Feature name	**Upgrade PipeWire and WirePlumber** to latest stable for each UCB release
Functional area	Audio/Video/Multimedia – SPEC-4934
Short definition	The AGL team must have the bandwidth to integrate the latest stable in each UCB release. Ideally that can be coordinated with the upstream Yocto project
Rationale for AGL	While the Yocto project is only making software available, the AGL community is actively contributing to the development of PipeWire and WirePlumber. The features delivered by the projects are key for AGL adopters New features must be easily accessible for all AGL members and adopters
Effort estimation	2-3 man-weeks per AGL release
Target AGL release	Royal Ricefish, Super Salmon

Feature name	**Upstream agl-service-audiomixer bits into WirePlumber** and improve mixer APIs, making them more generic and reusable
Functional area	Audio
Short definition	Agl-service-audiomixer is a relic of the old application framework and carries some mixer logic that could just be part of WirePlumber and maintained upstream, as it no longer depends on AGL-specific components. The idea is to also improve the internal APIs so that the controls are exposed in a more generic way. This would also make them more reusable in other projects
Rationale for AGL	This will reduce the maintenance burden of the AGL community on such a component that does not need to be AGL specific
Effort estimation	3-4 man-weeks
Target AGL release	Royal Ricefish

Feature name	**Improve sample camera application**: rear view backup overlay, object detection labels display, stitching for 360 camera view (multi-camera setup)
Functional area	Camera
Short definition	Add common camera features into the camera app, such as view backup overlay (wheels placement, body alignment, opened doors outline), and stitching for 360 camera view (multi camera setup)
Rationale for AGL	Camera input has been a big gap for AGL for too long. Compute vision and camera video stream rendering are common use-cases in most vehicles today. The AGL sample camera application must demonstrate what is feasible with the Open Source software components readily available today
Effort estimation	6-8 man-weeks
Target AGL release	Royal Ricefish (backup overlay), Super Salmon (360 stitching)

Feature name	**Architecture for multi-process machine vision processing and policies**
Functional area	Machine vision
Short definition	Augment PipeWire’s multimedia IPC mechanism to be able to carry analytics metadata and introduce the appropriate configuration and policies to implement multi-process, multi-camera machine vision systems. Also add object detection labels display on the camera app to demonstrate the functionality
Rationale for AGL	Machine vision is a fundamental technology in automotive today. Use-cases are becoming increasingly complex, while OEMs aim to reduce costs and hardware complexity (wiring, etc.). It is necessary to develop new frameworks to manage machine vision, dividing the workloads and orchestrating the resulting pieces.
Effort estimation	6-8 man-weeks (will depend on the hardware targeted)
Target AGL release	Super Salmon

Feature name	Implement the **stream status negotiation protocol** in PipeWire
Functional area	Audio/Multimedia - SPEC-4791
Short definition	When switching between media applications, users expect the stream of the application that goes into the background to be corked (i.e. silenced/paused). This unfortunately does not work as expected because the application remains unaware of the fact that it is corked, potentially resulting in bad behavior later on. To remedy the situation, an extension protocol in PipeWire needs to be developed to negotiate the streaming and corking state between the application and the session manager
Rationale for AGL	Improve user experience and API capabilities. This is a long-standing limitation of the existing implementation that risks degrading the user experience
Effort estimation	4 man-weeks
Target AGL release	Super Salmon

AUTOMOTIVE GRADE LINUX

Feature name	**Split window functionality** with gRPC & documentation for gRPC API
Functional area	Graphics / SPEC-4833, SPEC-4839
Short definition	Extend gRPC API to allow side-by-side windows being displayed simultaneously and prepare API gRPC documentation
Rationale for AGL	Many OEMs are choosing to let their IVI systems allow displaying multiple windows at the same time. For example, navigation and media player might be displayed together arranged simultneously on the same display output
Effort estimation	2 man-weeks (work is already in progress)
Target AGL release	Quirky Quillback

Feature name	**Head-up Display support**
Functional area	Graphics / SPEC-4910
Short definition	Add support for HUD devices where the output is possibly projected onto a irregular/concave display
Rationale for AGL	As requested by Toyota, HUDs are becoming prevalent in vehicles and AGL should use this opportunity to demonstrate navigation information together with additional IC data (speed/rpm/traffic?)
Effort estimation	Exploratory work; requires access to off-the-shelf HUD device to test. 1 man-week (once the hardware is available)
Target AGL release	Royal Ricefish, Super Salmon

Feature name	**Replace Waltham with VNC**
Functional area	Graphics / SPEC-4689
Short definition	Use VNC (or RDP) protocol for remote streaming in place of Waltham
Rationale for AGL	Waltham was designed early in Wayland’s life to be a remote analogue to Wayland. Since then, the automotive industry has aligned on VNC as the standard for remote content. To improve the client ecosystem and reduce maintenance burden, replace Waltham with VNC as the standard content-streaming solution for AGL.
Effort estimation	3 man-weeks (combined effort with doing something similar to Weston frontend)
Target AGL release	Royal Ricefish

Feature name	**Hybrid configuration for agl-compositor**
Functional area	Graphics / SPEC-4710
Short definition	Allow agl-compositor configuration to be composed from multiple fragments
Rationale for AGL	Align agl-compositor configuration with Yocto recipes. Yocto recipes are composable by layering fragments on top of each other (e.g. AGL base + OEM base + platform base + product), whereas agl-compositor currently requires a single configuration file. Adding support for fragments allows agl-compositor to use the same model for configuration as Yocto.
Effort estimation	4 man-weeks. This effort could be integrated as part of larger upstream task to have dynamic runtime configuration files as being defined upstream
Target AGL release	Super Salmon

Feature name	**New window manager policy**
Functional area	Graphics / SPEC-3436
Short definition	Add Lua and scripting support to agl-compositor
Rationale for AGL	The AGL window management framework was designed around the requirements of RBAC with a static policy. WirePlumber’s Lua scripting model has since become more popular; aligning with this model where the policy is expressed in a scriptable language would give a more coherent story for AGL. Users have also requested the ability to use external scripts to configure window management, especially for demo and development purposes.
Effort estimation	8 man-weeks (possibly a bit more)
Target AGL release	Super Salmon

Feature name	**Compositor render scale**
Functional area	Graphics / SPEC-3281
Short definition	Composition upscaling for high-resolution support
Rationale for AGL	Some platforms can display at a higher resolution than the GPU can render at. The most common example is being able to display some content (e.g. media or static) at 4K, but only being able to render at 1080p. This would allow the compositor to render at 1080p when direct display is not possible but upscale to 4K. In particular, this unblocks 4K displays on Raspberry Pi.
Effort estimation	3 man-weeks
Target AGL release	Royal Ricefish, Super Salmon