Virtual Open Systems

What's next in AGL virtualization: EG-VIRT communication mechanism and IO virtualization

Michele Paolino

AGL AMM Europe 2018

2018/10/18

Acknowledgement

This work has received funding from the European Union's Horizon 2020 research and innovation program, NGPaaS, under grant agreement No 761557. The work presented in this paper reflects only authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

NGPaaS

www.ngpaas.eu

Why virtualization in automotive?

- Number of automotive functions is exploding
- Software time to market needs to be reduced
- Updates are required to be performed faster and at a lower cost
- Cyber security is a threat
- Combination of different software ecosystems is extremely of interest

EG-Virt introduction

The Virtualization Expert Group (EG-VIRT) is the AGL team which aims to bring open source virtualization in production cars.

- Kicked-off at the beginning of 2017, EG-VIRT's first achievement has been the creation of the AGL virtualization infrastructure
- Meta-egvirt layer, agl-egvirt feature and KVM porting to the Renesas R-Car M3
- EG-VIRT is composed by virtualization professionals active in AGL
- Bi-weekly meeting are held on Wednesday
- Wiki page: https://wiki.automotivelinux.org/eg-virt

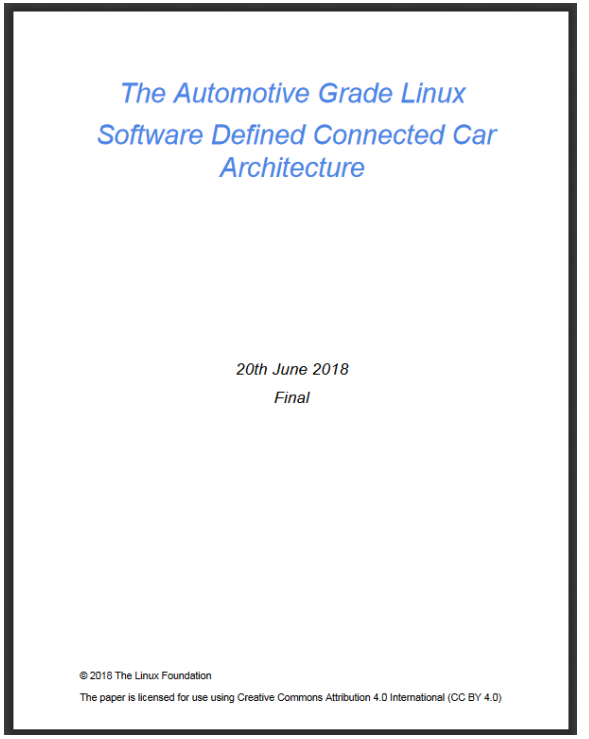
EG-Virt purpose

- Introduce virtualization in AGL and set the ground for virtualized open source autonomous vehicles
- Provide a reference virtualization platform for future automotive systems
- Enable Tier-1 companies and automakers to differentiate their offer by adding customization and extensions

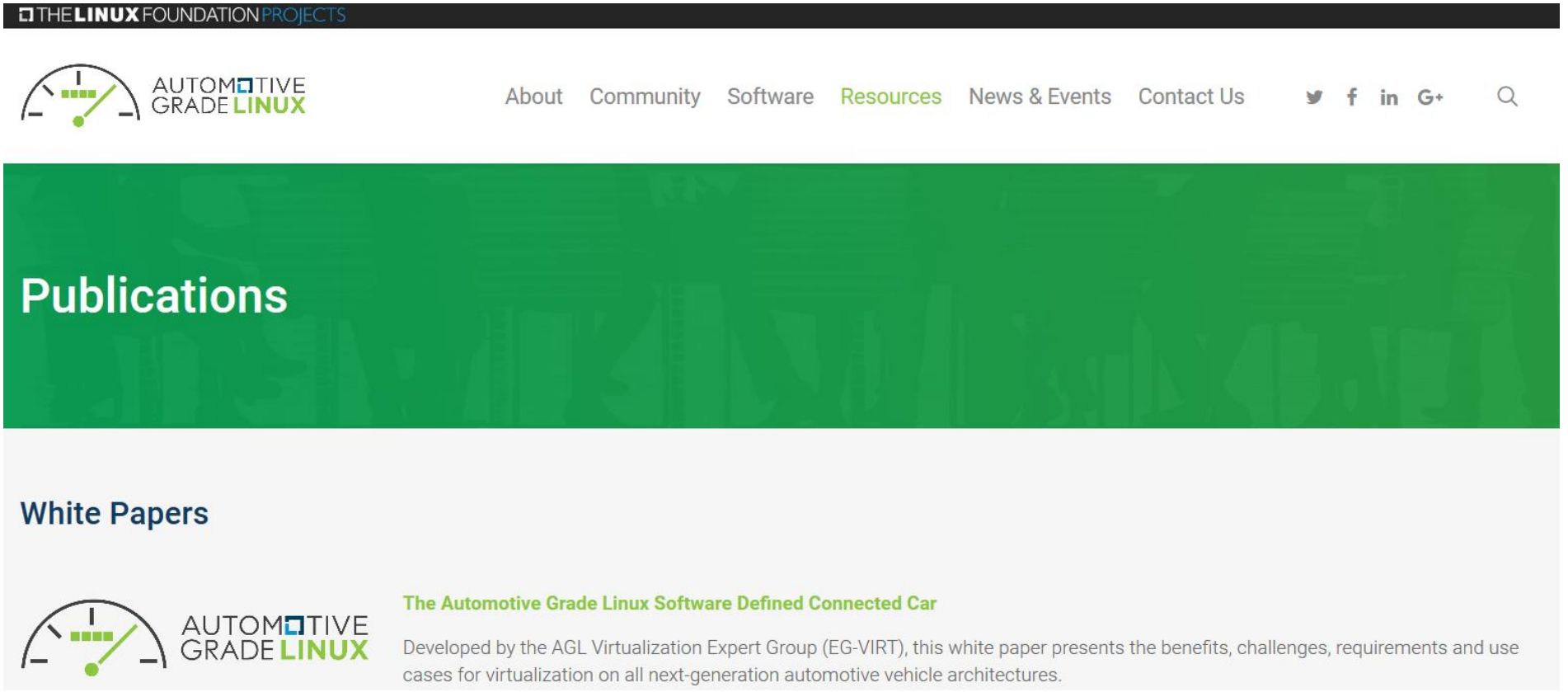
EG-VIRT white paper

The AGL software defined connected vehicle architecture

- Published during ALS2018
- Creative Commons 4.0 license (CC BY 4.0 international)
  - You are free to share and adapt.
  - You must give credit to AGL
- 15 authors



EG-VIRT white paper



Your feedback is needed!

EG-VIRT white paper key contributions

The key contributions of the work are:

- **Definition of the AGL EG-VIRT approach towards virtualization**
- Definition of the AGL architecture
- Identification of the AGL role in open source virtualization

How to approach virtualization for AGL? Mixed-criticality

- Applications with different level of criticality are targeted to coexist and run in a virtualized manner.
- AGL targets to consolidate applications different certification requirements.

How to approach virtualization for AGL? Modularity

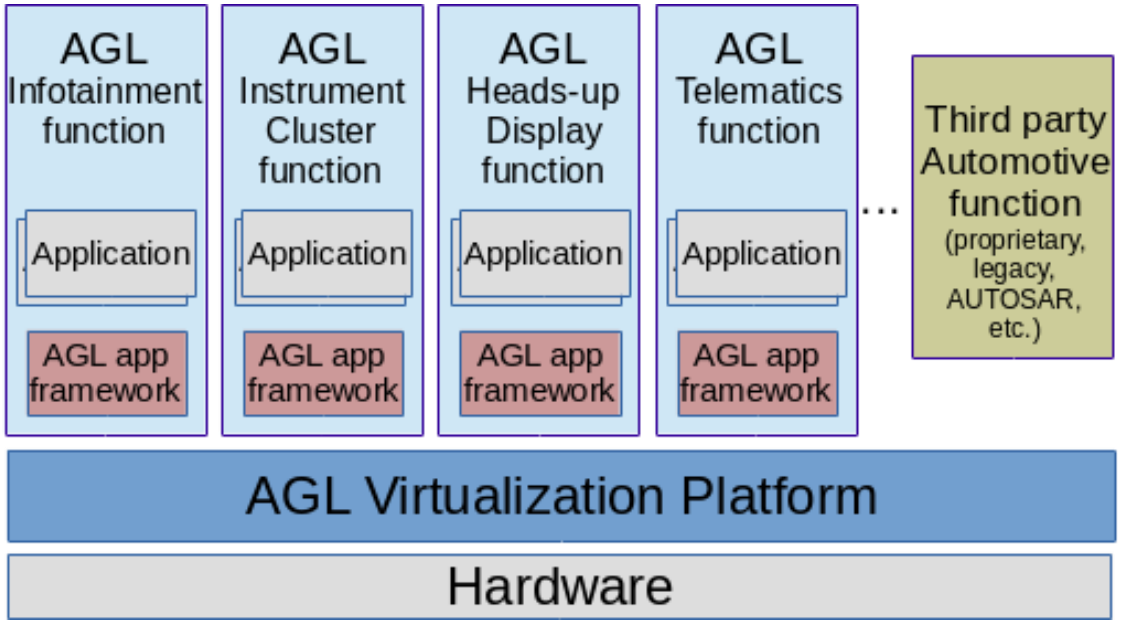
- Hypervisors, virtual machines, AGL Profiles and automotive functions are seen by the AGL architecture as interchangeable modules.
- The combination of these modules differentiates AGL products.
- To achieve modularity, interoperability will be required, especially between open and proprietary components.

How to approach virtualization for AGL? Openness

- There is no restriction in the way the AGL virtualization platform can be used, deployed and extended.
- The AGL virtualization architecture supports multiple hypervisors, CPU architectures, software licenses and can be executed as a host and guest.

How to approach virtualization for AGL?

Putting together modularity, openness and mixed criticality



EG-VIRT white paper key contributions

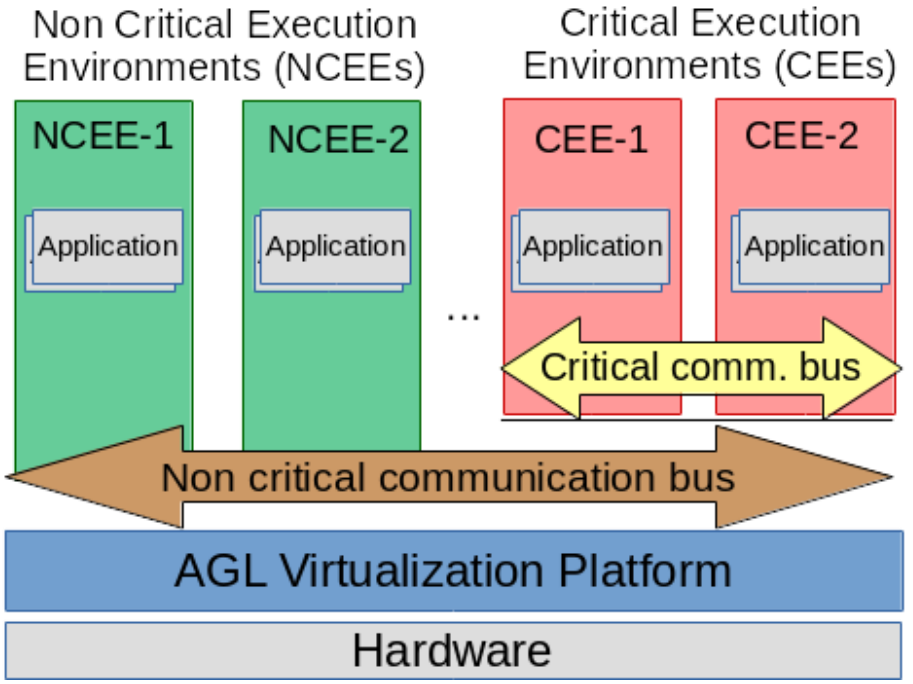
The key contributions of the work are:

- Definition of the AGL EG-VIRT approach towards virtualization
- **Definition of the AGL architecture**
- Identification of the AGL role in open source virtualization

The AGL virtualized architecture

The AGL virtualized architecture is composed by:

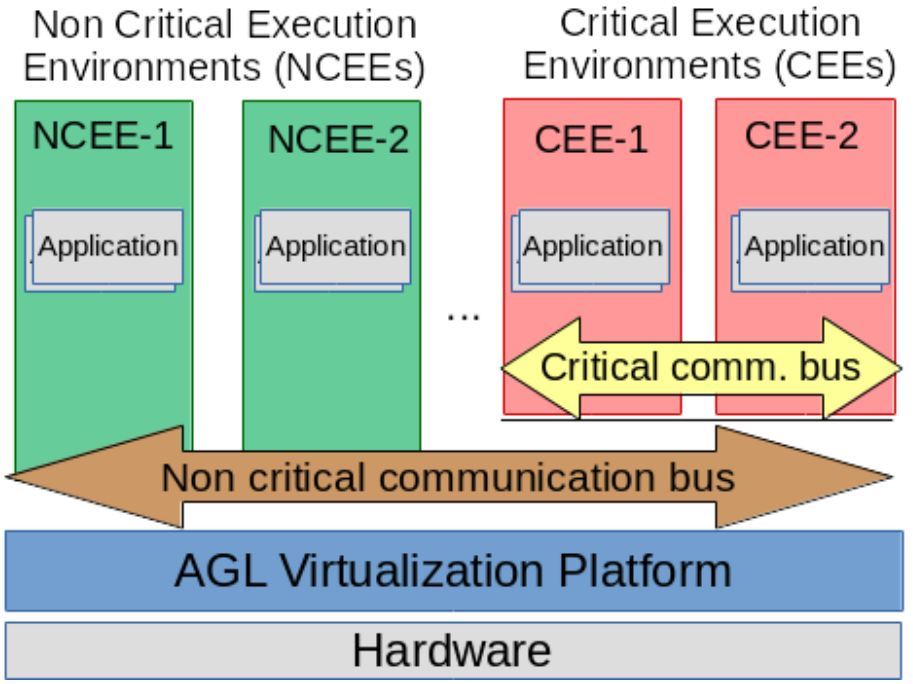
- Execution environments (EEs)
- Virtualization platform
- Communication buses



The AGL Execution Environments (EEs)

Execution environments (EEs) are software silos created by the virtualization platform:

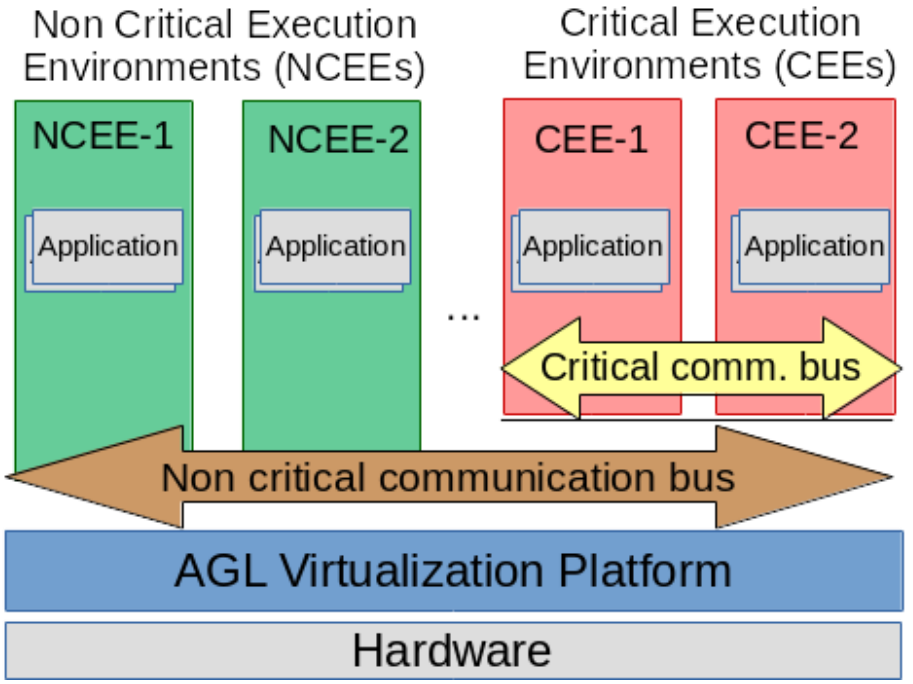
- They run isolated from each other
- EEs can be implemented as bare metal applications, virtual machines, etc.
- Can execute critical or non critical functions



The AGL Virtualization Platform

It a safe and secure execution of multiple EEs consolidating them in a single hardware/software platform

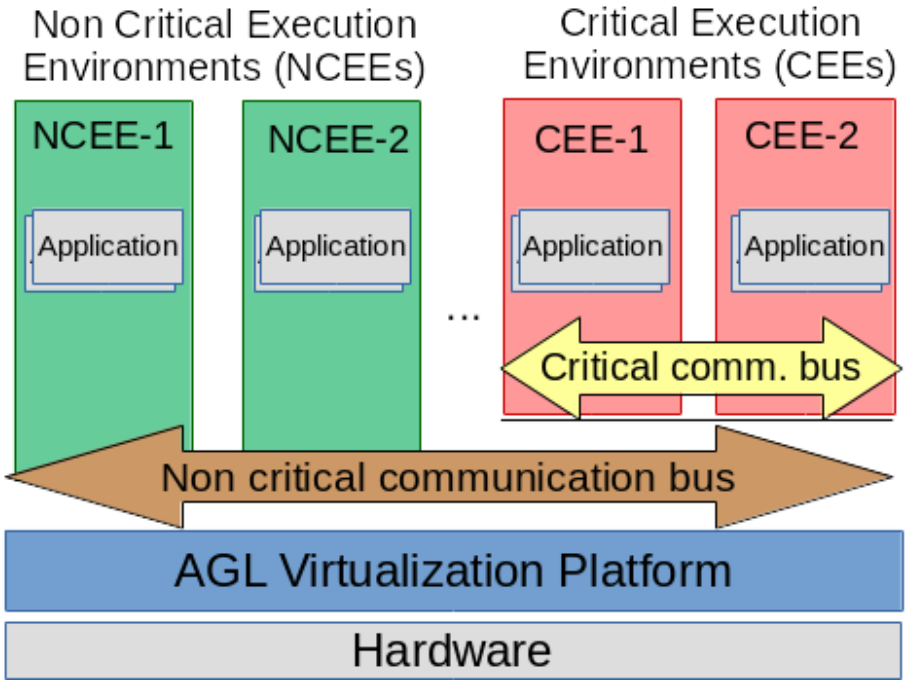
- Can be implemented as a hypervisor, system partitioner, container engine, etc.
- Has to use hardware mechanisms to properly isolate EEs
- Provide tools to enhance modularity, openness and mixed criticality



The AGL Communication Buses

Create connection endpoints between EEs as well as between the Virtualization Platform and each EE

- Have different level of criticality (requirements, objectives, etc.)
  - The Critical one needs to be securely isolated
  - The non critical requires high performance and bandwidth



EG-VIRT white paper key contributions

The key contributions of the work are:

- Definition of the AGL EG-VIRT approach towards virtualization
- Definition of the AGL architecture
- **Identification of the AGL role in open source virtualization**

The role of AGL in the open source virtualization developments

AGL's role is the one of virtualization technology integrator, aiming at supporting different virtualization technologies and to make them interoperable and interchangeable.

- AGL will not develop a new hypervisor but will leverage on existing open source solutions
- The developments will aim to enhance openness, modularity and portability of its platform

AGL: enhancing existing Virtual Platforms, EEs

As a consequence, the developments will aim to enhance openness, modularity and portability of its platform. Practical examples are:

- Development of new open/standard APIs
  - Communication, software defined automotive functions life management, etc.
- But also
  - Portable drivers for IO virtualization, test bench, image building tools for different virtualization solutions, etc.

AGL EG-VIRT challenges ahead

Key points identified by EG-VIRT as future works are:

- Design and develop standardized communication buses (critical and non critical).
  - This is seen as an enabler of virtualized automotive functions portability, interoperability, performance, security and safety.
- Add support for GPU virtualization

AGL EG-VIRT challenges ahead

Key points identified by EG-VIRT as future works are:

- **Design and develop standardized communication buses** (critical and non critical).
  - This is seen as an enabler of virtualized automotive functions portability, interoperability, performance, security and safety.
- Add support for GPU virtualization

Coomunication Buses: AGL EEs requirements

The AGL Execution Environments (EEs) needs to communicate with each other and with the outside world. They have different requirements:

- NCEE: High performance bandwidth (video streaming, Gbps)
  - Ethernet is a good candidate
- CEE: Security and resiliency for safety critical messages (messages, Kbps)
  - CAN is a key solution to support

Coomunication Buses: virtio

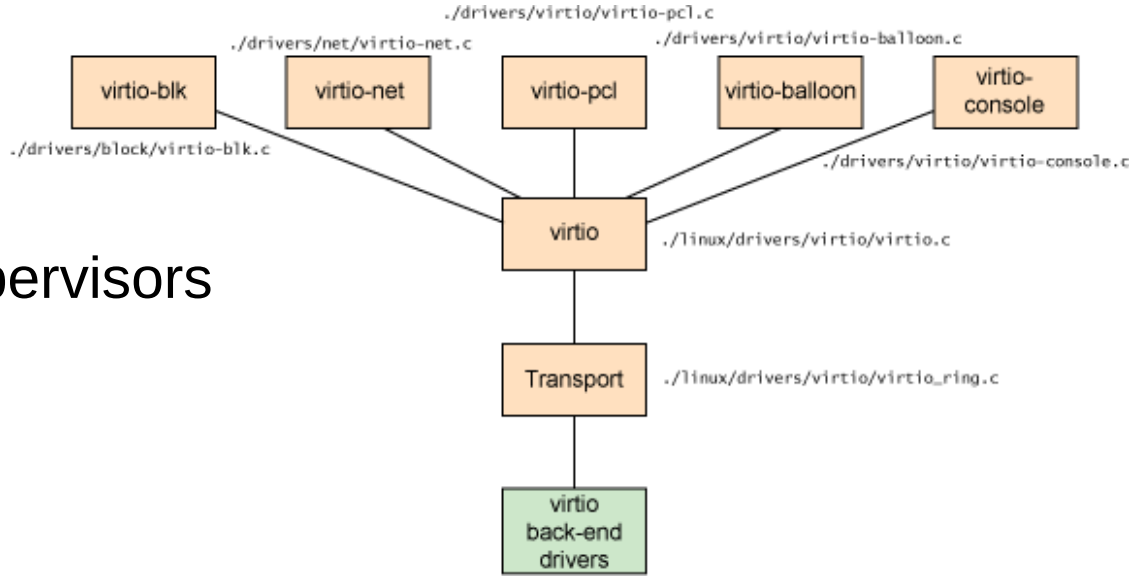
virtio is a standard (OASIS) for virtualizing IO devices

- It is based on bulk data transport and a DMA like memory model
- Separates the device and

the transport part

- Already supported by different hypervisors

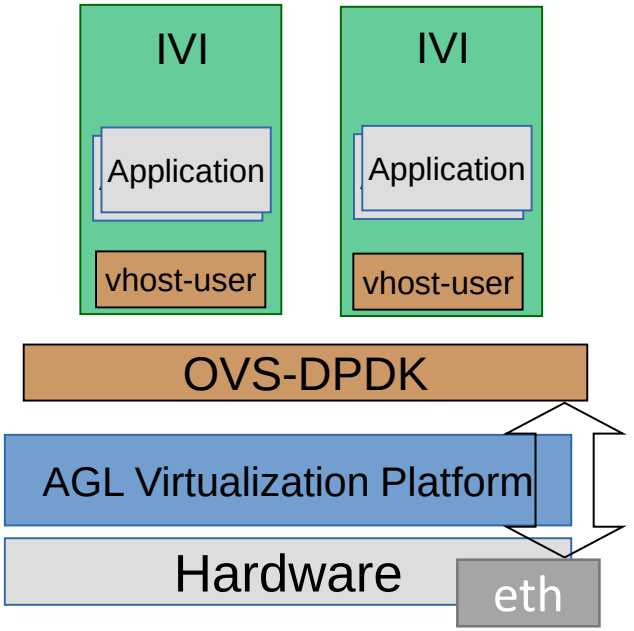
and operating systems



Ethernet vSwitch

Guest to guest ethernet-based communication is usually implemented through virtual switch engines

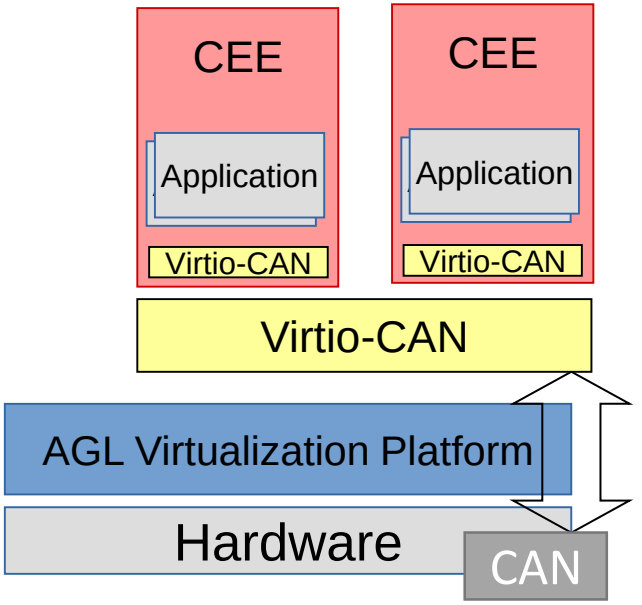
- Open Virtual Switch is one of the most used solutions in server infrastructures
- Software acceleration available with specific dataplane implementations (e.g., DPDK)
- Open Source, with an active community



virtio-CAN

virtio-CAN enables paravirtualizated support for CAN communication inside guests

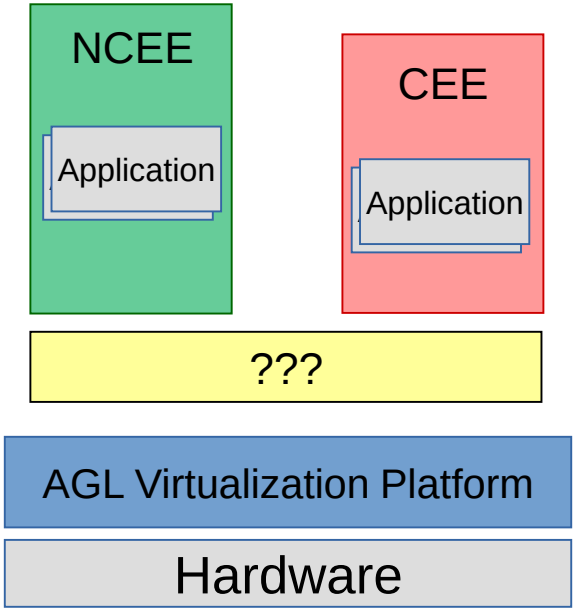
- It is based on the widely used SocketCAN linux framework
- Guests can communicate with each other and with external CAN devices
- Open Source



Communication buses challenges to overcome

Today we lack of optimal solutions for EE communication

- Determinism, high performance
- Security, QoS, etc.
- Hypervisor independence
- Support of legacy applications, portability



AGL EG-VIRT challenges ahead

Key points identified by EG-VIRT as future works are:

- Design and develop standardized communication buses (critical and non critical).
  - This is seen as an enabler of virtualized automotive functions portability, interoperability, performance, security and safety.
- **Add support for GPU virtualization**

GPU virtualization: AGL EEs requirements

The AGL Execution Environments (EEs) have different requirements

- NCEE: High performance graphics (3D acceleration, 60FPS)
- CEE: Availability of a secure area on the screen where to draw warning icon and safety critical messages (resilience, >30FPS ISO 15005)

GPU Virtualization

Different approaches are available and depend of the hardware capability:

- Direct assignment (no device sharing)
- Hardware assisted virtualization
- Paravirtualization/API remoting

GPU Virtualization – Direct assignment

Direct assignment allocates exclusively devices to an EE

- Provides best performance and isolation
- Multiple GPUs are required for multiple EEs
- Not the most efficient solution
- No device sharing

VFIO is one of the technologies that enables direct assignment

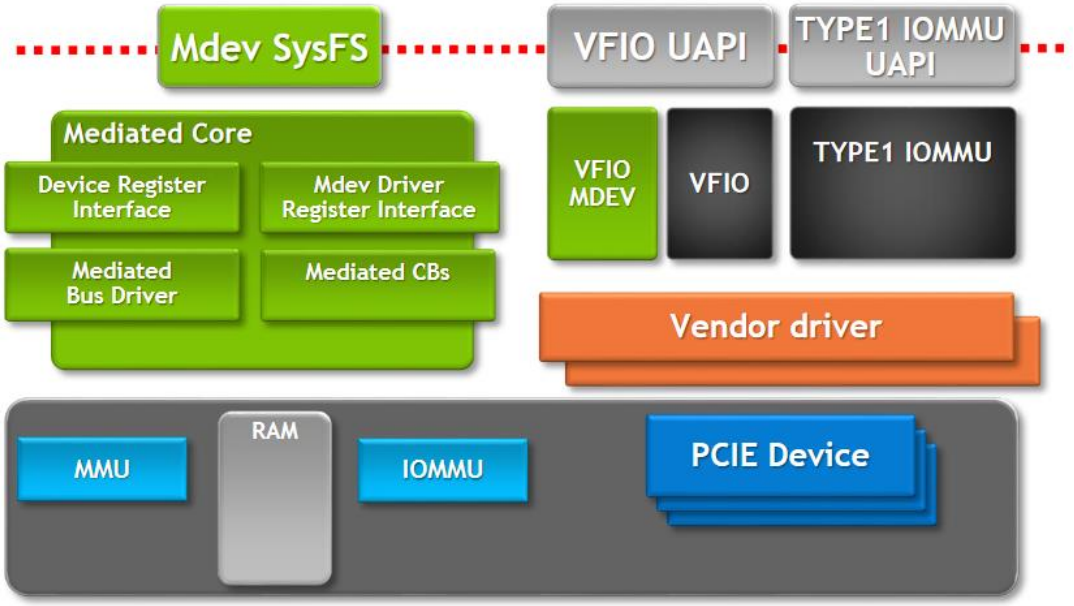
GPU Virtualization – Hardware assisted

GPU/display controller exposes multiple virtual instances of the hardware resources

- Hardware support required
- Not all the automotive platforms today have GPU/display controllers that support hardware assisted virtualization
- Direct assignment is used to allocate each virtual instance to an EE
- The number of EEs that can be connected could be limited by the availability of virtual instances
- Highly dependent from the GPU manufacturer (mdev framework, proprietary solutions, etc.)

mdev based GPU virtualization

Mediated device framework is a kernel feature that decouples vendor drivers from virtual instances



Source linux-kvm.org

GPU Virtualization – Paravirtualization

The support for virtualization is implemented in software.

- No hardware support required
- It is flexible, and can overcome hardware limitations in software
- Overhead due to the software implementation of virtualization support
- Can be implemented at the API level for specific applications, e.g., OpenGL (API remoting)

Project of interest in this field are virgil3D, virtio-GPU

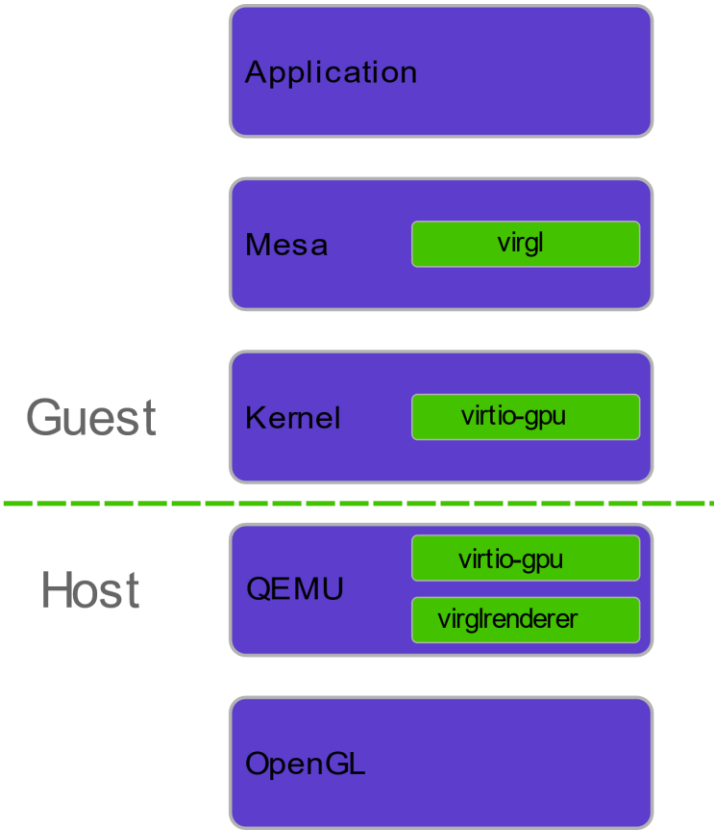
virtio-gpu

virtio-GPU provides a portable solution for GPU virtualization

- OpenGL based
- Mesa abstracts the hardware

implementation details

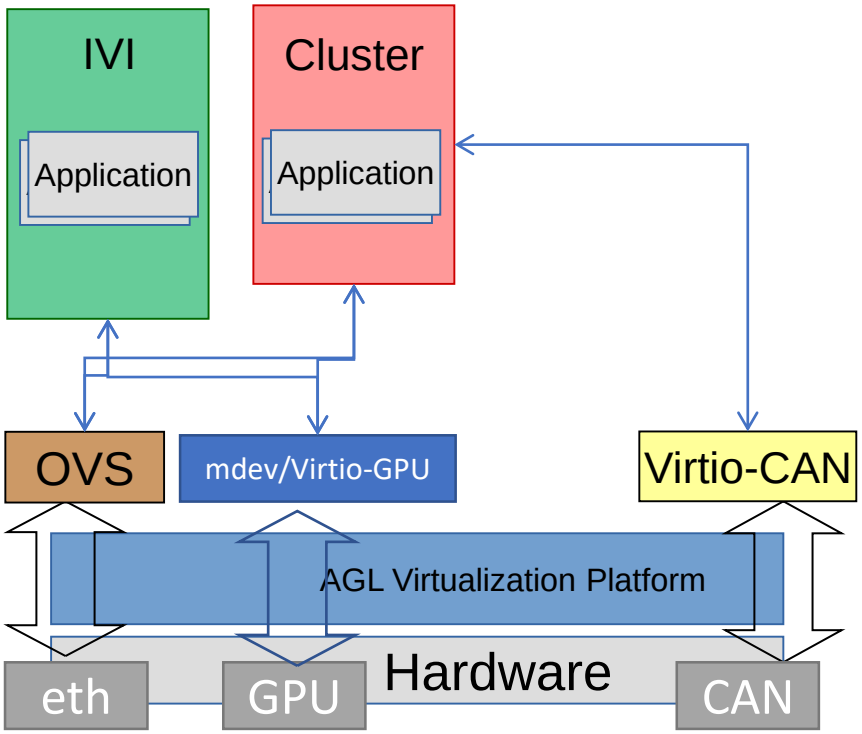
- Open Source



Source memcpy.io

EG-VIRT RFC

- Different implementation alternatives for the short term
- Feedback of the community is very important, join the discussion!



Thank You

Questions?

m.paolino@virtualopensystems.com

contact@virtualopensystems.com

Web: virtualopensystems.com

Products: http://www.virtualopensystems.com/en/products/

Demos: virtualopensystems.com/en/solutions/demos/

Guides: virtualopensystems.com/en/solutions/guides/

Research projects: virtualopensystems.com/en/research/innovation-projects/