Automotive Grade Linux Lessons Learned

Embedded Linux Conference Europe

August 27, 2025

Walt Miner

Senior Director AGL Community

The Linux Foundation

THE LINUX FOUNDATION

Who is This Guy?

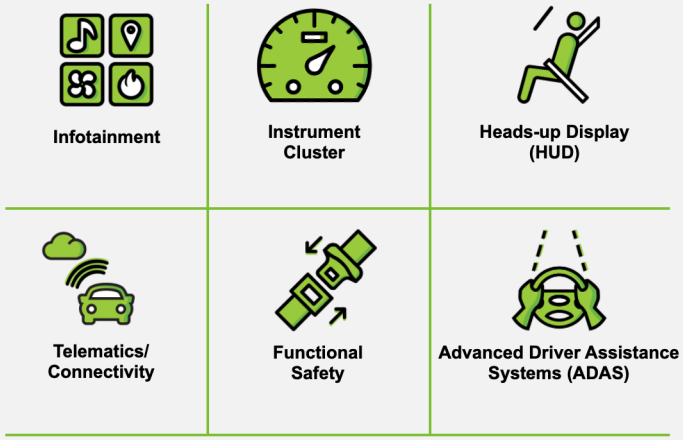
- AGL Development and Community Manager since 2014
- Previously at MontaVista and Mentor Graphics
- Continental AG
- Motorola Mobile Devices
- Motorola Telematics Communications Group
- Defense Electronics at ITT Avionics and Northrop Grumman

Intro to AGL

What is AGL?

AUTOMOTIVE GRADE LINUX

the only organization addressing all software in the car



- Non-profit organization
- Open source Linux-based collaborative project
- Hosted at Linux Foundation
- Collaborating to build the car of the future through *rapid innovation* by uniting the automotive and software industries

OEMs supporting AGL

Millions of vehicles on the road with AGL

AGL first shipped in the 2018 Toyota Camry

Vehicles with AGL:

- Toyota Camry, RAV4
- Toyota Prius Prime
- Lexus RX350, RX450
- Subaru Legacy, Outback
- Mercedes-Benz Vans
- More to come….

Camry image for depiction purposes only, actual vehicle may vary.

Device Profiles

- IC Expert Group has multiple IC configurations including bare metal Linux and container versions
- IVI Expert Group bringing embedded Flutter into vehicle
- Gateway device – data collection and aggregation for telematics, V2C and telemetry using VSS

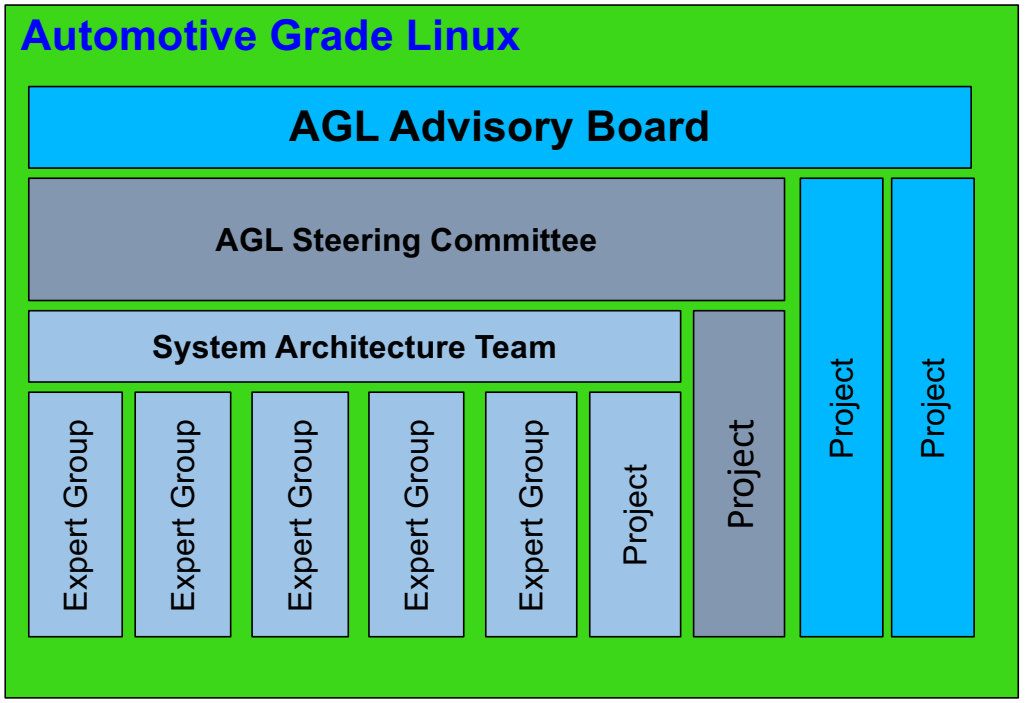
History of AGL

History of AGL

- AGL was initiated by the Linux Foundation in 2012.
- AGL was founded with the support of automotive OEMs including Toyota, Nissan, Hyundai, and Jaguar-Land Rover
- First two years efforts were bifurcated – JLR doing one thing, Japanese members working separately
- 2014 - Dan Cauchy brought on board as executive director
- 2015 – Moved from Tizen and OBS to Yocto as primary upstream. First All-Member Meeting held. IVI System Requirements Specification released. 60% member growth
- 2016 – Introduced Unified Code Base (UCB) with first release: Agile Albacore

AGL Governance

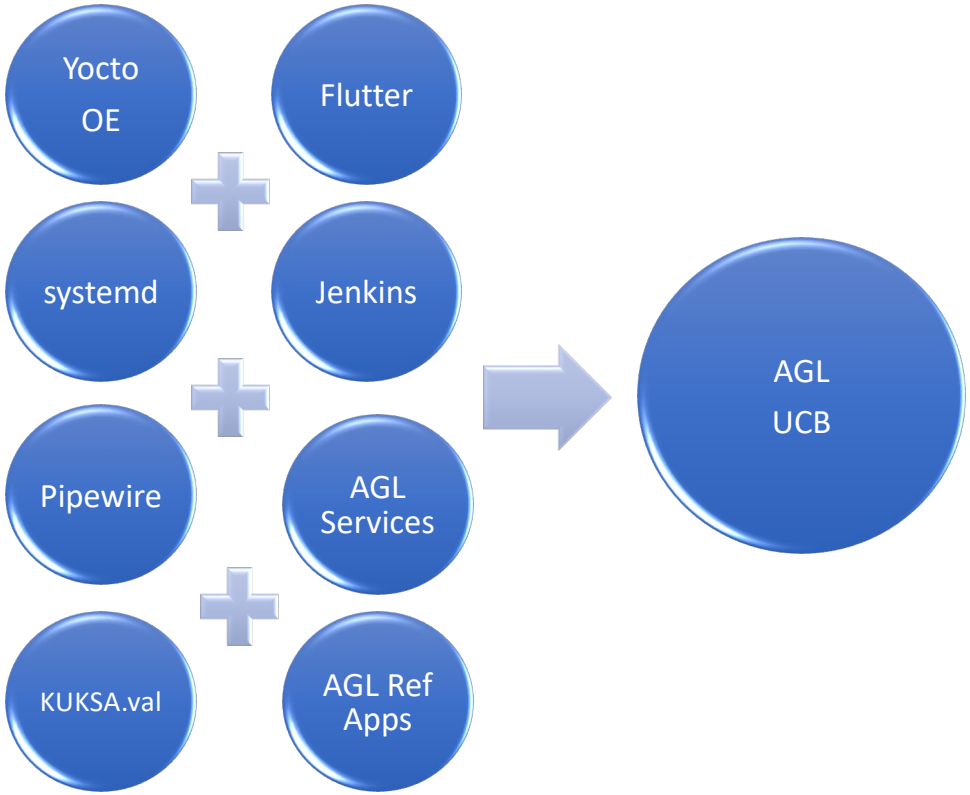
- Bottom up approach, most technical decision making is in the hands of the Expert Groups
- Advisory Board responsible for overall direction, budget, strategy
- Steering Committee oversees AGL technical activities, software feature priorities
- System Architecture Team oversees end-to-end system and consistency
- Expert Groups can run one or more projects
- Projects may be directly managed by SAT, SC, AB



Lesson One – Get on the same page

- Way too many mail lists
  - Cut down to a single developer mail list that covers everything (no patches)
  - Add new mail lists very sparingly (e.g., no EG specific list)
- Initially Focused on writing a spec with no owner or coordination
  - Pushed to get spec done then everyone focused on UCB

AGL Distro “Unified Code Base”



- *Unifying the best of Open Source into a single shared code base for the entire industry!*
- *Reduce fragmentation, focus on innovation and new features!*
- *Completely open source and transparent!*
- *Fully customizable; supported by an ecosystem of Tier Ones and service providers*

AGL Aquarium

Two releases per year, every year since 2016!

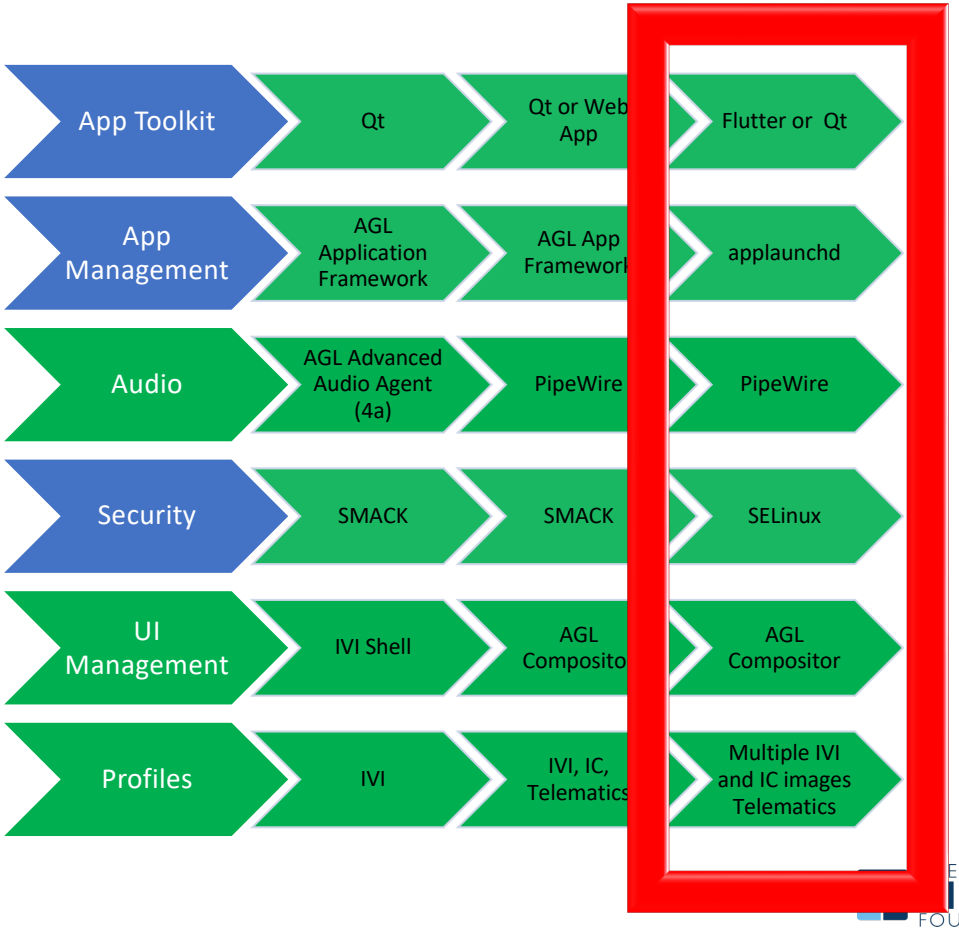
AGL Release	Code Name	Release Date	Fish
AA (1.0)	Agile Albacore	January 2016	
BB (2.0)	Brilliant Blowfish	July 2016	
CC (3.0)	Charming Chinook	January 2017	
DD (4.0)	Daring Dab	July 2017	
EE (5.0)	Electric Eel	January 2018	
FF (6.0)	Funky Flounder	Sept 2018	
GG (7.0)	Grumpy Guppy	March 2019	
HH (8.0)	Happy Halibut	August 2019	
II (9.0)	Itchy Icefish	March 2020	
JJ (10.0)	Jumping Jellyfish	Sept 2020	
KK (11.0) ……..	Kooky Koi	Feb 2021	
PP (16.0)	Prickly Pike	Aug 2023	
QQ (17.0)	Quirky Quillback	Feb 2024	

Lesson 2 – Embrace Change

- UCB is dramatically different in 2025 from what was first released in 2015
- Change should be organic via new or maturing technology

AGL Transformation

- Pre-UCB AGL was Tizen IVI and OBS based
- 2022 saw us transition AGL Reference Apps and Service Binders to newer technologies
- Supplemented Qt reference Apps with Flutter
- Replaced Vehicle Signal Manager with kuksa.val and VSS
- Applaunchd created – now using gRPC!



Make Use of Open Source Experts

- AGL changed from an FTE donation to a cash contribution model in 2015 – easier for companies to contribute cash then developers
- Analyzed potential packages for gaps in automotive use cases (audio, Bluetooth, connectivity, and others)
- Engaged package maintainers and open source experts to close the gap upstream rather than maintain code out of tree

Code First/ Upstream First

- AGL has invested in automotive software components not available anywhere else
- Continually evaluating open source technologies to find best in class for automotive use cases
- AGL has invested (provided developers and upstreamed code) in open source projects such as Pipewire, Yocto, Lava, and others.
- Willing to collaborate with anyone who brings code

History of AGL

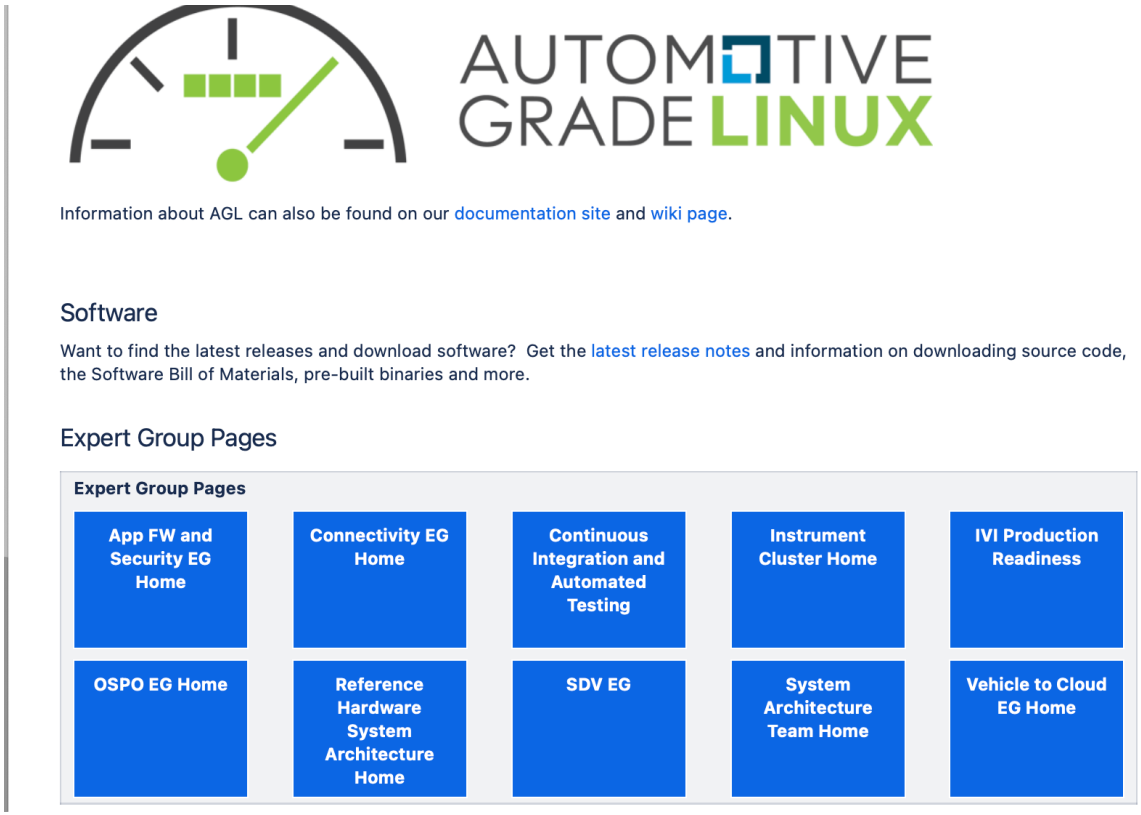
- 2016 – Introduced Unified Code Base (UCB) with first release: Agile Albacore
- 2017 - Toyota Adopts AGL in production vehicles starting with the 2018 Camry.
- 2021 – Virtualization Expert Group formed – focus in expanding VirtIO for automotive use cases
- 2022 – Flutter Embedder developed and maintained by Toyota adopted for user experience
- 2023 – Software Defined Vehicle Expert Group formed
- 2024 – OSPO EG formed
- 2025 – UCB Terrific Tuna will be version 20 – ten years after starting work on the UCB

Lesson 3 – Educate the Industry

- Very few automotive OEMs or their Tier One suppliers had OS contribution processes when we started the project
  - Same is probably true for aerospace companies and their suppliers
- It is still very hard for most automotive companies to contribute code – OSPO EG is an attempt to fix that
- We have had 2+ year waits for promised contributions to materialize

Active Expert Groups

- Find latest activities for all Expert Groups on our Confluence Page



Open Source Program Office (OSPO) EG

- The Linux Foundation has numerous resource available for open source projects and OSPOs
- AGL forming an OSPO EG led by Toyota to address the special issues in the automotive community
  - Encourage participation in OSS communities
  - Contribution process for automotive companies
- Biweekly meetings

Lesson 4 - Engagement

- Engaging outsiders is hard, but worthwhile
- A common open source myth is that there are hundreds of developers waiting to volunteer on your project – you need to find the few that are really there if there is no clear way to monetize the project
- Make multiple channels available for engagement (email, IRC, Discord, Slack, developer calls, meetups)
- Real automotive hardware is very expensive - find lower cost hardware (RPI, Beagle Bone, Rockchip) or emulators to get people started

Lesson 5 – Dev boards are not production boards

- AGL is code first – rely on board vendor for BSPs
- Yocto kernel update cycles very slow
- Need to solve the lag in BSP updates from LTS kernels to automotive hardware

Q&A

THANK YOU