

AGL Long Term Support

Goals of this effort

- Define what conditions are required to met before AGL declares a long-term support version
- Define what will be maintained as part of an LTS version and what AGL device creators can expect from AGL LTS.
- Estimate the costs of an LTS so the AB is aware of the long-term commitment.

Current situation - Releases are supported for 6 - 9 months with patch updates, typically 4 - 6 updates per release.

Patch updates (AGL 6.0.1 to AGL 6.0.2) may include:

- Minor Yocto version updates (e.g. 2.6.1 to 2.6.2)
- Security fixes
- Fixes to AGL Services and Apps.
- New reference apps
- BSPs for new reference or community boards
- Updated BSPs for existing reference or community boards

Patch updates do not include:

- Major Yocto version updates (e.g., 2.6.2 to 2.7.0 or rocko to sumo)
- Updates to stable AGL APIs

Minor version updates (AGL 6.0.2 to 6.1.0) may include

- Minor Yocto version updates (e.g. 2.6.1 to 2.6.2)
- Updates to stable AGL APIs
- Fixes to AGL Services and Apps
- BSPs for new reference or community boards
- Updated BSPs for existing reference or community boards

Stated desired for declaring a LTS version for 4-5 years of support

- 2-3 years prior to start of production, then 2 - 3 years of support post SOP.
- Will declare an LTS once every two to three years at maximum.
- Advisory Board approval required for an LTS to be declared since it requires a serious long-term funding commitment

Requirements and Assumptions for an LTS version of AGL

- **Code Maintained Upstream of AGL**
 - Probably want to limit which components are maintained. For example bluez may not be included because it is a replaceable component that we think will be used by the device creators.
 - Can only accept upstream components into AGL that are maintained in a reproducible manner (this is an impossible requirement to meet today. Yocto is working on making their builds reproducible but the feeling is that they may be a year away from that goal. In particular we are interested in core-image-weston from Yocto.)
 - Gain agreement from commercial stakeholders, especially SOC vendors, that they will support the AGL LTS.
 - For SOC vendors this will requirement an agreement to maintain their BSP, kernel version, and graphics stack for the life of the AGL LTS version.
 - For OEM and Tier One members this will requirement an agreement that they will contribute fixes back to the AGL LTS effort.
- **Code Maintained by AGL**
 - Source code and recipes will be maintained in such a way that builds are reproducible over the lifetime of the LTS version.
 - Host system will be limited to the container that is shipped with the initial LTS version. AGL will ensure that the container is supported on the latest host system. (i.e. As Fedora or other distributions evolve the AGL container will still be able to be built)
 - AGL demo may not be maintained but we still need to build a binary that runs on the supported platforms so that we can test it.
 - Test image or images that allow AGL to be tested
 - Once we have defined the APIs that are supported in the AGL LTS we can use the recipes to derive the upstream packages that need to be maintained and reproducible.
 - APIs will be stable over the lifetime of the LTS version.
 - APIs will be well specified and documented in a common format
 - Security patches and bug-fixes will be backported from latest AGL version to LTS version.
- **Hardware Considerations**
 - Need to limit the number of hardware platforms that are supported for the LTS version only to the boards that are being put into production by members.
 - Limit to Reference Hardware System Architecture compliant architectures?
 - Instrument cluster reference hardware?
 - AGL devices manufacturers are responsible for maintaining BSPs for their production hardware.

- **Other Considerations**

- Governance of the LTS version. How are decisions made as to when an LTS is declared, what is maintained as part of the LTS version, when does LTS go EOL, approval of releases, etc.
- Patch workflow - Need a strategy for how to accept patches into the LTS that may be different from the development model we use today.
- Cost and funding by the AGL Advisory Board over the lifetime of the LTS.

General LTS in open source industry

- Not done by the project itself beyond two years typically. Usually is taken over by a commercial company
- Yocto provides no long-term support beyond 18 - 24 months. Usual expectation is that a commercial company can be contracted to maintain packages beyond that time-frame.

Next Steps

- Gather and consolidate requirements for review by the SAT and TSC by the June 6 SAT call. Team will consist of Walt, Panasonic, Denso, Toyota, VW, MB.
- Notes from meeting in Berlin on 26 Sep
 - AI: identify base services (What is in core-image-minimal that is available in all profiles.)
 - * lifecycle
 - * apps vs platform (Which apps are maintained as part of the LTS)
 - *
 - * API definition & documentation & Schema
 - * AI: iot.bzh will follow-up on <https://jira.automotivelinux.org/browse/SPEC-1903>
 - * e.g. what about new things like: <https://fuchsia.googlesource.com/docs/+ea2fce2874556205204d3ef70c60e25074dc7ffd/development/languages/fidl/tutorial.md>
 - * Follow-up on SAT
 - * API versions
 - * -> APPFW / Jose
 - * 'core' LTS vs. IVI or Cluster LTS
 - * depends on changes required e.g. by IC