



Automotive Grade Linux

AGL Video Architecture and Graphics Update - Collabora Proposal

December 8th 2020

Agenda

- › Discuss framework for AGL video/camera support
- › Update on compositor progress and plans

AGL video project scope: PipeWire

- › Stream setup and configuration framework
- › Container-ready stream exchange framework
- › Stream policy management
 - › with WirePlumber
- › Application isolation & security
- › Low-level, real-time processing capabilities

AGL video project scope: Similarities with Audio

- › Principles are not different, just the data
 - › Input / processing / output
 - › Streams: local, remote, in network
 - › Policy restrictions
- › Reusing the same infrastructure makes sense

AGL video project scope: Benefits

- › Security: hardware isolation from apps & containers
- › Resource sharing
 - › ex. front camera shared between IVI and deep learning containers
- › Streaming
 - › Zero-copy streaming between containers
 - › Secure capture from compositor through PipeWire

AGL video project scope: First Steps

1. Enable video capture support in AGL PipeWire builds
2. Build a demo camera application for experimentation
3. Establish further requirements and document next goals

AGL video project: Goals, scope and hardware

- › Specify initial use cases
 - › i.e. rear-view camera, dash camera, deep learning, 360 degrees top-view, ...
- › Need automotive video capture hardware
 - › Anything with V4L2 drivers would work out-of-the-box
 - › Specialized sensors, network video streams, ...
- › Video playback hardware (& software) matters
 - › Camera video should be passed through to the display
 - › Need support from hardware, kernel, graphics drivers

AGL video project: Requirements and features

- › Automotive requirements
 - › Latency, security, safety
 - › Discuss, understand, iterate on the design
 - › Reflect on PipeWire, upstream as much as possible
- › Additions may be needed in PipeWire & WirePlumber
- › Bring GStreamer up to speed to support applications

AGL video project: Challenges

- › Supporting time-to-rear-view-camera usecase
 - › No support in Linux kernel for power-on presentation of rear view
 - › ‘Live source’ proposal for DRM/KMS prepared but not advanced
 - › Extremely platform-specific, would require specific enablement

AGL video project: Challenges

- › Complex camera processing pipelines
 - › libcamera project created to support dynamic camera ISP pipelines
 - › Further development required to support this on all platforms
- › Zero-copy sharing with machine-learning platforms
 - › No standard for zero-copy sharing of large image or video buffers between acquisition hardware and machine-learning processing hardware
 - › Further investigation and development required to enable this

AGL compositor update: 2020 achievements

- › AppFrameWork integration, client shell + demo apps ported/updated
- › supports Qt/HTML5 toolkits, HTML5 through chromium/ozone
- › policy engine with customisable static policies w/ RBA implemented as a policy
- › rebooted and updated Waltham, updated transmitter/receiver demo apps
- › adding screenshots within CIAT for basic application testing (integration phase)
- › support and help out both regular users and OEM vendors on various inquiries and questions about Wayland and compositor related topics

AGL compositor update: 2021 plans

- › Possibly uprev libweston to new upstream: depends on Renesas BSP schedule
 - › Kurokawa-san, please provide an update on the schedule
- › Continue to support IVI/PR and IC groups for all usecases
- › Support for verifying compositor within CIAT
- › Investigate multiple backends for dynamic remote-output support using PipeWire, to integrate with IC system
 - › Efficient cross-domain, buffer sharing mechanism between IC <-> IVI
- › Content highlighting and multiple gestures handling to migrate content to other outputs

Legal Notices

The Linux Foundation, The Linux Foundation logos, and other marks that may be used herein are owned by The Linux Foundation or its affiliated entities, and are subject to The Linux Foundation's Trademark Usage Policy at <https://www.linuxfoundation.org/trademark-usage>, as may be modified from time to time.

Linux is a registered trademark of Linus Torvalds. Please see the Linux Mark Institute's trademark usage page at <https://lmi.linuxfoundation.org> for details regarding use of this trademark.

Some marks that may be used herein are owned by projects operating as separately incorporated entities managed by The Linux Foundation, and have their own trademarks, policies and usage guidelines.

All other trademarks are the property of their respective owners. Use of such marks herein does not represent affiliation with or authorization, sponsorship or approval by such owners unless otherwise expressly specified.

The Linux Foundation is subject to other policies, including without limitation its Privacy Policy at <https://www.linuxfoundation.org/privacy> and its Antitrust Policy at <https://www.linuxfoundation.org/antitrust-policy>. Each as may be modified from time to time. More information about The Linux Foundation's policies is available at <https://www.linuxfoundation.org>.

Please email legal@linuxfoundation.org with any questions about The Linux Foundation's policies or the notices set forth on this slide.