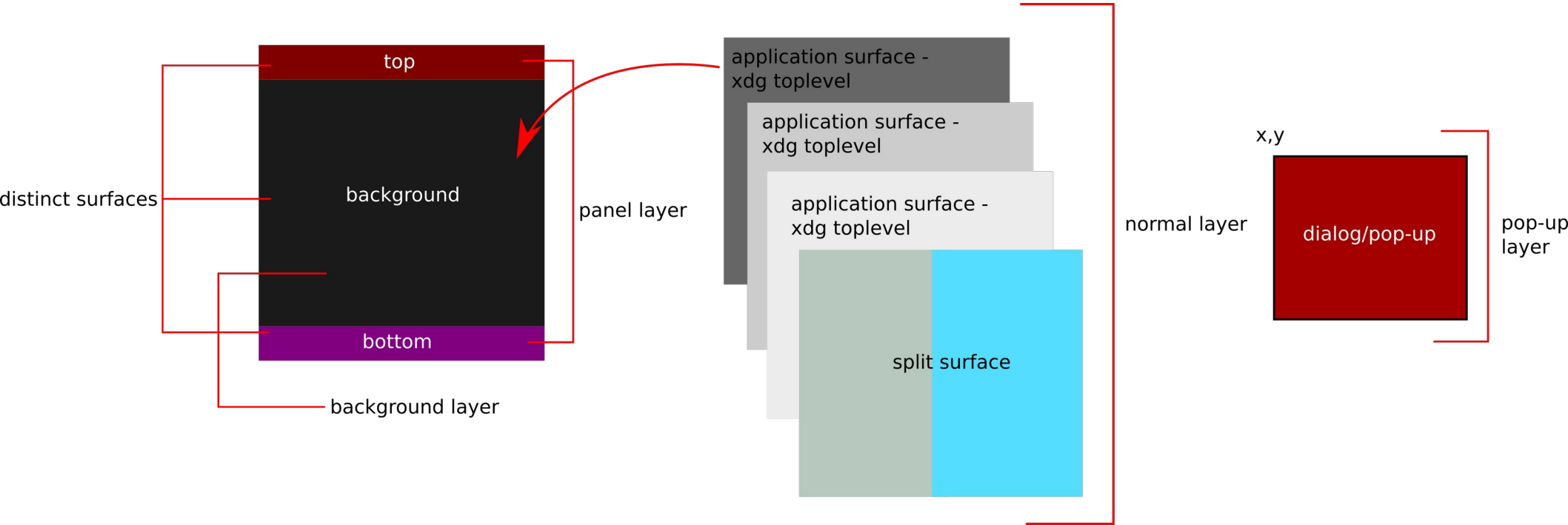


The AGL Wayland compositor

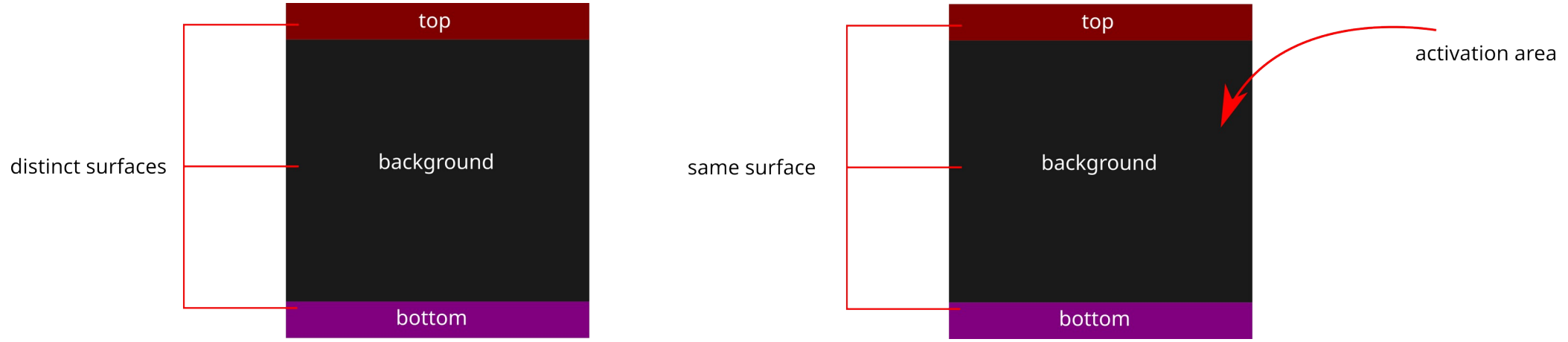
- Libweston-based: include libweston.h + start-up code = Wayland compositor
- All AGL/IVI functionality provided by two private extensions:
 - **agl-shell (agl-shell-desktop)** → only the HMI implementing the client side protocol
 - **gRPC proxy** – additional window management
- clients: all toolkits implement the desktop-like, xdg-shell protocol, use it to perform app identification

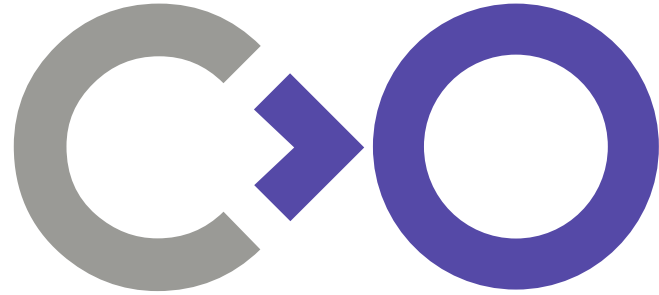


The AGL Wayland compositor - agl_shell



The AGL Wayland compositor – agl-shell – single surface





Thank you!

marius.vlad@collabora.com



COLLABORA

Open First