

Proposed 2023 Development Roadmap

Current Status

Development in 2022 included:

- Reworking the applaunchd application enumeration and management daemon to use systemd units for applications instead of direct spawning. This enabled the original goal of using systemd sandboxing features for applications. As well, applaunchd was changed to start as unique user separate from root and agl-driver to improve security.
- Adding a gRPC API to applaunchd in addition to the original D-Bus API, with an eye to removing the D-Bus API as soon as all demo applications are converted to using it. At present, only the HTML5 demo applications remain to be converted.
- Adding a gRPC API version of the legacy application framework's radio binding.
- Reworking the legacy application framework's agl-service-hvac binding into a demo VIS client that uses the VSS HVAC signals instead of an AGL specific HVAC API.
- Reworking the legacy application framework's agl-service-audiomixer binding into a demo VIS client that uses the VSS master volume signal instead of an AGL specific mixer API. This is viewed as a stopgap until support for a gRPC version of the previous audiomixer API can be added.
- Upgrading KUKSA.val to 0.2.5 to pick up a more stable version of its gRPC API and upgrade to VSS 3.0.
- Initial implementation of a gRPC version of the agl-compositor's agl-shell API.
- Adding SELinux support, but in non-enforcing mode in the demo builds.

Proposed Development Tasks

Development in 2023 is likely to be evolutionary, with a focus on extending and improving the use of gRPC in the demo stack. Proposed tasks:

(Updated February 7, 2024 with status)

1. Create a repository to hold the shared AGL gRPC API definitions so that server and client implementations can source them from a common location. Some investigation will be required with respect to the most straightforward way of integrating this with the server and client implementations already in hand (e.g. integration with Meson) - **Repo name agl-grpc-protos. Scott getting a better handle on using proto files so it should be ready. Not ready for Quillback.**
2. Finish converting the HTML5 demo homescreen and launcher to using the applaunchd gRPC API and removing the D-Bus API from the applaunchd codebase - **completed for Pike**
3. Finish investigation into a minimal set of advanced systemd sandbox features to turn on by default in the application templates - **completed for Pike. Documentation needs to be addressed.**
4. Update the Flutter and HTML5 demo homescreen applications to use the latest agl-shell v3 features with respect to application start up and surface activation, matching the Qt demo homescreen features. On the Flutter side, this may involve work in the Flutter embedder. - **Pike completed with agl-shell v4 for Web and Flutter.**
5. Investigate converting at least some of the demo homescreen applications to use the new agl-compositor gRPC API instead of the agl-shell wayland protocol. -**completed for Quillback for Flutter, Web apps, and Qt.**
6. Define a gRPC version of the legacy application framework's agl-server-audiomixer API, extend the new agl-server-audiomixer daemon to implement it, and update the Qt demo mixer application to use it - ~~Not started. Probably will get completed for Quillback.~~ **Decided not to use this approach. Eliminated per role volume controls and switched to using VSS. In the new flutter app, the mixer was replaced with more basic audio controls (bass, treble, fader, balance)**
7. Investigate the effort required to define and implement gRPC APIs for the previous agl-service-bluetooth and agl-service-network bindings, with an eye to following through with implementing them if the effort seems reasonable. The end goal of this would be to enable development of a Flutter demo settings application and potentially enhancing the HTML5 demo settings application - **Not started**
8. Investigate solutions for managing authorization tokens and potentially SSL certificates for VIS and/or gRPC. One option that needs to be considered is integrating use of systemd-auth to pass things to daemons and applications. - **Not started**
9. Investigate enabling SELinux enforcing mode in the demo builds. - **Not started**

Related Connectivity EG tasks:

1. Keep following upstream KUKSA.val development and upgrading AGL to the latest releases. - **Pike updated to KUKSA 0.4.0**
2. Convert all existing KUKSA.val VIS WebSocket usage to the gRPC API. - **Started for Pike. Completed in Quillback.**
3. Investigate potentially switching from the original KUKSA.val server to the Rust-based reimplemention (known as the KUKSA.val "databroker"). There are signs that the upstream community is starting to switch focus to the databroker, and the use of Rust is a potentially interesting technology demo to provide to AGL members. - **Complete. Rust mix-in layer completed and databroker is available in Pike**