HW isolation for automotive environment

BoF

Michele Paolino
m.paolino@virtualopensystems.com

AGL All Member Meeting 2016,
2016-09-07, Munich, Germany

http://www.tapps-project.eu/
Michele Paolino, Virtualization R&D software architect at Virtual Open Systems. He presented “ARM TrustZone and KVM Coexistence with RTOS For Automotive” at the ALS summit 2015 (June 1st, Tokyo)

Virtual Open Systems is a high-tech start-up company active in open source virtualization solutions and custom services for complex mixed-criticality automotive. In July 2016 the company announced VOSYSmonitor 1.0, a high performance certifiable ARMv8 monitor implementation.

This work is done in the context of the H2020 Trusted APPs for CPS (TAPPS) project (www.tapps-project.eu).
Recently a new set of automotive platforms have been released. All of them have in common:

- Powerful CPUs (e.g., quad core ARMv8) and GPU (hundreds of cores)
- Several GBs of RAM memory
- Support for advanced technologies for security (e.g., TrustZone), virtualization (extended CPU ISA), multimedia, etc.

*How to use all this power?*
Virtualization can be used to provide HW isolation (memory, interrupts, etc) and to achieve Electronic Control Unit (ECU) functions consolidation and portability. This BoF aims at:

- Continuing the discussion started on the AGL ML to the define AGL virtualization requirements, architecture, etc.
- Finding contributors from OEM, software and car manufacturer members
- Proposing to create a new AGL EG for virtualization
Virtualization enables the execution of different operating systems concurrently.

➢ Security and easy third party application support
➢ ECU consolidation (reduced hardware and wiring costs)
➢ Ease of software updates
➢ Migration and portability
Full Virtualization is the ability of a system to run different partitions concurrently with unmodified software. VMs that exploit the CPU virtualization extensions (i.e. ARM VE, Intel VT), are hardware isolated regarding:

- Memory
- Interrupts
- Exceptions

Notable examples of open source full Virtualization solutions are KVM and XEN.
Virtualization means also a set of new challenges for AGL:

- Mixed criticality environment with RT requirements
- Certification
- Security and trustworthiness of the software (device sharing, etc.)
- High performance and hardware acceleration virtualization (object recognition, DRM encoding, 3d acceleration, etc.)
Proposed Requirements and Architecture

With the intent of fostering the discussion and find a solution agreed by the AGL community, a new architecture for the AGL virtualization has been proposed, focusing on the below requirements:

- Single RTOS and single non critical environment running concurrently
- Single RTOS and multiple non critical environments running concurrently
- Multiple RTOSes and multiple non critical environments running concurrently
The proposed architecture targets Intel and ARMv8 architectures, and leverages on:

- CPU Virtualization extensions (Intel VT or ARM VE)
- Security Extensions (ARM TrustZone)
TrustZone safely runs two OSes by defining a secure operational mode completely isolated from the rest of the system:

- The two OSes are fully independent
  - if the IVI part crashes, the safety critical OS runs normally
- TrustZone implements a secure context switch mechanism through the TrustZone Monitor

![Diagram of TrustZone architecture](image)
The Monitor firmware in ARM TrustZone is of pivotal importance. It implements:

- Secure world boot
- S-EL1 payload dispatcher
- Secure/Normal world isolation Initialization
- SMC (Secure monitor Call) Handling
- PSCI for secondary core bring-up

ARM Trusted Firmware (open source) and VOSYSmonitor (certifiable) are examples of such implementation.
Virtualization is not enough to efficiently isolate safety critical systems.

- VENOM, CVE-2015-3456, is a security vulnerability in the QEMU virtual floppy drive
- It allows an attacker to escape from the VM isolation (step 1)
- VENOM could open access to the host and all other VMs, potentially giving adversaries significant elevated access to the adjacent systems (step 2)
Proposed architecture: Single RTOS/non critical environment

<table>
<thead>
<tr>
<th>ARMv8</th>
<th>Intel x86</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pros</strong>: easier certifiability, native execution of both Linux/RTOS (no virtual drivers needed), no virtualization overhead, isolation stronger than a hypervisor</td>
<td><strong>Pros</strong>: certifiability, hypervisor isolation</td>
</tr>
<tr>
<td><strong>Cons</strong>: Hypervisor needed to run multiple OSes in EL1</td>
<td><strong>Cons</strong>: virtualization overhead, VMs can affect security of the RTOS</td>
</tr>
</tbody>
</table>

**ARMv8**
- TrustZone monitor
- EL0
- SEL0
- Linux/AGL
- RTOS
- EL1
- SEL1
- EL2
- EL3

**Intel x86**
- Hypervisor
- VMX root
- RTOS
- SEL0
- SEL1
- EL0
- EL1
- EL2
- EL3
- Linux/AGL
- Ring 0
- Ring 3
Proposed architecture: Single RTOS/multiple non critical environments

**ARMv8**

- **Pros**: Multiple OSes, if the hypervisor fails the RTOS is not affected, open source solutions available, no virt. overhead for RTOS
- **Cons**: virt. overhead for VMs, VMs can affect security of other VMs/hypervisor

**Intel x86**

- **Pros**: Multiple OSes, open source solutions available
- **Cons**: virt. overhead for VMs and RTOS, VMs can affect security of the RTOS, other VMs and of the Hypervisor
Proposed architecture: Multiple RTOSes/non-critical environments

**Pros:**
- Multiple RTOSes strongly isolated by non-critical services, short certification chain (Monitor+RTOS),
- Single point of failure in the Monitor (e.g., ATF is ~20K LOC)

**Cons:**
- Single point of failure in the hypervisor (e.g., XEN is ~100K LOC)
AGL has a long way to go, in order to lead the introduction of virtualization in automotive.

- Requirements and use cases definition
  - when is it needed? which are the use cases of interest? what is it needed?
- Analysis of the SoA technologies and open source projects (KVM, QEMU, XEN, libvirt, etc)
  - Type 1 vs Type 2, comparison with other technologies such as containers, TrustZone, etc.
- Identify the potential extensions from open-source upstream projects according to the identified requirements
- Extensions development and integration in AGL
Conclusion

With this BoF a new EG for AGL virtualization has been proposed. The objective is to find 5/10 engineers interested in contributing and participating to the discussion.

➢ Comment/propose changes in the wiki page https://wiki.automotivelinux.org/bof-hypervisor
➢ Submit your comments/questions by email in the AGL mailing list or through the #automotive IRC channel
➢ Contact me at m.paolino@virtualopensystems.com

If you are one of them, you can easily join the discussion:
Thank you!