Wayland/HomeScreen
API based Architecture Proposal

A new graphic architecture to foster user experience inovation
1st technical Contributor

- **Application Development**
  - SDK (binary distro + development ready docker container)
  - Application Framework (live cycle, cybersecurity provisioning)
  - Application Binder Framework (APIs exposure & protection)

- **Integration**
  - Yocto recipes
  - Releases automation & Testing (CI)
  - Renesas boards support
  - Security (MAC, Cynara, Systemd, CGroups, NameSpaces,...)

- **Low Level Services**
  - Audio Manager
  - Connectivity
  - Signaling & Events
  - SOTA
  - Secure Boot & Trusted Zone

- **Community Support**
  - Documentation (kickstart, developer samples, guides ...)
  - White Papers & Conferences (Genivi, AGL, Fosdem, ELC, ...)
  - Renesas Community support
Requirements

- Applications
  - Consistent & Efficient API
  - Independence to graphic toolkit, skills, hardware, ...
  - Multi user+screen+language
  - Long term API stability (XDG protocol ?)
  - Available competency on the market
  - Enable independent software vendor & application portability
  - Scripting Language for quick prototyping

- Platform
  - Efficient, limit resource consumption as much as possible
  - Flexibility to change things (graphic toolkit, homescreen, hardware mapping, ...)
  - Attached to a well supported mainline project
  - Built in Security (Privileges for static ACL & Policies for dynamic ACL)

- Manufacturer
  - Capability to leverage unique feature (input device, head display, ...)
  - Development lower cost and faster time to market
  - Capability to create a unique user experience
  - Long term maintenance
X11 versus Wayland Arch
CES-2017 Homescreen Architecture

https://wiki.automotivelinux.org/homescreen

Wayland/Homescreen API based Architecture Proposal

Feb-2017
Main Wayland Compositors

- Weston
  - Official Wayland reference implementation
- Enlightenment
  - Samsung/Tizen
- Gnome
  - Community driven rely on GTK+ & Mutter
- QT-Compositors
  - The Qt Company
WESTON UI
Weston +/-

- Official reference Wayland compositor
- Does not enforce any graphic toolkit
- Support XDG-protocol
- Is available as an embedded library

- Does not provide any advanced feature
- Ship with almost no tools
- Not an easy community to deal with
Enlightenment EFL

Samsung Tizen Phone
EFL/Enlightenment

- Working solution on millions of Tizen devices
- +/- Maintain and lock by Samsung
- Small, Fast, well adapted to embedded

- Unique/Untypical programmatic logic
- No developer available on the market
Gnome GTK+

Mutter Window manager
GTK/Gnome

+ Huge existing community of developers
+ Native Wayland since Fedora-25
+ Scripting language for UI [Javascript]

- Window Mgr [Mutter] not X11 clean
- Community focus at large on desktop
- Big piece of software
QT Compositor

Based Qt Automotive Suite
QT +/-

- + Well know and understood from embedded
- + Proven viable when embedded
- + Compositor fully integrated with API

- Commercial License or GPL-V3
Why a new architecture?

• Current
  • Is too complex
  • Since it conception many things changed
  • Not integrated with AGL framework
  • Not secure
  • No clean isolation UI/Business Logic

• Proposal
  • Fully leverage XDG protocol
  • Leveraging existing advance Wayland shell/compositor
  • Isolation of Homescreeen UI & Logic
  • Native integration with AGL framework
  • Security Built In
AGL-DD Proposed Logic
Conclusion

• Toolkit Graphic/Compositor
  • Weston: leave a lot of freedom but also plenty of work
  • Gnome: Best for demo, but could be heavy in production
  • Enlightenment: Best for production, but depends on EFL
  • QT: Commercial/GPLv3 license limitation
  • Current should be fully rewritten without IVI-shell
  • Security [ACL static=>Cynara dynamic=>Policy Engine binding API]

• Applications
  • Should support any graphic toolkit
  • Should talk XDG-protocol
  • Should accept to run with security on

• Homescreen
  • Provide a scriptable version for demos
  • Allow people to easily change user experience
  • Should allow separation of business logic from UI
  • Should be implemented as a standard AGLapp
  • Security should be compatible with AGL-FW