Interaction between AUTOSAR and non-AUTOSAR Systems on top of a Hypervisor

Pierre-Antoine Bernard | 7th AUTOSAR Open Conference | Detroit, October 23rd 2014
Introduction

Pierre-Antoine Bernard
Senior Software Engineer

OpenSynergy is a global provider of software solutions for embedded automotive systems.

• Located in Germany (Berlin – Headquarter) and United States (American Fork, UT)
• Provider of a software platform based on a hypervisor (virtualization)
Interaction between AUTOSAR and non-AUTOSAR Systems on top of a Hypervisor
Pricing pressure requires multiple functions integrated on a single ECU.

Challenge: Software Integration

amount of AUTOSAR and non-AUTOSAR software-based functions

cost per function

number of ECUs

time

Virtualization

Pricing pressure requires multiple functions integrated on a single ECU
Major concepts have been introduced in AUTOSAR 4.x to address the challenge of software integration:

• Memory Partitioning
• Multi-core Architectures
• Enhanced BSW Allocation
• Dual MCU
AUTOSAR vs non-AUTOSAR

- Modular software architecture
- Scalability to different vehicle and platform variants
- Support of different functional domains (no infotainment)
- Support of applicable automotive international standards
- Fulfillment of strong security and safety requirements

- Modular infotainment architecture
- Reuse of available open source software components
- Robust development environment
- Highly customizable systems
- Challenge to meet safety and security automotive requirements
Virtualization extends the AUTOSAR concepts by making possible the integration of AUTOSAR and non-AUTOSAR systems on a single ECU and still satisfying automotive requirements:

- Real-Time
- Fast Boot
- Security
- Safety
- Certification
Virtualization – Use Case

Telematic Unit (TU)

GPS

Application Layer

RTE

BSW

Hypervisor

Multi-core SoC

CAN
Virtualization – Use Case

Infotainment Head Unit (HU)

Instrument Cluster (IC)

HU

IC

Application Layer

RTOS

OpenGL

RTE

BSW

Hypervisor

Multi-core SoC

CAN
The Hypervisor is an abstraction layer between hardware and virtual machines (VM) acting as the underlying technology for the virtualization technology. Dedicated CPU cores can be assigned to VMs or CPU cores can be shared between VMs (vCPU concept).

The Hypervisor is responsible for isolating the hardware resources between the VMs and for controlling access to I/O.
The Hypervisor runs in the highest **privileged CPU mode** and provides communication **primitives** for the communication between VMs.
Hypervisor – Communication Layers

<table>
<thead>
<tr>
<th>Communication Layer</th>
<th>Inter-VM Communication Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication Services</td>
<td>Virtual Ethernet Device (Connect multiple VMs to a virtual Ethernet network)</td>
</tr>
<tr>
<td>Communication Concepts</td>
<td>Virtual Device Service (Provide virtual device emulation to the VMs)</td>
</tr>
<tr>
<td>Hypervisor Communication Primitives</td>
<td>Synchronous Messaging (Hypercalls or System call to the hypervisor)</td>
</tr>
<tr>
<td></td>
<td>Shared Memory (Read/write access to shared memory area)</td>
</tr>
<tr>
<td></td>
<td>Communication Framework (Provide high level signal communication services)</td>
</tr>
<tr>
<td></td>
<td>Data Channel Service (Provide data communication channels between the VMs)</td>
</tr>
</tbody>
</table>
Virtual device emulation makes possible the **reuse** of a communication stack (CAN, Ethernet) within a VM to **ease** the software integration.
Example – CAN – SocketCAN

Linux

Application

Communication Stack

CAN Driver

Virtual CAN Bus

Application Layer

OS

RTE

Com

PduR

CanTP

CanIf

Can

CanTrcv

Hypervisor

Multi-core SoC

16
**Signal** based communication with **low data overhead** is required for an efficient communication (Automotive Communication Framework).

Diagram showing the integration of Linux, CDD library, Data Channel Driver, Application Layer, RTE, Com, PduR, CanTP, CanIf, Can, CanTrcv, Hypervisor, and Multi-core SoC/I/O components.
The scope of the IOC (Inter-OsApplication Communicator) module can be **extended** to the scope of **inter-VM** communication (vendor extension).
Thank you!
Contact

OpenSynergy GmbH
Rotherstraße 20
D-10245 Berlin
Germany
Phone +49 30 60 98 54 0-0
E-Mail info@opensynergy.com

OpenSynergy GmbH
Starnberger Str. 22
D-82131 Gauting / Munich
Germany
Phone +49 89 8934 13-33
E-Mail bluetooth@opensynergy.com

OpenSynergy, Inc. (USA)
765 East 340 South
Suite 106
American Fork, Utah 84003
Phone +1 619 96 21 725
E-Mail bluetooth@opensynergy.com