

connected cars

By MICHELE PAOLINO, AUTOMOTIVE GRADE LINUX VIRTUALIZATION EXPERT GROUP

JUNE 25, 2018

This blog is part one in a three part series. Look for part two next week.

Automotive Grade Linux (AGL) is one of the most innovative community involved in embedded systems developments. Such innovation, as officially recognized by Embedded Computing Design (<http://www.embedded-computing.com/articles/2018-top-embedded-innovator-dan-cauchy-automotive-grade-linux>), has its root in the open source and code-first approach and consists of delivering a de-facto Linux-based standard platform that significantly reduces automotive software time to market and licensing costs. This platform is a disruptive innovation in the automotive industry today, where OEMs and Tier-1 companies are continuously challenged to provide smarter, cheaper, and more connected functionalities.

The AGL software platform provides a vertical solution starting from the operating system layer (Linux) up to APIs, software-development kit (SDK), and reference applications. However, autonomous driving and software defined vehicles require the execution of several hundreds functions with different functional safety requirements. In fact, users want to see the automatic pilot function running together with their mailbox application, or to be able to let children play the latest 3D video games on the backseats while using the rearview camera. In this context, there is a need to extend the AGL platform to support an additional technology allowing the safe and efficient combination of mixed critical functions: virtualization.

What is virtualization?

Virtualization is a technique used to create multiple virtual execution environments by means of resource abstraction or partitioning. Historically designed as a technology for the server market, with the increasing computational power of embedded hardware platforms, virtualization also found applications in embedded systems with hypervisors, system partitioners, and containers. The hypervisor is a software component running close to the operating system layer that creates hardware resources abstraction for the execution of different operating systems concurrently in the form of virtual machines (VMs). System partitioners do not use abstraction, but partition system resources to create siloed virtual environments for the execution of operating systems directly on the (partitioned) hardware.

Containers, on the other hand, achieve the same objective of hypervisors and system partitioners, but without using specific capabilities of the hardware platform. Their use in AGL is limited to workloads without safety or real time requirements for this reason.

ADVERTISEMENT



Android and Human Interaction: Warp Speed Ahead

Mouser takes a look at the advances in technology surrounding androids in the fields of robotics, and cybernetics. Innovative companies like Intel are developing solutions and components that offer advances in programming, artificial intelligence (AI), computing, storage and power.

[CLICK HERE](#)

AGL automotive virtualization

In modern vehicles there is a need to isolate the different functions securely, with support for ISO 26262 safety certification, and in a way that they can't interfere with each other. Moreover, users are expecting an increasing number of these functions to be installed on the fly, customizable and upgradable over the air. In this scenario, adding a new electronic control unit (ECU) for each new function in the car architecture does not apply anymore: it does not scale with the increasing number of functions, it cannot provide the customization level required by the users, and it exponentially increases costs (hardware, wiring, maintenance, etc.).

For AGL, virtualization is the software defined connected car enabler technology because of its ability to host the execution of several (and isolated) functions concurrently, by offering at the same time run-time configurability, a flexible and scalable architecture in a secure and efficient way. AGL is working to expand its platform with a set of virtualization technologies and, thanks to a team of virtualization professionals inside the community that constitutes the Virtualization Expert Group (EG-VIRT), published the white paper, [The AGL Software Defined Car Architecture \(https://www.automotivelinux.org/wp-content/uploads/sites/4/2018/06/agl_software_defined_car_jun18.pdf\)](https://www.automotivelinux.org/wp-content/uploads/sites/4/2018/06/agl_software_defined_car_jun18.pdf).

The objective of this work is to pave the way towards an open source de-facto standard for software defined connected cars.

Michele Paolino is a senior virtualization architect at Virtual Open Systems, a company involved in open source virtualization extensions development. He has contributed to several open source projects and is today leading the Automotive Grade Linux Virtualization Expert Group, aiming to bring open source virtualization in automotive. He holds a Master degree in Computer Engineering from the Alma Mater Studiorum University of Bologna with a thesis about virtualization solutions for linux-based heterogeneous SoCs and programmable manycore accelerators. Today, his activity focuses on mixed-critical virtualization. He is working as principal architect for the Virtual Open Systems' [VOSySIoT \(http://www.virtualopensystems.com/en/products/vosysiot-edge/\)](http://www.virtualopensystems.com/en/products/vosysiot-edge/), [VOSYSmcs product \(http://www.virtualopensystems.com/en/products/vosysmcs/\)](http://www.virtualopensystems.com/en/products/vosysmcs/)s, which are mixed critical virtualization solutions for the IoT edge and automotive markets; he is also Innovation manager for the [NGPaaS project \(http://ngpaas.eu/\)](http://ngpaas.eu/).

ADVERTISEMENT

(<http://www.embedded-computing.com/articles/eliminating-complexities-in-embedded-system-development>)

Eliminating complexities in embedded system development

Using a model-driven approach and adopting agile development can reduce or eliminate complexities in the em...





Start the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS

Name

Be the first to comment.

Recommended



Eliminating complexities in embedded system development

Using a model-driven approach and adopting agile development

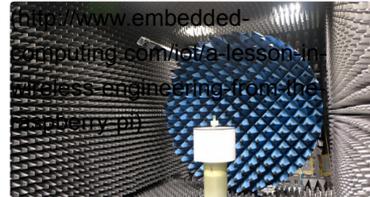
Read Article



Server technology in a small package

The fact that there are so many different Industrial Internet of Things (IIoT) applications that require edge or fog computers calls

Read Article



A lesson in wireless engineering from the Raspberry Pi

How did the engineers behind the Raspberry Pi Zero W tackle antenna design, given its compact

Read Article



2018 Top Innovator Cauchy, A Grade Lin

Vehicles are be defined network

Read Article

[Return to Home \(http://www.embedded-computing.com/\)](http://www.embedded-computing.com/)