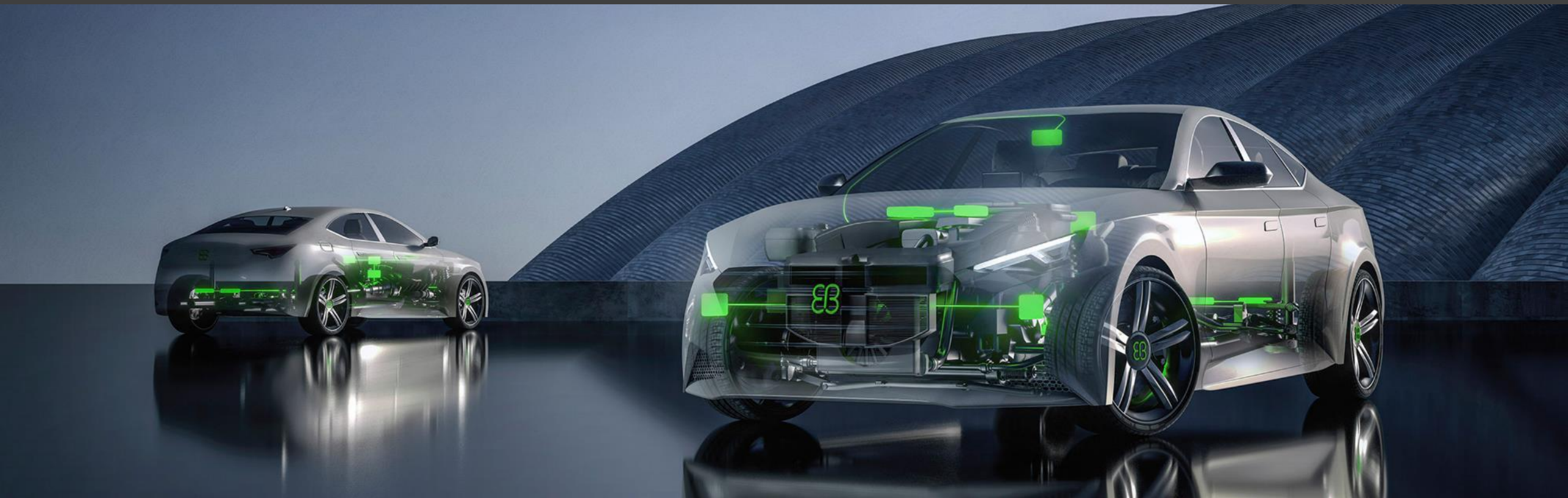
ST Developers Conference:
How to Efficiently Separate Automotive Functions Within a Multi-Core System with EB tresos AutoCore Hypervisor

Christoph Dietachmayr, Solution Manager
September 12th, 2019

Elektrobit



Agenda

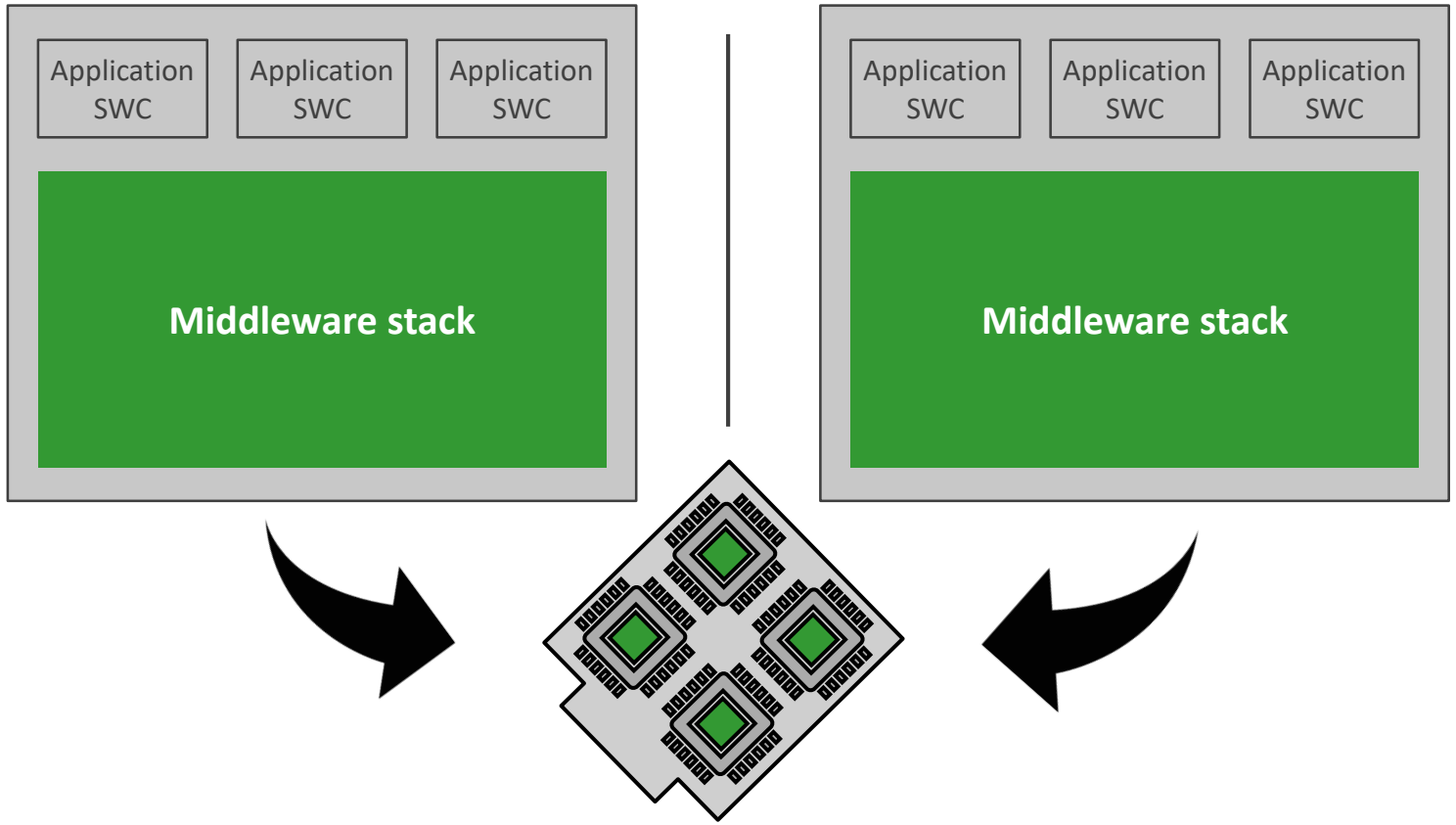
- Motivation
- Challenge
- General solution
- Specific Implementation
  - EB tresos AutoCore Hypervisor
  - EB corbos Virtual Ethernet Switch
- Hypervisor Comparison
- Summary

Motivation

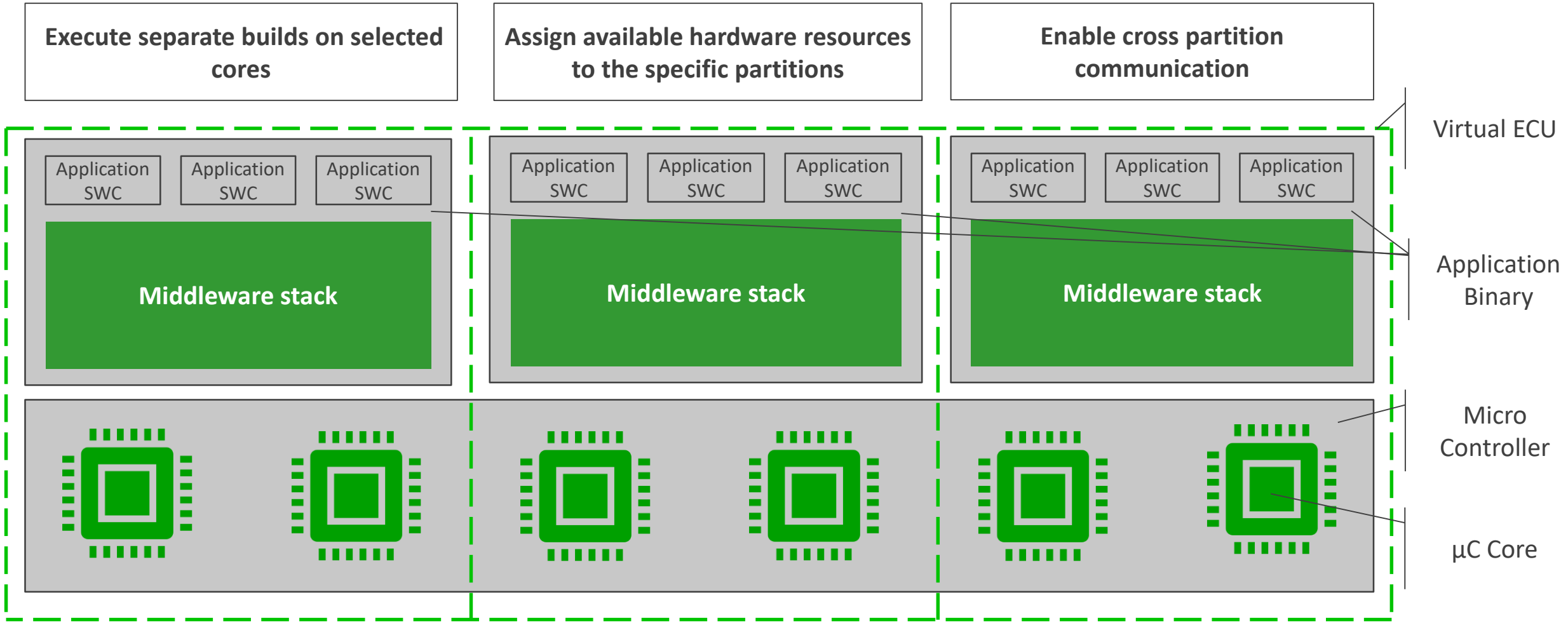
Market	• New vehicle architecture
Business model	• Software as a product
Efficiency	• Shorter development cycle
Legacy migration	• New E/E architecture
Development	• Distributed development

The Challenge

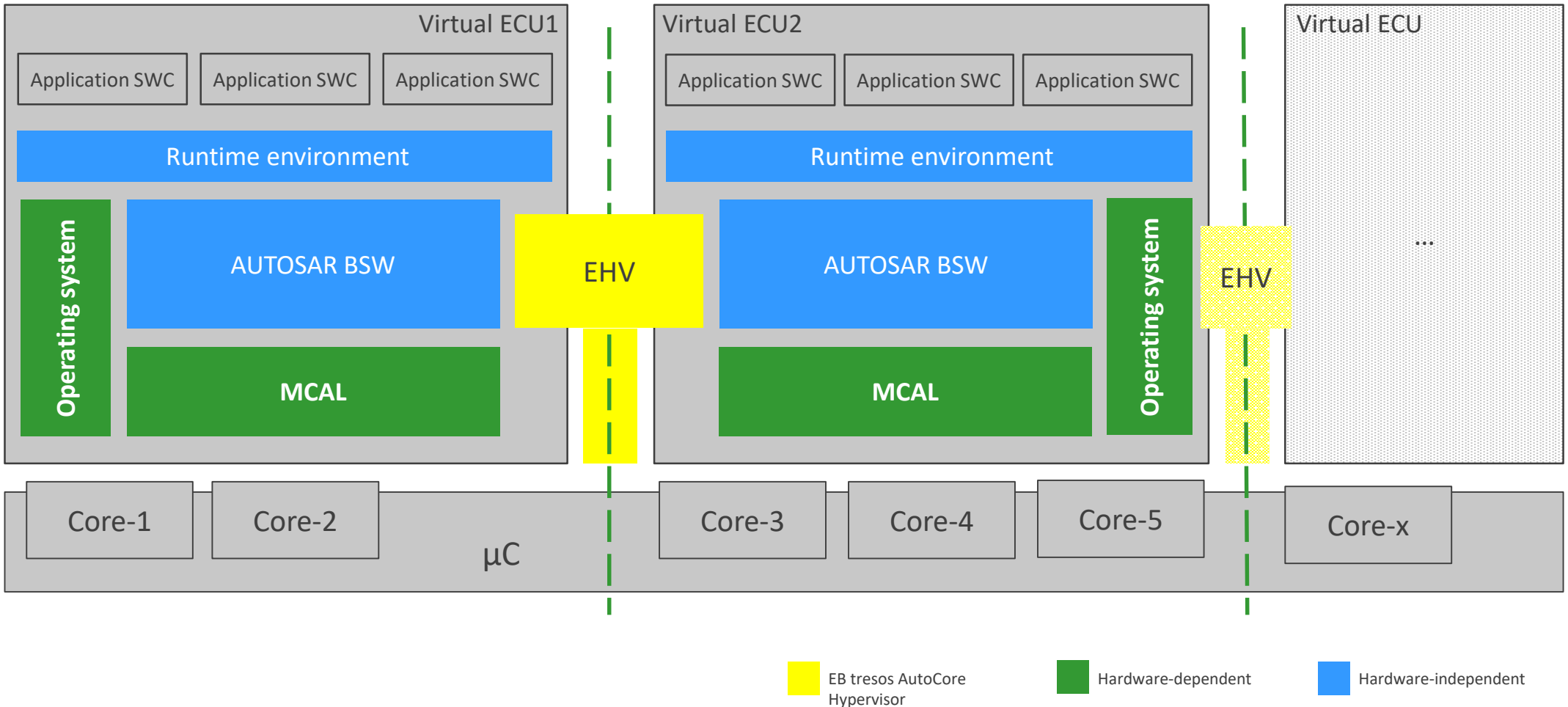
- Homologation of functional application sets (OBD/Non-OBD)
- Consolidation of functional application sets on one ECU
- Separation of safety criticality levels on one ECU (different ASIL levels)



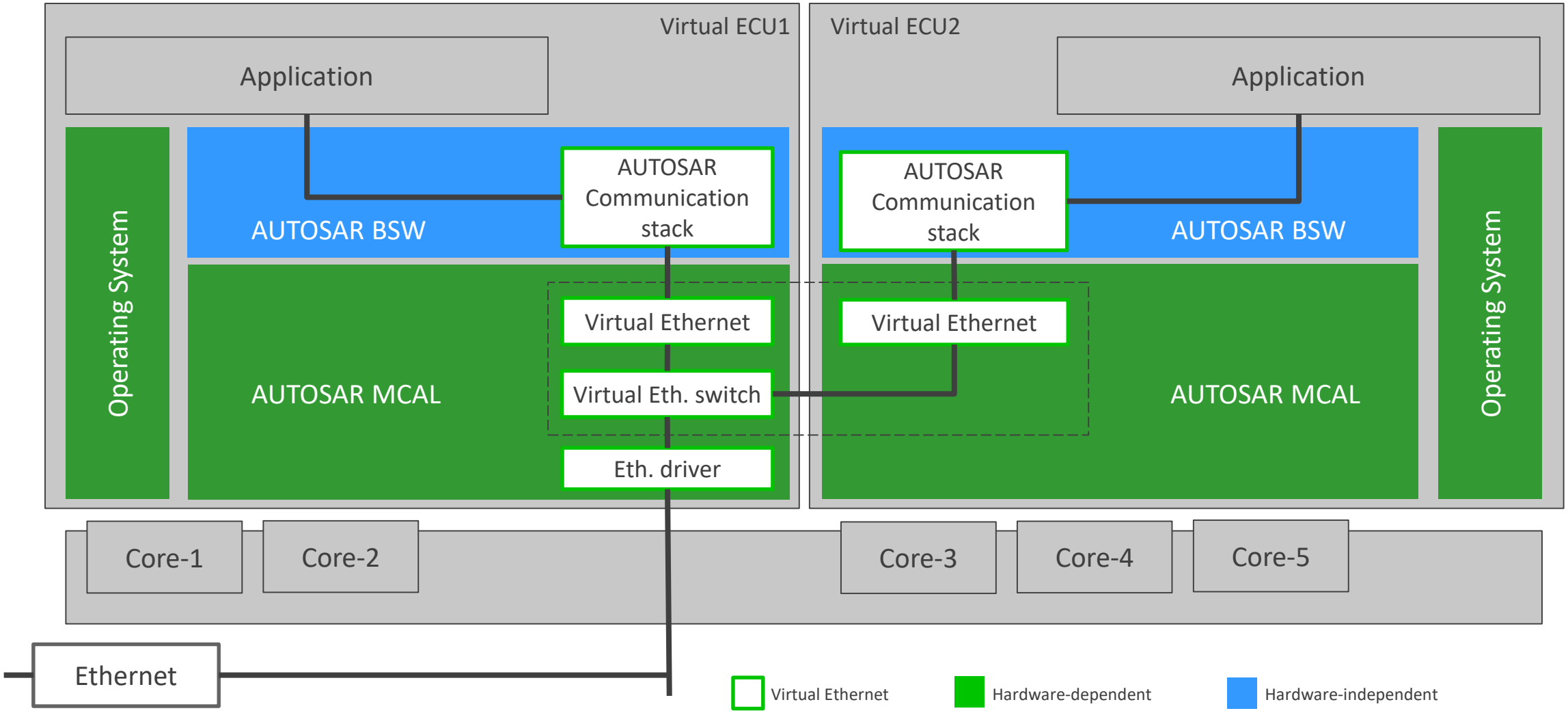
Solution Concept with a Multi-Core Processor - Overview



Static Separation Mechanism (w/o MMU)



Virtual ECU Communication with Virtual Ethernet Switch



Virtual Ethernet Switch

General

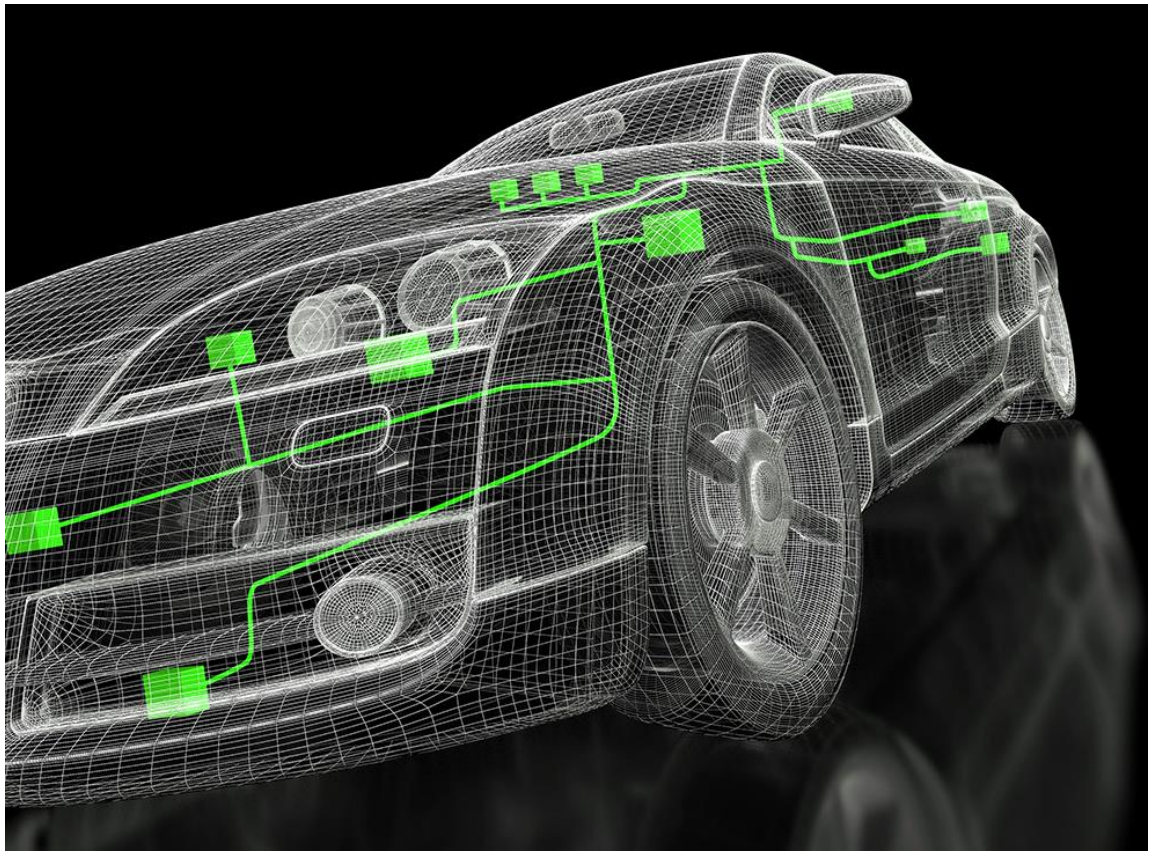
- Transmission/reception from cores or from outside
- Support of QoS (Tx/Rx) and TimeSync

Performance

- Performance optimized data path for Tx/Rx
  (e.g. zero-copy/minimize copy operations, lock time)
- Deterministic Tx/Rx forwarding delay

Safety

- Freedom from interference (timing and execution, memory, communication)
- Support of coexistence with safety related components

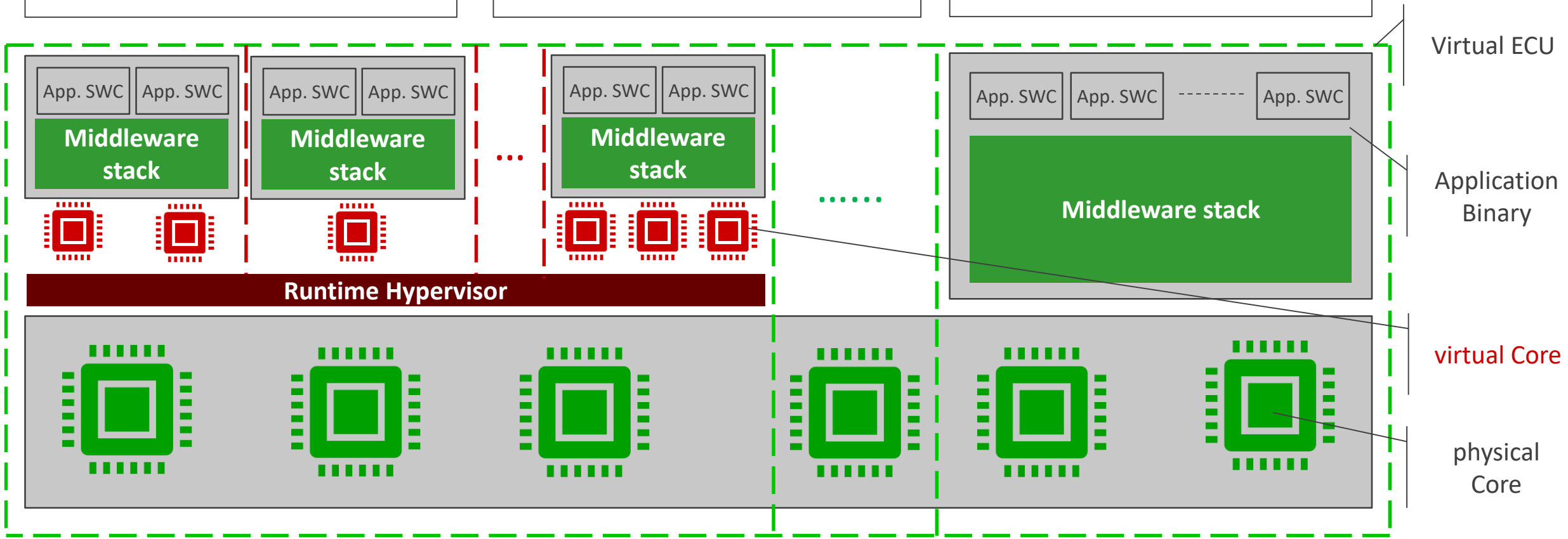


EB tresos AutoCore Hypervisor
Full support for Cortex-R52 like architectures

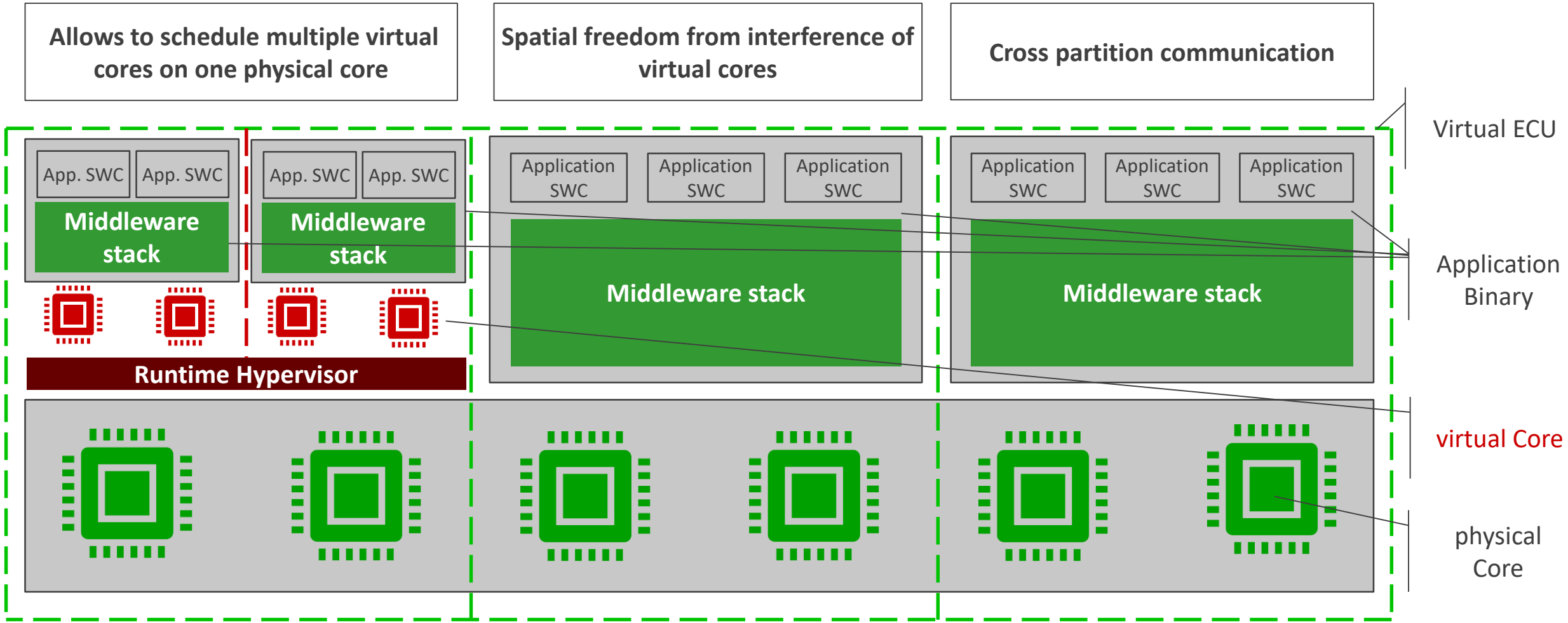
Allows to schedule multiple virtual cores on one physical core

Spatial freedom from interference of virtual cores

Cross partition communication



EB tresos AutoCore Hypervisor in Action
Potential Configuration on Stellar



Features – EB tresos AutoCore Hypervisor

Core separation

- Virtual ECU separation
- Containing individual code
- OBD/non OBD on separate cores
- Safety OS (ASIL-D) and AutoCore OS (QM) on same ECU but different cores
- Static configuration of memory regions
- Booting of separate OS instances
- Setup of virtual ECU containment
- Configuration of interrupts

Inter-VECU communication

- Library with static communication channels
- Communication via shared memory
- OS-independent inter-core locks
- Cross-core interrupt trigger
- Call-out functions

Safety

- Freedom from interference between separation domains using hardware separation features such as registers, memory, peripherals, and code
- Support of coexistence with safety-related components

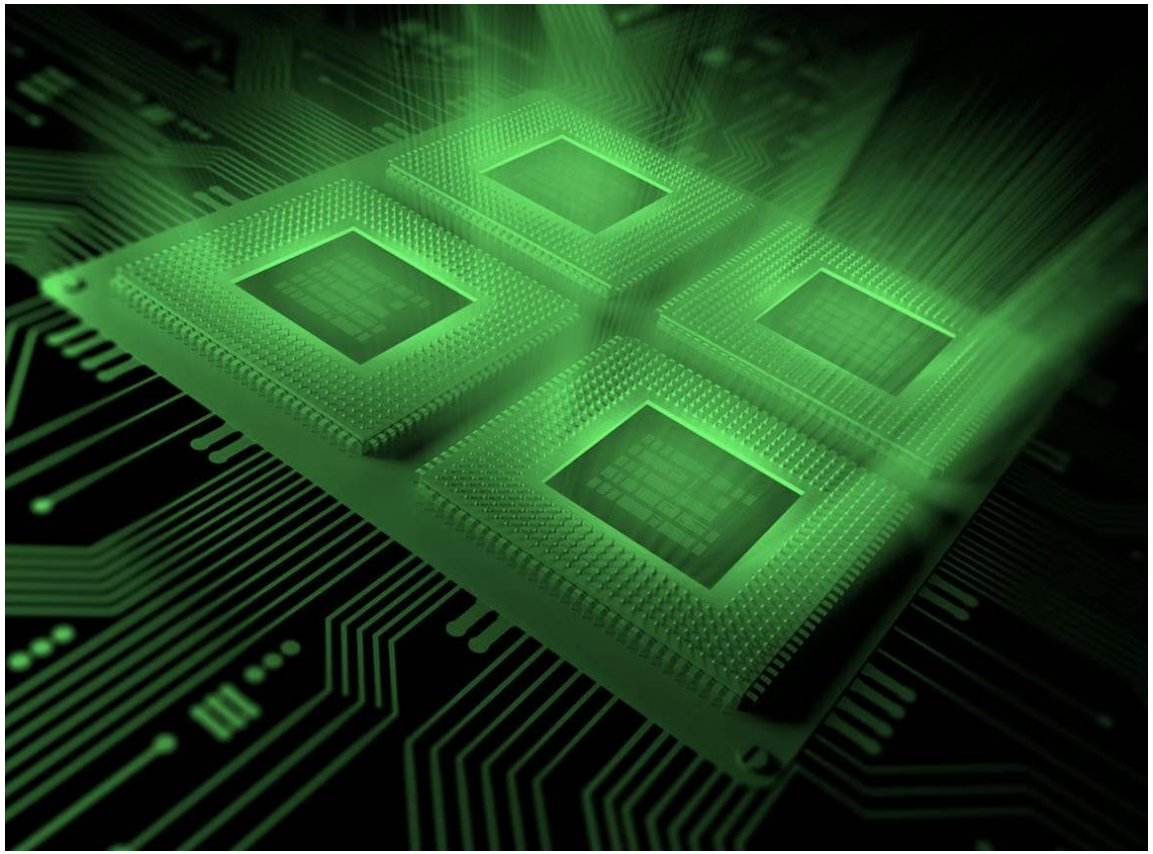
Further OS support

- EB tresos Safety OS supports multiple instantiations
- Support for containment domains with configurable number of CPU cores

Integration of ST SR6x7 Centauri

The following features of µ controller are utilized:

- Hypervisor EL2 MPU
- NOC-Firewall
- Interrupt controller
- Core-specific timer
- Shared and private memory
- Fast inter-core communication
- Inter-core locking



Hypervisor comparison

EB tresos AutoCore Hypervisor

- Hard real-time capable
- Usage of MPU
- Minimal virtualization
- Less processor overhead

Dynamic hypervisor

- Virtualization of hardware resources
- Usage of MMU
- Overhead due to dynamic allocation
- Usage of dynamic OS (Linux/QNX)

Summary

EB tresos AutoCore Hypervisor contributes towards:

- Reduction of ECUs in the vehicle
- Legal separation of software components
- Flexibility for peripherals (e.g. CAN, FR, …)

EB tresos AutoCore Hypervisor allows:

- Smooth inter-core communication also for inter-VECU communication
- Hard real-time behavior
- High hardware utilization
- Low power consumption
- Exchange of software on a single-core (VECU vs RECU)

Get in touch!

christoph.dietachmayr@elektrobit.com
www.elektrobit.com

Elektrobit

Virtualization Approach on the ST STELLAR MCUs

Khaldoun Albarazi

Market Development Engineer

Automotive Product Division

STMicroelectronics

life.augmented

ST Developers Conference

September 12th, 2019
Santa Clara Convention Center - Mission City Ballroom
Santa Clara, CA

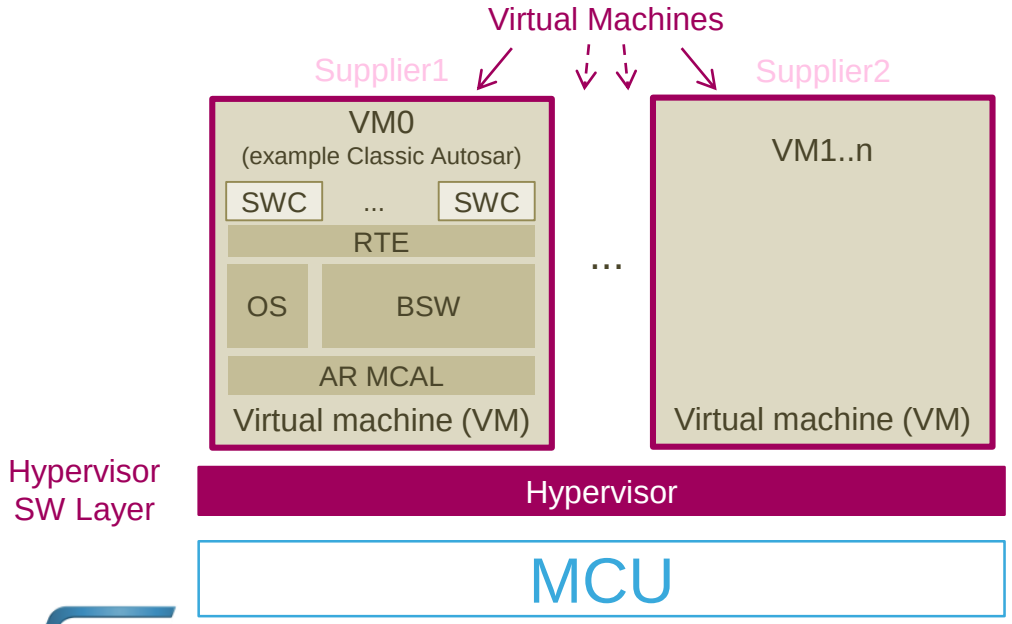
Forewords – Embedded Virtualization

Concept Introduction

State of the art SW Technology for Integration and Isolation of Multiple Applications onto one single Processor (MCU/MPU),
allowing multiple SW with different ASIL levels to run on same MCU

There are other ways to isolate applications but virtualization offers additional key benefits:

- Reduce Interaction between different SW suppliers, easier integration of multiple independant SW
- Correspondingly easier SW updatability (each supplier / application SW independantly from each other)



Drivers for embedded Virtualization in Automotive

SW complexity reduction taking advantage of the introduction of more performing MCUs (STELLAR) and moderate to no cost advantage of technology beyond 40nm

- re-combine distributed functionality in a single ECU
- Domain Controller architecture to tackle multiple challenges
- SW modularity & update-ability

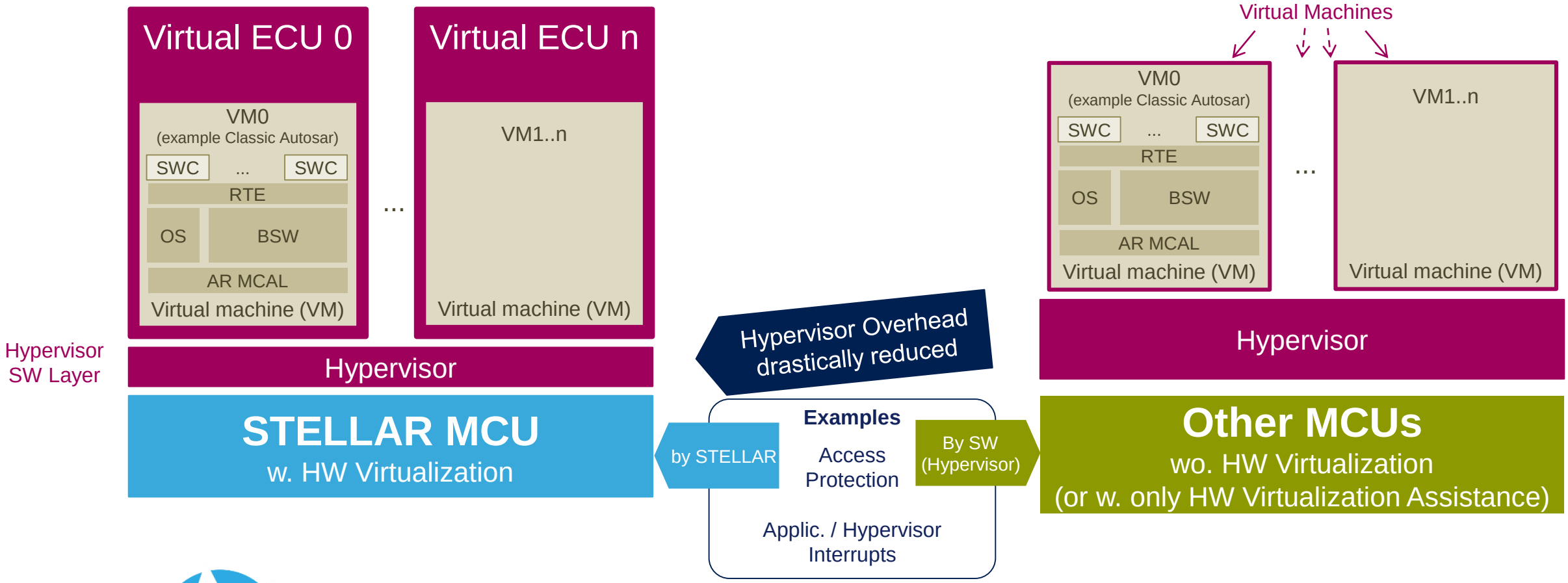
Implications

- Integrated software functions have different functional safety level missions and are developed by different suppliers
- Freedom from interference must be insured in a simple manner
- Possibility to update one software function independently from the others (no risk on the others and no need to fully requalify)

Requirement for real-time (boot & runtime) is maintained for embedded systems

life.augmented

Unique Hardware/Real-Time Virtualization



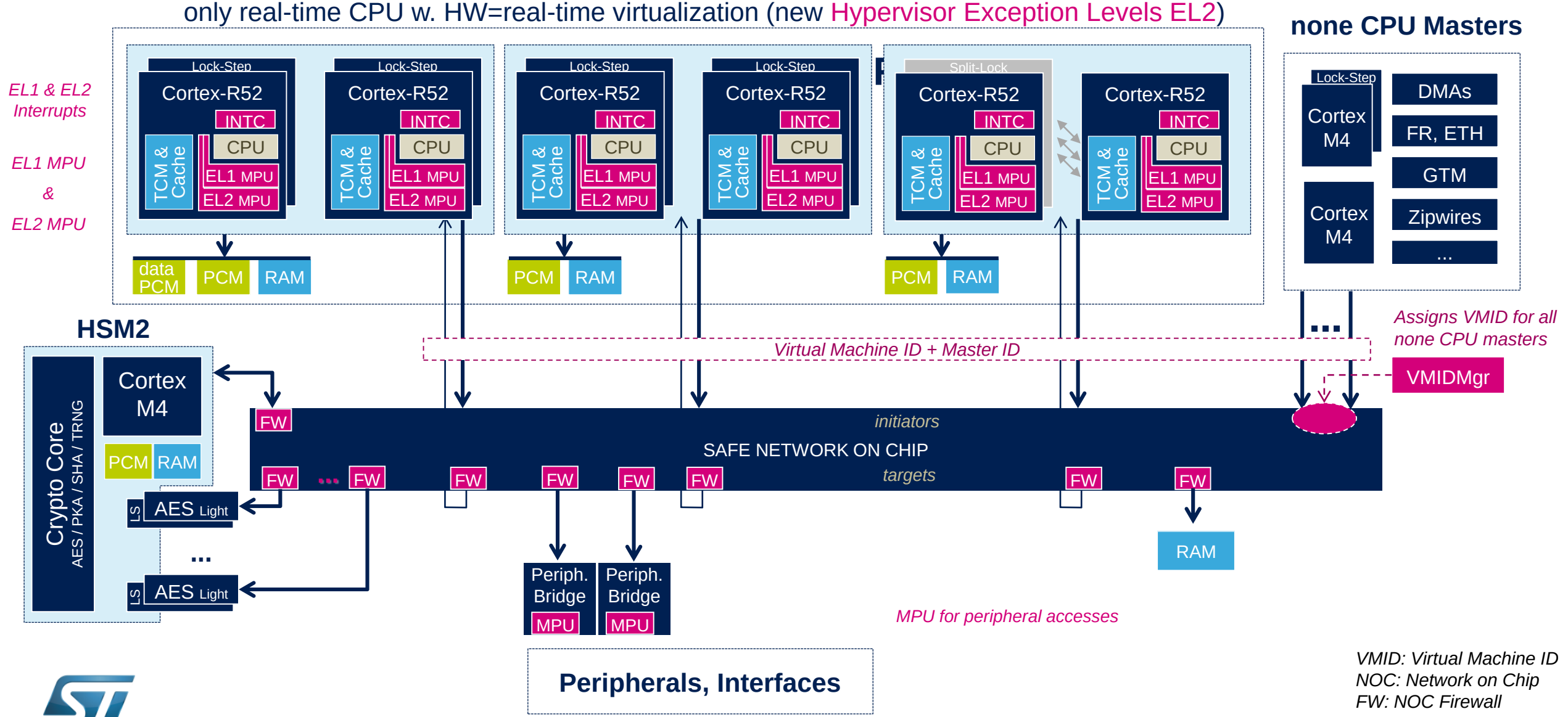
High Performance Virtualization	Big Overhead for Virtualization Support
Real-time Virtualization	No Real-Time capability
Deterministic Virtualization	Not Deterministic for none real-time CPU

life.augmented

STELLAR Real-time Deterministic Virtualization Architecture

arm CORTEX-R52 Clusters

only real-time CPU w. HW=real-time virtualization (new Hypervisor Exception Levels EL2)



life.augmented

Interrupt and Access Protection Infrastructure based on VMID / Master ID (VMID / Task ID for R52)

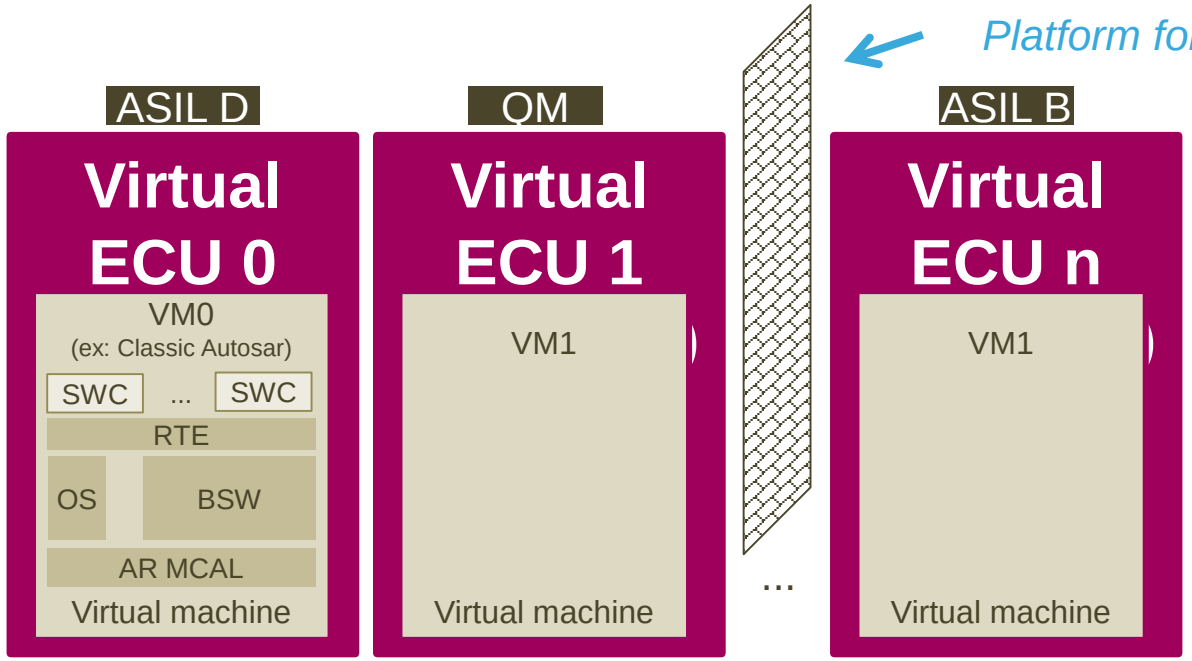
STELLAR MCUs Integration Platform

1st Automotive MCU w. real-time Virtualization Platform for ECU Integration & Domain Controllers

Turn-key Safety Isolation Platform for mixed ASIL level Applications („modular „Safety“)

Multi SW vendor Platform

Modular Software Development & Updates



Unprecedented Real time & Safe Performance

Secure Platform

STELLAR enabled High Performance Virtualization / Hypervisor

life.augmented

Thank You!

Contact

Khaldoun Albarazi
Market Development Engineer
Automotive Product Division
STMicroelectronics
Khaldoun.Albarazi@st.com

life.augmented