TAPPS

Trusted Apps for open CPSs

Cyber Security through Virtualization

Alvise Rigo

a.rigo@virtualopensystems.com

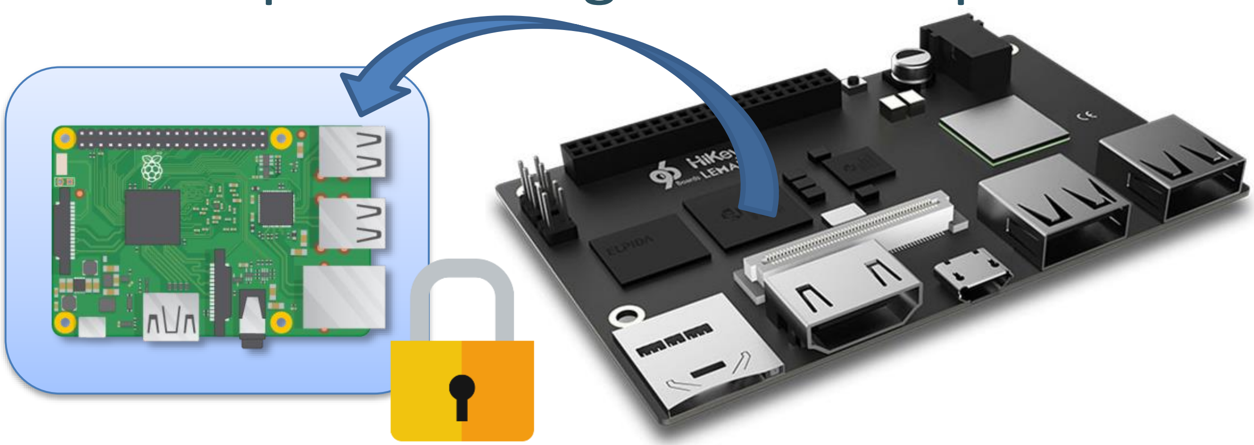
Virtual Open Systems

TAPPS Workshop Ispra 2017-05-10

Co-funded by the Horizon 2020 Framework Programme of the European Union under grant agreement no 645119

Virtualization

- Creation of a “virtual” copy of a resource
  - Virtualizing several resources (CPU, memory, etc.), an entire system can be executed in a *isolated execution environment*
  - For instance, a virtualized Raspberry Pi system (as **guest**) can run inside an HiKey board (the **host**)
  - But also multiple OSs as guests on top of an host OS

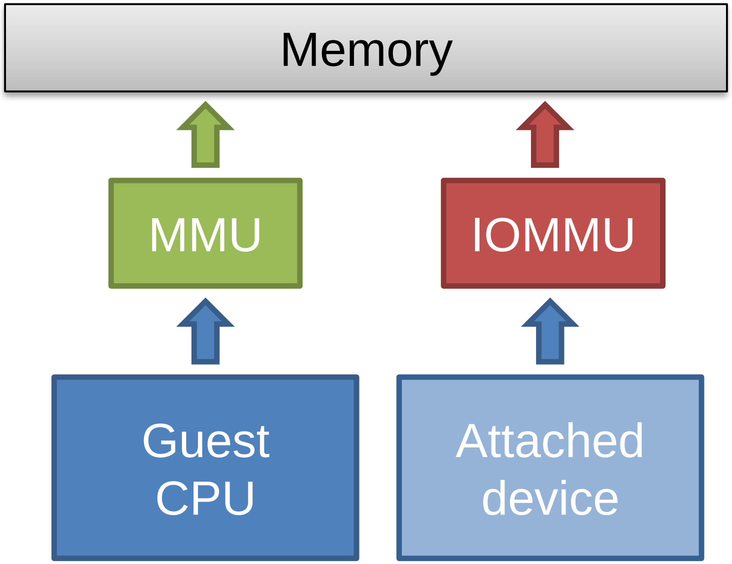


Virtualization

- Types of virtualization
  - Full virtualization
  - Para-virtualization
- Types of hypervisor
  - Type I (Xen)
  - Type II (KVM)
- Virtualization is different from emulation
  - Usually, devices are emulated, CPU are virtualized
  - A virtualized CPU is much more performant than an emulated one
- KVM is the in-kernel virtualization solution for Linux

Benefits of Virtualization

- Main benefit: **Security**
  - Spatial isolation between guest OS(s) and host OS
  - The execution of the guest will not affect the execution of the host OS
  - The hardware MMU and IOMMU allow to enforce the isolation

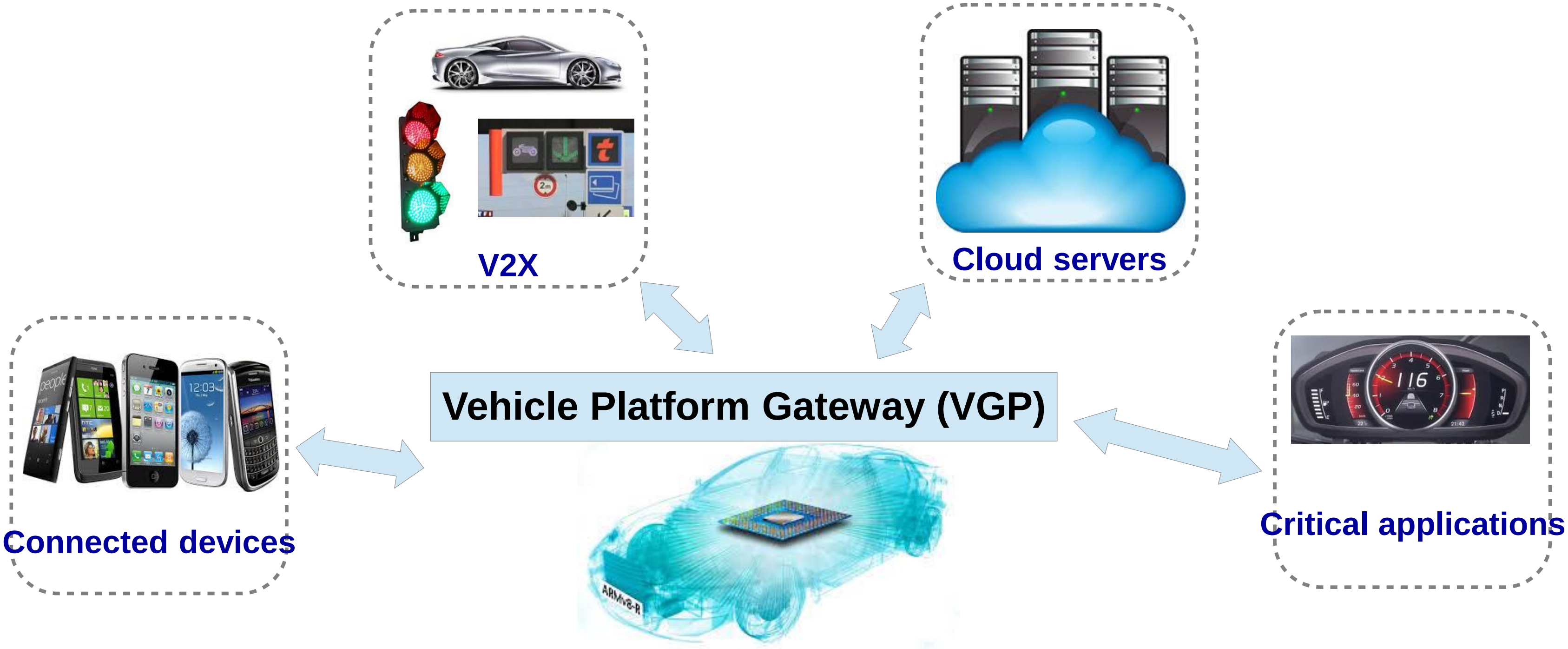


Benefits of Virtualization

- Several functions can be run inside virtual machines (VMs)
  - A web server and a MySQL server can be run in their dedicated VM
  - Malicious attacks to both these services shall not compromise the host OS
- In the context of CPS, virtualization brings interesting advantages like the **consolidation** of different functionalities on a common hardware platform

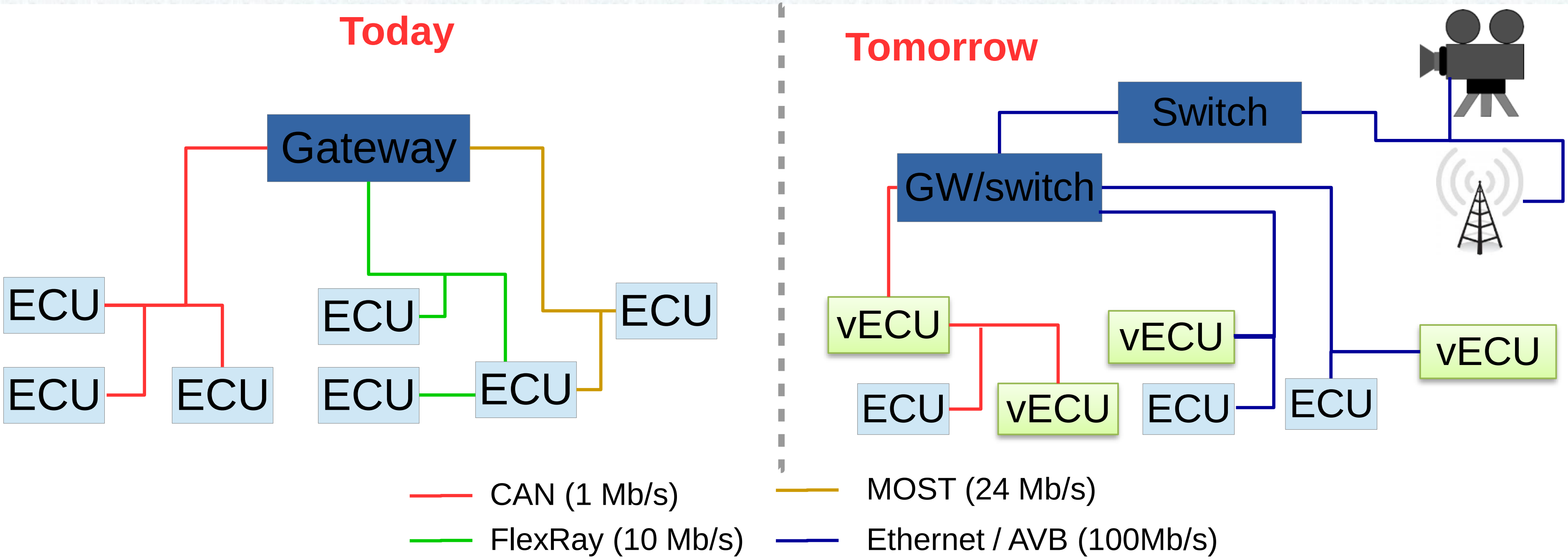
Automotive example: Connected-cars

The main challenge connected cars is the integration of information (e.g., IVI, V2X, connected devices, etc.) with critical data flows:



VGP must support interconnection with external applications while ensuring in-vehicle buses secure access to ECUs, which contains critical applications

Connected-cars: From Gateway to Backbone Arch



(Source: Automotive Gateways – Bridge & Gateway from FlexRay/CAN/LIN to AVB Networks - BOSCH)

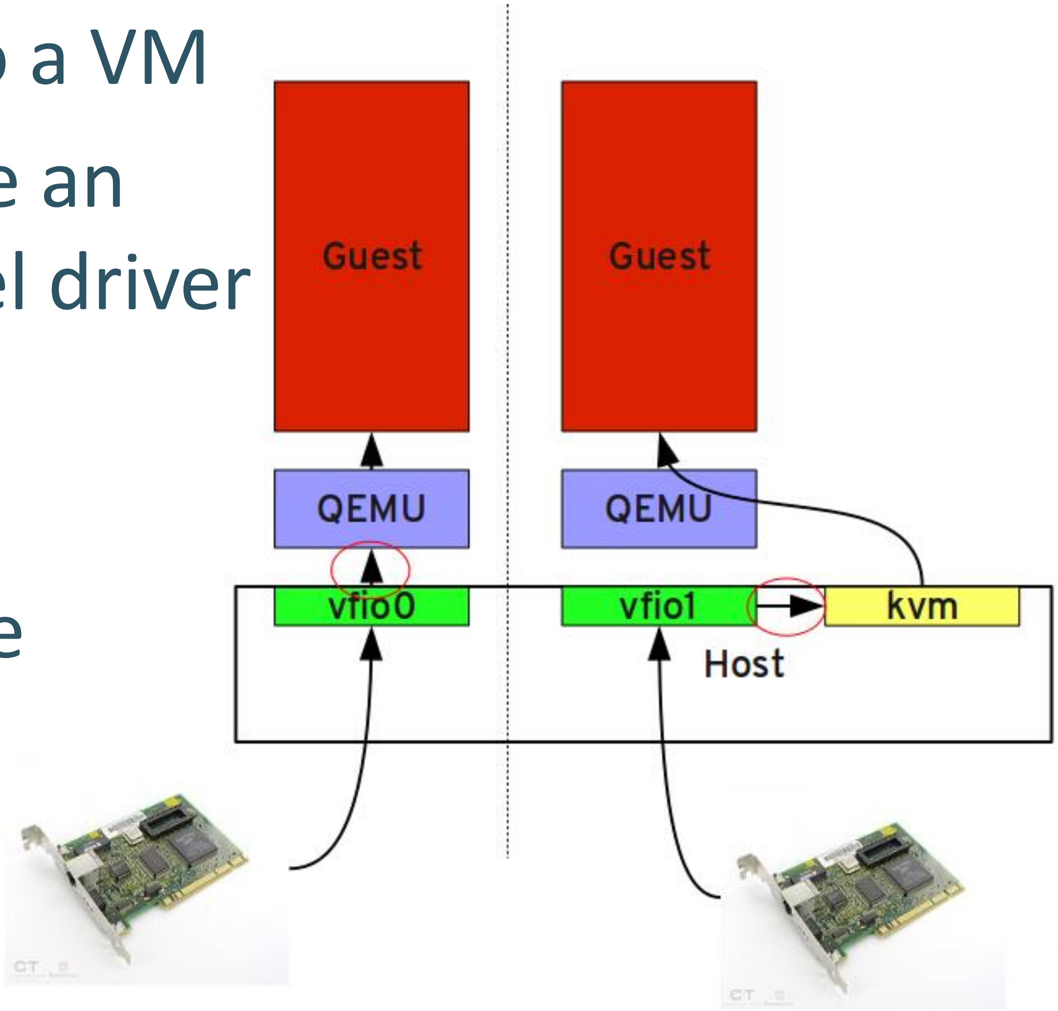
- New Connected cars' functionalities add an amount of streaming data and control signals, which cannot be handled by the current infrastructure.
- The future car will become an Ethernet networking based platform.

Performance Challenges in Virtualization

- Very few devices are optimized for virtualization
  – Only few network cards (SR-IOV specification)
- Device emulation
  – Always forces a guest-to-host switch
  – The emulation of the device adds overhead
  – Emulation vulnerabilities (VENOM, CVE-2015-3456)
- Two alternative solutions to emulation
  – Direct Device Attachment/Pass-through
  – API remoting

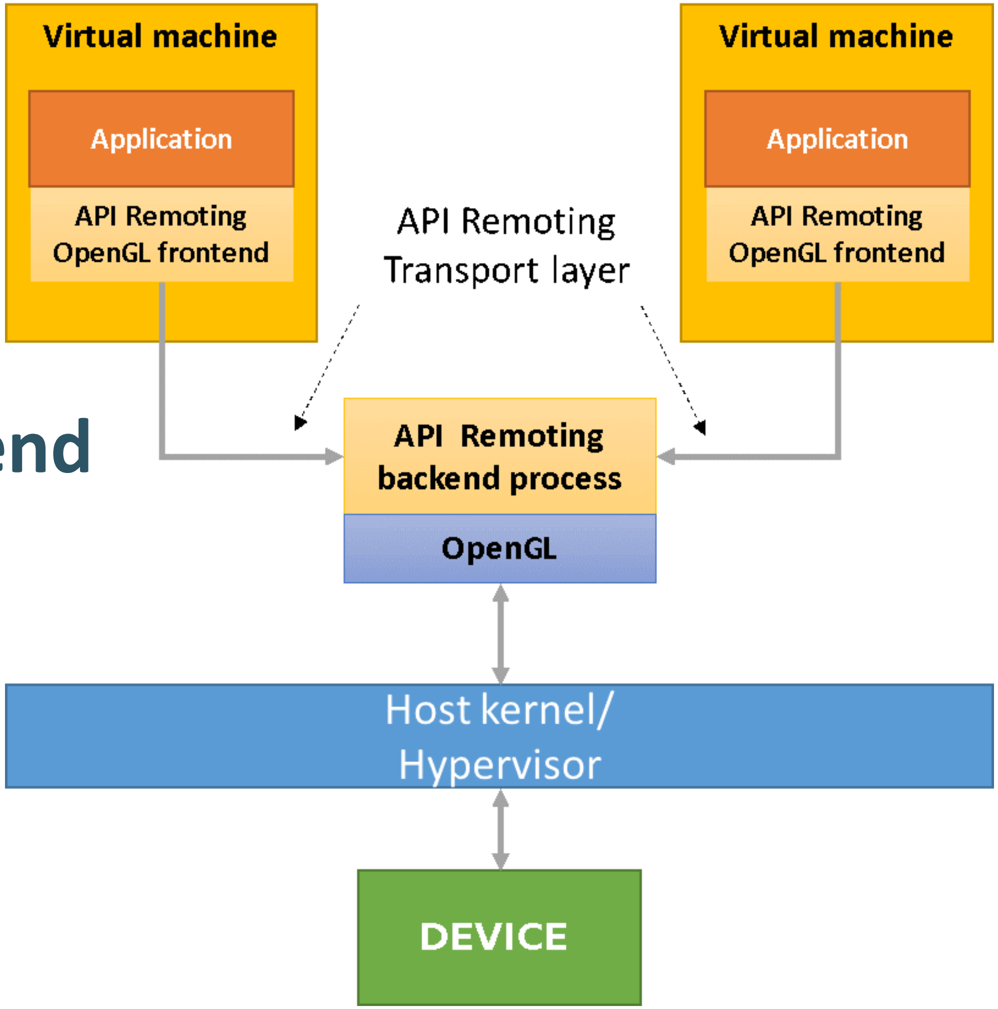
Device Pass-through (VFIO)

- Assign a device to a VM
- ✚ The VM can use an unmodified kernel driver
- ✚ Almost native performance
- ▬ Only one device per VM



API remoting

- The **frontend** is run inside the VM
- The OS that has direct access to the device runs the **backend**
- ▬ Performance overhead
- ✚ Many users (VMs) supported

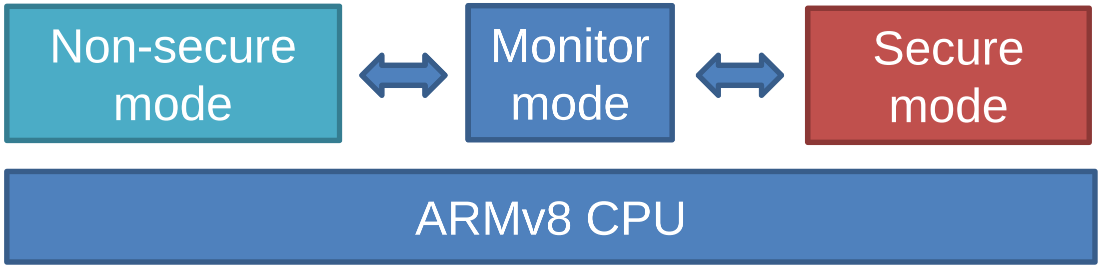


Mixed-criticality in CPS

- Virtualization provides isolation and permits to achieve consolidation
- However, this is not enough for mixed-criticality CPSs
  - Non-commercial hypervisors hardly can ensure a proper execution of an RTOS

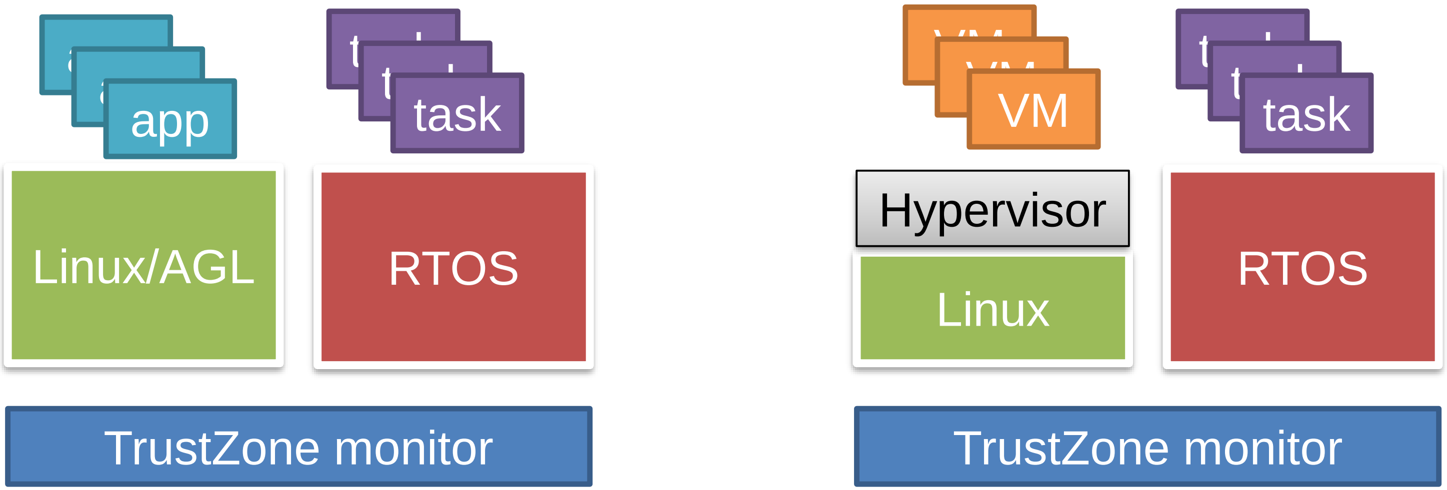
ARM TrustZone

- Separation of the CPU in two different execution states: Secure and Non-secure
- Possibility to add a Secure compartment to the “normal”, Non-secure one
- Context switch between the two modes is done by the monitor, in the most privileged way
  – Similar to what happens with Virtualization where the “monitor” is Linux KVM



Mixed-criticality in CPS

TrustZone monitor as key solution



Single GPOS
Single RTOS

Multiple GPOS
Single RTOS

- RTOS running in baremetal without virtualization overhead
- VMs can not affect the RTOS

VOSYSmonitor

High-performance Implementation of ARMv8 monitor layer

VOSYSmonitor is a Virtual Open Systems proprietary firmware (C/ASM), running in the Secure Monitor mode (EL3) of ARMv8 processors (64-bits), which enables co-execution of virtualized IVI systems with a safety critical real time OS on the same platform and/or core.

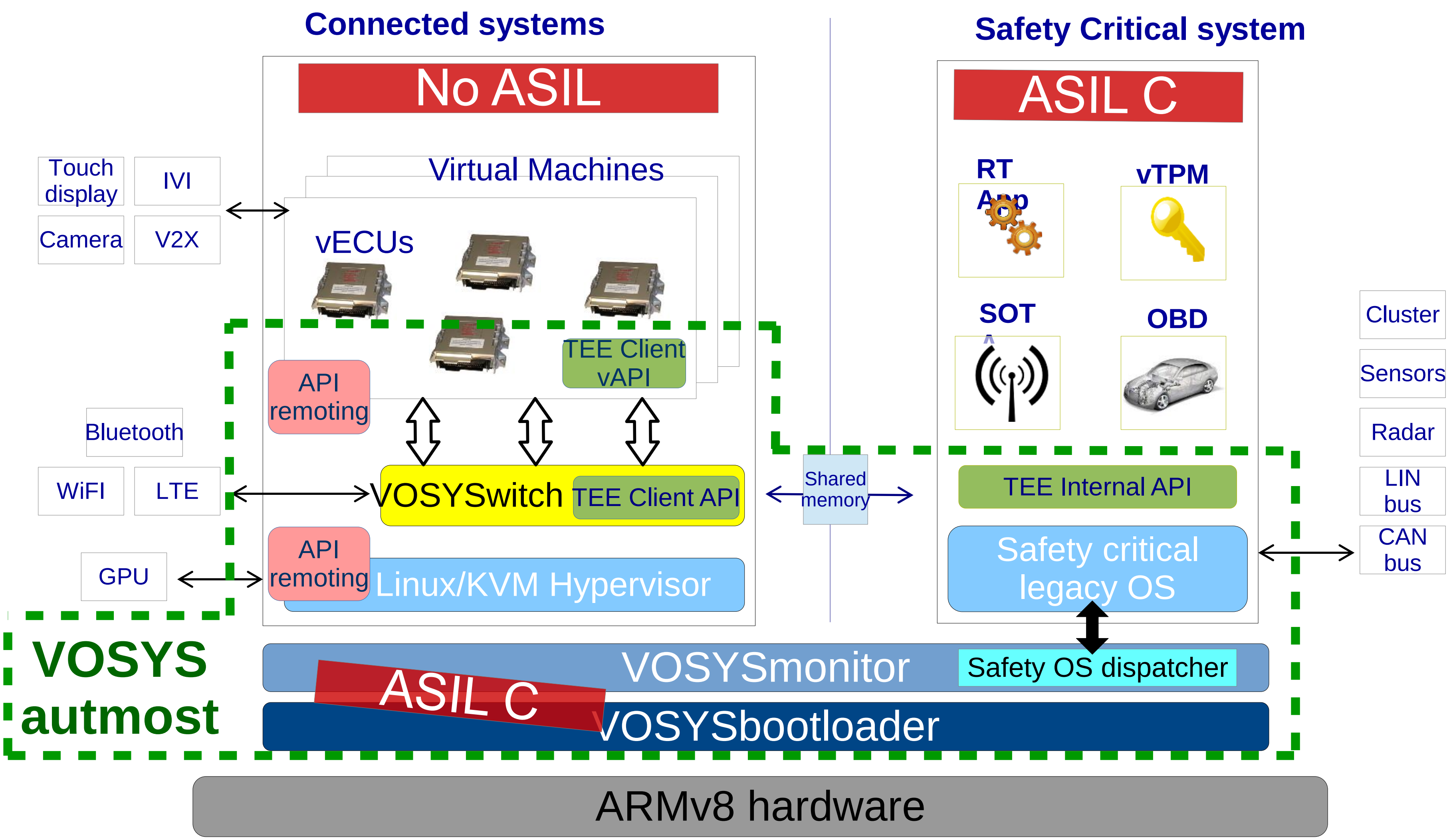
– Certifiable firmware running in secure EL3 mode
– Safety critical RTOS isolation using TrustZone
– Provide virtualization features for IVI systems
– Saves/restores PSCI state
– Controls ARMv8 hardware exception mechanisms such as interrupts (FIQ, IRQ), Normal (IRQ) as well as Aborts.
– Modular and scalable architecture

VOSYSmonitor features

The VOSYSmonitor design has been focused to meet the following requirements:

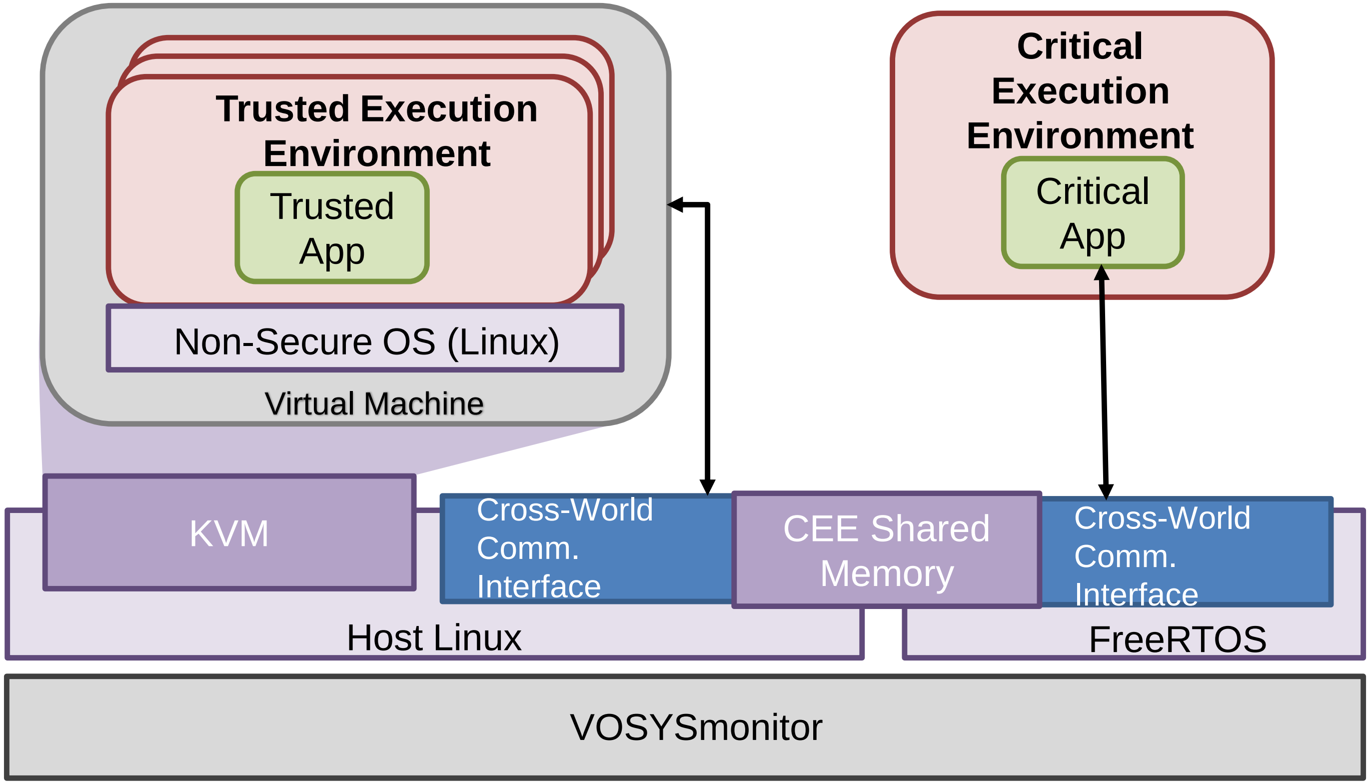
- Enable concurrent execution on the same hardware of an RTOS (critical applications) and a GPOS (KVM virtualization)
- Support complete RTOS resources (Memory, Peripherals, etc.) isolation from GPOS illegal access
- Complete RTOS boot in less than 60ms (VOSYSmonitor boot impact target is 1%)
- Minimize the interrupt latency impact – RTOS/GPOS Context switching time must be lower than 1us

Use case automotive example



The TAPPS approach

- Introduction of a monitor layer to the Virtualization concept
  - Definition of Execution Environments: Normal world (TEE and REE) and Secure world (CEE)



TAPPS’ challenges

- New challenges similar to Virtualization
  - Examples: zero-copy shared memory for inter-world communication and data passing
  - Resource partitioning/sharing

Shared memory: guest vs host

- Shared memory between host and guest
- Implemented through a QEMU device with corresponding Linux kernel driver
- Big chunk of memory exposed as MMIO register
- With some QEMU magic, this can be done without copies

Shared memory: guest vs Secure world

- Shared memory enables both worlds to access a common region of memory
- The implementation of a shared memory mechanism introduces some challenges that can be summed up with the following list:
  - Visibility of a given segment of memory to both worlds
  - Reachability of the shared memory (32bit vs 64bit)
  - Fixed location of the shared memory: the memory should not be moved nor swapped by any of the OSes
  - Coherency between cores with respect to the data written to the shared memory (MMU/Cache)

Shared memory: guest vs Secure world

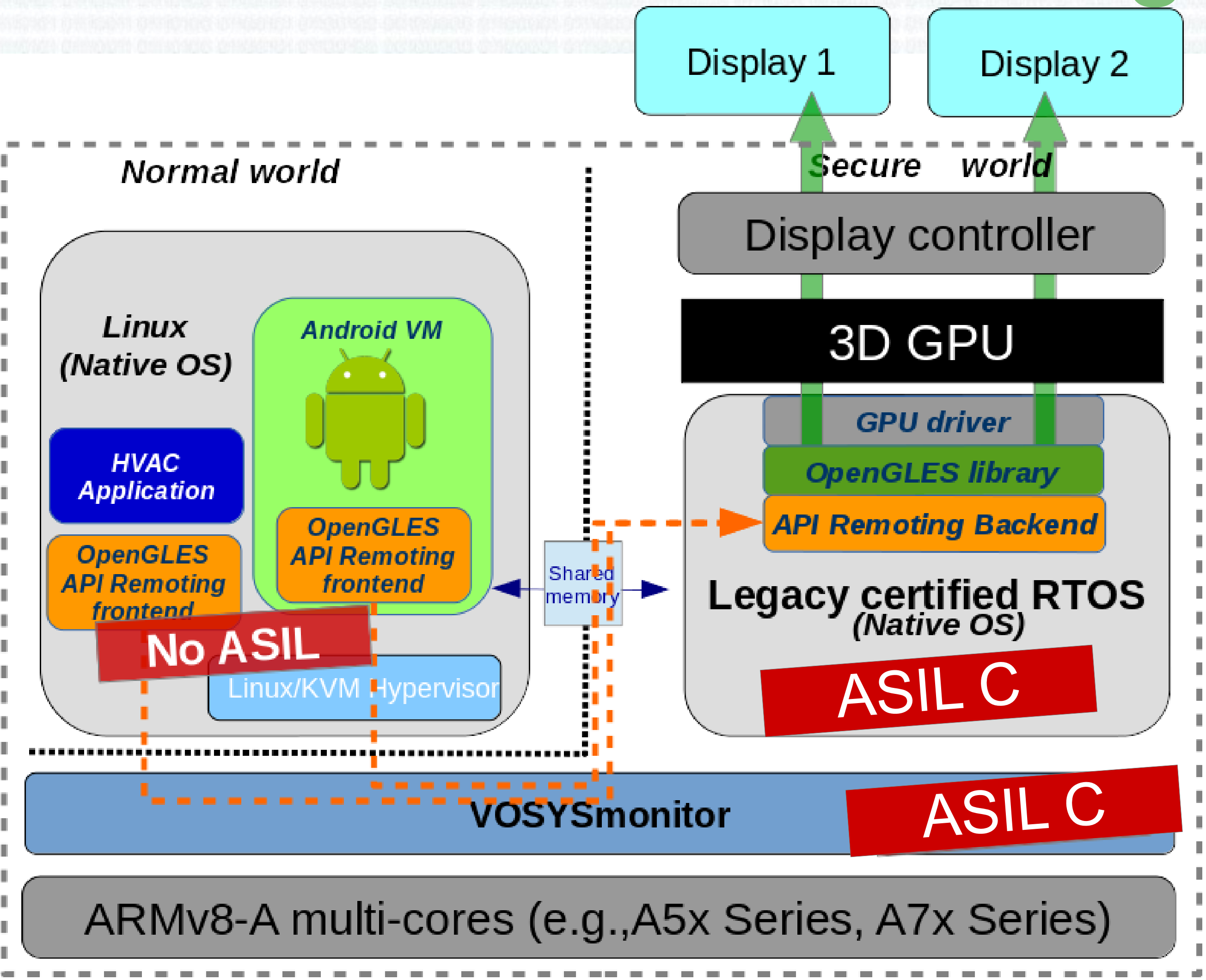
- In the context of TAPPS, a kernel driver has been implemented to setup the shared memory between Secure and Non-secure worlds
- QEMU allocates the shared memory and publish it to the Secure side using the kernel driver
- Finally, it exposes it to the guest, through the dedicated device

GPU sharing

In addition, the virtualization infrastructure developed during the TAPPS project allows for extensions in the direction of the GPU virtualization

- GPU virtualization unlocks interesting use cases for CPSs. For instance, in the automotive domain:
  - Display of emergency icons
  - Multiple displays, each handled by a VM (IVI system, dashboard, etc.)
- API remoting is a solution to share the GPU with VMs and the Real-time OS

GPU sharing



- A stub OpenGLES library is installed in both Linux host and Android guest
- It forwards the GL ES API calls to the backend running in the RTOS
- The backend links to the native OpenGL ES library and executes the calls in the RTOS
- The shared memory is leveraged as a transport medium between the front and backend

Virtual Open Systems products and services for mixed-critical CPS

TAPPS Exploitation directions from Virtual Open Systems developments:

- VOSYSmonitor: commercial ASIL C ARMv8 TrustZone monitor layer
- VOSYSmonitor integration services within consolidated IVI and cluster ECUs with RTOS and virtualized GPOS (AGL, etc.)
- Commercial software components and services related to GPU sharing, shared memory

Further TAPPS Dissemination enabler directions from Virtual Open Systems developments:

- Leading the effort of Linux Foundation AGL Virtualization work-group (EG-VIRT) - https://wiki.automotivelinux.org/eg-virt

The End

Partners of TAPPS

fortiss
ST life.augmented
TTTech
Virtual Open Systems
actility Making Things Smart
FCSR Fondazione Centro San Raffaele
T.E.I. of Crete
ENERGICA
crit SCOUTING YOUR NEXT TECHNOLOGY
OSPEDALE SAN RAFFAELE

Third parties

Contact

Alvise Rigo // Virtual Open Systems

a.rigo@virtualopensystems.com

TAPPS

Trusted **Apps** for open CPSs

Co-funded by the Horizon 2020 Framework Programme of the European Union under grant agreement no 645119