AUTOMOTIVE · RAILWAY · AVIONICS
MULTICORE SYSTEMS

aramis

Two Architecture Approaches for MILS Systems in Mobility Domains

HIPEAC-Conference, January 20, 2015
Daniel Adam (BMW R&T, Speaker), Sergey Tverdyshev (SYSGO), Timo Sandmann (KIT), Carsten Rolfes (Fraunhofer AISEC)

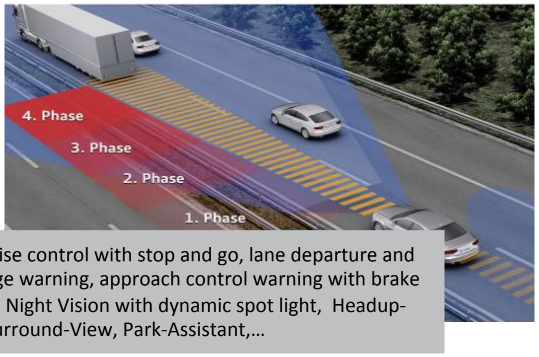
GEFÖRDERT VOM
Bundesministerium
für Bildung
und Forschung

ARAMIS - MAIN OBJECTIVES

- **Objective 1 - Common Solution across the domains:** The ARAMiS partners have common interest, to define the logical view on the safety critical embedded multicore computer architecture in a common form as much as possible.
- **Objective 2 - Use of standard-based modeling language**: A modeling language shall be used/defined that allows to understand the constructed architecture easily and allows the definition of design methods which can be easily followed in order to define the architecture at the system scope and to refine it in the disciplines of software and hardware.
- **Objective 3 - Consequence of the constraints due to the safety requirements**: A fault tree analysis must be performed on the defined system architecture, which results in safety requirements. These will impact the use of the multicore of the multicore computer. For example this will result in failure detection and management (redundancy management) functions and in the allocation of the safety levels of the defined system functions.
- **Objective 4 - Consequence of the constraints due to the security requirements:** A threat analysis must be performed on the defined system architecture, which results in security requirements. These will impact the use of the multicore of the multicore computer. For example this will result in functions that will detect the of compromise system security and in the allocation of the security levels of the defined system functions
- **Objective 5- Segregation of safety-critical functions grouped on a multi-core platform:** The segregation of functions grouped on a multicore platform must be addressed particularly. Many new aspects are arising and are influencing the design of the logical architecture. These are e.g. safety requirements, like the decision to group system functions on the cores, the avoidance of propagation of errors across cores and thus applications, the realization of redundancy and redundancy management of system functions, and the need of independence for specific functions or mechanisms of monitoring.
- **Objective 6 - Efficient Parallelization**: The technique and design for parallelization of application must be addressed to effectively gain significant performance by involving multicore platforms. This influences directly the decomposition and deployment strategies.
- **Objective 7: Concurrent access to common resources**: Solutions for the problem of the concurrent access of common resources and its consequences on the determinism must be provided. It must be particularly analyzed for race conditions, for the influence on the current communication strategies and patterns

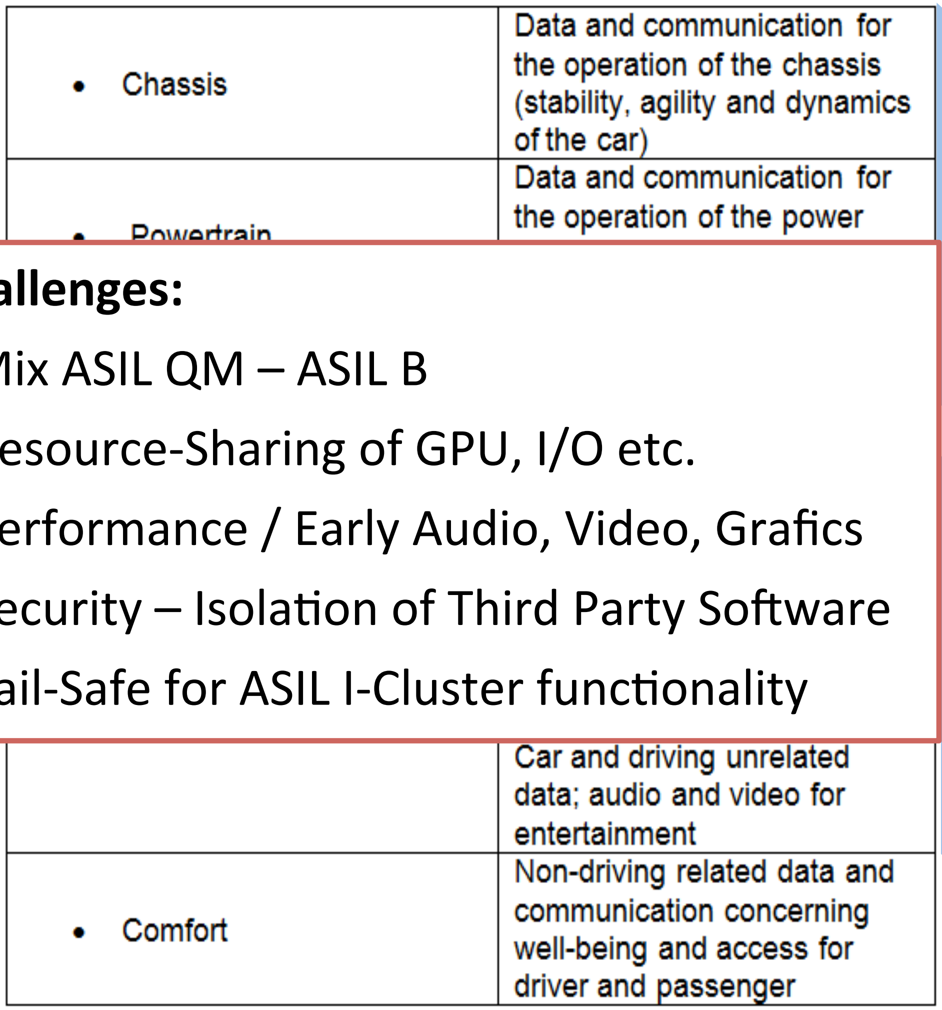
AUTOMOTIVE INDUSTRY TOPICS

- Environmental Protection and Energy-Efficiency
  - Increasing importance of emission reduction specially for congested areas result in complex engine and power management strategies (e.g. coordination of engine system hybrid vehicles, partial networking)
  - Legislative Process: Cost penalties when exceeding fleet consumption limits
  - Intermodal traffic management using networked vehicle information and infrastructure components which serves as a data collector. In the future intermodal transport scenarios are possible by proposing alternative cross domain transport solutions to reach the target destination I time.
- Active Safety and reduction of traffic accidents
  - Further reduction of traffic accidents results in an increasing number of assistance systems to control the vehicle and support the driver
  - In the future use adhoc Car2Car, Car2Infrastructure or Car2Backend networks for exchanging safety critical information
- Information and Communication (“Infotainment”)
  - Networking the vehicle with backend systems (“Cloud”) to improve driving experience and enable the availability of personalized data and services for a seamless living environment “atHome”, “atWork”, “atVacation”
  - Integration of mobile devices



Active Cruise control with stop and go, lane departure and lane change warning, approach control warning with brake activation, Night Vision with dynamic spot light, Headup-Display, Surround-View, Park-Assistant,…

AUTOMOTIVE DOMAIN CHARACTERISTICS



• Chassis	Data and communication for the operation of the chassis (stability, agility and dynamics of the car)
• Powertrain	Data and communication for the operation of the power
	Car and driving unrelated data; audio and video for entertainment
• Comfort	Non-driving related data and communication concerning well-being and access for driver and passenger

Challenges:

- Mix ASIL QM – ASIL B
- Resource-Sharing of GPU, I/O etc.
- Performance / Early Audio, Video, Grafics
- Security – Isolation of Third Party Software
- Fail-Safe for ASIL I-Cluster functionality

Runtime Environment: AUTOSAR 4.x

Static Configuration

Safety: ISO 26262 ASIL QM-ASIL D

Hard Realtime Requirements

Different Suppliers

Runtime Environment: Different GPOS, RTOS

Dynamic Configuration

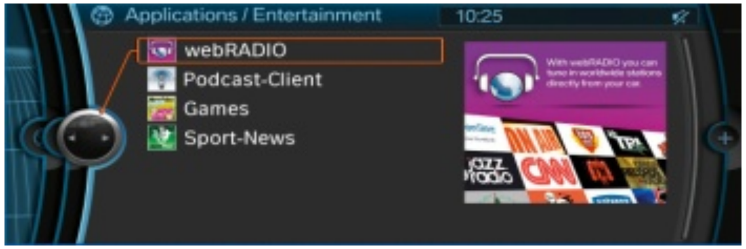
Safety: ISO 26262 ASIL QM-ASIL B

Security Requirements

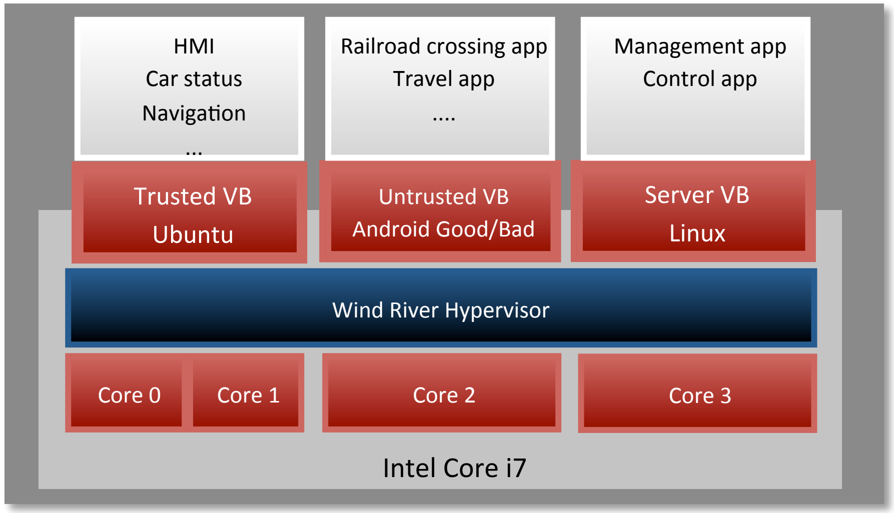
Early Audio, Video

Different Suppliers

VIRTUALIZED CAR TELEMATICS (VCT) DEMONSTRATOR - GOALS



- Segregation: isolation of applications of different safety- or security-levels (MILS)
- Virtualization as key technology to use multicore platforms in embedded systems
- Centralization / consolidation of functions into infotainment domain unit
- Re-use of existing software



HMI
Car status
Navigation
...

Railroad crossing app
Travel app
....

Management app
Control app

Trusted VB
Ubuntu

Untrusted VB
Android Good/Bad

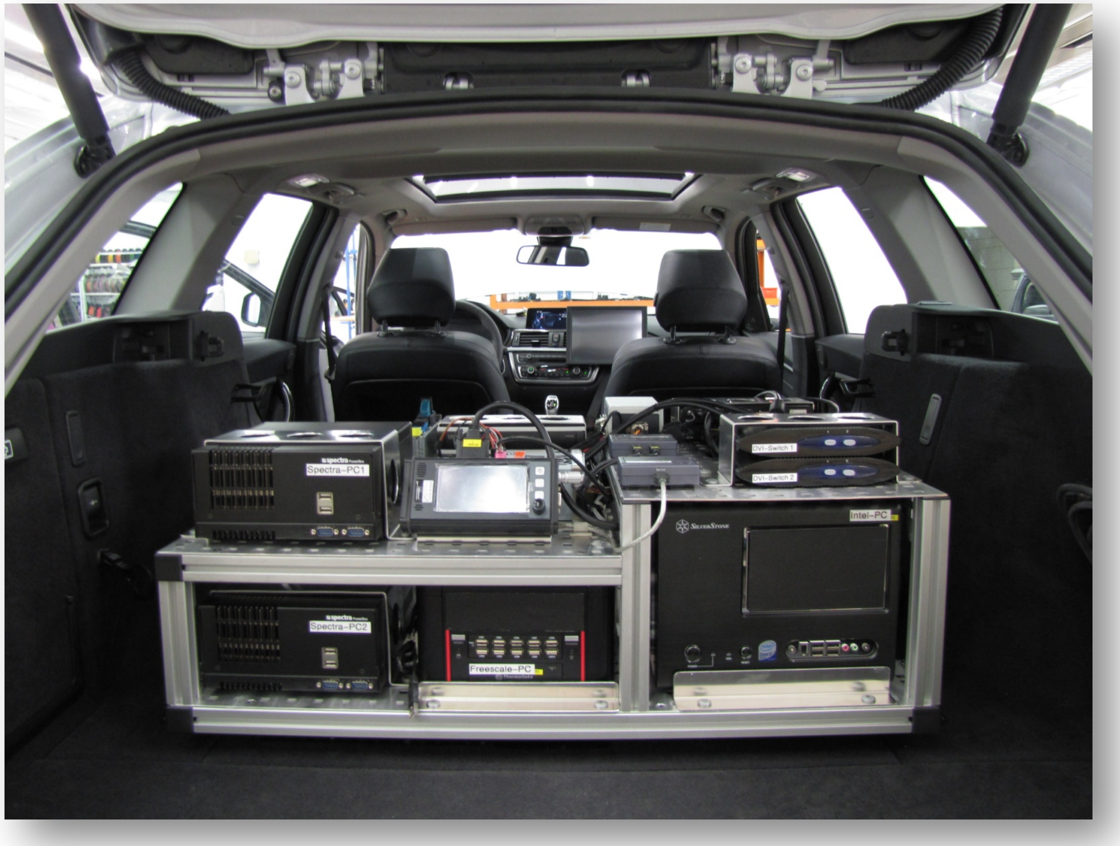
Server VB
Linux

Wind River Hypervisor

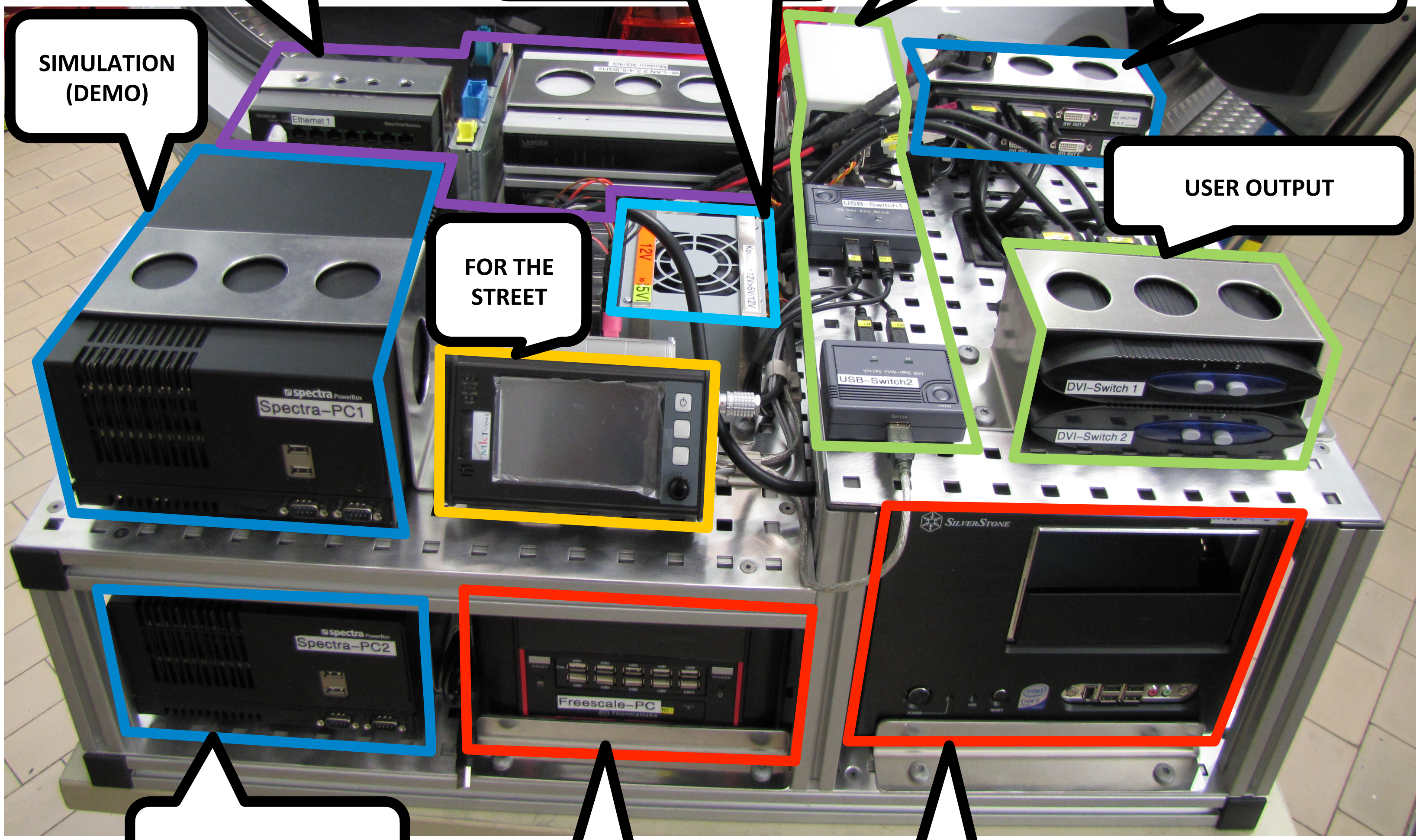
Core 0
Core 1
Core 2
Core 3

Intel Core i7

VIRTUALIZED CAR TELEMATICS (VCT) DEMONSTRATOR



A DEEP FOCUS ON THE DEMONSTRATOR



INTERNAL/EXTERNAL COMMUNICATION

POWER SUPPLY

USER INPUT

PRESENTATION (DEMO)

SIMULATION (DEMO)

USER OUTPUT

FOR THE STREET

VISUALIZATION (DEMO)

PLATFORM B

PLATFORM A

BOTH PLATFORMS AT A GLANCE

	Platform A	Platform B
Main Goal	▪ Isolation and resource sharing for applications of **different safety- or security**-levels	▪ To implement a **hardware based**, **low cost** multi-context TPM that is capable of serving virtualized machines running on a multicore CPU architecture.
Focus	CoProcessor	Security on Multicore without special hardware
Hypervisor	Wind River	SYSGO
Hardware	Intel i7 + Xilinx FPGA	i.MX 6 + Xilinx FPGA

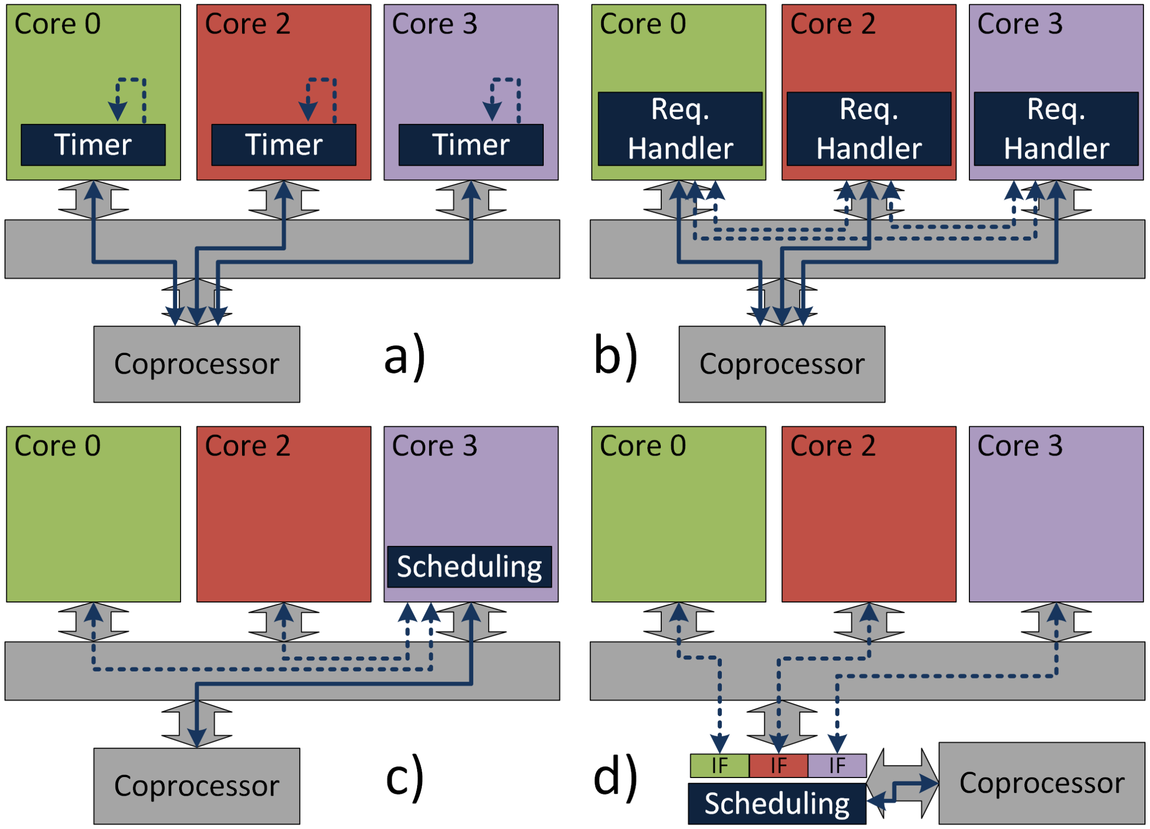
PLATFORM A

GOALS OF PLATFORM A

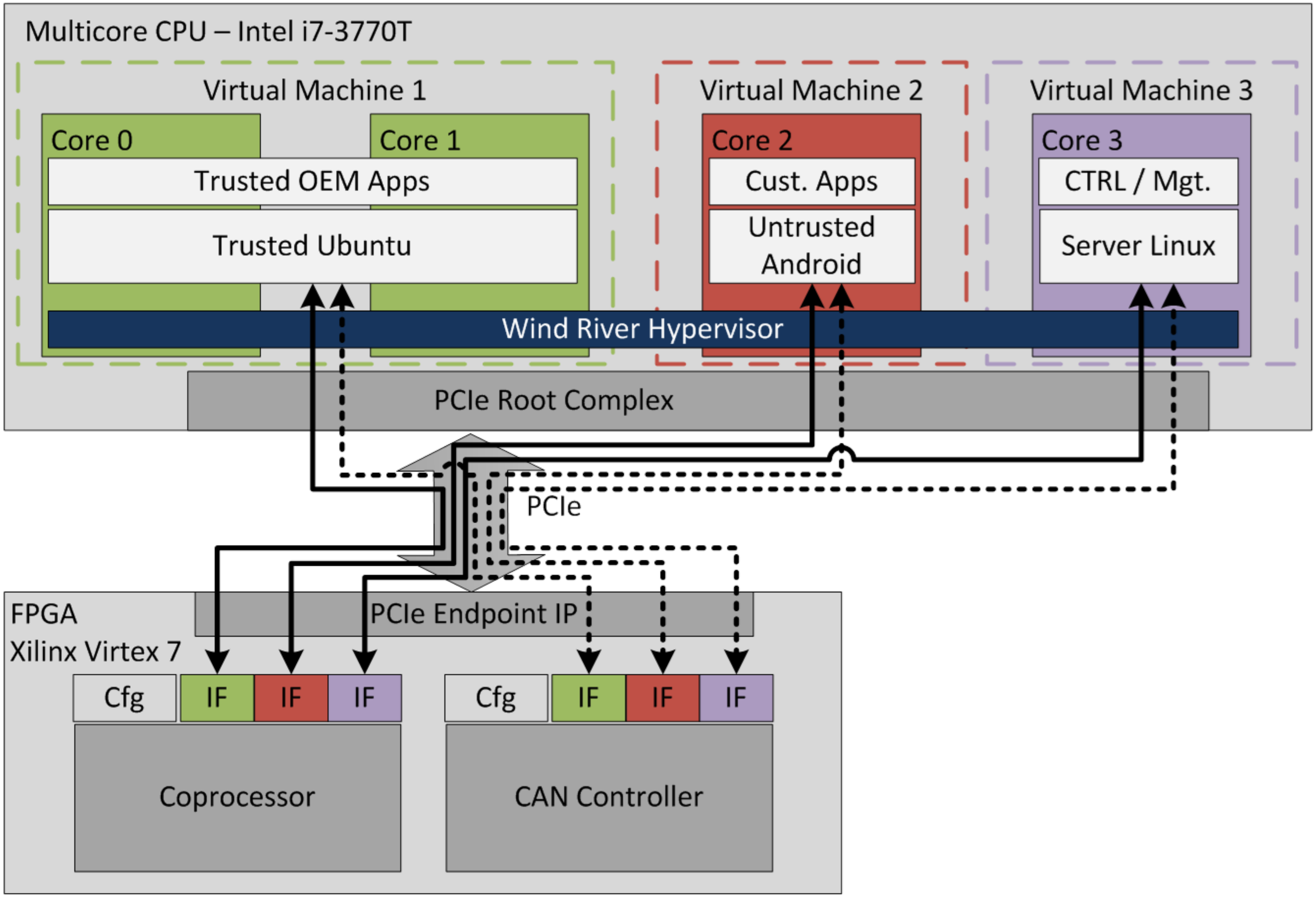
- Isolation and resource sharing for applications of **different safety- or security-levels**
- **Dynamic mapping** of user-oriented 3D-graphics on combi-display / headunit
- **Dynamic relocation** of content depending on vehicle status
- Usage of android-apps by providing of a **segregated partition for „insecure“** applications

SHARED COPROCESSORS IN MULTICORE SYSTEMS

- Resource sharing mechanisms
  - a) time-based
  - b) request-based / cooperative
  - c) proxy partition / hypervisor
  - d) hardware scheduling, transparent for partitions
- Requirements in safety-critical systems
  - efficient usage of multicore architecture
  - different priorities of partitions
  - predictability of behavior at concurrent accesses
  - quality-of-service assertions
  - portability to different multicore architectures



VCT COMPUTER ARCHITECTURE



PLATFORM B

GOALS OF PLATFORM B

- To implement a **hardware based**, **low cost** multi-context TPM that is capable of serving virtualized machines running on a multicore CPU architecture.

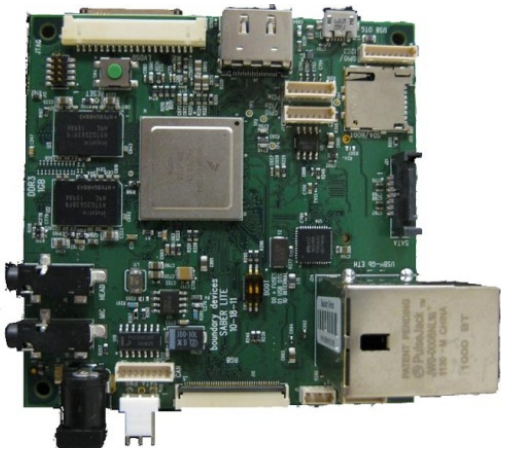
- Virtual Machine Manager – an interface between TPM and application processors.

- Tasks of a VMM
  - Secure context switching
  - Scheduling the TPM
  - Part of the trusted software stack which is verified using trusted boot

- Developing a Demonstration Setup (**Multi-context TPM ⬄ PikeOS**) dedicated to automotive COTS hardware

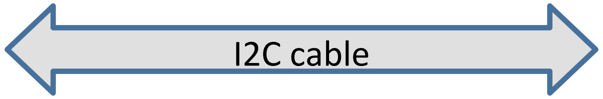
DEMONSTRATOR SETUP

Hypervisor/ VMs



I2C master

I2C slave



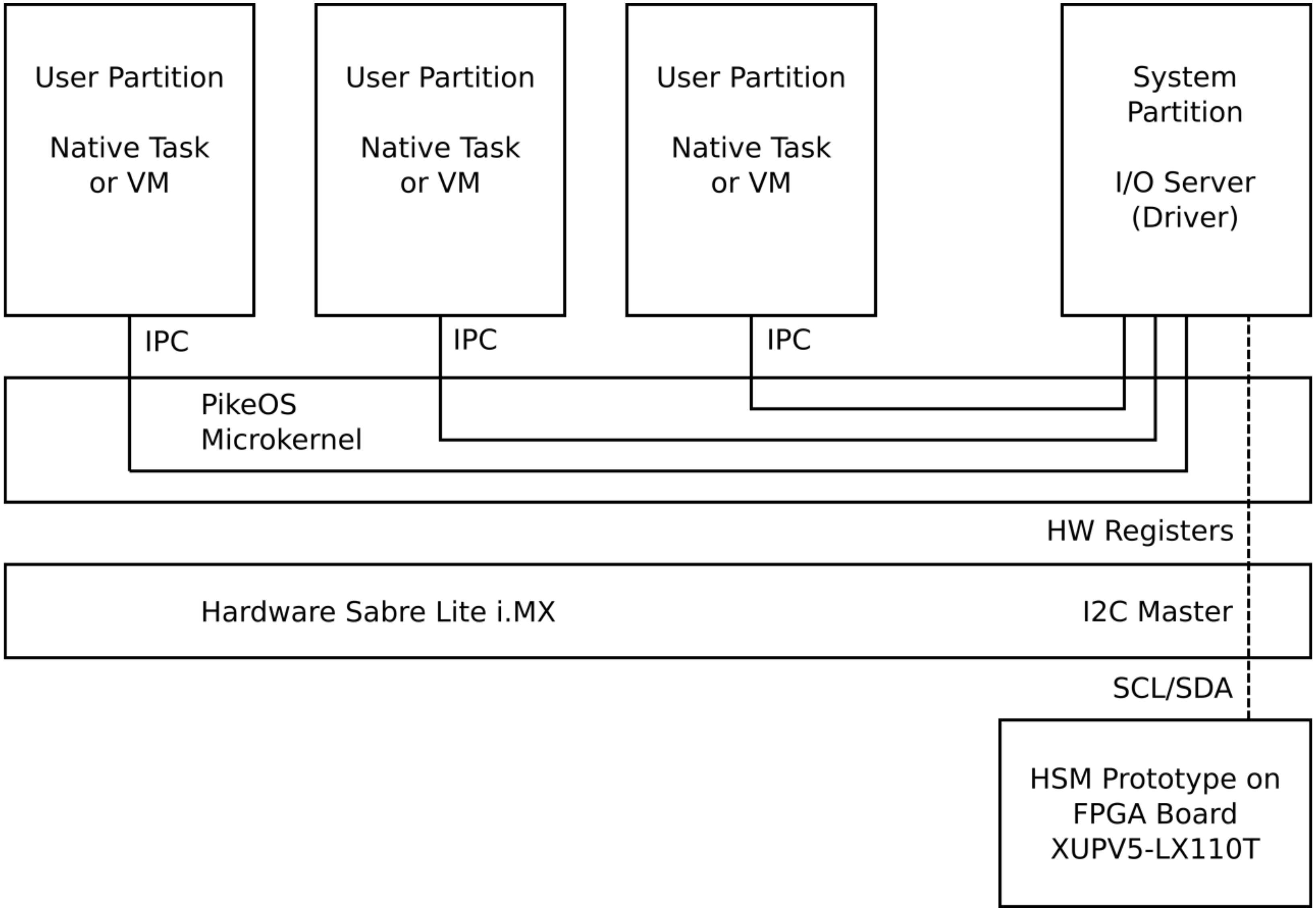
I2C cable

HSM

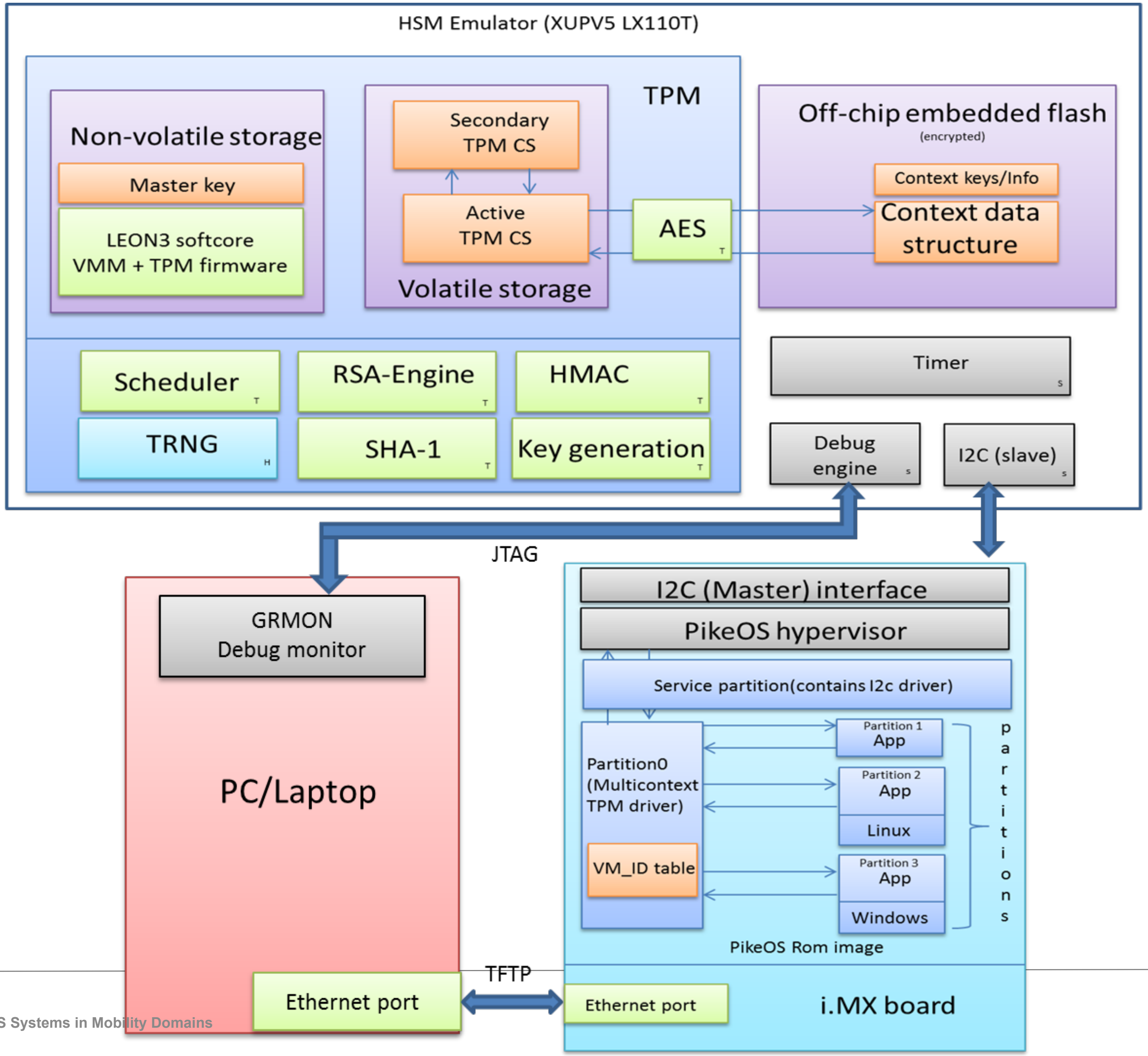
- i.MX6 quadcore
- Virtualized System: PikeOS Partitions running OS

- Xilinx Virtex5 FPGA
- LEON softcore with VMM + TPM 1.2
- H/W Accelerators(TRNG)

SYSTEM MILS ARCHITECTURE WITH PROXY



SYSTEM ARCHITECTURE (MILS): FOCUS TPM



HSM Emulator (XUPV5 LX110T)

TPM

Non-volatile storage

Master key

LEON3 softcore VMM + TPM firmware

Secondary TPM CS

Active TPM CS

Volatile storage

AES

Off-chip embedded flash (encrypted)

Context keys/Info

Context data structure

Scheduler

RSA-Engine

HMAC

TRNG

SHA-1

Key generation

Timer

Debug engine

I2C (slave)

JTAG

PC/Laptop

GRMON Debug monitor

Ethernet port

TFTP

I2C (Master) interface

PikeOS hypervisor

Service partition(contains I2c driver)

Partition0 (Multicontext TPM driver)

VM_ID table

Partition 1 App

Partition 2 App

Linux

Partition 3 App

Windows

partitions

PikeOS Rom image

Ethernet port

i.MX board

CONCLUSION

- Successful integration **multi-context TPM ⬄ PikeOS**
  - No PikeOS Changes
  - Only modification on the TrouSerS lib to support multi context TPMs

- **Future Work**
  - Monitoring the number of writes in flash
  - Fast flashes for storing context data
  - Implement cryptographic modules of TPM 1.2 emulator in hardware
  - Master key to be stored in a shielded region of the on-chip ROM

BACK

GENERELL SUMMARY

- **MILS Systems will be part of future Automotive ECUs**
  - Increased computing power with better energy-efficiency
  - Support for centralization and more degrees of freedom for new E/E-architectural approaches
  - Automated Driving will increase the number of high-Peformance functionality
  - Increased reliability separating functions on cores
  - Increases safety supporting ASIL-decomposition
  - Enable virtualization scenarios to support scalability

ARAMiS will focus on the challenges looking for comprehensive solutions for Automotive, Avionics and Railway.

BACKUP

POTENTIAL USE-CASES

- Software activation/Electronic Payment

  Establish a secure connection (SSL or TLS)

  Secure data exchange between users and merchant (vehicle <-> OEM)

  Manage user credentials related to payment account

  Certificate for software license has to be issued

- Network Attestation

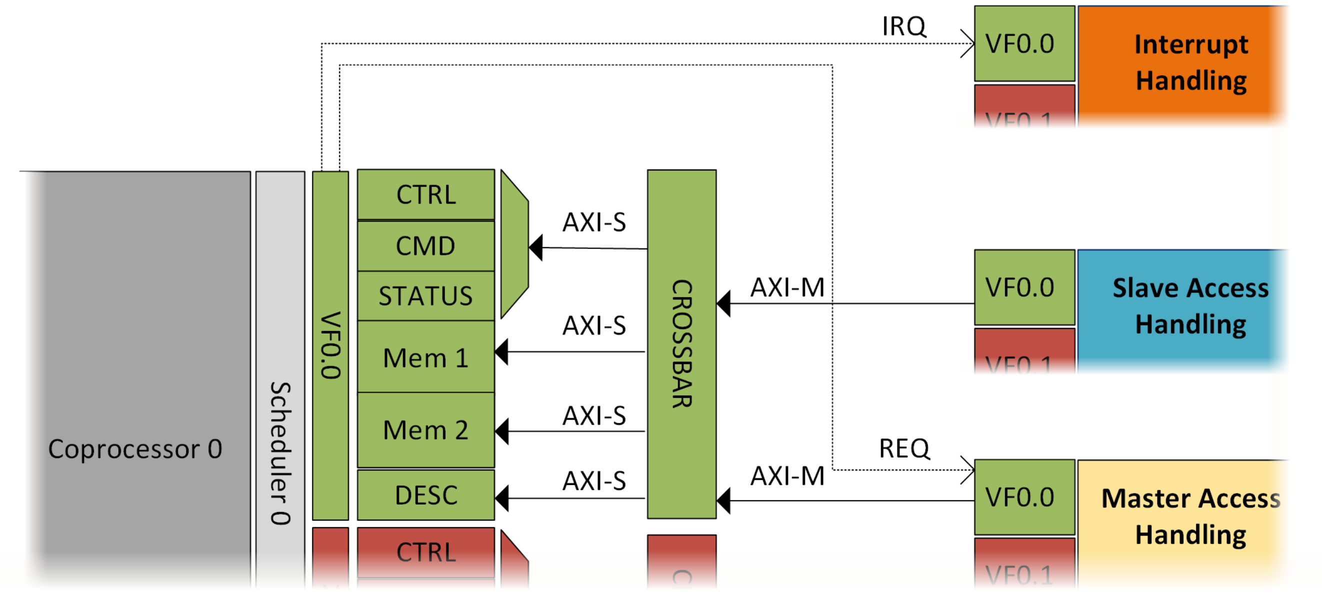
  Only platforms owned by enterprise are allowed to access network

  Platform configuration of client verified (vehicle is in trusted state)

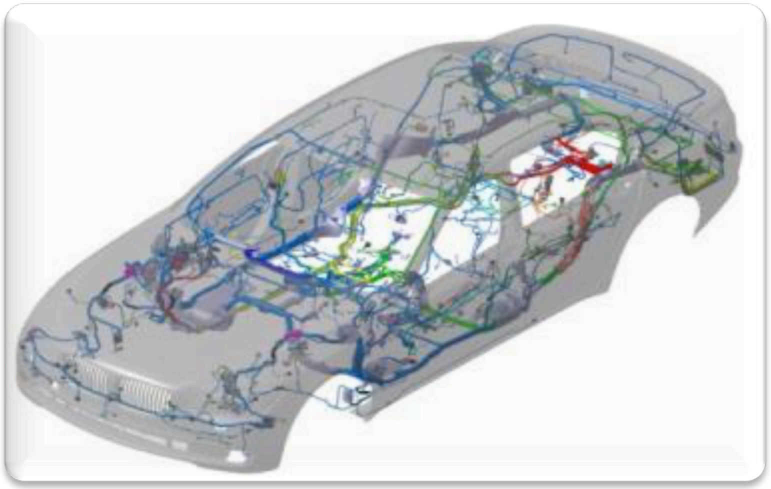
  Access granted to use network

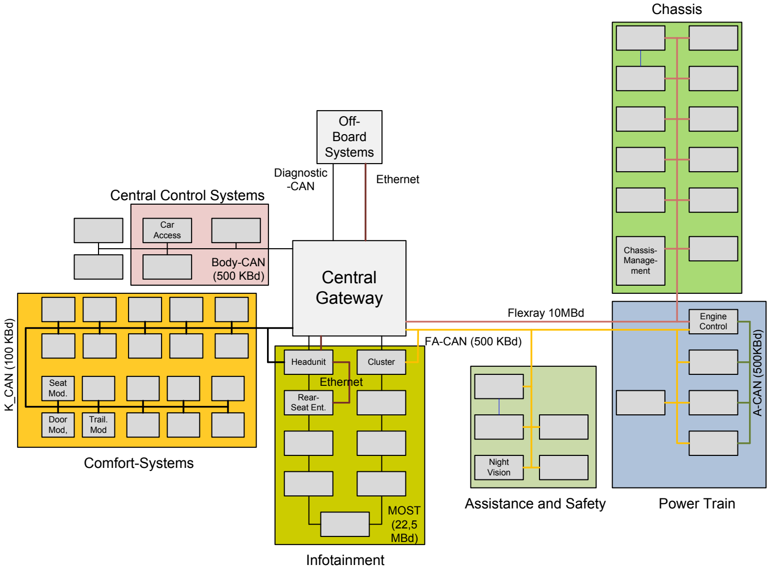
VIRTUALIZED COPROCESSOR INTERFACE

- Generic interface architecture for shared coprocessors
  - PCIe connection for Virtex-7 FPGAs, PCIe SR-IOV compatible
  - Support for slave- and DMA-accesses
  - Interrupt handling
  - Porting to further platforms (Zynq) currently in progress
- Virtual interfaces to realize spatial segregation
- Scheduling modules enforce temporal segregation



AUTOMOTIVE E/E-ARCHITECTURE TODAY





7 series wiring harness

Length	2751 m
Connecting Plugs	520
Weight	43 kg

Electronic Control Units (ECU)

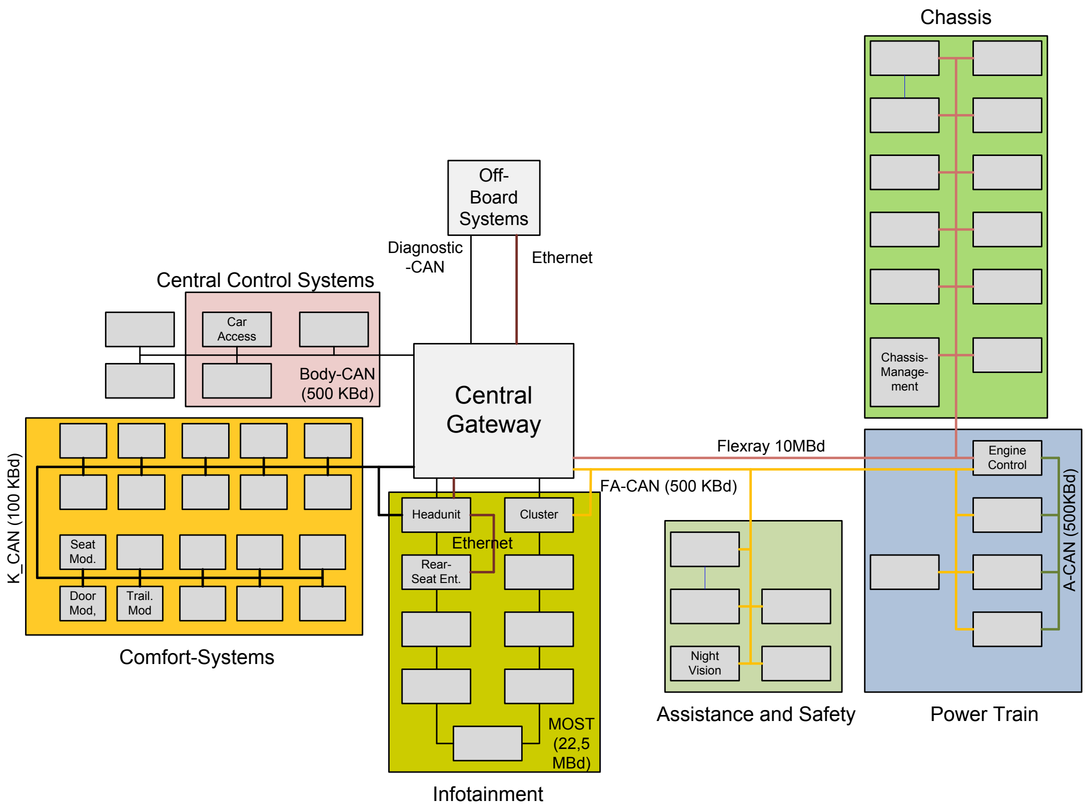
No. ECUs	28 … 74
CPUs	ca. 230
GPUs	> 5
Power PCs	3
Busse	CAN, LIN, MOST*, Flex Ray, Ethernet

Other

Software	3-5 GB Application 15-20 GB Data

AUTOMOTIVE E/E-ARCHITECTURE TODAY

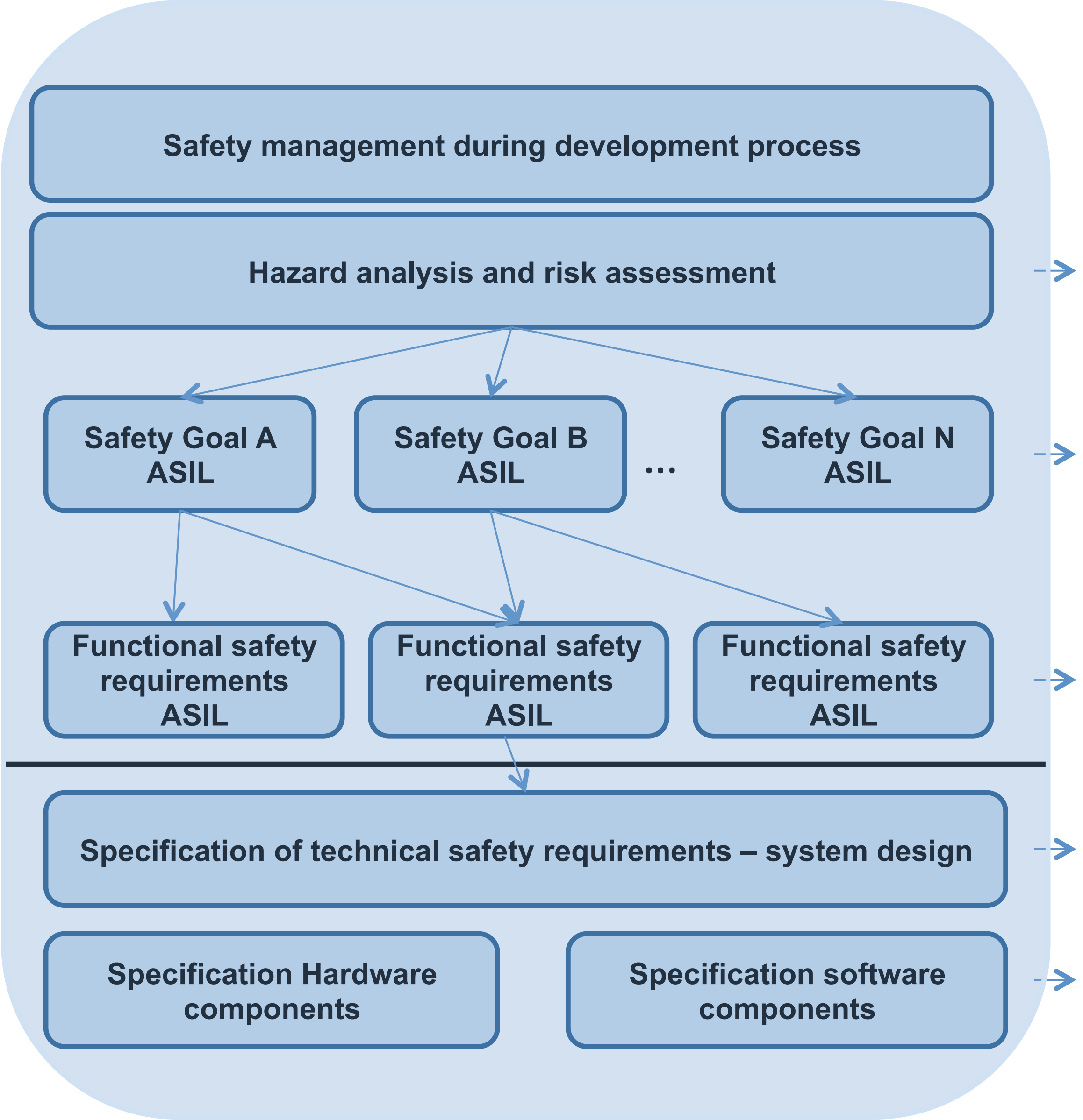
Typical logical bus-topology without LIN-Subsystems



• Chassis	Data and communication for the operation of the chassis (stability, agility and dynamics of the car)
• Powertrain	Data and communication for the operation of the power train (engine, gearbox, etc.)
• Driver Assistance	Autonomously operating data and communication (without user intervention) supporting the driving situation
• Infotainment	Data and communication interacting with the driver concerning the operation and driving situation of the car Route and traffic-related Information Car and driving unrelated data; audio and video for entertainment
• Comfort	Non-driving related data and communication concerning well-being and access for driver and passenger

Typical classification of automotive domains

ISO 26262 – FUNCTIONAL SAFETY IN AUTOMOTIVE


Safety management during development process
Hazard analysis and risk assessment
Safety Goal A ASIL
Safety Goal B ASIL
…
Safety Goal N ASIL
Functional safety requirements ASIL
Functional safety requirements ASIL
Functional safety requirements ASIL
Specification of technical safety requirements – system design
Specification Hardware components
Specification software components

Identify and categorize hazard = f (vehicle state, road conditions, environmental conditions, ..)
ASIL = f (severity, probability of exposure, controllability)
ASIL QM … ASIL D, ASIL D highest safety integrity level

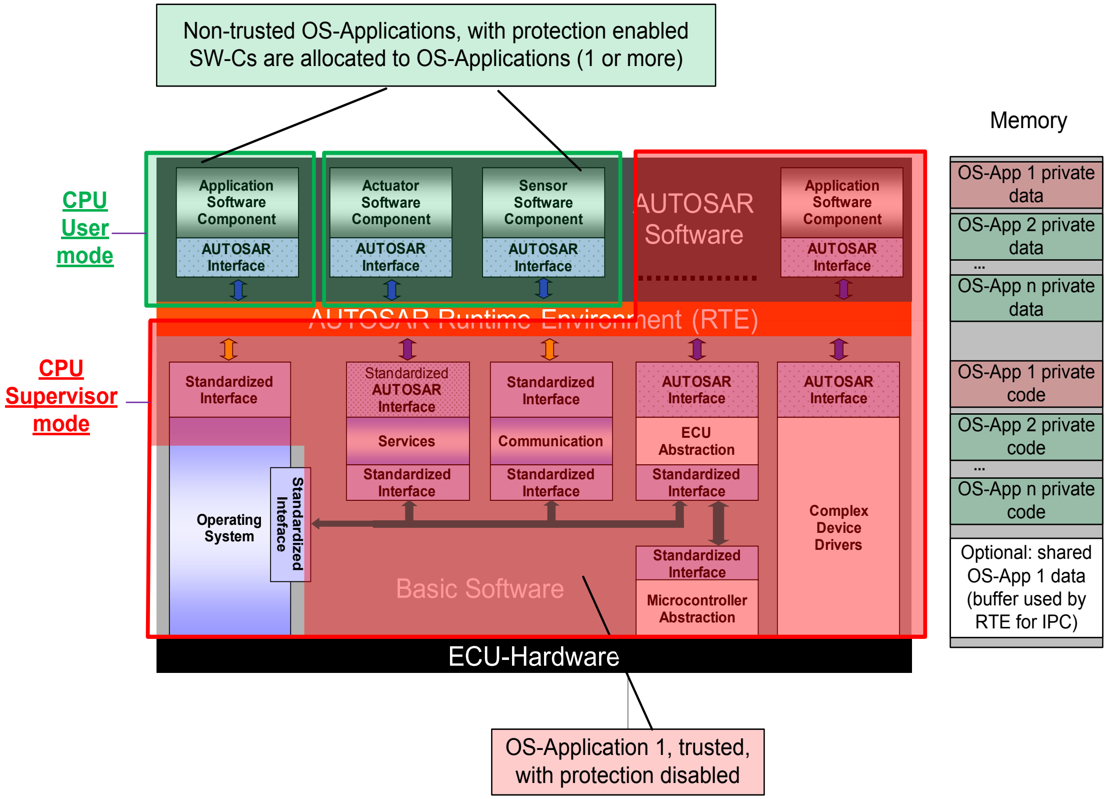
Determine safety goal for each hazard event with ASIL evaluated in hazard analysis

Contain Safety measures fulfill safety goals including safety mechanism like monitoring system, fault detection and driver warning, safe state transition etc.

Technical: System Level

Technical: Component Level

NON-INFOTAINMENT SOFTWARE-PLATTFORM: AUTOSAR



Non-trusted OS-Applications, with protection enabled
SW-Cs are allocated to OS-Applications (1 or more)

OS-Application 1, trusted,
with protection disabled

- Safety Features in AUTOSAR
  - Memory Protection:

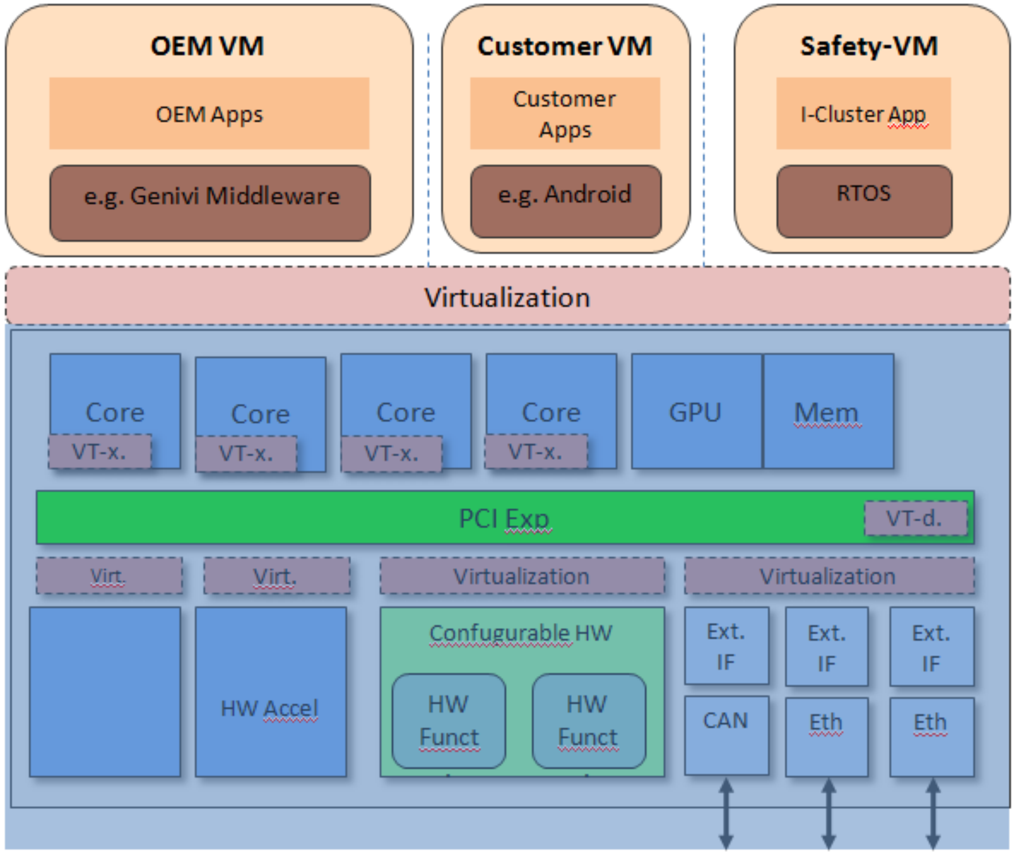
    Separate SW-applications in “OS applications” (trusted, untrusted) – support from MMU/MPU
  - Timing Determinism Features:

    Execution time monitoring, synchronized time base, means for synchronized execution
  - End-to-End Protection Library:

    Data protection
  - Program Flow Monitoring:

    Controls the temporal and logical behavior of applications.

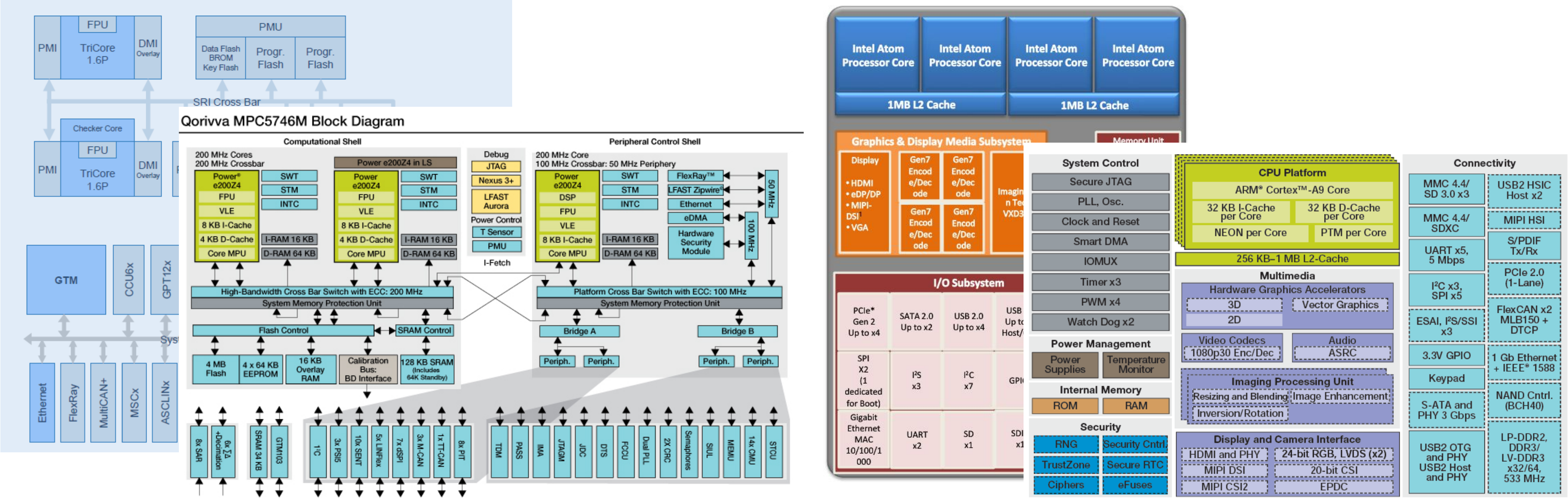
INFOTAINMENT HEADUNIT SOFTWARE-PLATTFORM



Example for an infotainment headunit characteristics

- Partitioning of Headunit regarding criticality of functions:
  - OEM-VM: Qualified OEM Apps.
  - Customer-VM: Standard OS with 3rd-party Software without validation
  - Safety-VM: Apps with safety or timing requirements.

TYPICAL HARDWARE-PLATTFORM

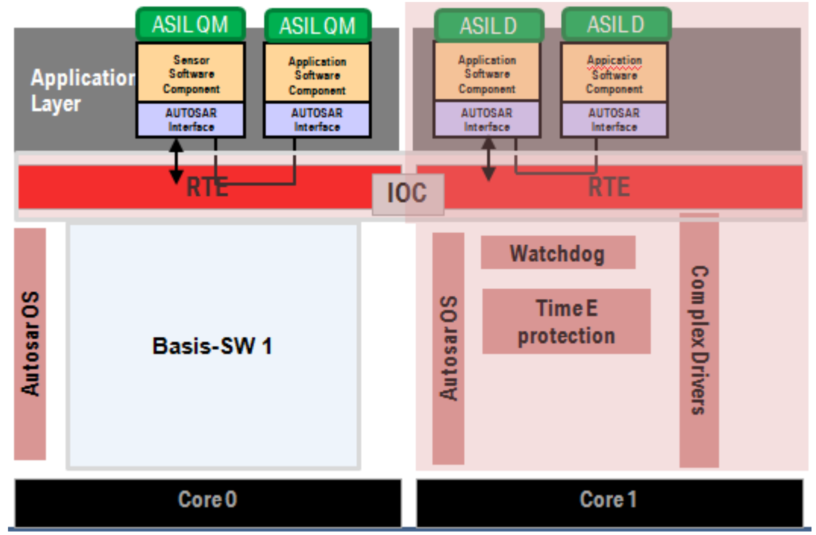


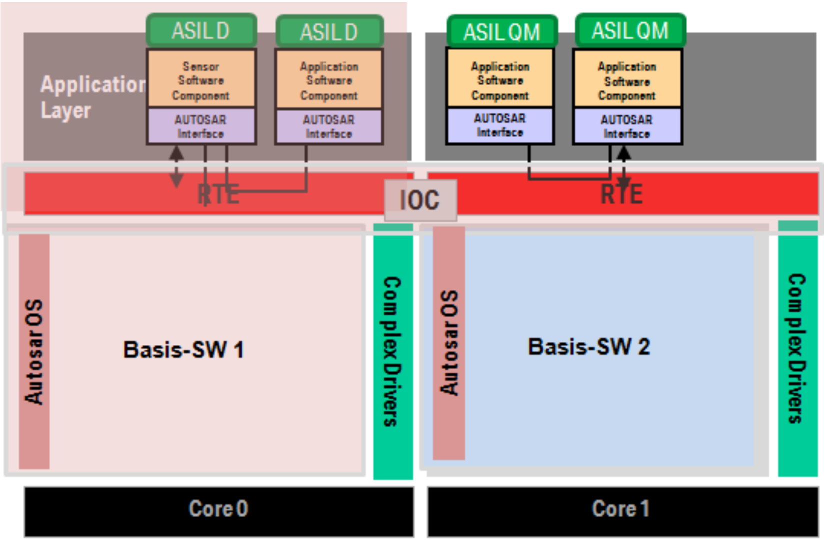
Different multicore architecture are of interest for different automotive domains…

➡

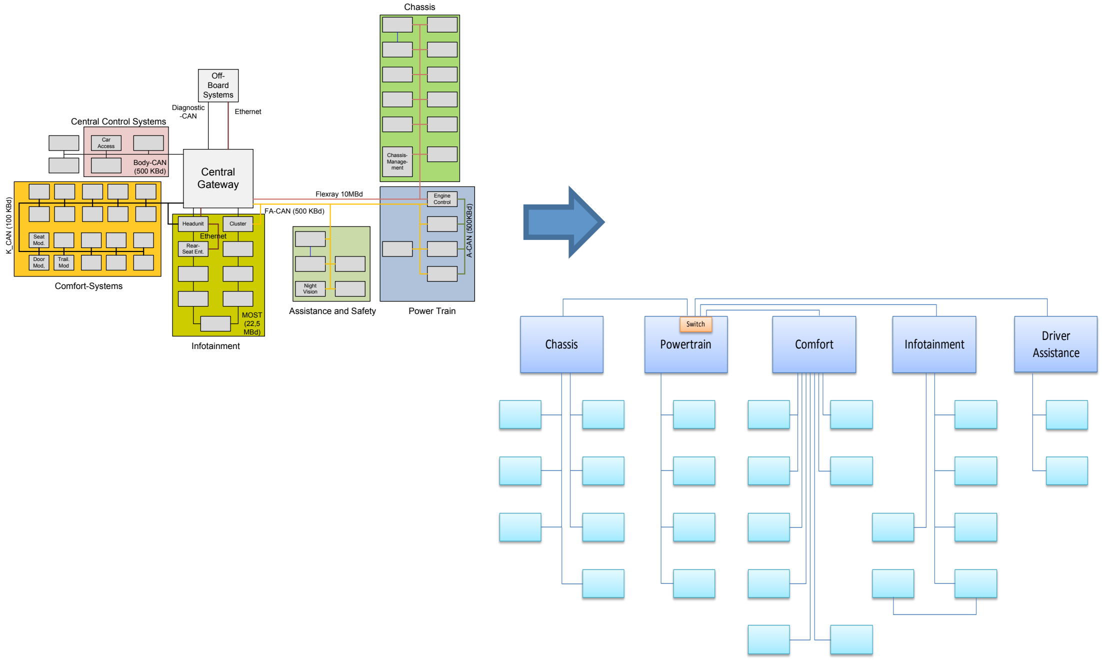
A particular software platform or multicore hardware-design has great influence on the whole system characteristics.
Based on the objectives ARAMiS will focus on the mapping process of logical architecture suggestions to technical solutions (SW, HW) under the conditions of existing designs and domain (Avionic, Automotive) requirements.

MULTICORE AUTOSAR USAGE SCENARIOS



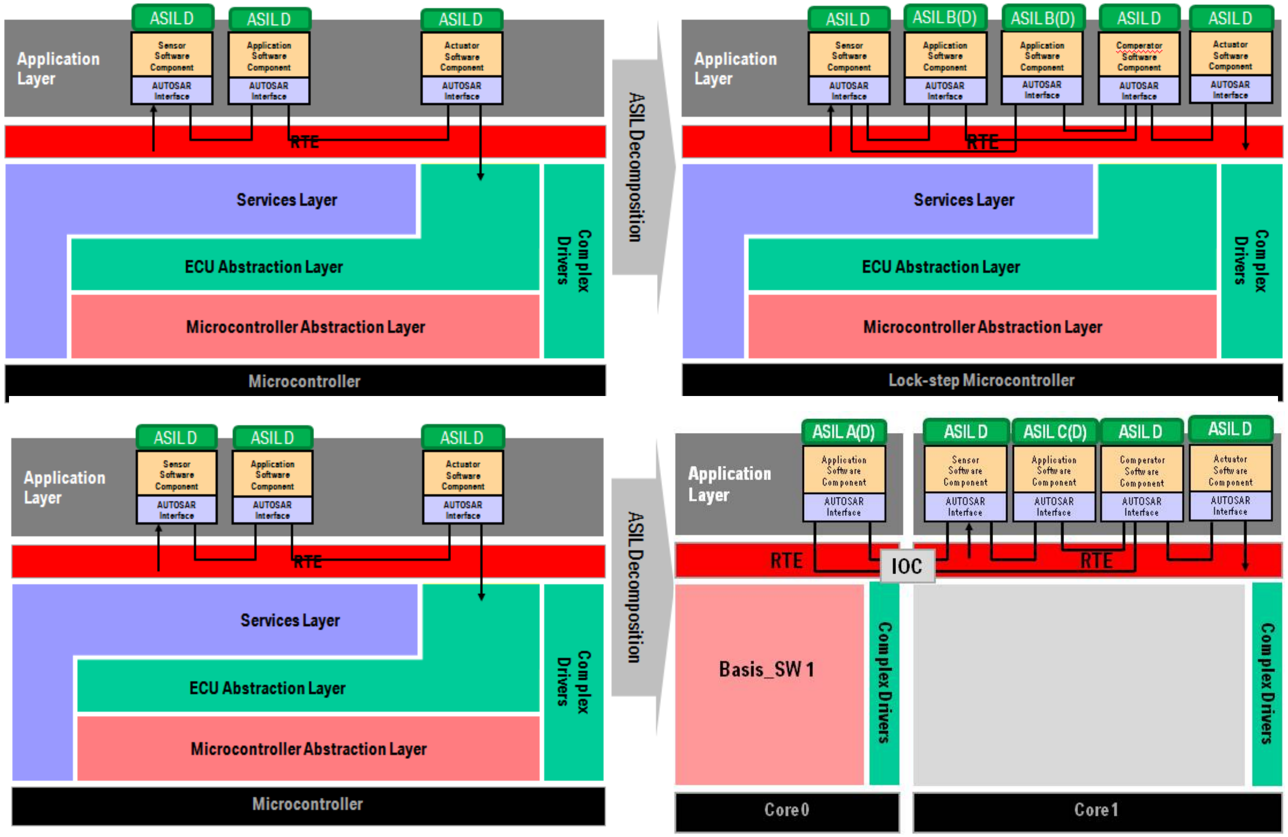


Usage scenario	• **Centralization**
Goal Achieved	• Reduce number of ECUs • Decrease number of networks and bussystems • Reduce complexity of networked functionality through domain specific functional centralization • Increase Safety via Core separation



Vision: „Domain-Controller“

MULTICORE AUTOSAR USAGE SCENARIOS

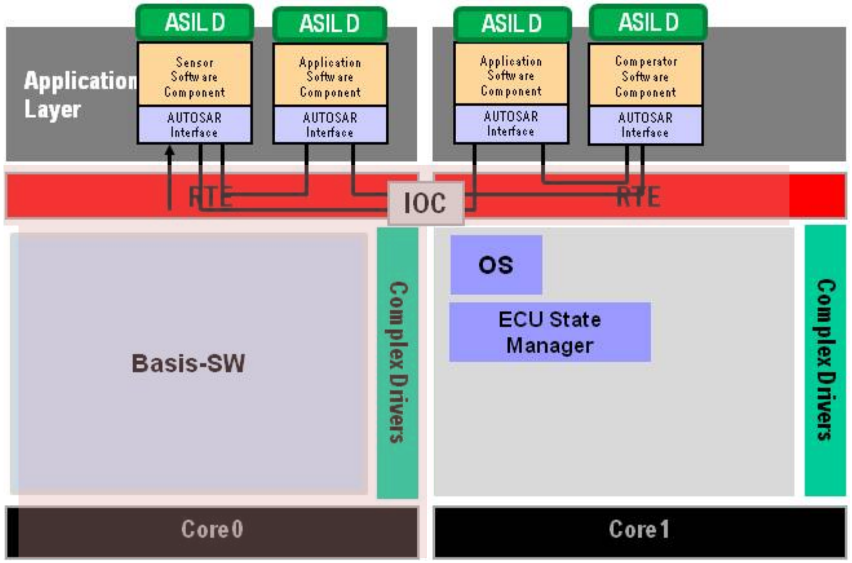


- Inheritance rules for integrity levels lead to a spread of the high integrity levels on the whole physical network

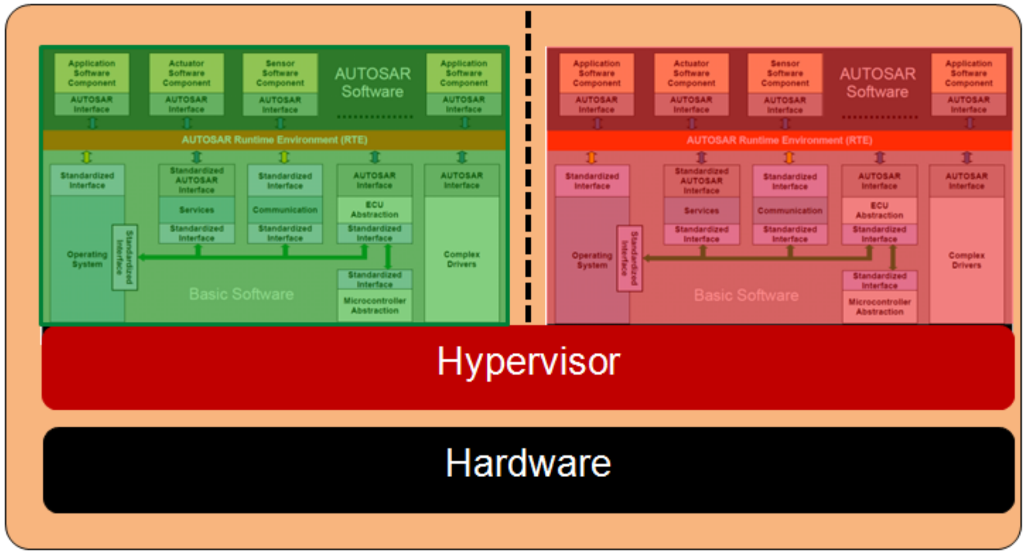
- **ASIL decomposition and safety criticality analysis become absolutely necessary**

Usage scenario	• **ASIL-Decomposition**
Goal Achieved	• Lower development costs • Better options for ASIL decomposition using intelligent app-distribution and “parallel redundancy” • Increase Safety via Core separation

MULTICORE AUTOSAR USAGE SCENARIOS



Usage scenario	• **Dedicated Use Of Cores (e.g. as I/O-controller and "number cruncher")**
Goal Achieved	• High Peformance



Usage scenario	• **Safe Virtualization, Scalability**
Goal Achieved	• Reduce configuration effort for scalability scenarios • Supplier specific isolation • ASIL specific isolation

SOME CHALLENGES WHEN CHANGE FROM SINGLE CORE -> MULTI-CORE

- Very often requests to shared resources on Autosar single core systems are realized by sounding the critical section with interrupt-blocking – this will not work with Multicore
  - Performant synchronization mechanism with hardware support is necessary (Spin lock with shared memory, message passing (IOC in Autosar), HW-support for atomic “test-and-set” function.
- Support for cache coherency in hardware or software
- MPU/MMU should support IO Protection
- Optimized and safe inter-core communication
- Peripheral-Access should not be the bottleneck. Number of cores are limited by I/Os.
- Tooling: Support for SWC-mapping to optimize core load, minimize inter-core communication and allow energy management to power-down cores
- Energy-management mechanism on SW- and HW-level
- Reuse of existing code – need for automated migration options

SOME CHALLENGES FOR VIRTUALIZATION

- MPU/MMU support for spatial separation in the IO-space
- The MPU should contain sufficient registers to contain the architectural state (register sets) of the hypervisor and the guest.
- Hardware-support for shared IO-Devices (e.g. CAN-Bus)
- GPU should support scheduling and memory protection (IO-MMU)
- Hardware support that allows each interrupt or trap to be directed either to a guest or to the hypervisor with no time penalty
- Small trusted code base
- Hypervisor should allow qualification based on ISO 26262
- Self-Monitoring system to capture status of partitions and trigger fail-safe mechanism.

SUMMARY

- **Multicore will be part of future Automotive ECUs**
  - Increased computing power with better energy-efficiency
  - Support for centralization and more degrees of freedom for new E/E-architectural approaches
  - Automated Driving will increase the number of high-Peformance functionality
  - Increased reliability separating functions on cores
  - Increases safety supporting ASIL-decomposition
  - Enable virtualization scenarios to support scalability

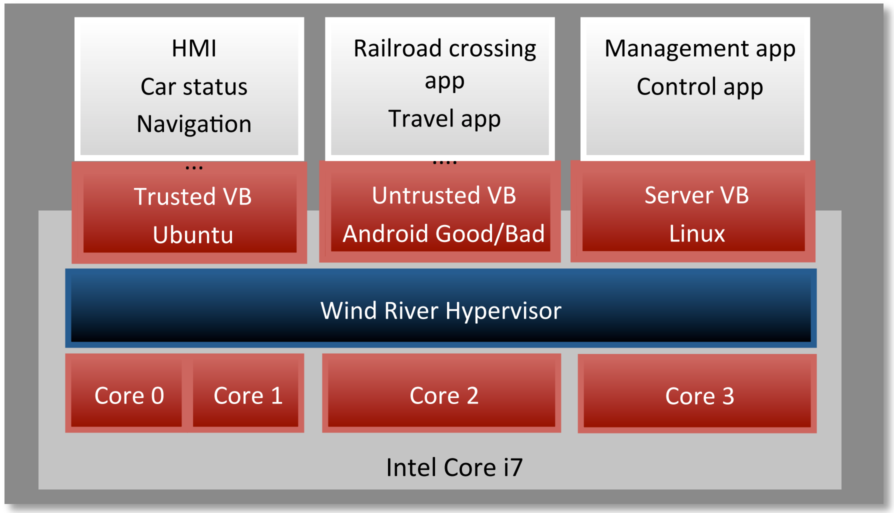
There is still some homework to do for overall use in series production

⇒

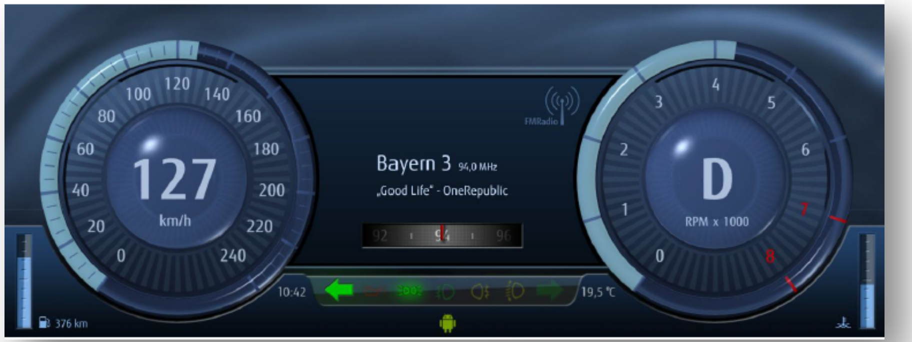
ARAMiS will focus on the challenges looking for comprehensive solutions for Automotive, Avionics and Railway.

VIRTUALIZED CAR TELEMATICS (VCT) DEMONSTRATOR

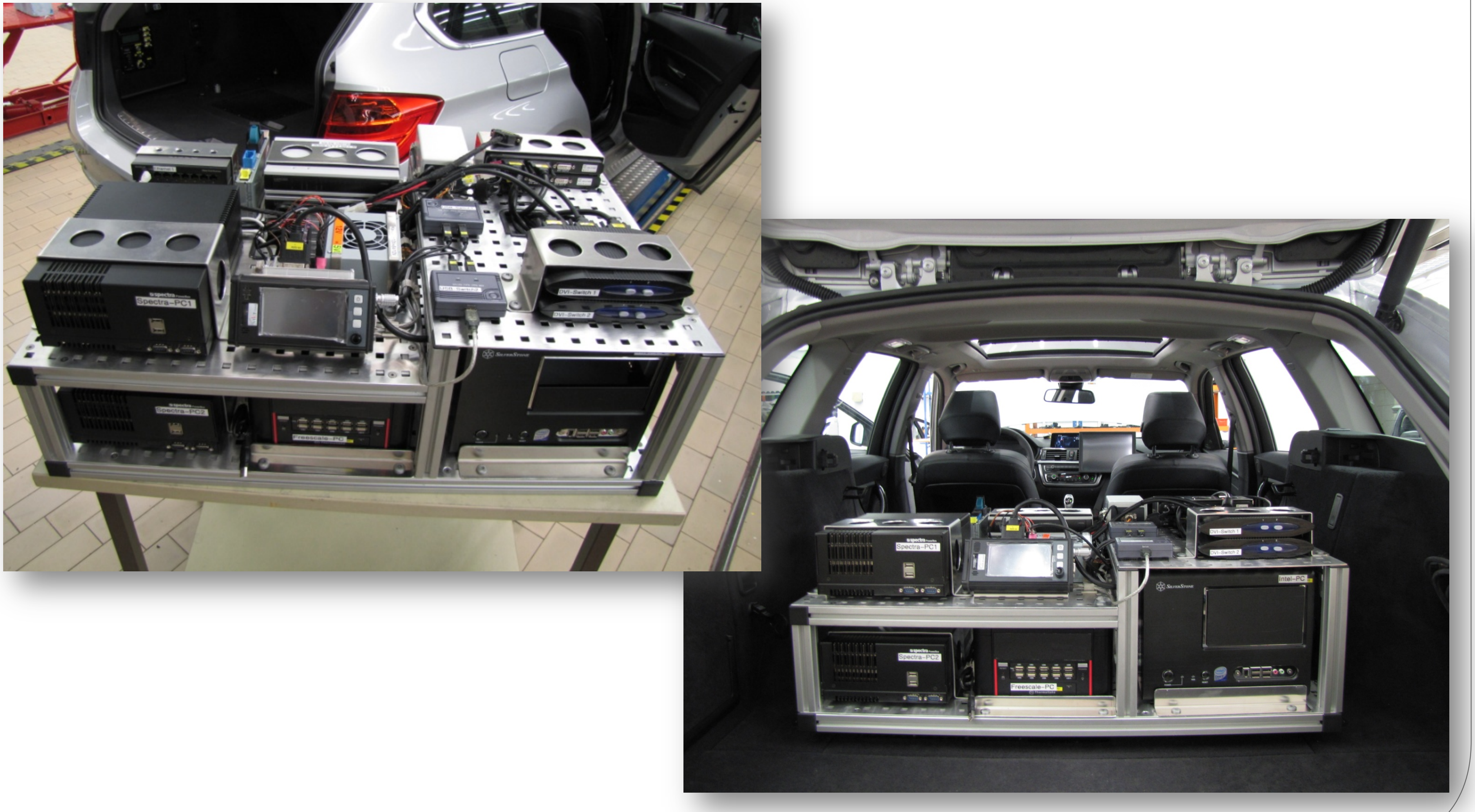
- Virtualization as key technology to use multicore platforms in embedded systems
- Centralization / consolidation of functions into infotainment domain unit
- Segregation: isolation of applications of different safety- or security-levels
- Re-use of existing software



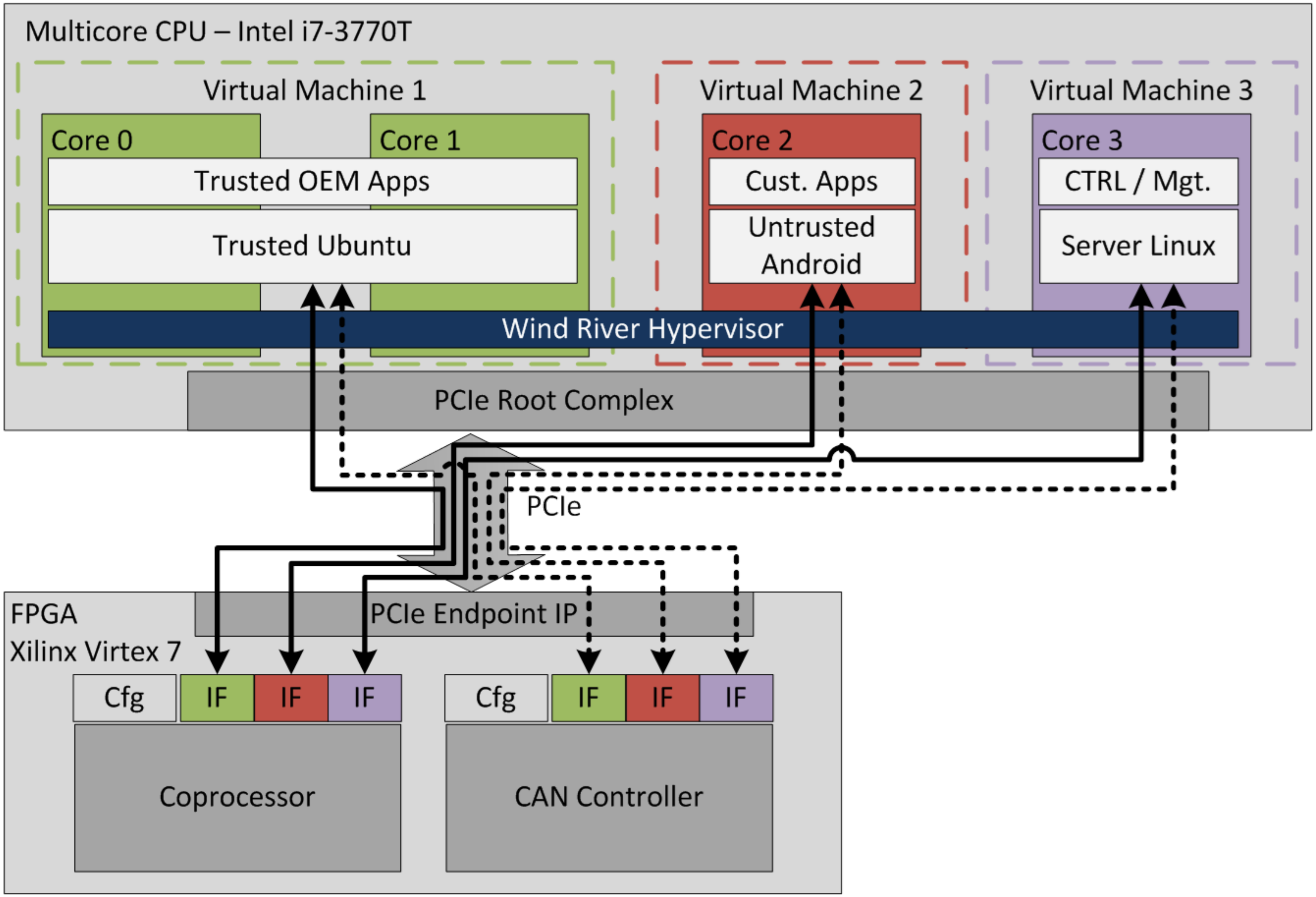
- Goals
  - Dynamic mapping of user-oriented 3D-graphics on combi-display / headunit
  - Dynamic relocation of content depending on vehicle status
  - Usage of android-apps by providing of a segregated partition for „insecure“ applications



VIRTUALIZED CAR TELEMATICS (VCT) DEMONSTRATOR

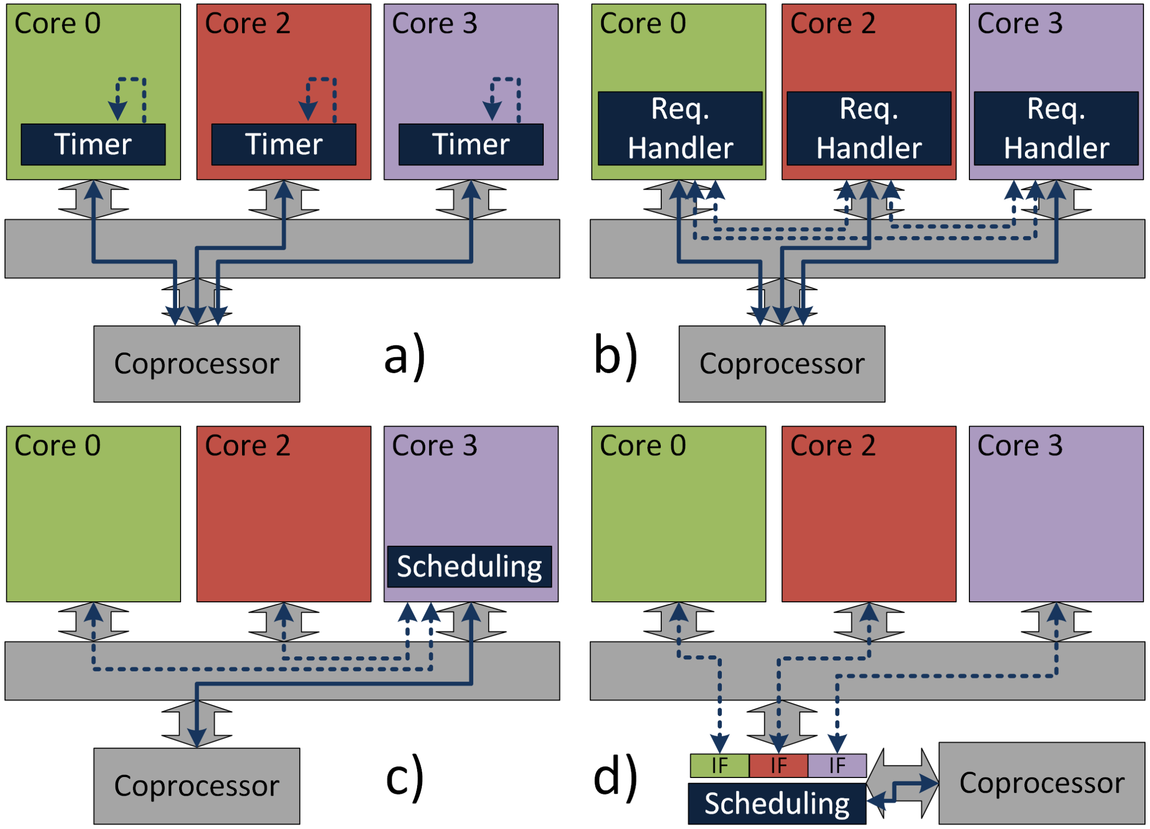


VCT COMPUTER ARCHITECTURE



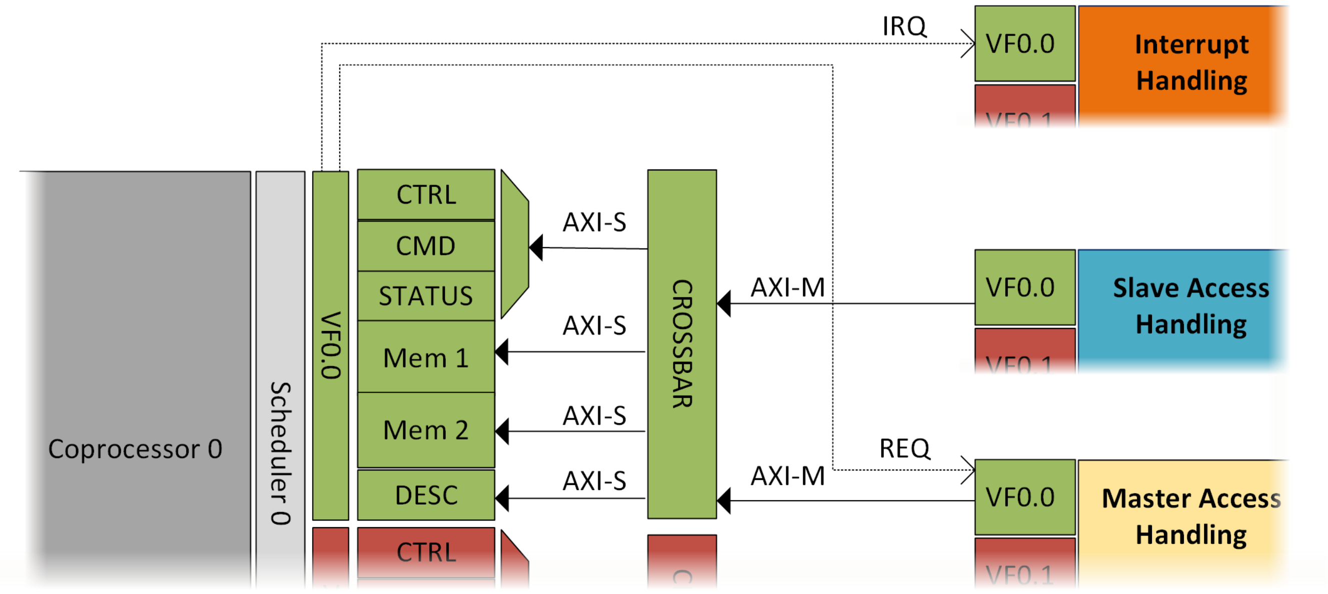
SHARED COPROCESSORS IN MULTICORE SYSTEMS

- Resource sharing mechanisms
  - a) time-based
  - b) request-based / cooperative
  - c) proxy partition / hypervisor
  - d) hardware scheduling, transparent for partitions
- Requirements in safety-critical systems
  - efficient usage of multicore architecture
  - different priorities of partitions
  - predictability of behavior at concurrent accesses
  - quality-of-service assertions
  - portability to different multicore architectures

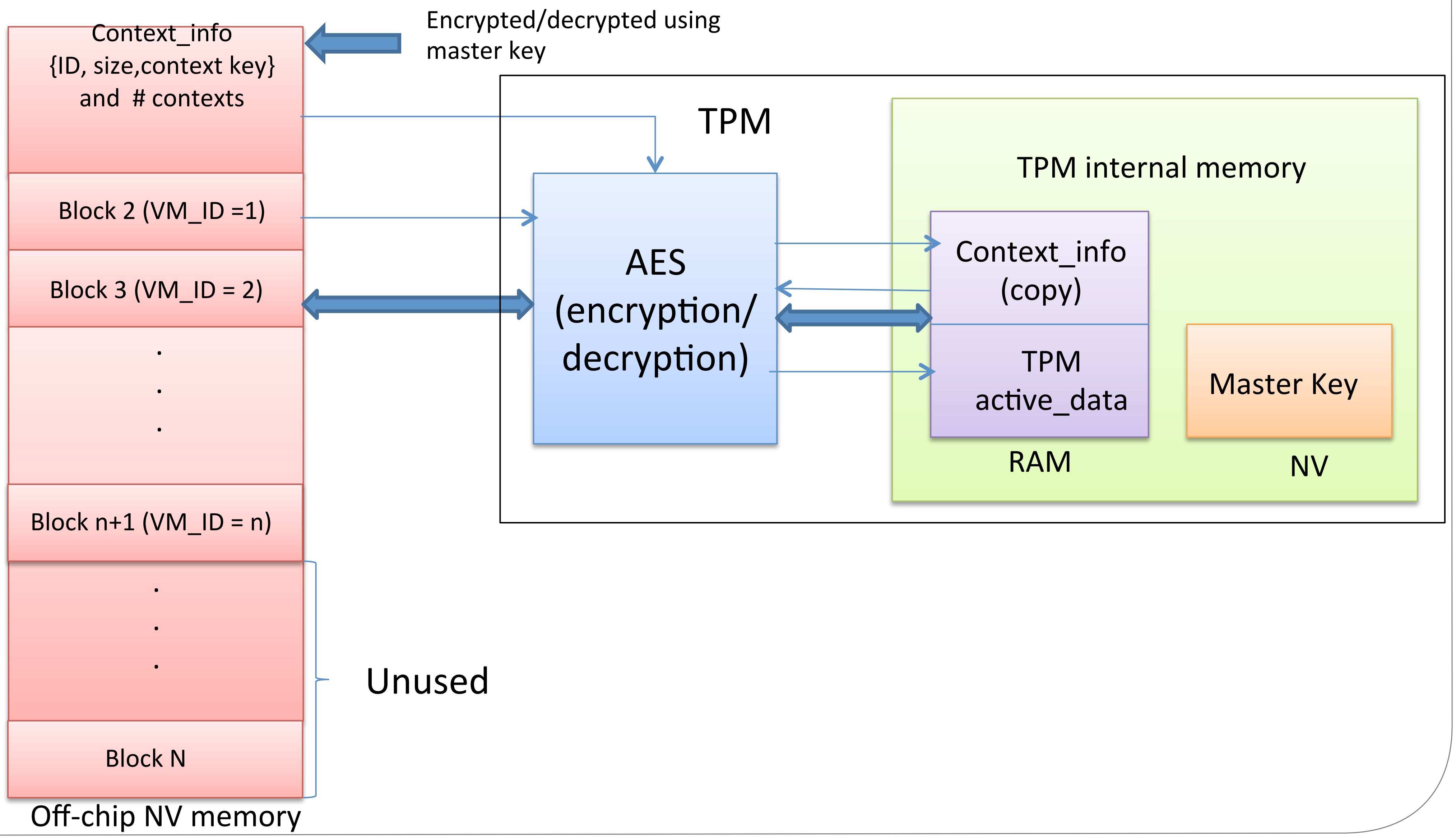


VIRTUALIZED COPROCESSOR INTERFACE

- Generic interface architecture for shared coprocessors
  - PCIe connection for Virtex-7 FPGAs, PCIe SR-IOV compatible
  - Support for slave- and DMA-accesses
  - Interrupt handling
  - Porting to further platforms (Zynq) currently in progress
- Virtual interfaces to realize spatial segregation
- Scheduling modules enforce temporal segregation



SECURE CONTEXT SWITCHING
ON-CHIP MASTER KEY APPROACH


Context_info {ID, size,context key} and # contexts

Encrypted/decrypted using master key

Block 2 (VM_ID =1)

Block 3 (VM_ID = 2)

Block n+1 (VM_ID = n)

Block N

Unused

Off-chip NV memory

TPM

AES (encryption/ decryption)

TPM internal memory

Context_info (copy)

TPM active_data

RAM

Master Key

NV