Automotive in Open Source: Yocto Project Opens the Doors to Connected Cars

- Cars have been connected for at least a decade now, but most drivers and consumers are looking for more innovation. Unlike other consumer industry products, the car industry tries to invent nearly everything on a brand and model basis, often stifling software re-use and resulting innovation for developers to contribute. Consumer demand and open standards are quickly changing this. Recent development of common Open Source software and programming interfaces (APIs) will enable cars that benefit from innovation in adjacent industries such as smartphones and the internet of things.

- This talk will highlight the breakthrough in open software platforms now available for production use by automakers and suppliers. Projects pioneered by GENIVI Alliance and Linux Foundation AGL have been united by using Yocto Project. Prior build systems led to fragmentation between suppliers which is exactly what the auto industry is trying to avoid.

Research - 12 Expert Interviews (invited 20)
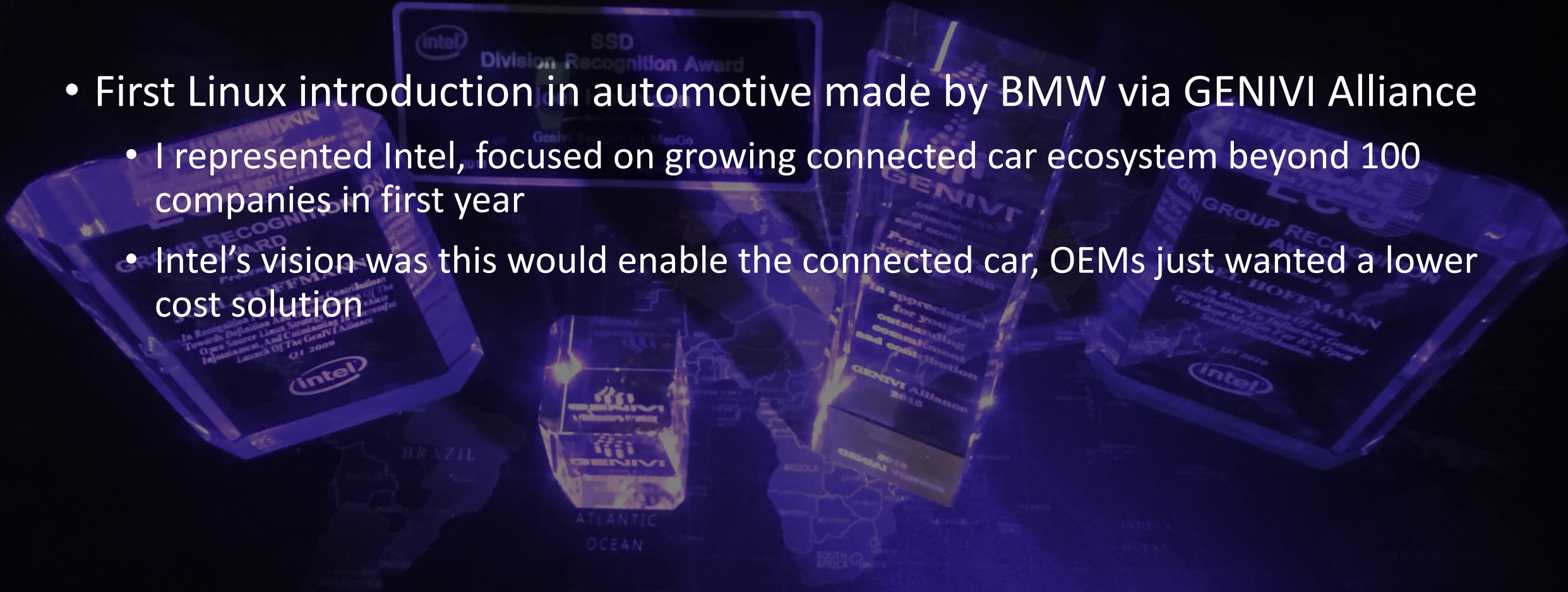
Research - 12 Expert Interviews (invited 20)

- ⭐ Volvo (software strategist)
- ⭐ GENIVI (exec director)
- ⭐ GENIVI (program manager)
- ⭐ Intel (community manager)
- ⭐ Intel (former strategic planner)
- ⭐ Intel (software architect)

- ⭐ JLR (systems architect)
- ⭐ Luxoft (managing director)
- ⭐ Renesas (senior manager)
- ⭐ Renesas (principle engineer)
- ⭐ CodeThink (chief executive)
- ⭐ CodeThink (principle consultant)

My Background Relevant to Automotive

My Background Relevant to Automotive

- First Linux introduction in automotive made by BMW via GENIVI Alliance
  - I represented Intel, focused on growing connected car ecosystem beyond 100 companies in first year
  - Intel's vision was this would enable the connected car, OEMs just wanted a lower cost solution

My Background Relevant to Automotive

- First Linux introduction in automotive made by BMW via GENIVI Alliance
  - I represented Intel, focused on growing connected car ecosystem beyond 100 companies in first year
  - Intel's vision was this would enable the connected car, OEMs just wanted a lower cost solution
- Alliance to scale was initiated in 2009 by Intel, BMW, Wind River and 5 others (GM and PSA were OEM leads)

My Background Relevant to Automotive

- First Linux introduction in automotive made by BMW via GENIVI Alliance
  - I represented Intel, focused on growing connected car ecosystem beyond 100 companies in first year
  - Intel's vision was this would enable the connected car, OEMs just wanted a lower cost solution
- Alliance to scale was initiated in 2009 by Intel, BMW, Wind River and 5 others (GM and PSA were OEM leads)
- Effort to align with broader ecosystem was a goal, thus Moblin > MeeGo > Tizen IVI projects were incorporated as reference implementations

My Background Relevant to Automotive

- First Linux introduction in automotive made by BMW via GENIVI Alliance
  - I represented Intel, focused on growing connected car ecosystem beyond 100 companies in first year
  - Intel's vision was this would enable the connected car, OEMs just wanted a lower cost solution
- Alliance to scale was initiated in 2009 by Intel, BMW, Wind River and 5 others (GM and PSA were OEM leads)
- Effort to align with broader ecosystem was a goal, thus Moblin > MeeGo > Tizen IVI projects were incorporated as reference implementations
- These early IVI projects used OBS, not Yocto Project, I watched each of them fade away in favor of "BitBake" recipes of Open Embedded
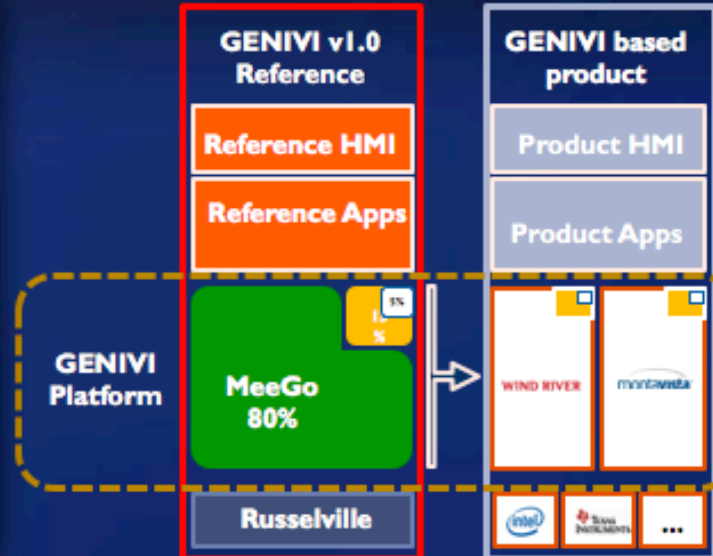
4

GENIVI: Driving Industry Evolution Within Automotive

Based on a Community; No single supplier Dependency

Adopt – Adapt – Create for MW » Transform into Product

MeeGo Software Platform Chosen by the GENIVI Alliance

By *Linux_Foundation*
Created 07/23/2010 - 07:04

MeeGo Software Platform Chosen by the GENIVI Alliance

SAN FRANCISCO, July 23, 2010 – The Linux Foundation, the nonprofit organization dedicated to accelerating the growth of Linux, today announced the GENIVI Alliance [1] has chosen MeeGo [2] as the basis of its next reference release for In-Vehicle Infotainment (IVI).

MeeGo is an open source platform hosted by The Linux Foundation that brings together Intel Corporation and Nokia's previous projects and is designed for computing device types including smartphones, netbooks, tablets, mediaphones, connected TVs and IVI systems. Adoption by this major automotive alliance is a testament to the cross-device, cross-architecture advantages of the MeeGo platform. GENIVI is a nonprofit industry alliance with founding members BMW Group, Delphi, GM, Intel, Magneti-Marelli, PSA, Visteon and Wind River.

IVI is a rapidly growing and evolving field that encompasses the digital applications that can be used by all occupants of a vehicle, including navigation, entertainment, location-based services, and connectivity to devices, car networks and broadband networks. MeeGo will provide the base for the upcoming GENIVI Apollo release that will be used by members to reduce time to market and the cost of IVI development. MeeGo's platform contains a Linux base, middleware, and an interface layer that powers these rich applications.

"We are pleased to see GENIVI choose MeeGo to power their software platform," said Jim Zemlin, executive director at The Linux Foundation. "MeeGo has been built from the ground up for these types of applications. Because MeeGo is a truly open platform, the work GENIVI will do to extend the platform can benefit the project and all who use it. For developers, this is a great opportunity to harness the power of the MeeGo APIs to target a variety of devices and architectures and extend their work on handset applications toward vehicles."

"We selected MeeGo as the open source basis for our platform because it is technically innovative and can provide the cross architecture build support we require for our references," said Graham Smethurst, President of GENIVI. "Working with MeeGo we expect to establish a solution that effectively merges IVI needs with those of the other MeeGo target device categories."

"Access to a thriving ecosystem and engaging applications for an in-vehicle infotainment system is vital to our customers," said Gerulf Kinkelin, GENIVI Director and Innovation Area Manager for Electronics and Telematices, PSA Peugeot Citroen. "PSA supports GENIVI's decision and believes the rapid innovation associated with open source and a rich network of contributors available with MeeGo can enable us to define and deliver a faster time-to-market, outstanding IVI experience in our upcoming products."

An initial release of the MeeGo platform is available now from http://www.meego.com/downloads [3].

History of Yocto Project (YP) in automotive Linux

History of Yocto Project (YP) in automotive Linux

- Wind River (Intel) proposed to maintain a YP recipe (Meta-IVI) representing GENIVI spec and compliance as baseline, not full distribution

History of Yocto Project (YP) in automotive Linux

- Wind River (Intel) proposed to maintain a YP recipe (Meta-IVI) representing GENIVI spec and compliance as baseline, not full distribution
- MeeGo-IVI was selected by GENIVI, then discontinued, opening up OSV enablement of multiple architectures and suppliers within GENIVI (ARM, Canonical)

History of Yocto Project (YP) in automotive Linux

- Wind River (Intel) proposed to maintain a YP recipe (Meta-IVI) representing GENIVI spec and compliance as baseline, not full distribution

- MeeGo-IVI was selected by GENIVI, then discontinued, opening up OSV enablement of multiple architectures and suppliers within GENIVI (ARM, Canonical)

- GENIVI moved its focus away from delivering full code stack, instead onto expert teams defining specs and software baselines

History of Yocto Project (YP) in automotive Linux

- Wind River (Intel) proposed to maintain a YP recipe (Meta-IVI) representing GENIVI spec and compliance as baseline, not full distribution
- MeeGo-IVI was selected by GENIVI, then discontinued, opening up OSV enablement of multiple architectures and suppliers within GENIVI (ARM, Canonical)
- GENIVI moved its focus away from delivering full code stack, instead onto expert teams defining specs and software baselines
- Alternate baseline and build system struggled to gain broad adoption among Tier1 suppliers, despite likely advantages

History of Yocto Project (YP) in automotive Linux

- Wind River (Intel) proposed to maintain a YP recipe (Meta-IVI) representing GENIVI spec and compliance as baseline, not full distribution
- MeeGo-IVI was selected by GENIVI, then discontinued, opening up OSV enablement of multiple architectures and suppliers within GENIVI (ARM, Canonical)
- GENIVI moved its focus away from delivering full code stack, instead onto expert teams defining specs and software baselines
- Alternate baseline and build system struggled to gain broad adoption among Tier1 suppliers, despite likely advantages
- Automotive Grade Linux gained traction among automotive supporters of Tizen-IVI until IVI development ceased, AGL pickup up from there on full stack distro model

Influence of YP on Automotive Linux Alignment

Influence of YP on Automotive Linux Alignment

- YP enables but does not implement any sort of interoperability of automotive OS implementations and their implied application interfaces

Influence of YP on Automotive Linux Alignment

- YP enables but does not implement any sort of interoperability of automotive OS implementations and their implied application interfaces
- Since YP became ubiquitous, there is greater possibility of alignment than when using OBS in some project and YP in others

Influence of YP on Automotive Linux Alignment

- YP enables but does not implement any sort of interoperability of automotive OS implementations and their implied application interfaces
- Since YP became ubiquitous, there is greater possibility of alignment than when using OBS in some project and YP in others
- There are now only two main sources for governance of automotive Linux, AGL and GENIVI, compared to dozens that existed prior to YP popularity

Influence of YP on Automotive Linux Alignment

- YP enables but does not implement any sort of interoperability of automotive OS implementations and their implied application interfaces
- Since YP became ubiquitous, there is greater possibility of alignment than when using OBS in some project and YP in others
- There are now only two main sources for governance of automotive Linux, AGL and GENIVI, compared to dozens that existed prior to YP popularity
- AGL dictates a specific software stack, does not demand compliance with GENIVI

Influence of YP on Automotive Linux Alignment

- YP enables but does not implement any sort of interoperability of automotive OS implementations and their implied application interfaces
- Since YP became ubiquitous, there is greater possibility of alignment than when using OBS in some project and YP in others
- There are now only two main sources for governance of automotive Linux, AGL and GENIVI, compared to dozens that existed prior to YP popularity
- AGL dictates a specific software stack, does not demand compliance with GENIVI
- GENIVI offers members guidance on approved components but allows OSV and Tier1 customization - not attempting to force a full standard across carmakers

What's next for YP in Automotive and What Else Should be?

What's next for YP in Automotive and What Else Should be?

- Reference Kit concept now includes IoT (Intel) and General Purpose (Poky)
  - Someone could sponsor a project to build and test an Auto RefKit

What's next for YP in Automotive and What Else Should be?

- Reference Kit concept now includes IoT (Intel) and General Purpose (Poky)
  - Someone could sponsor a project to build and test an Auto RefKit
- Better documentation of license scanning beyond package level is needed
  - GENIVI members are interested in helping, LiD project, SPDX

What's next for YP in Automotive and What Else Should be?

- Reference Kit concept now includes IoT (Intel) and General Purpose (Poky)
  - Someone could sponsor a project to build and test an Auto RefKit
- Better documentation of license scanning beyond package level is needed
  - GENIVI members are interested in helping, LiD project, SPDX
- YP is capable of building product, however automotive Tier1 long term maintenance (7-10 years) is unproven process
  - Need more automotive requirements while keeping competencies in the right communities (AGL could help YP)

Lessons learned

Lessons learned

- Connected Cars will need more consistency of architecture to achieve services across makes, models, and geographies

Lessons learned

- Connected Cars will need more consistency of architecture to achieve services across makes, models, and geographies

- Fragmentation is not resolved simply by using common tools - requires decisions made outside of YP, such as AGL

Lessons learned

- Connected Cars will need more consistency of architecture to achieve services across makes, models, and geographies

- Fragmentation is not resolved simply by using common tools - requires decisions made outside of YP, such as AGL

- Product development needs are different than R&D needs and may require added robustness to build systems

Lessons learned

- Connected Cars will need more consistency of architecture to achieve services across makes, models, and geographies

- Fragmentation is not resolved simply by using common tools - requires decisions made outside of YP, such as AGL

- Product development needs are different than R&D needs and may require added robustness to build systems

- Owning software in large scale product development (like auto) is a complex thing, look for many friends that will help!

OPEN SOURCE
LEADERSHIP SUMMIT
FEBRUARY 14–16, 2017 | LAKE TAHOE, CA

THE LINUX FOUNDATION