



# Open Source in Every Car with Automotive Grade Linux

*Embedded Linux Conference Europe  
2016*

Walt Miner ( [@VStarWalt](#) )

Community Manager, [AGL](#), [The Linx Foundation](#)

# Who Is This Guy?

---

- Linux Foundation / AGL Dev Manager since 2014
- Prior 15 years a mix of Tier 1 Automotive Suppliers and Mobile Devices
  - MontaVista / Mentor Embedded
  - Continental BU Infotainment and Connectivity
  - Motorola Mobile Devices
  - Motorola Telematics
- Defense Aerospace



# Git Commits BB and CC

Commits	Name	Company
458	Jose Bollo	IoT.BZH
341	NuoHan Qiao	Fujitsu Ten
70	Stephane Desneux	IoT.BZH
64	Ran Cao	Fujitsu Ten
59	Manuel Bachmann	IoT.BZH
58	Jan-Simon Moeller	Linux Foundation
55	Fulip Ar Foll	IoT.BZH
35	Yanhua GU	Fujitsu Ten
34	Christian Gromm	Microchip
27	Yannick Gicquel	IoT.BZH
20	Tadao Tanikawa	Panasonic
15	Leon Anavi	Konsulko
7	Kotaro Hashimoto	Mitsubishi Electric
6	Yuta Doi	Witz
5	Stephen Lawrence	Renesas

Commits	Name	Company
5	Andre Magalhaes	Collabora
4	Phong Tran	Renesas
3	Anton Gerasimov	Advanced Telematics
3	Jens Bockage	Mentor
2	Carlos Alberto Perez	Igalia
2	Tomoki Sekiyama	Hitachi
1	Wataru Natsume	ADIT
1	Philippe Coval	Samsung
1	Tasuku Suzuki	Qt Company
1	Damian Hobson-Garcia	Renesas

\* 15 Jan 2016 – 01 Sep 2016

1260 Total Commits  
18 Companies

# Today's Goals

---

- Educate you on what AGL is all about
- How to access source code and documentation
- Generate interest in developer community to participate in AGL



# Introduction to Automotive Grade Linux

*Or “AGL is changing the way automotive manufacturers build and manage software”*

# Automotive Grade Linux

---

*Collaborating to build the car of the future through rapid innovation*

[\*http://AutomotiveLinux.org\*](http://AutomotiveLinux.org)

***“If Linux is in the car, we want it all to be based on AGL, no matter what the function.”***

**Dan Cauchy, Linux Foundation,  
Automotive Linux Summit 2015**

***AGL is the only organization that plans to address IVI,  
instrument cluster, telematics, HUD, control systems, ADAS.***

# Charter: AGL is “Code First”

---

- AGL is a Linux Foundation Open Collaborative Project
- Leverage Linux and Open Source technologies
- Build standardized platform(s) and app framework for the entire industry and for all functions in the vehicle
- Develop ~80% of the starting point for a production project
- AGL is a “Code First” organization
- Work with upstream projects
- Educate the industry in open source collaboration and best practices



# *AGL is changing the future of driving*

---



*AGL has 8 major OEM supporters*

# AGL Members – Over 80 companies!

## Platinum Members

**DENSO**



**Panasonic**

**RENESAS**

**TOYOTA**

## Gold Members

**HONDA**  
The Power of Dreams

**NTT DATA**  
NTT DATA MSE Corporation

## Silver Members

**AISIN AW CO.,LTD.**

**Codethink**

**Continental**

**FUJITSU TEN**

**irdeto**

**MITSUBISHI ELECTRIC**



**Pioneer**

**QUALCOMM**

**WIND**

## Bronze Members

**Advanced Driver Information Technology**

**Advanced Telematic SYSTEMS**

**ALPS**

**AutoVISTA**

**bright box**

**中国移动 China Mobile**

**cinemo**

**COLLABORA**

**ENEA**

**ETRI**  
Electronics and Telecommunications Research Institute

**Eureka, Inc.**

**Ford**

**FORGEROCK**

**FUJITSU**

**GlobalLogic**  
Leaders in Software R&D Services

**HARMAN**

**HI CORP.**

**HITACHI**  
Inspire the Next

**HYUNDAI MOBIS**

**igalia**

**intel**

**JAGUAR LAND ROVER**

**JVC KENWOOD**  
creates excitement & pace of mind

**Konsulko Group**

**Linaro**

**mccloudware**

**MEDIA TEK**

**Mentor Automotive**

**MICROCHIP**

**micware**

**MITSUBISHI MOTORS**

**KOMPTAC**

**MOVIMENTO**

**NEC**

**NVIDIA**

**NXP**

**OBIGO**

**OPENSYNERGY**

**ORACLE**

**SUBARU**

**Suntec**

**SYMBIO**

**TEXAS INSTRUMENTS**

**Qt** The Qt Company

**TOSHIBA**

**Ubiquitous**

**UIEVOLUTION**

**VeriSilicon**

**Virtual Open Systems**

**WITZ**

# AGL is changing the industry

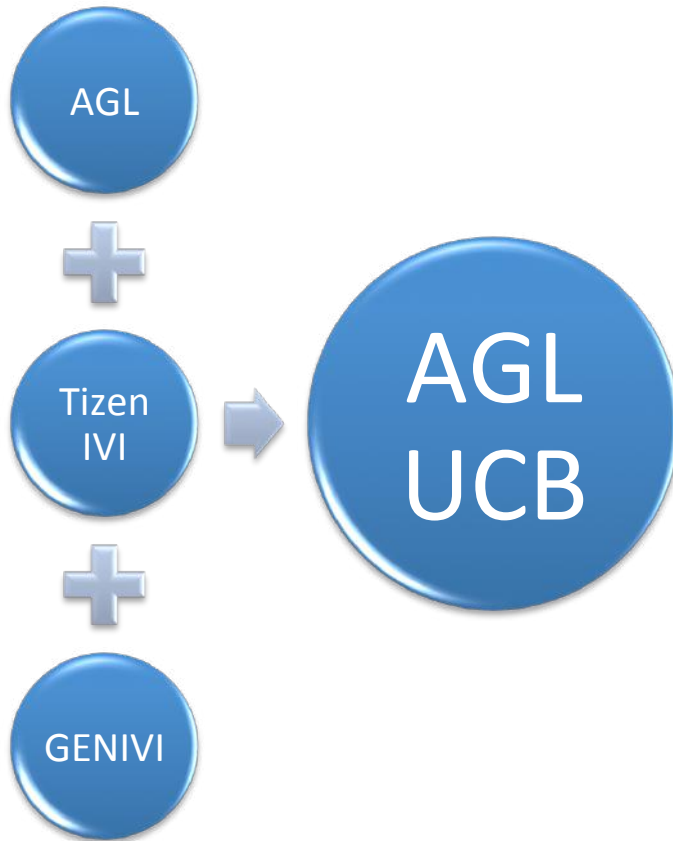
---

- We are creating a new software development methodology for automotive using open source
- We are changing the way automotive manufacturers build, manage, and treat software
- We are building an AGL ecosystem and supply chain, all using the same code base
- AGL will change:
  - The way consumers interact with the vehicle
  - The way vehicles interact with other vehicles
  - The way vehicles interact with the cloud



# AGL Roadmap

# AGL Distro “Unified Code Base”



- *First Release announced at CES Las Vegas in January*
- *Unifying the best of AGL, Tizen IVI and GENIVI into a single code base for the entire industry!*
- *Reduce fragmentation, focus on innovation and new features!*
- *Yocto/Poky based with AGL specific layers*

# Thanks for all the fish...

---

- AGL Releases:

AA – Agile Albacore – Jan 2016



BB – Brilliant Blowfish – July 2016



CC – Charming Chinook – Jan 2017



DD – Daring Dab – July 2017

# CES AGL Demo Video



The screenshot shows the Automotive Grade Linux website. At the top left is the logo, which consists of a speedometer icon with a green needle and the text "AUTOMOTIVE GRADE LINUX". To the right of the logo is a search bar and a navigation menu with links for "About", "Community", "Software", and "News". Below the navigation menu is a news article titled "AGL Shows Demo at CES 2016". The article has a "News" tag and a short paragraph of text. Below the text is a video player with the title "Automotive Grade Linux (AGL) Software Demo @ CES". The video player shows a hand interacting with a tablet displaying a software interface. Below the video player is a section titled "About Automotive Grade Linux (AGL)" with a paragraph of text and a link to the website.

**AUTOMOTIVE GRADE LINUX**

About Community Software News

## AGL Shows Demo at CES 2016

News

During the 2016 Consumer Electronics Show in Las Vegas, the Automotive Grade Linux project showed a demonstration of the [new Unified Code Base distribution](#). Here is a video of the demonstrator in action as shown by Dan Cauchy, General Manager of Automotive at the Linux Foundation.

Automotive Grade Linux (AGL) Software Demo @ CES

**About Automotive Grade Linux (AGL)**

Automotive Grade Linux is a collaborative open source project that aims to accelerate the development and adoption of a fully open software stack for the connected car. Leveraging the power and strength of Linux at its core, AGL is uniting automakers and technology companies to develop a common platform that offers OEMs complete control of the user experience so the industry can rapidly innovate where it counts. The AGL platform is available to all, and anyone can participate in its development. Learn more: <http://automotivelinux.org/>.

Automotive Grade Linux is a Collaborative Project at The Linux Foundation. Linux Foundation Collaborative Projects are independently funded software projects that harness the power of collaborative development to fuel innovation across industries and ecosystems [www.linuxfoundation.org](http://www.linuxfoundation.org)

- We posted a video of the CES AGL UCB Demo online:
  - <https://www.automotivelinux.org/news/news/2016/01/agl-shows-demo-ces-2016>
- ALS Video Coming Soon!

# Brilliant Blowfish



- Released July 15, 2016 (Version 2.0.0)
- Version 2.0.2 released October 7
- Upgraded to Yocto 2.0
- Additional BSPs
- IVI Audio Manager
- IVI Layer Manager
- Automated Test Improvements



# Brilliant Blowfish



- Reference BSPs – Fully supported by manufacturer, CI, etc.
  - ✓ Renesas R-Car 2 - Porter board – **Full ALS demo**
  - ✓ QEMU – demo code available – not shown at ALS
- Community BSP – Best effort by AGL with minimal support
  - ✧ NXP – i.MX6 – SABRE – **ALS demo available**
  - ✓ NXP - i.MX6x – Wandboard – issues with graphics drivers
  - ✓ Intel - Minnowboard Max - demo code available – not shown at ALS
  - ✓ TI - Jacinto 6 - Vayu board – **ALS demo available**
  - ✧ QCOM – Dragonboard 610-c – no demo available
  - ✧ Raspberry PI – no demo available

# Reference or Community BSP?

---

- Reference board
  - BSP available as part of AGL Core Distribution
  - BSP maintained by board manufacturer
  - Documentation and Kick-start guide available for downloading and building code and running the AGL demo code.
  - SDK Released and maintained
  - Manufacturer provides at least two boards for AGL Continuous Integration and Automated Test (CIAT) infrastructure
  - Continuous Integration
    - Daily snapshot builds available from AGL Jenkins
  - Test and QA
    - Sponsoring company sets up test nodes in Lava
    - Full AGL CIAT test suite is run
    - Test results reported.
    - Expect >90% pass

# Reference or Community BSP?

---

- Community board
  - Hobbyist boards that are not automotive specific
  - Older automotive specific boards that are no longer sponsored / maintained by the manufacturer
  - Best effort by the community
  - AGL will have “featured” community BSP(s) as proposed by the community and designated by the SAT
- See [https://wiki.automotivelinux.org/agl-distro#supported\\_hardware](https://wiki.automotivelinux.org/agl-distro#supported_hardware) for list of boards

# Charming Chinook



- Target December 15, 2016
- SDK for AGL App Developers
- Reference AGL Apps
- AGL Compositor
- AGL Home Screen Reference App in Qt and HTML5
- Device Profiles for Telematics, IC, ADAS
- IP Network Manager with WiFi and LTE

# AGL Documentation

---

- MD with web publishing for all AGL documentation
- Use git/ gerrit for version control and reviews
- AGL Security Spec
- Move Requirements Spec from DOORS NG

# SDK for App Developers

---

- Available for reference boards with published images that include graphics drivers
- Enables rapid AGL application development ( download SDK and write “Hello World” in less than 1 hour)
- Support for Qt and HTML5
- IDE with debugging supported (optional for CC)
- Documentation
- No Yocto knowledge is needed or assumed for SDK users

# AGL Compositor

---

- Currently using Weston and IVI shell as the compositor.
  - Does not meet automotive requirements.
  - Modified from desktop environment
  - No good alternative for automotive is available as open source
- Other option considered was to use Qt compositor, but this is not desirable in the AGL core distribution
- Ideally a member company would donate a solution we can build upon

# IP Network Manager with WiFi and LTE

---

- ConnMan made it into BB
- UI and device management is needed for CC
- Reference Application(s)



# Daring Dab

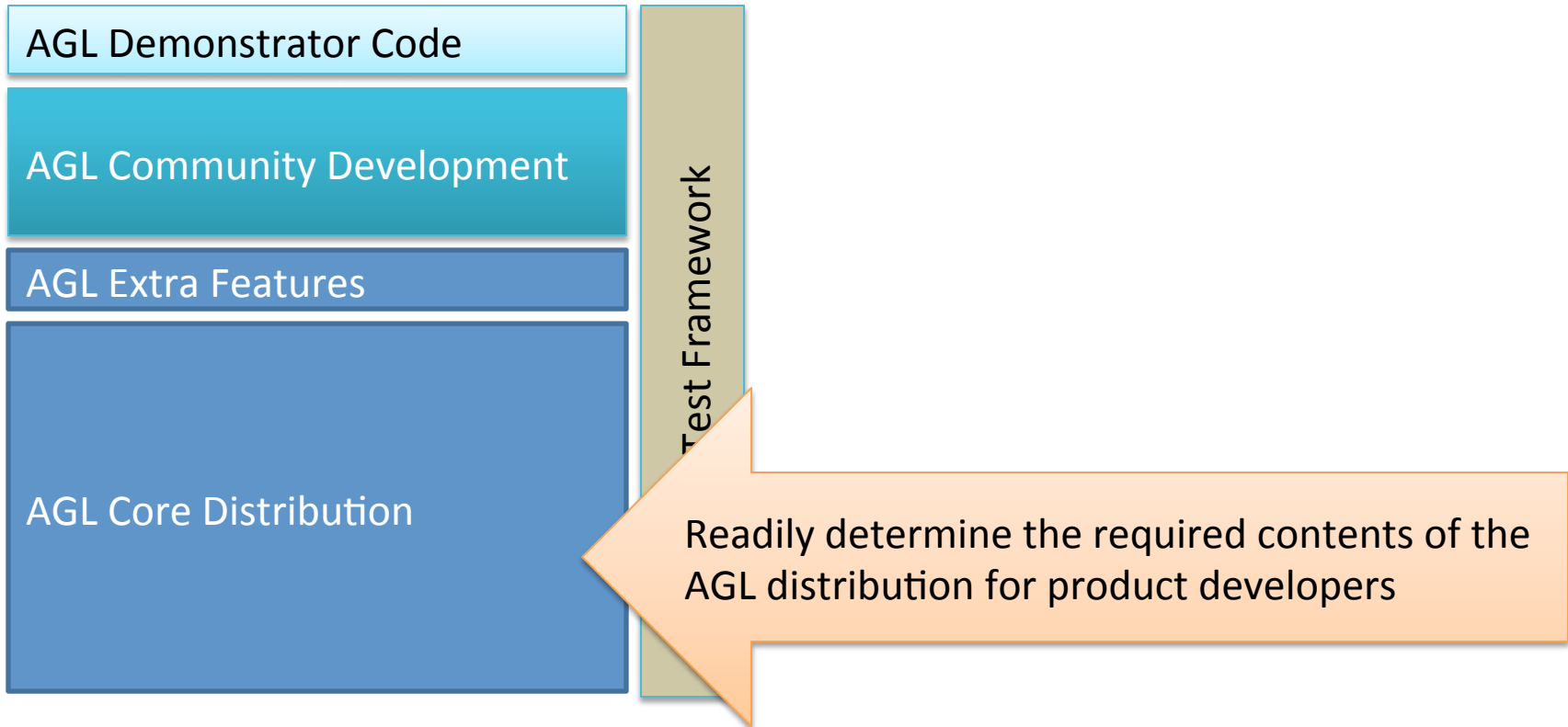


- Smart Device Link
- Navigation API
- Speech Services API
- Browser Engine API

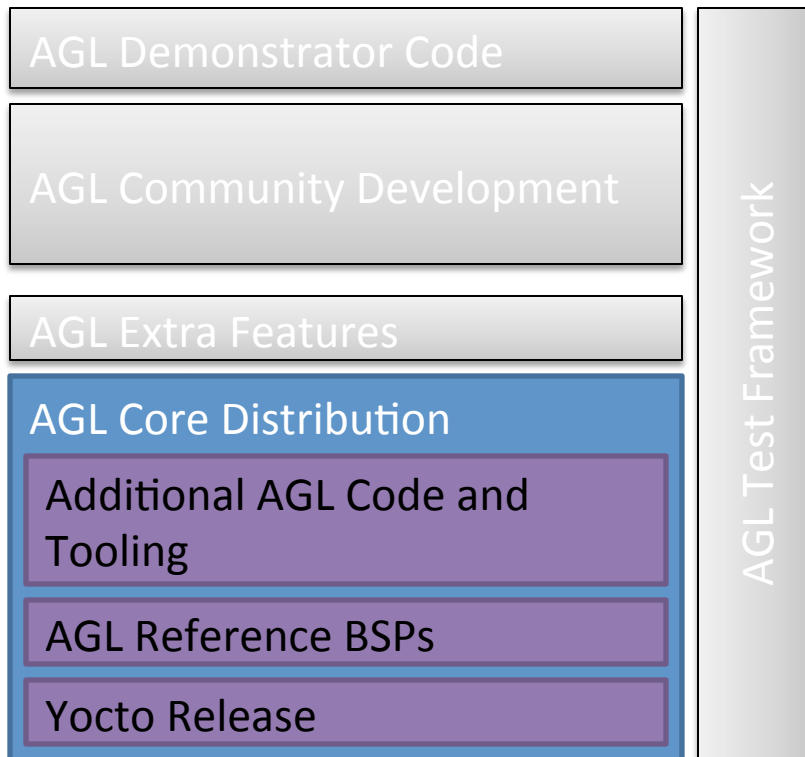
# CODE STRUCTURE

---

# Software Configuration Requirements

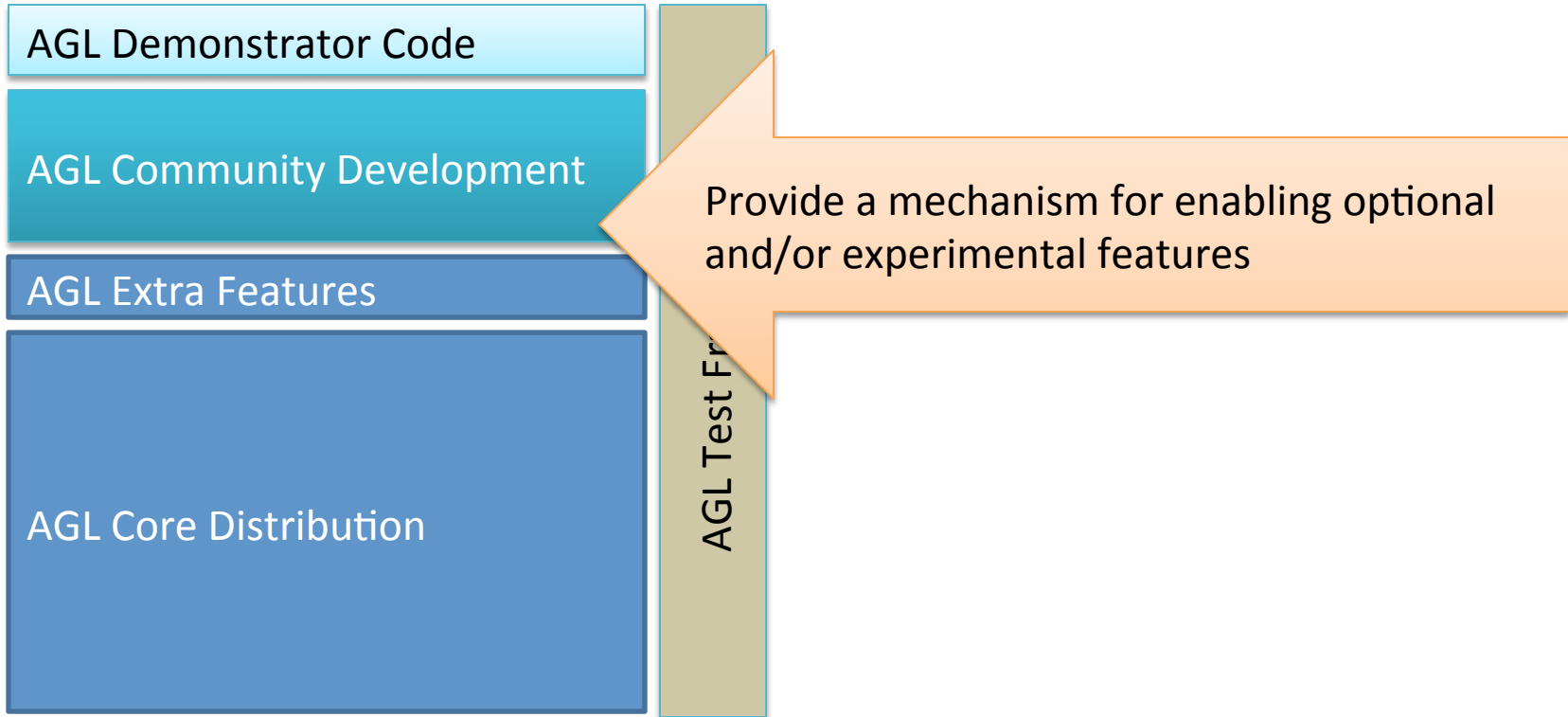


# AGL Core Distribution

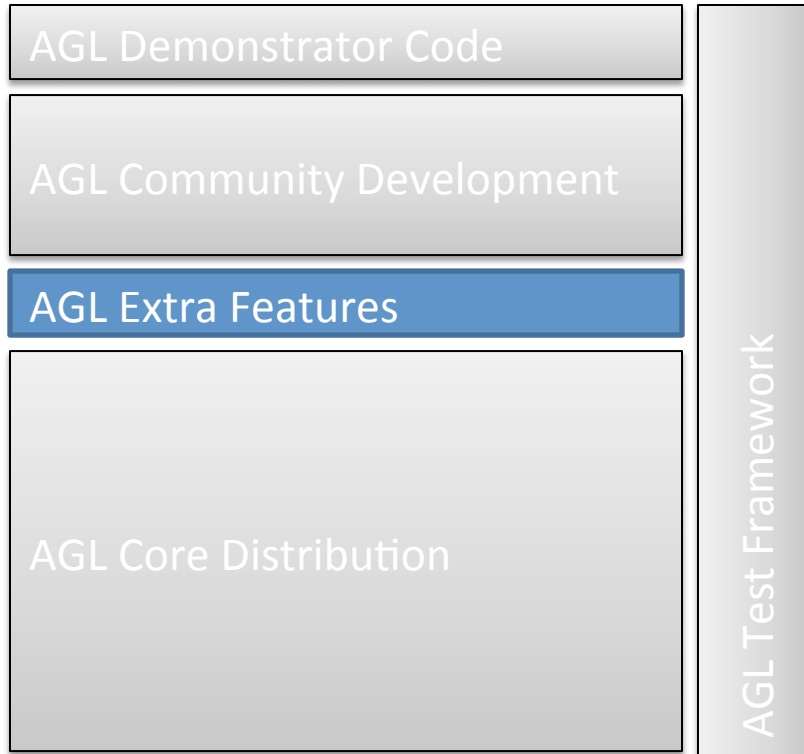


- Stable Yocto release
- Reference BSPs fully supported by the board manufacturer or chip vendor
- Documentation and tooling for building and deploying reference BSPs
- Tooling to allow selection of optional features in the core build
- Test results provided using AGL Test Framework
- Fully supported with updates for at least 6 months
- Defined by Yocto layer – meta-agl

# Software Configuration Requirements

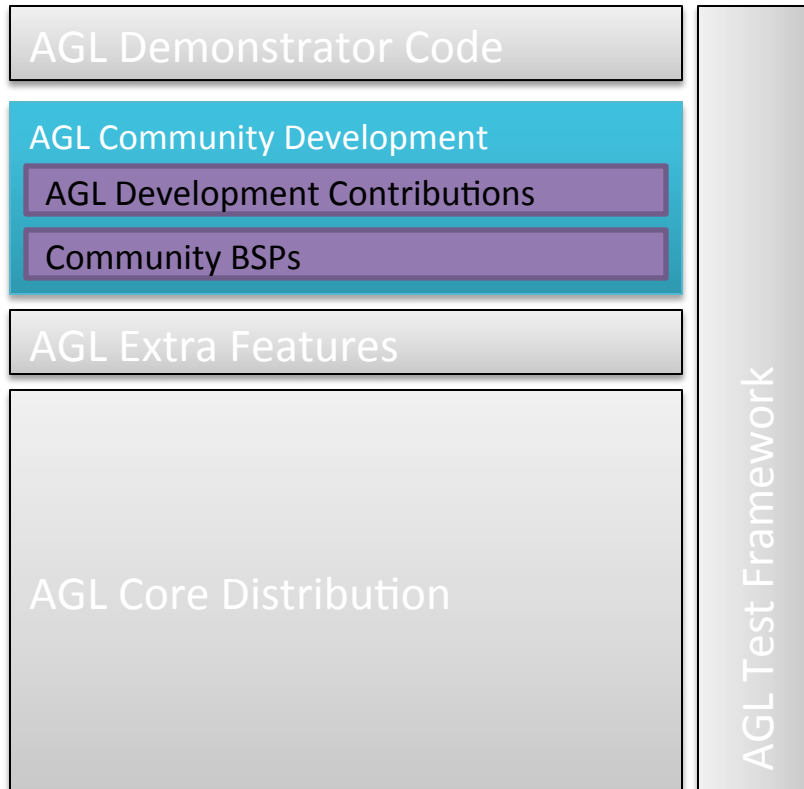


# AGL Extra Features



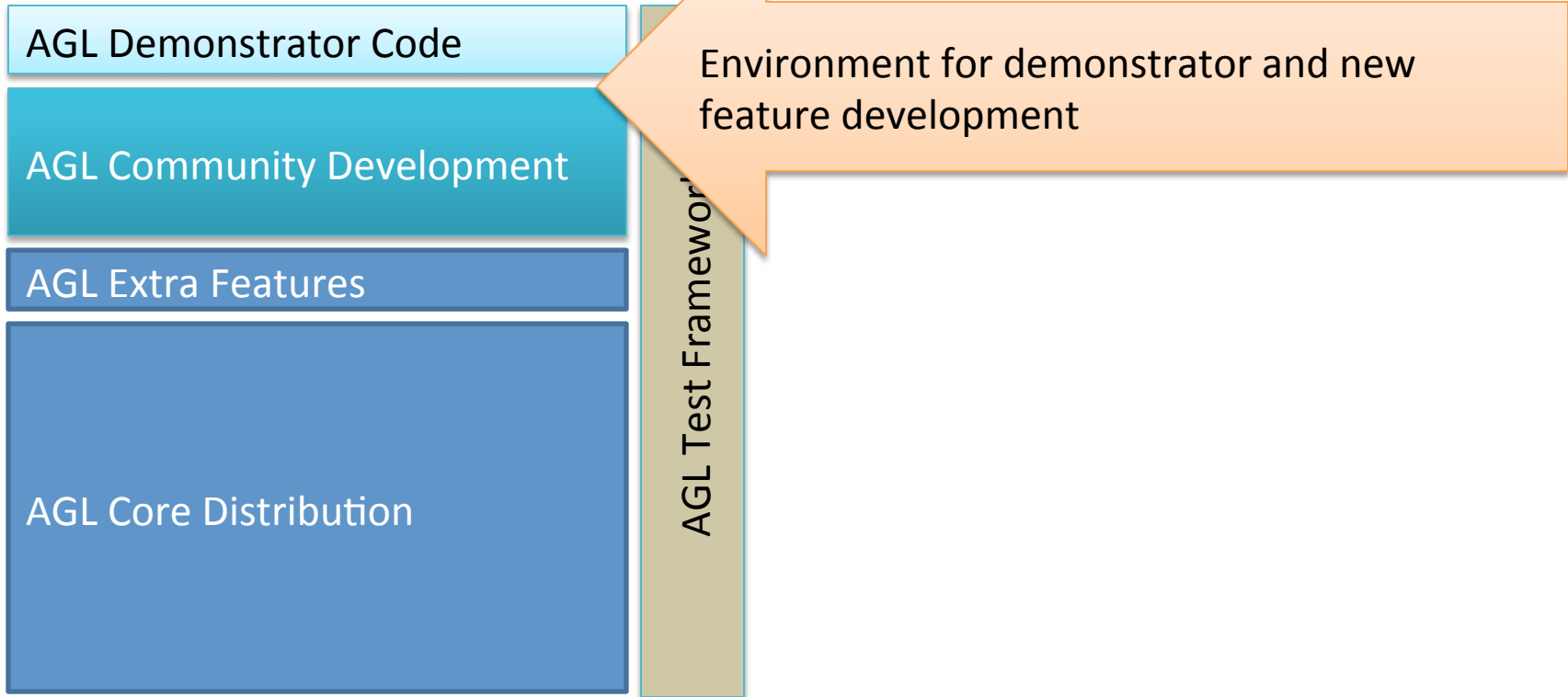
- Builds on AGL Core Distribution
- Features are fully tested and supported as part of AGL release
- AGL environment set up provides extra features that may be enabled by device creators
- Device profiles (e.g., Telematics, ADAS) will be provided in AGL Extra Features
- Yocto layer – meta-agl-extra

# AGL Community Development



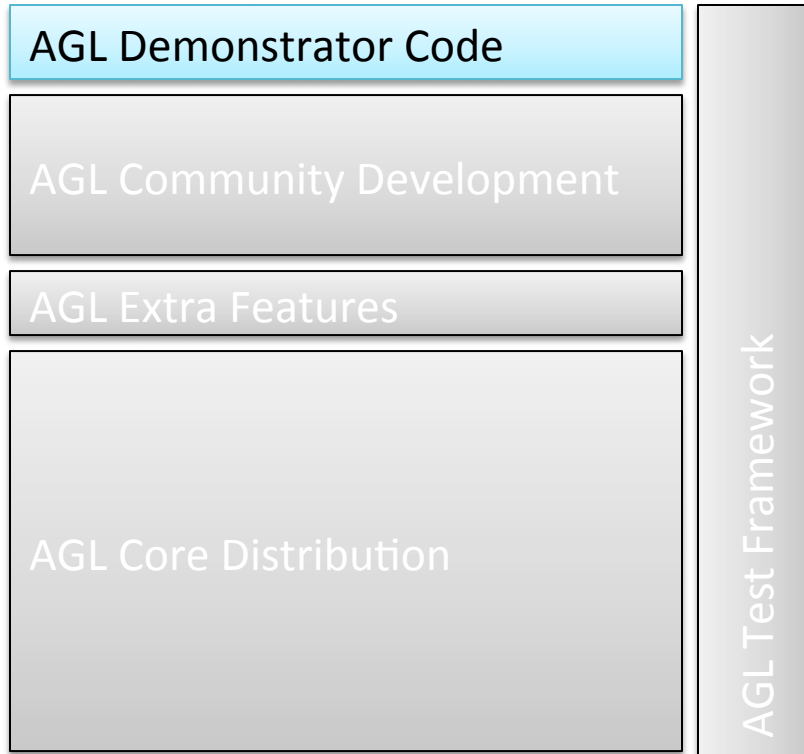
- Place for developing code that may eventually make it into AGL Core or Extra Features
- Snap shot builds for experimental features to facilitate collaboration
- Community BSPs without official support
- Snap shot builds may be provided for Community BSPs
- No formal QA – basically whatever the community can provide
- Defined by Yocto layer – meta-agl-devel

# Software Configuration Requirements



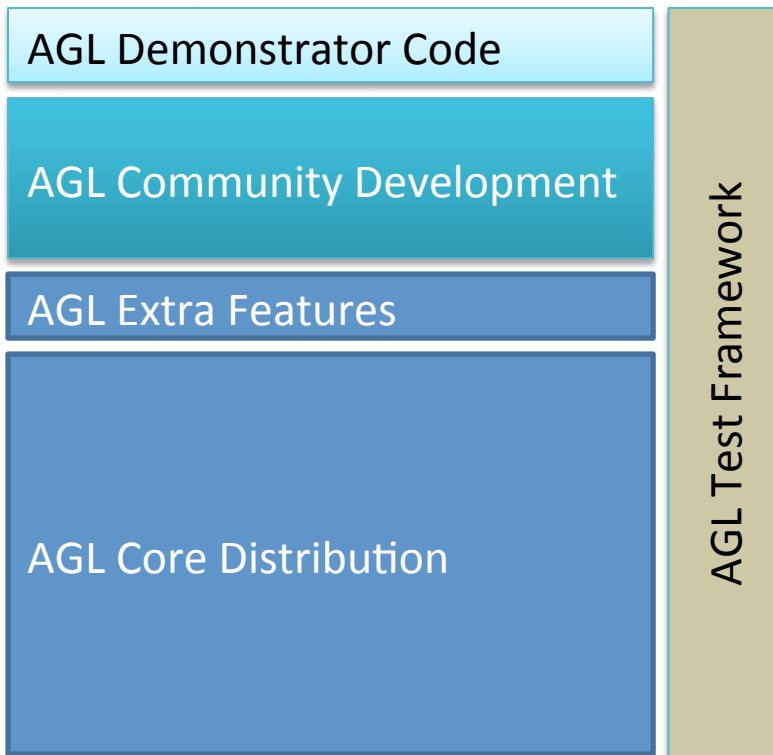


# AGL Demonstrator Code



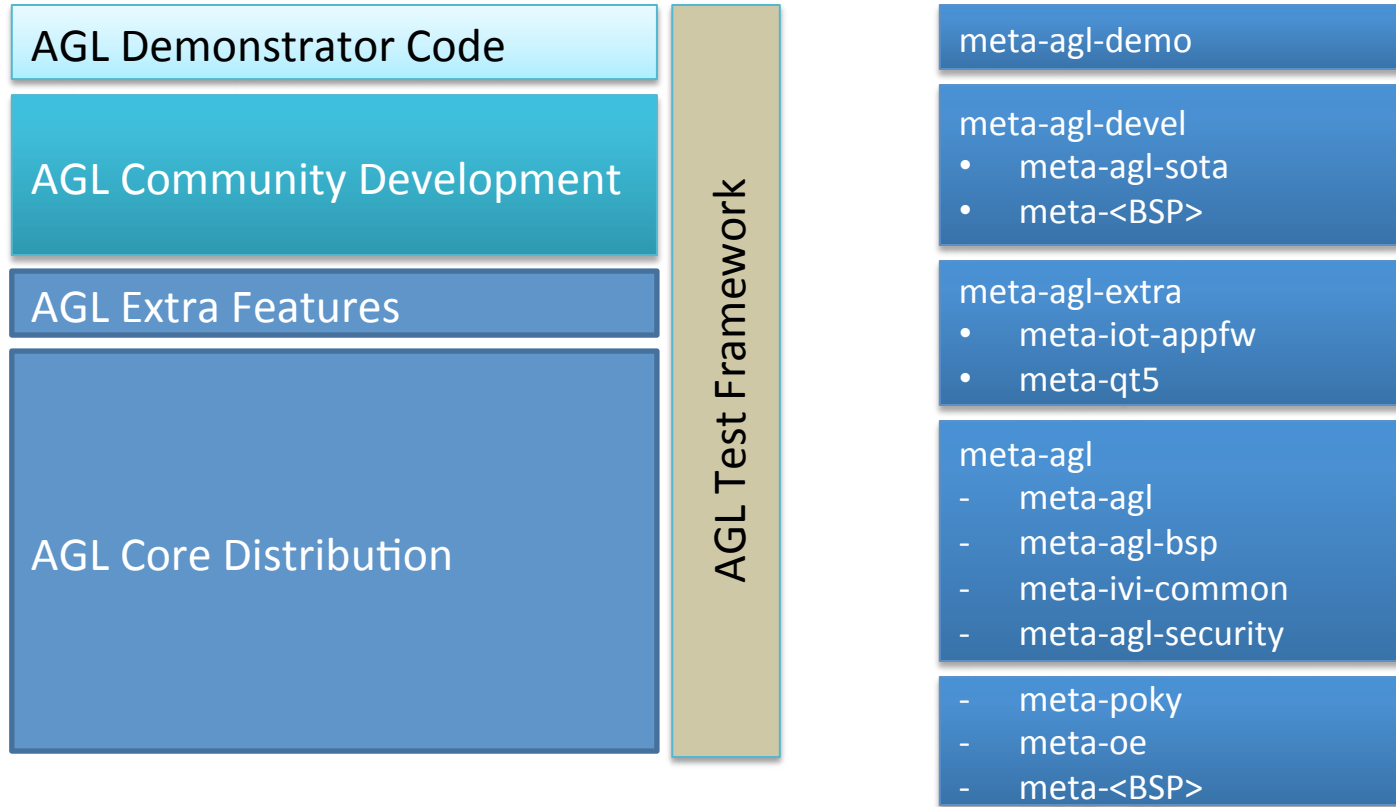
- Code developed to demonstrate specific features and/or releases of AGL
- CES 2016
- Automotive Linux Summit 2016
- Intended for “one shot” development
- Provided “as-is”
- Yocto layer – meta-agl-demo

# Release Management



- Twice per year release of AGL Distribution includes
  - AGL Core Distribution and Extra Features
  - All code and tooling with test results
  - Full test results for reference BSPs
  - As-Is demo code, Community Developed features, and BSPs
- Support biannual releases with code fixes for six months
- Long term support (2+ years) for selected releases
- Daily snapshot builds for specific configurations
- Pre-release candidates to allow developer collaboration and coordinated testing

# AGL Yocto Layers



# Get The Code

---

- Pre-built binaries and source tar balls available
  - <https://www.automotivelinux.org/software/download>
- Latest Source Code and Build Instructions
  - <https://wiki.automotivelinux.org/agl-distro/source-code>

# Build Options

---

- Once you have the repos set up use  
\$ source meta-agl/scripts/aglsetup.sh -h
- To determine available boards and build options
- Example – Build QEMU AGL Demo  
\$ source meta-agl/scripts/aglsetup.sh -m  
qemux86-64 agl-demo agl-netboot agl-appfw-  
smack

\$ bitbake agl-demo-platform

# Summary

Source Location	Layer	QA Performed	Release Support	Daily Build and CI Builds
Staging (or remote)	Meta-agl-demo	N	N	Y
Staging (or remote)	Meta-agl-devel	N	N	Y
Src (or remote)	Meta-agl-extra	Y	Y	Y
Src (or remote)	Meta-agl	Y	Y	Y

- Examples
  - ALS and CES Demo apps belong in meta-agl-demo
  - Meta-agl-sota belongs in meta-agl-devel
  - Meta-iot-appfw belongs in meta-agl-extra



# Getting Involved with AGL

# Reference Apps for AGL App FW

---

- Show your app in the official AGL demo at CES
- Call for participation for CES 2017

<https://wiki.automotivelinux.org/agl-distro/ces-2017-demo>

- Selection by Steering Committee by 15 Oct



# Reference HW for CES

---

- Expanding reference hardware platforms for CES demos.
- Call for participation for CES 2017
- Check mail list this week for qualifications necessary to participate
- Selection by Steering Committee by 15 Oct

# Getting Involved

---

- Most subsystems in need of developers and maintainers particularly user space
- Application developers needed
- Weekly developer calls on Tuesdays at 13:00 UTC
  - Info at <https://wiki.automotivelinux.org/dev-call-info>
- Check Jira for open issues and tasks that need to be done
  - <https://jira.automotivelinux.org/>

# Contribution Process

---

- Code development process is documented
  - <https://wiki.automotivelinux.org/agl-distro/contributing>
- Process continues to evolve as we mature

# Git and Gerrit

---

- AGL uses git for version control and gerrit for code reviews
- Code and patch submissions are via gerrit and use the gerrit review and merge process
- These can be found at
  - <https://gerrit.automotivelinux.org>
  - <https://git.automotivelinux.org>

# Git and Gerrit

---

- The AGL gerrit setup is divided into three main repository groups
  - AGL - contains the recipes for building the AGL distribution
  - src - contains source code repositories where AGL is the upstream. This code is officially part of the AGL distribution
  - staging - contains source code repositories where AGL developers can work on new features or components that can eventually be included in the AGL distribution
- Complete descriptions and links to gerrit can be found at <https://wiki.automotivelinux.org/agl-distro/contributing>

# Continuous Integration

---

- Using Jenkins for Continuous Integration
- Patches
  - All changes submitted to gerrit are built immediately by Jenkins.
  - Successful build gives +1 to new code in Gerrit
  - Build failure -1 in gerrit
- Daily Snapshot builds
  - Available for reference BSPs
  - May add community BSPs later this year
  - <https://download.automotivelinux.org/AGL/snapshots/master/>

# Automated Test

---

- Fuego (LTSI Jenkins Test Automation) being integrated into process
- More information
  - <https://wiki.automotivelinux.org/agl-testframework>



# AGL Expert Groups

*Or “You don’t have to be an expert to work in an Expert Group”*



# Team Overview

---

- System Architecture Team and Expert Groups working on new feature requirements and architecture
- Component teams own software not specifically assigned to an EG
  - Common Libraries and OS
  - Kernel and Device Drivers
- Each team has a dedicated wiki page
  - Link to roadmap and project backlog from wiki

# App Framework and Security EG

---

- *Application lifecycle (install, run, remove, applications)*
- *SDK and application developer experience both in security and APIs*
- *Security framework (SELinux, SMACK, AppArmor, etc.), policies, and strategy for the distribution*
- Network and vehicle firewalls in conjunction with the Connectivity EG
- Software Update and secure update
- Diagnostic log and trace
- Secure boot

<https://wiki.automotivelinux.org/eg-app-fw>

# UI and Graphics EG

---

- *AGL Compositor, Layer Manager, and GPU interface*
- *Multimedia video manager (including multi-display and display sharing) and audio manager, and media manager/player.*
- Browser Engine
- Speech Recognition
- Navigation

<https://wiki.automotivelinux.org/eg-ui-graphics>

# Connectivity EG

---

- *Vehicle Connectivity (CAN, MOST, LIN, AMB)*
- *Network and vehicle firewalls*
- Cloud Connectivity (Iotivity)
- Connected Car
- Bluetooth, Wifi, NFC
- Smart Device Link (SDL)
- Remote Vehicle Interactions (RVI)

<https://wiki.automotivelinux.org/eg-connectivity>

# CI and Automated Test EG

---

- *Build and smoke test of Gerrit submissions on all hardware*
- *Daily snapshot build and testing*
- *Device tests on real hardware*
- *Test environments such as JTA and Lava*
- Test suites such as LTP
- UI testing (OpenQA)

<https://wiki.automotivelinux.org/eg-ciat>



## Q&A

*Tweet questions to @VStarWalt*



**THANK YOU**