The road from GPU-powered prototypes to production ECUs

Elektrobit

2018-03-26

Status Quo – Cars as computers with wheels

1:1 relation between functionality and embedded device is standard

Since Years:

- Up to 100 small ECUs (Electronic control unit) serving a dedicated function like window lifter or climate control
- Several and highly specialized communication channels between ECUs (CAN, CANFD, FlexRay, LIN, Ethernet)

Today:

- Multi-core processors in use on ECUs serving multiple functions
- Software for ECUs is statically configured and flashed with a single image containing all applications and basic software

Challenge:
Integration of software of several parties cross company

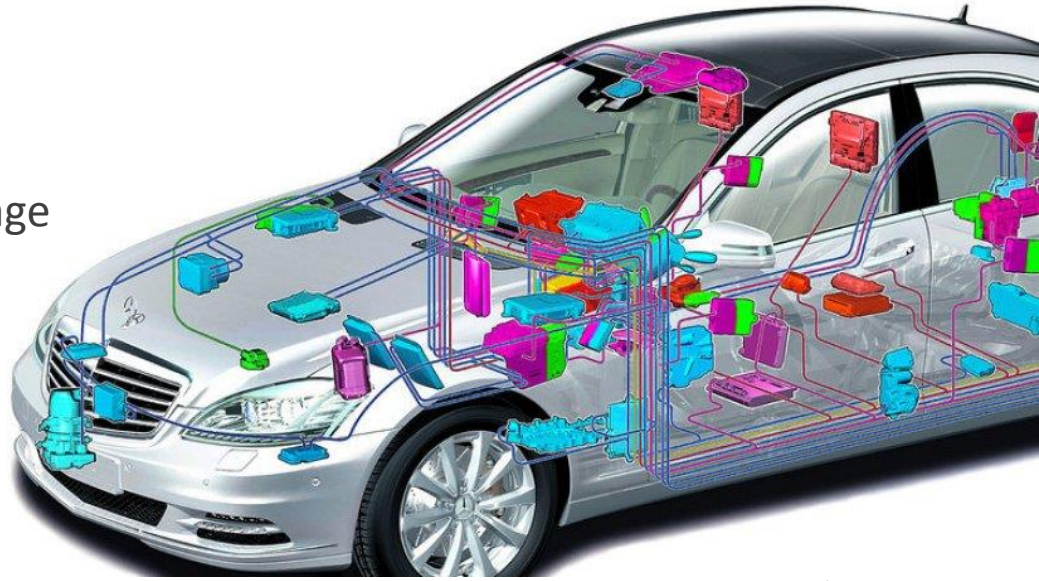
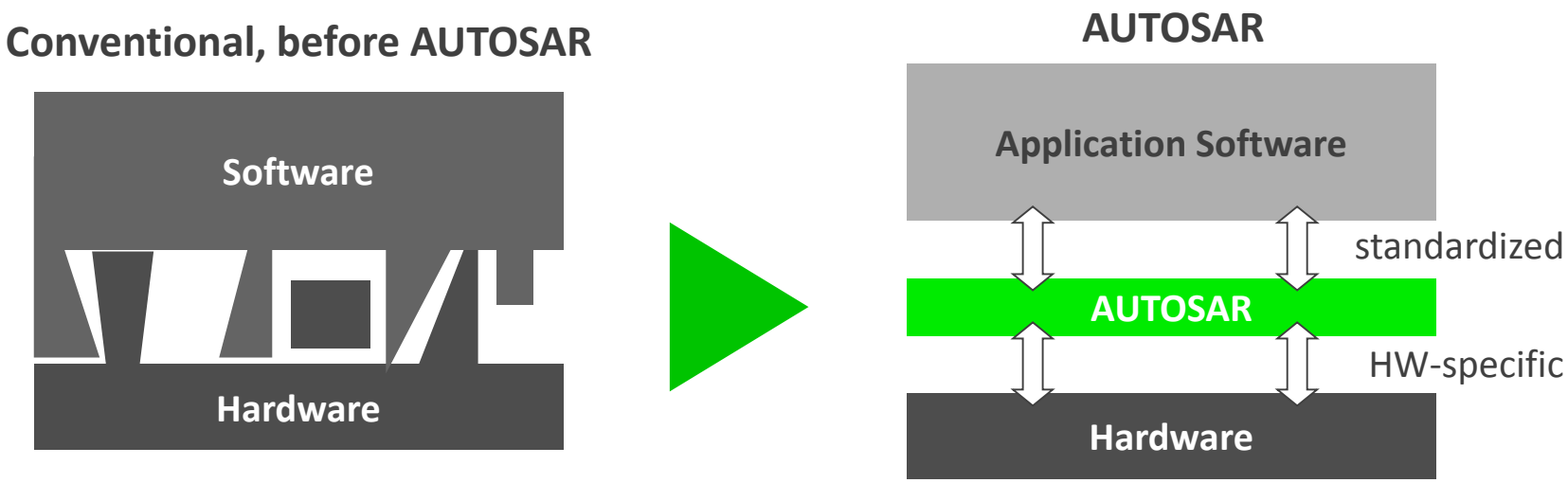


Photo: Hamburger Abendblatt /Daimler AG

Status Quo – AUTOSAR Standard enabling work split



Decouple application software from hardware

Standardize software interfaces

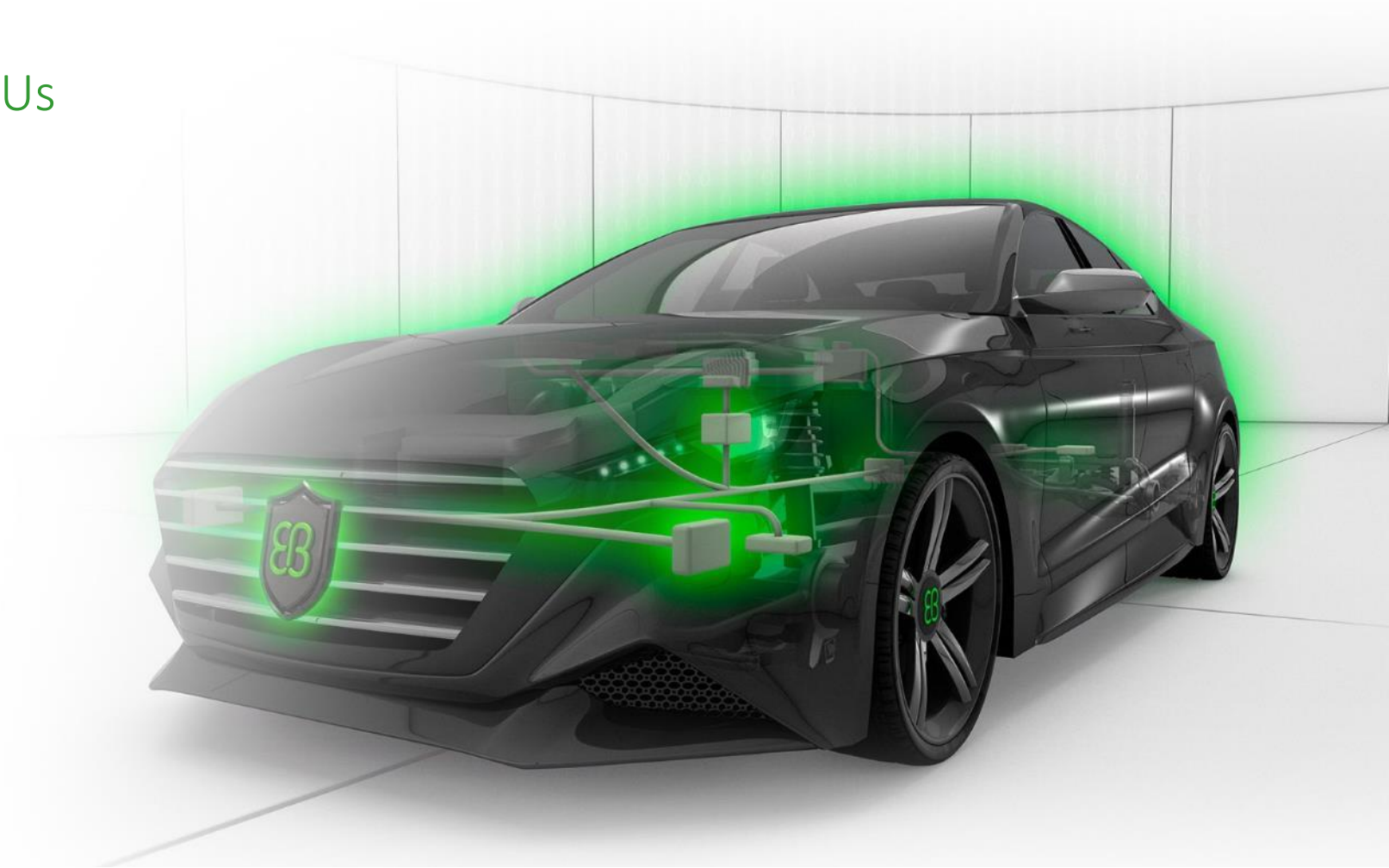
Standardize configuration concepts

Design the complete vehicle application software over all ECUs

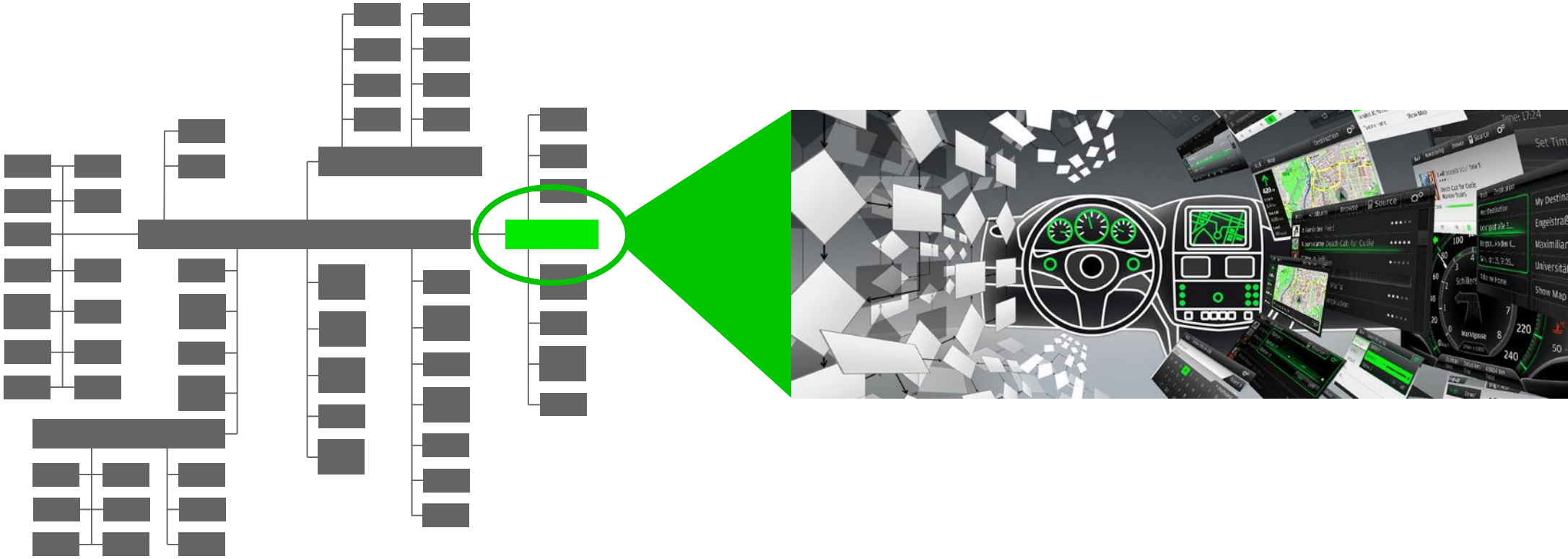
Status Quo – Vehicle infrastructure

Solutions for Classic AUTOSAR ECUs

- Basic software and services
- Tooling for standardized configuration
- Single- and multi-core operating systems
- **Functional Safety solutions**
  ISO 26262 certified up to ASIL-D
- **Embedded Security solutions**
- **Certifiable, e.g. Onboard Diagnostics**
- Automotive Ethernet software



Status Quo – Today's Vehicle infrastructure



GPUs only in use in the infotainment domain!

Copyright: IBM

How can we achieve the next level of mobility?

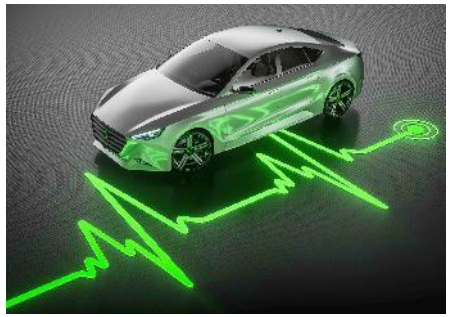
Requirements and challenges

Software over-the-air-updates

- New vehicles features
- Updates and patches
- Silent testing
- Security

Dynamic deployment

- User-driven updates
- Network accessible sensors & actuators



Remote analytics and diagnostics

- Car as a sensor
- Remote diagnostics
- Fleet campaigns

Developer oriented, target independent environment

- Environment independent software
- Easy qualification and deployment
- Small, encapsulated and exchangeable software services (mircoservice)

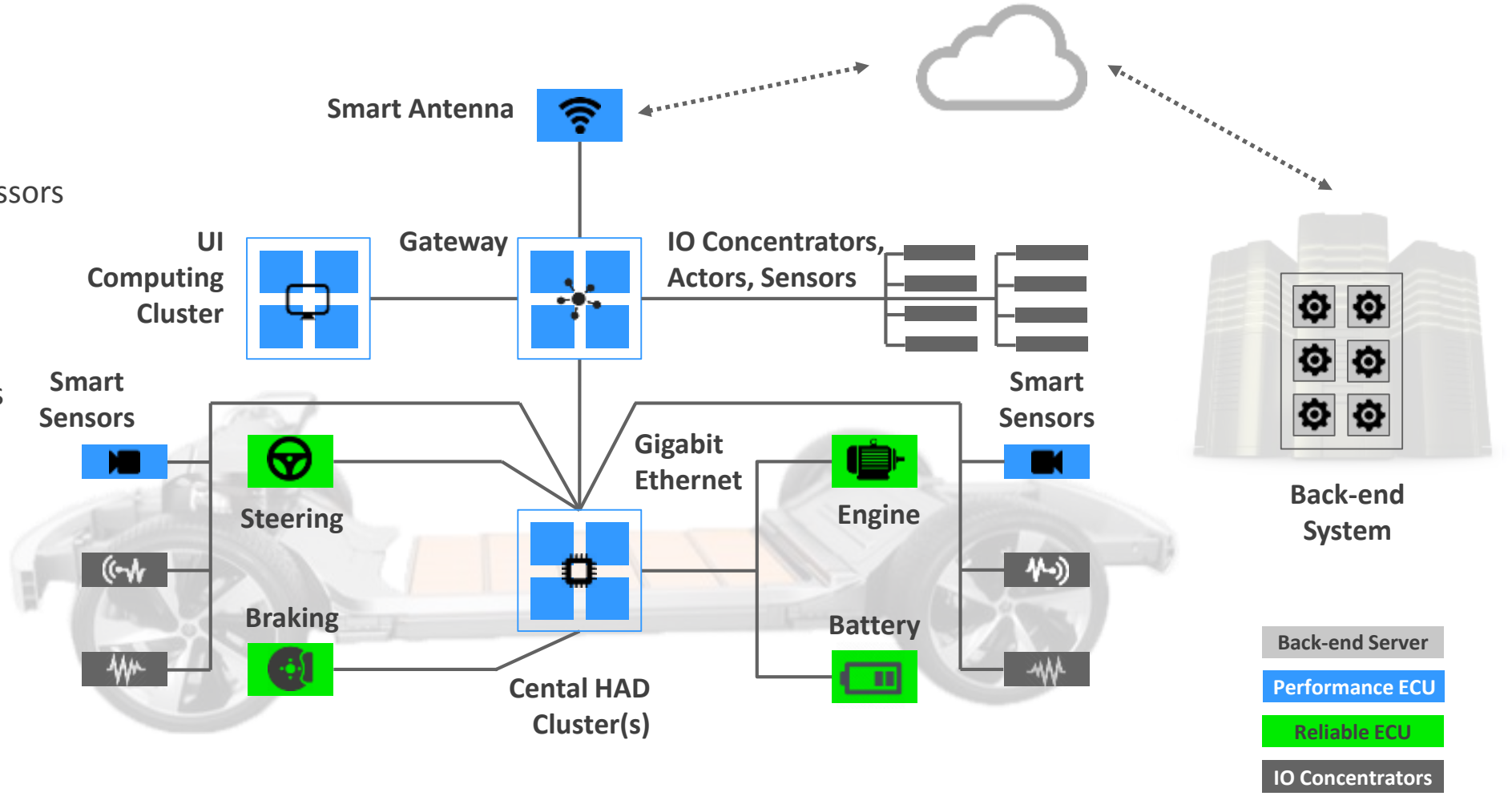
Dependable systems

- Safety
- Security
- Availability
- Reliability
- Maintainability

Elektrobit

Possible next generation vehicle infrastructure architecture

- Mix of:
  classic microcontrollers and powerful “cellphone” processors
- Mix of:
  classic automotive networks and cloud connectivity
- Mix of:
  safety-relevant functions and apps installed by the user



Elektrobit

Sensor fusion system requirements spanning across

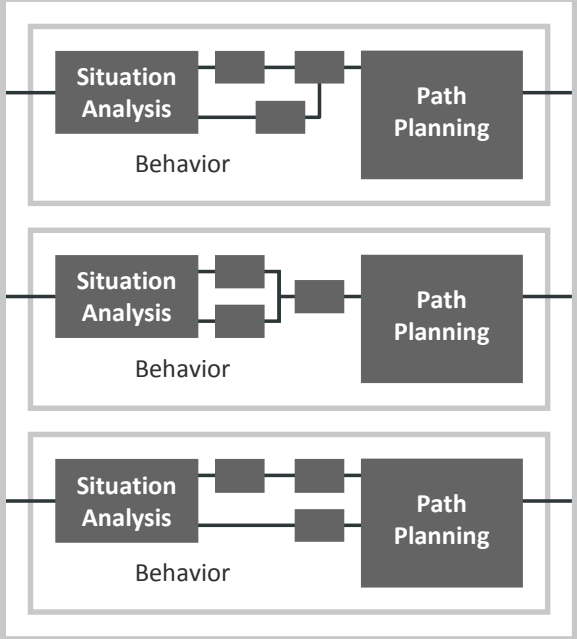
Vehicle Abstraction - Sensors

Sensor Data Fusion

- Positioning
- Object Fusion
- Grid Fusion
- Road and Lane Fusion
- Vehicle Database

Function Specific Views

Situative Behavior Arbitration



Motion Management

- Trajectory Control
- Longitudinal Control
- Lateral Control

HMI Management

Vehicle Abstraction - Actuators

Safety Management

Infineon

NVIDIA

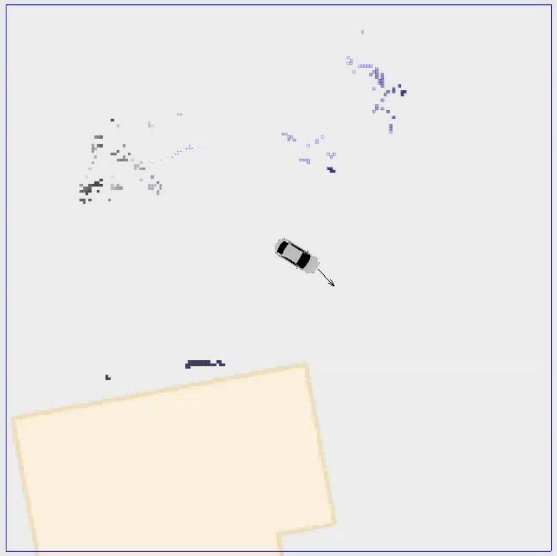
NVIDIA

Infineon

Grid Fusion in action

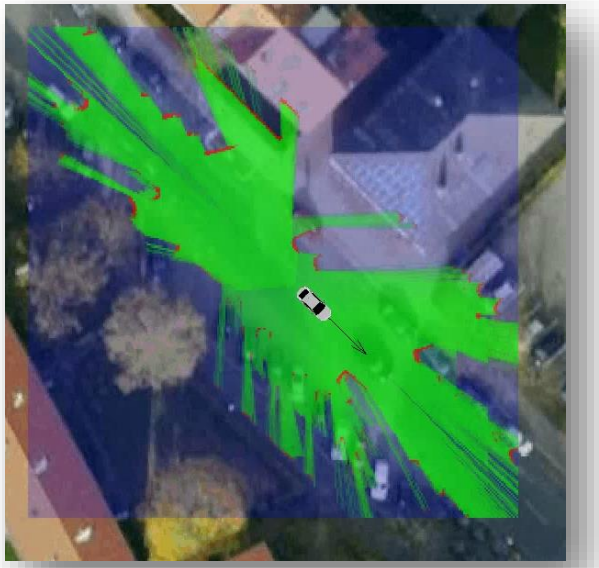
Bayes based Grid Fusion

- Traditional occupancy fusion
- Storing probability of occupancy
- Needs less memory due to smaller state vector



Height Map

- Storing height value and confidence
- Processes 3D data
- Use-case: ramp/hole detection, preprocessing for other fusion algorithms



Demster-Shafer based Grid Fusion

- State of the art occupancy fusion
- Storing evidence of occupancy/free
- Various sensor models
- Use-case: free space detection, path planning

Elektrobit

Sensor fusion system

Vehicle Abstraction - Sensors

Sensor Data Fusion

Positioning

Object Fusion

Grid Fusion

Road and Lane

Vehicle Data

Situative Behavior Arbitration

Motion Management

Scalable inputs

Sonar
Sensor Interface

Sonar
Sensor Adapter

Lidar
Sensor Interface

Lidar
Sensor Adapter

Radar
Sensor Interface

Radar
Sensor Adapter

Grid Fusion

Algorithm core

Scalable, standardized interfaces

Interface 1
high volume
for central ECU use

Interface 2
mid volume
for FlexRay / Ethernet

Interface 3
low volume
for CAN

For such application we need (parallel) computing power!

Elektrobit

Sensor fusion system

Vehicle Abstraction - Sensors

Sensor Data Fusion

Positioning

Object Fusion

Grid Fusion

Road and Lane

Vehicle Data

Situative Behavior Arbitration

Motion Management

Scalable inputs

Sonar Sensor Interface → **Sonar** Sensor Adapter

Lidar Sensor Interface → **Lidar** Sensor Adapter

Radar Sensor Interface → **Radar** Sensor Adapter

Grid Fusion

Algorithm core

Scalable, standardized interfaces

Interface 1
high volume
for central ECU use

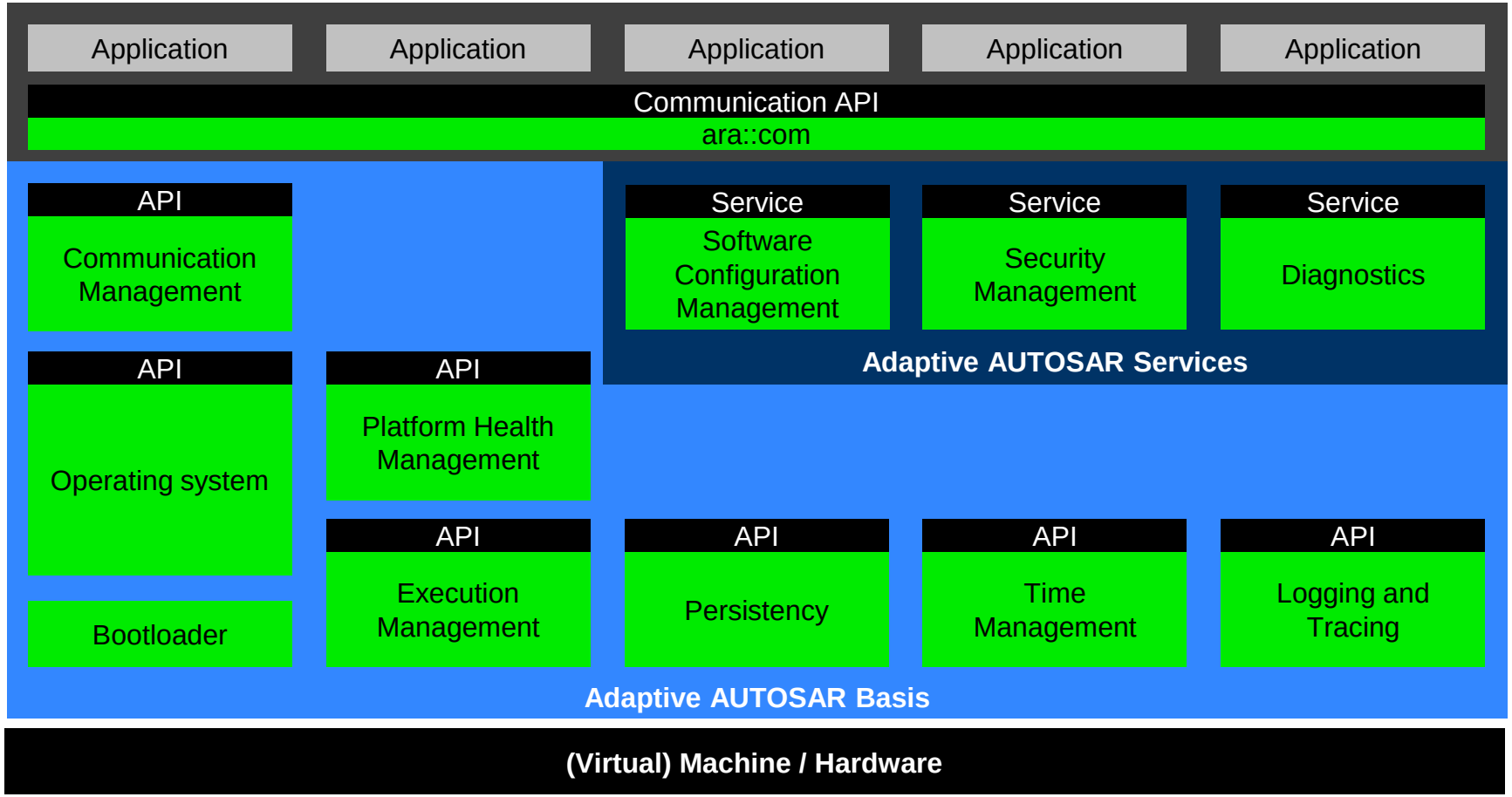
Interface 2
mid volume
for FlexRay / Ethernet

Interface 3
low volume
for CAN

How to achieve system abstraction?

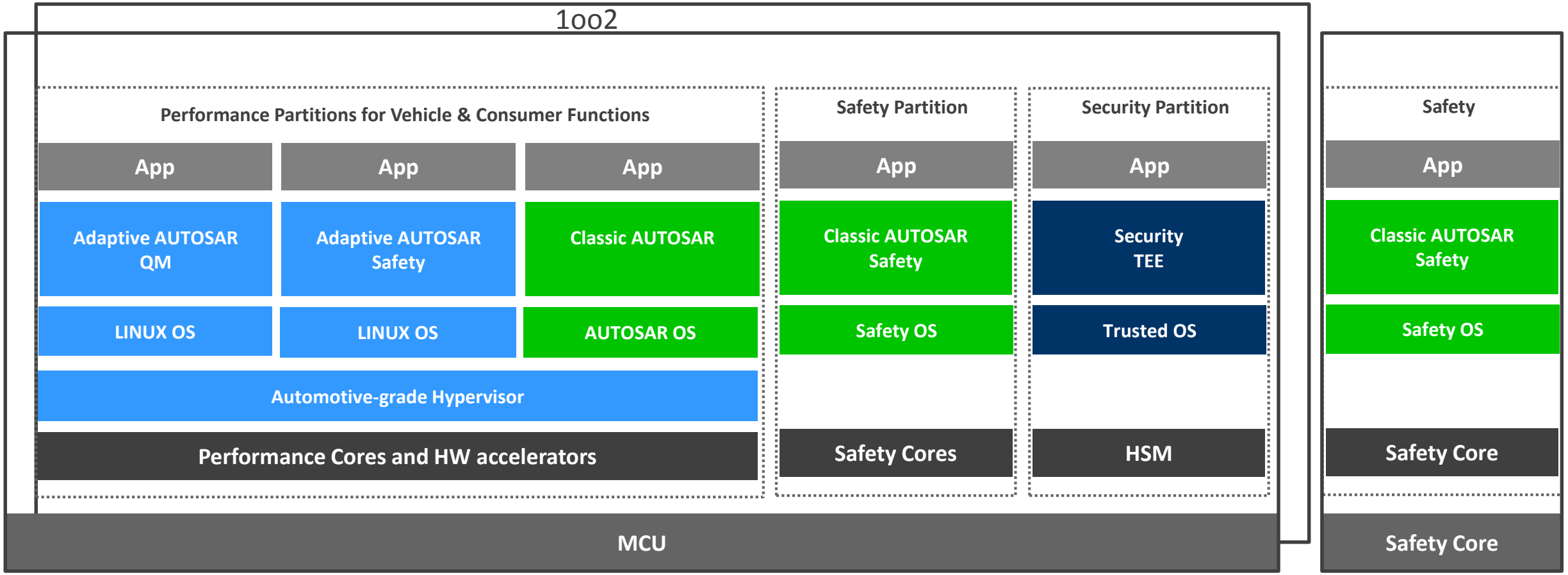
Elektrobit

Next generation of standardized basic software Adaptive AUTOSAR



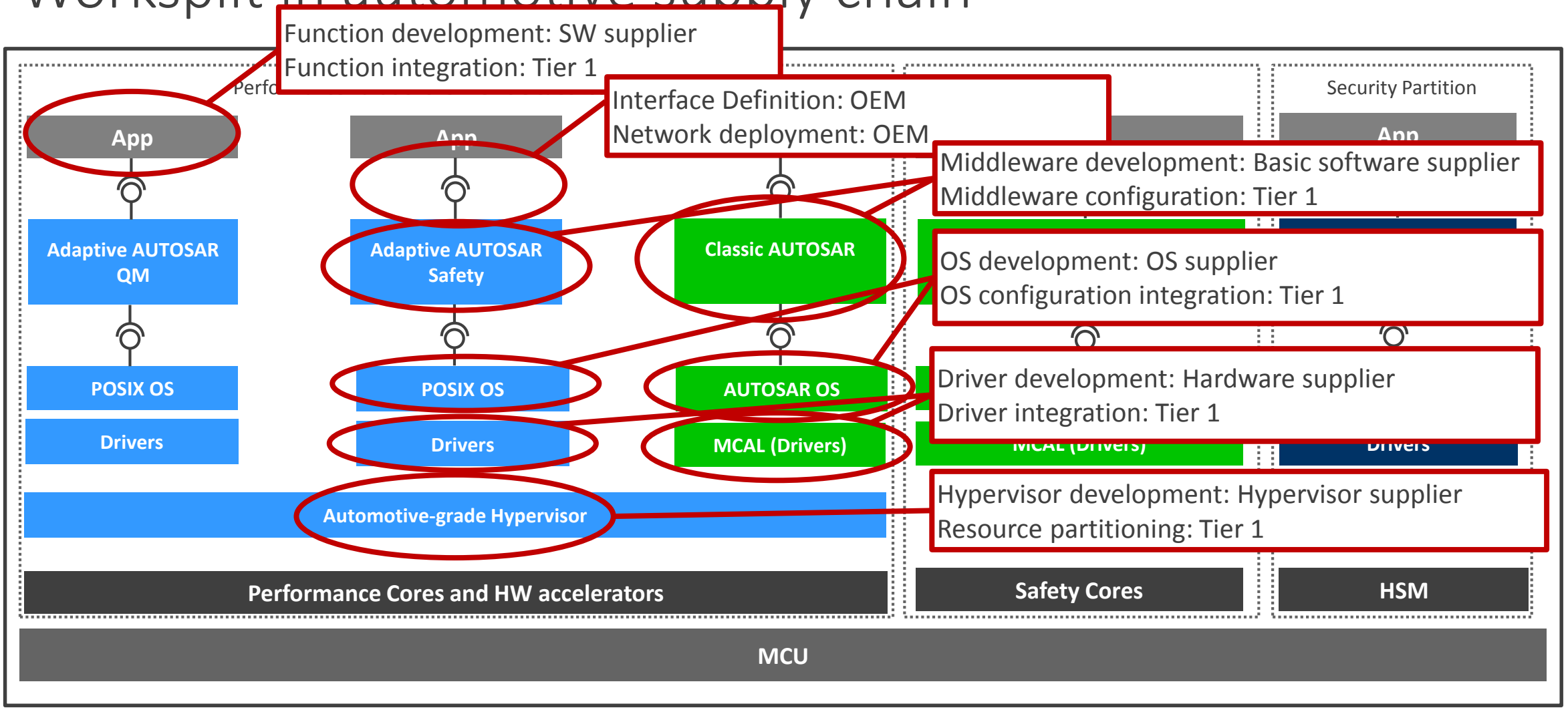
Elektrobit

Possible high performance controller architecture for SOP in 2019



Elektrobit

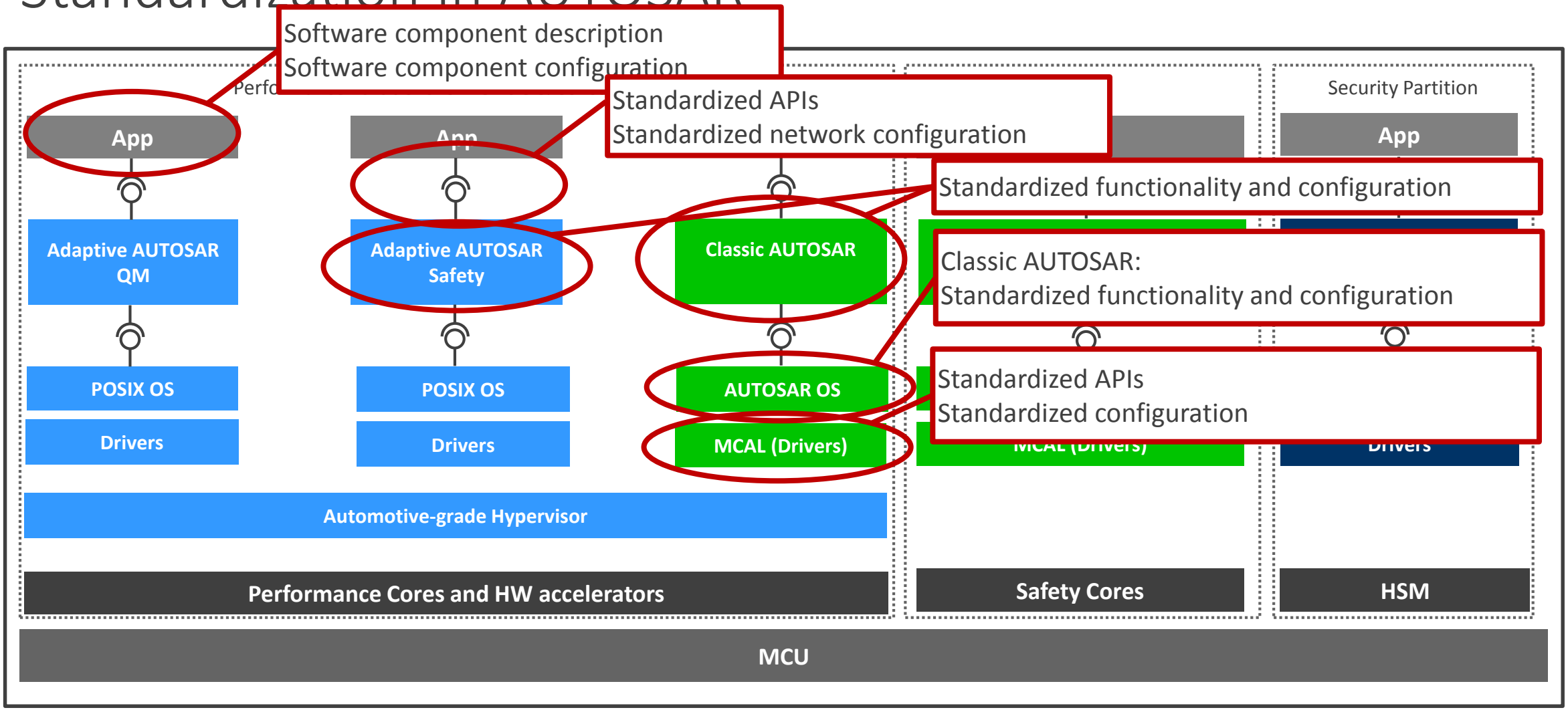
Worksplit in automotive supply chain



16

Elektrobit

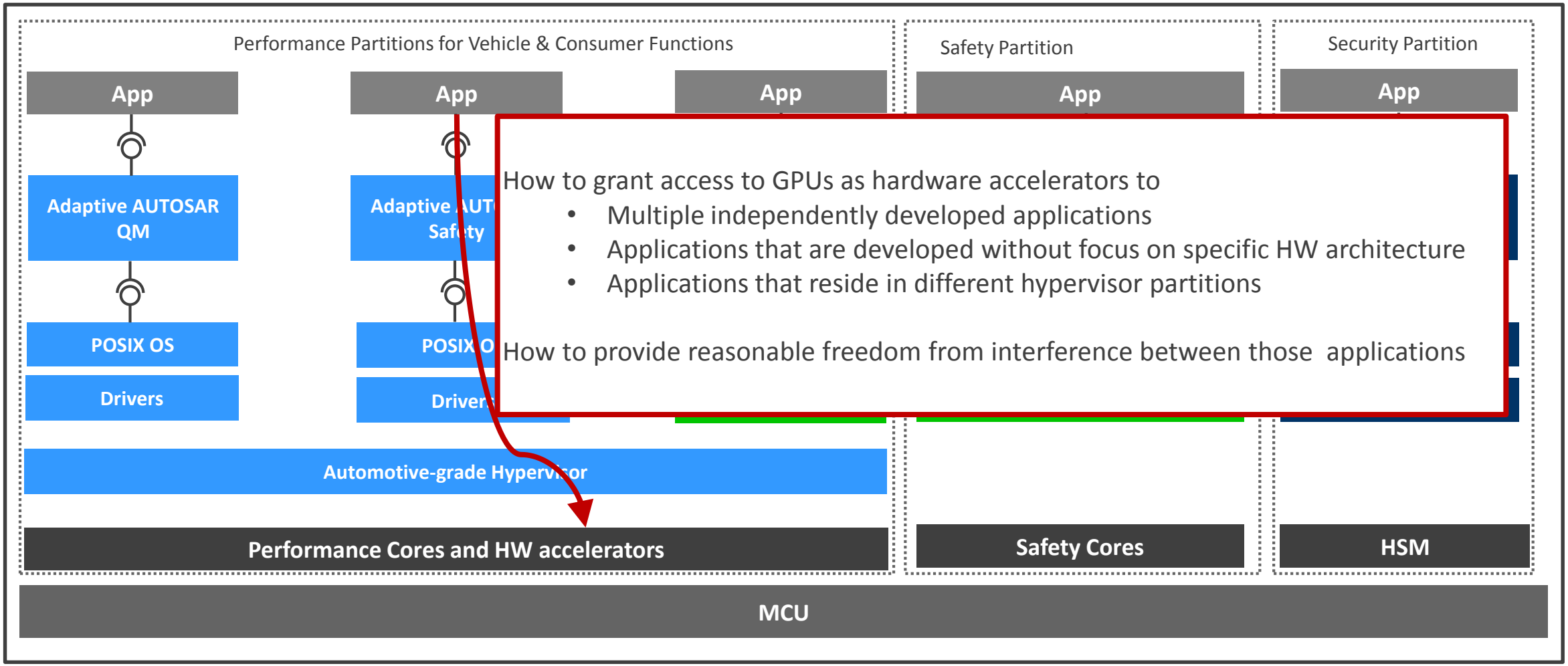
Standardization in AUTOSAR



17

Elektrobit

How to use GPUs for automotive applications?



How to grant access to GPUs as hardware accelerators to

- Multiple independently developed applications
- Applications that are developed without focus on specific HW architecture
- Applications that reside in different hypervisor partitions

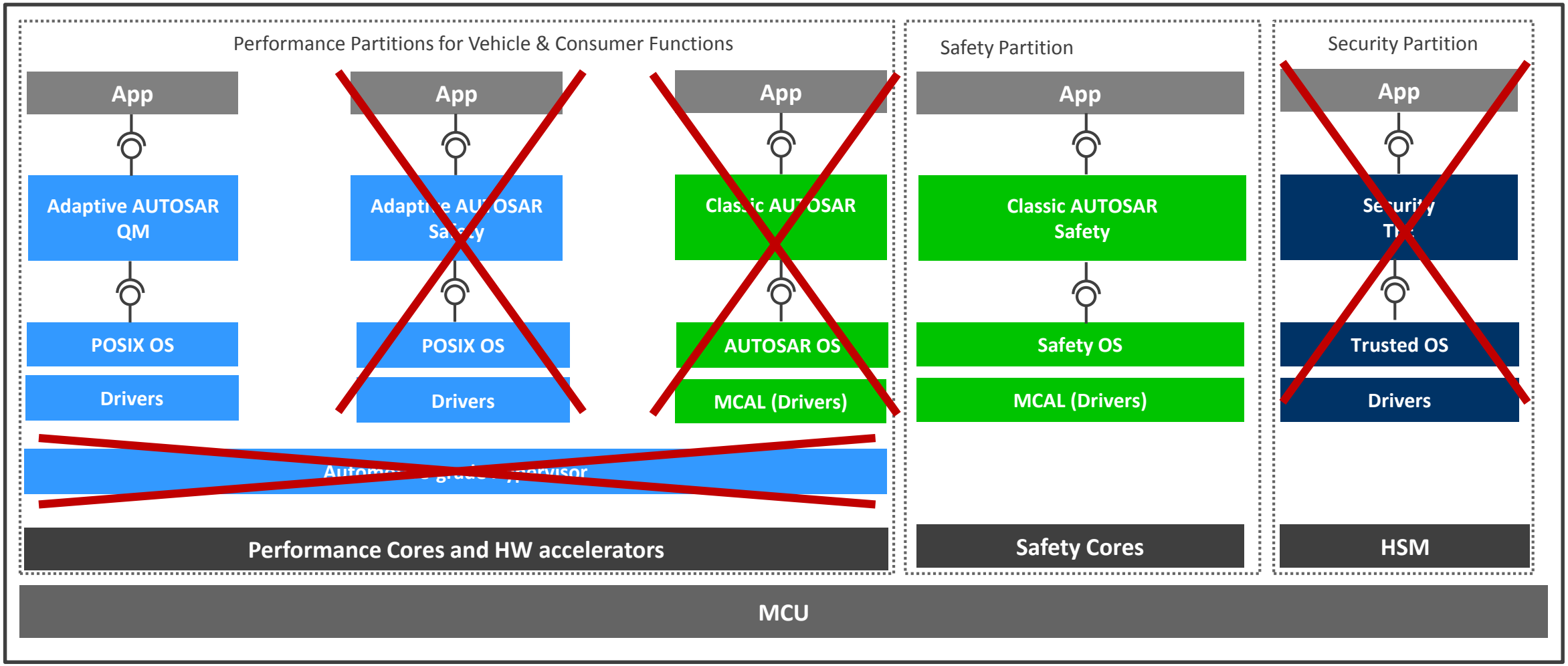
How to provide reasonable freedom from interference between those applications

Integrating GPUs in software infrastructure

Elektrobit

Elektrobit

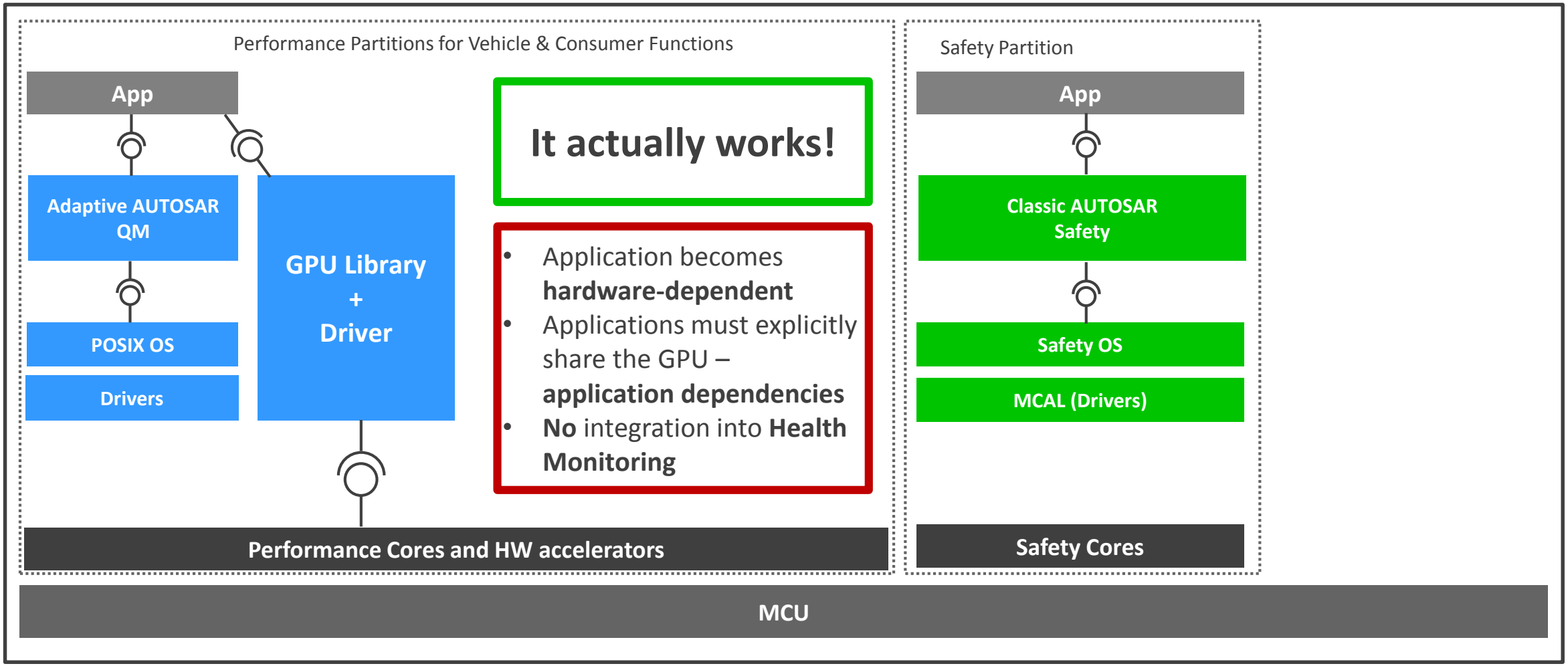
Current integration in (most) prototypes



20

Elektrobit

Current integration in (most) prototypes

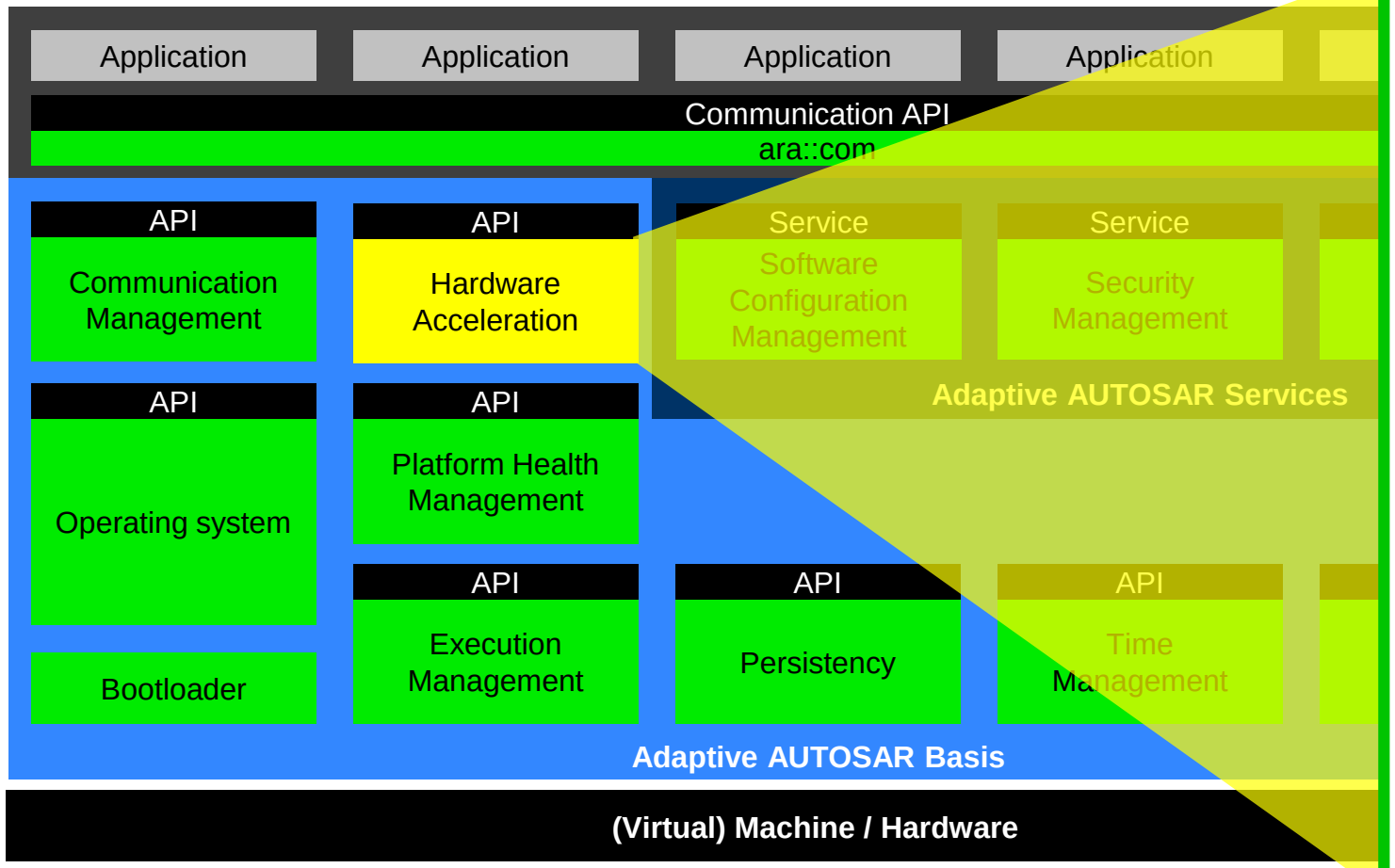


It actually works!

- Application becomes **hardware-dependent**
- Applications must explicitly share the GPU – **application dependencies**
- **No** integration into **Health Monitoring**

21

Standardized basic software – Adaptive AUTOSAR



Goal:

- Uncouple application design from the hardware it is deployed upon

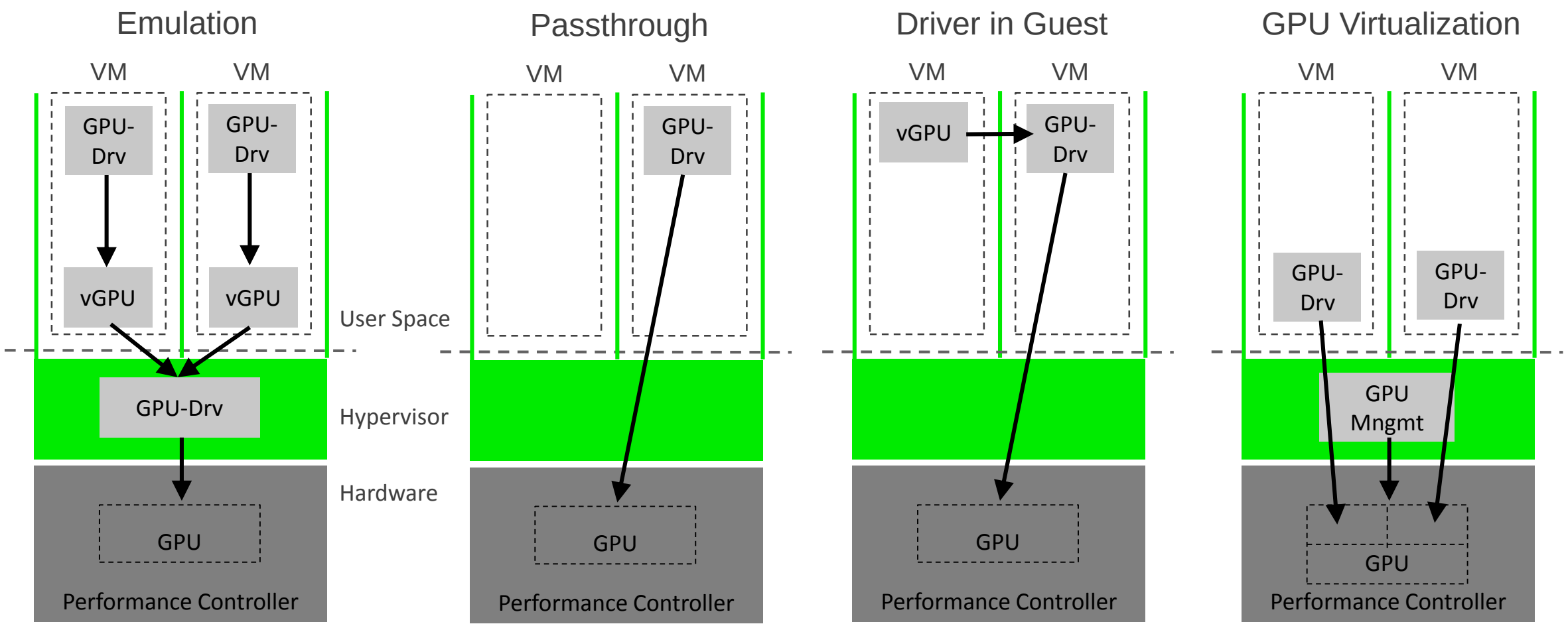
Functionality to be included:

- Libraries for using GPUs for ADAS applications
  - E.g. Tensorflow, Parallel STL, OpenCV
  - Build on top of e.g. CUDA, SYCL

Likely not standardized before 03/2019

Elektrobit

Sharing a GPU across virtual machines



Are we done yet?

What have we achieved so far:

- Standardization of APIs and abstraction from hardware
  - Allows to develop applications independent of ECU projects
- Enabling GPU virtualization
  - Allows sharing hardware resources in a more efficient manner

Integration should work. Now let's make things safe and secure.

The reality: Automotive recalls in the US

Data courtesy of NHTSA

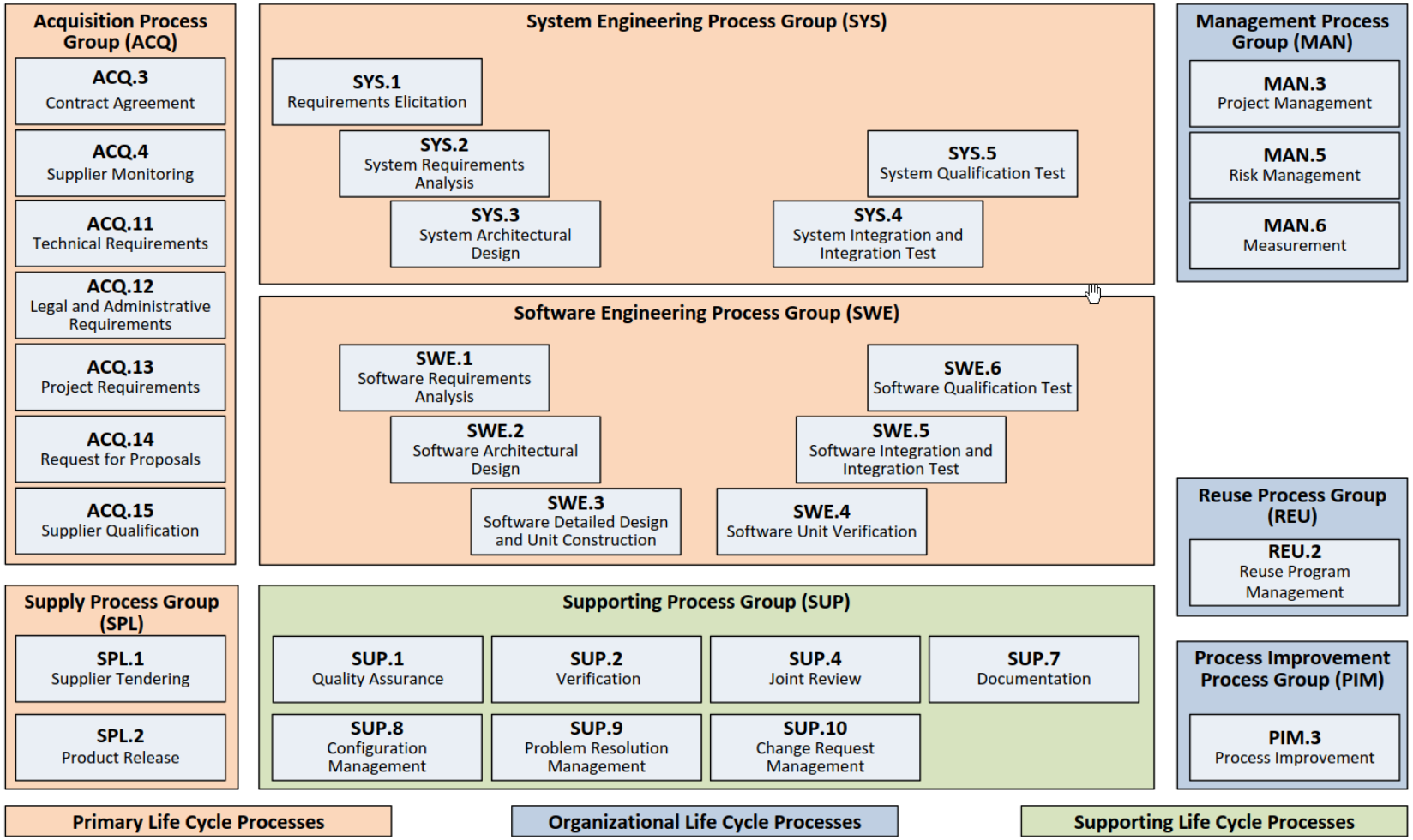
	Vehicles affected (Milion)	Vehicles affected (Per day)	Recalls (Total)	Vehicles sold (Million)	Recalls per sold vehicle
2014	51.0	139,726	779	16.45	3.1
2015	51.2	140,274	868	17.39	2.9
2016	53.2	145,753	927	17.46	3.0

- Software-related recalls have gone from less than 5 % in 2011 to 15 % by end of 2015[1]
  ➔ roughly 7.68 Million software related car recalls in 2015
- Software vendors are liable for their products in the automotive sector

[1]Stout Risius Ross (SSR), Automotive Warranty & Recall Report 2016

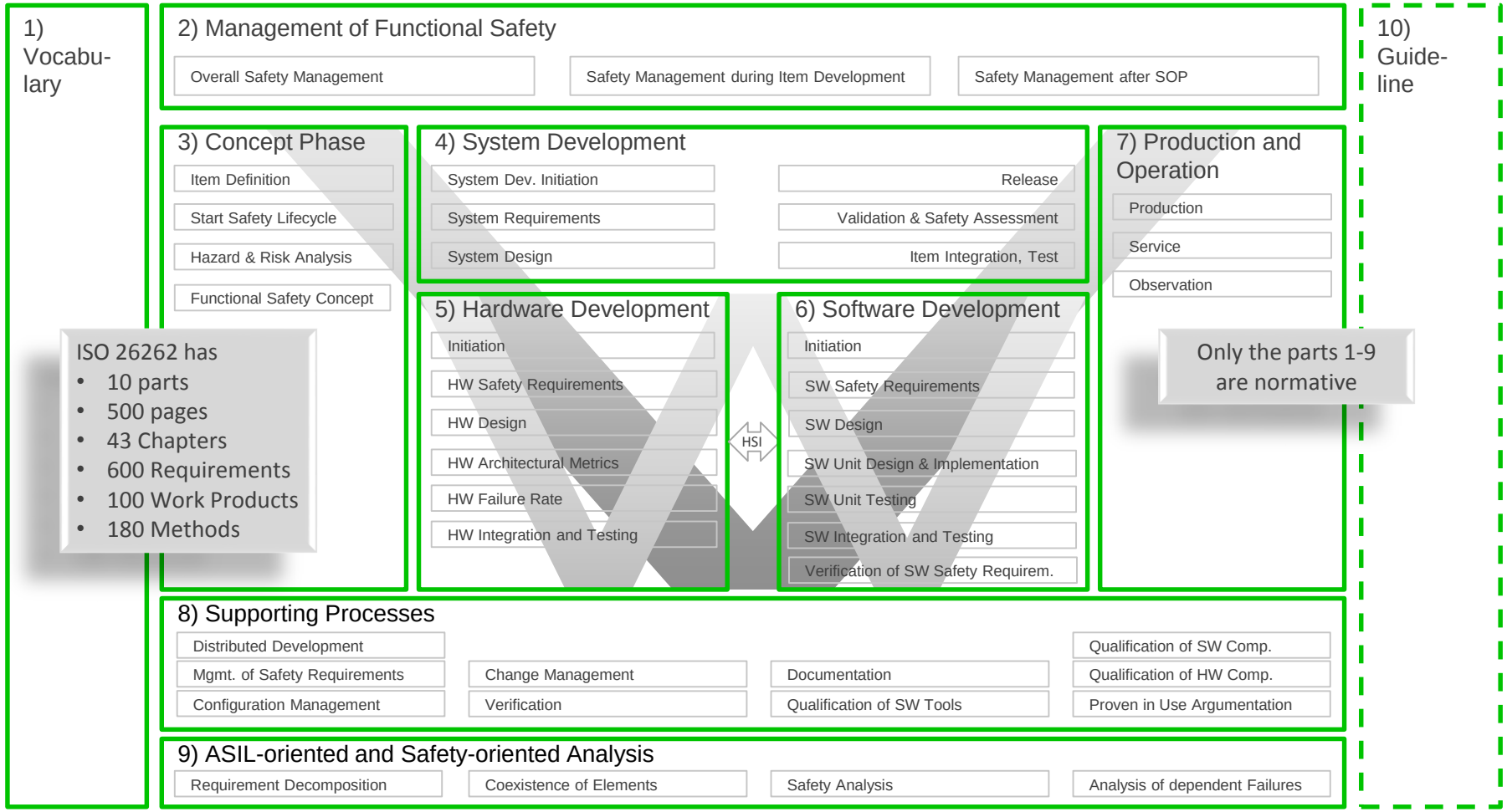
Software development for automotive (ASPICE)

- Quality- and verification driven
- Common standard: Automotive SPICE v3.1 for development processes
- In addition: most OEMs have further standards
- Process assessments are regularly performed by customers:
  OEM -> Tier-1 -> Tier-2 -> ...
- A Tier-1 is fully liable for the complete product, including the software

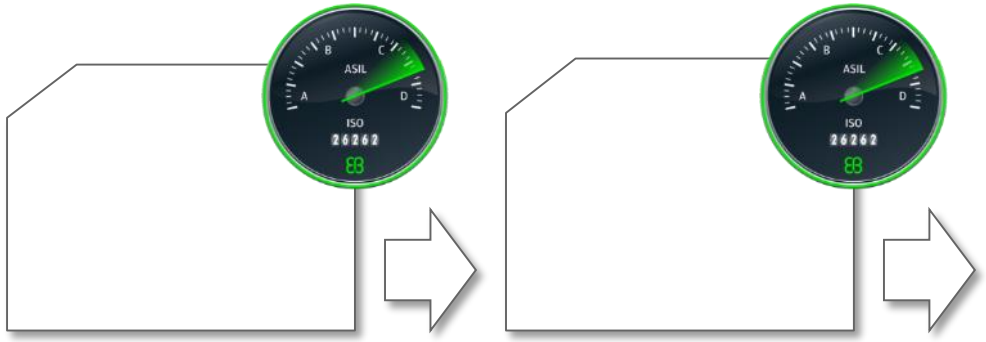


Elektrobit

Functional Safety according to ISO 26262



Functional Safety Requirements (ISO 26262)



Functional Description

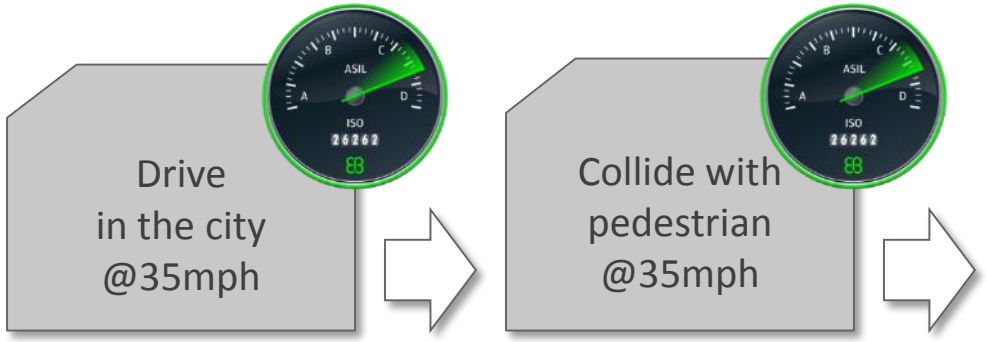
Hazard & Risk Analysis

Safety Requirements

Automotive Safety Integrity Level ASIL

ASIL = Severity x Exposure x **Controllability**

Functional Safety Requirements (ISO 26262)

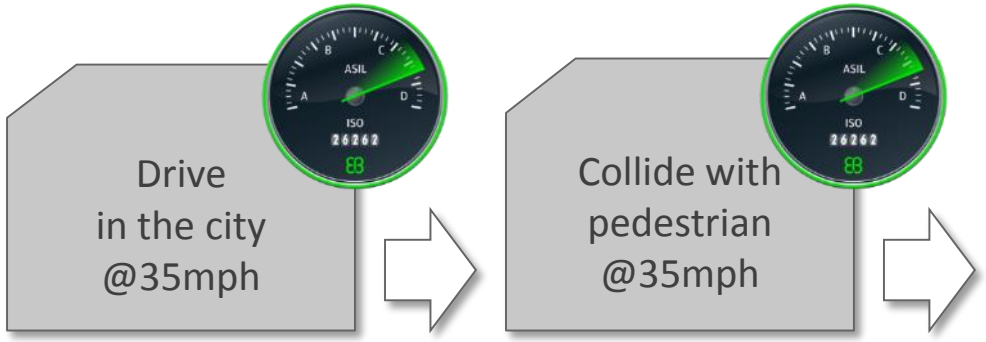


Functional Description **Hazard & Risk Analysis**

Safety Requirements

Automotive Safety Integrity Level ASIL
ASIL = Severity x Exposure x **Controllability**

Functional Safety Requirements (ISO 26262)



Functional Description **Hazard & Risk Analysis**

Safety Requirements

Automotive Safety Integrity Level ASIL
ASIL = Severity x Exposure x **Controllability**

		C1	C2	C3
S1	E1	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	A
	E4	QM	A	B
S2	E1	QM	QM	QM
	E2	QM	QM	A
	E3	QM	A	B
	E4	A	B	C
S3	E1	QM	QM	A
	E2	QM	A	B
	E3	A	B	C
	E4	B	C	D

Hazard and Risk Assessment in a nutshell

Hazard
Potential source of harm, caused by malfunctioning behavior of the item
Example: Not detecting a pedestrian on the street

Severity (S0-S3)
Impact of the harm
*Example: Life-threatening injuries => **S3***

Operational situation
Scenario that can occur during a vehicle's life, in which the hazard creates a considerable risk
Example: Driving at speed > 35 mph on a street operating with an activated Highway Chauffeur

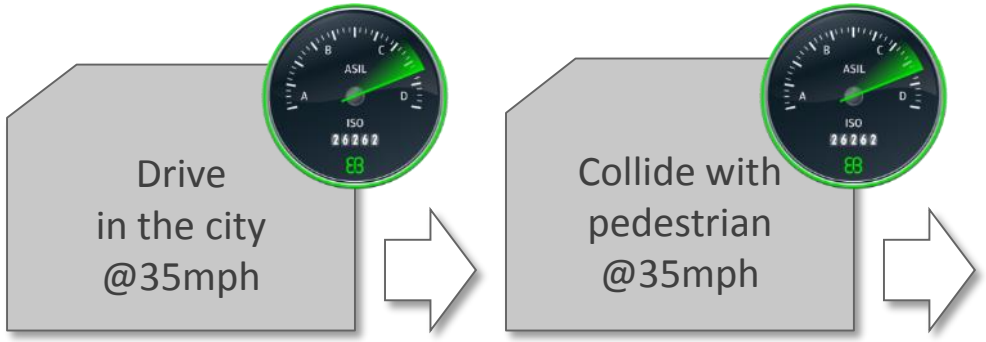
Exposure (E0-E4)
Probability of the operational scenario
*Example: High probability => **E4***

Possible avoidance actions
Things that the driver or other involved person are able to do in order to avoid the harm
Example: Driver immediately brakes with full power

Controllability (C0-C3)
Probability that one of the involved persons is able to avoid the harm
*Example: Normally not controllable => **C3***

		C1	C2	C3
S1	E1	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	A
	E4	QM	A	B
S2	E1	QM	QM	QM
	E2	QM	QM	A
	E3	QM	A	B
	E4	A	B	C
S3	E1	QM	QM	A
	E2	QM	A	B
	E3	A	B	C
	E4	B	C	D

Functional Safety Requirements (ISO 26262)



Functional Description **Hazard & Risk Analysis**

Safety Requirements

Severity 3: Severe injuries expected.

Exposure 4: Pedestrians in car's path likely. = **ASIL-D**

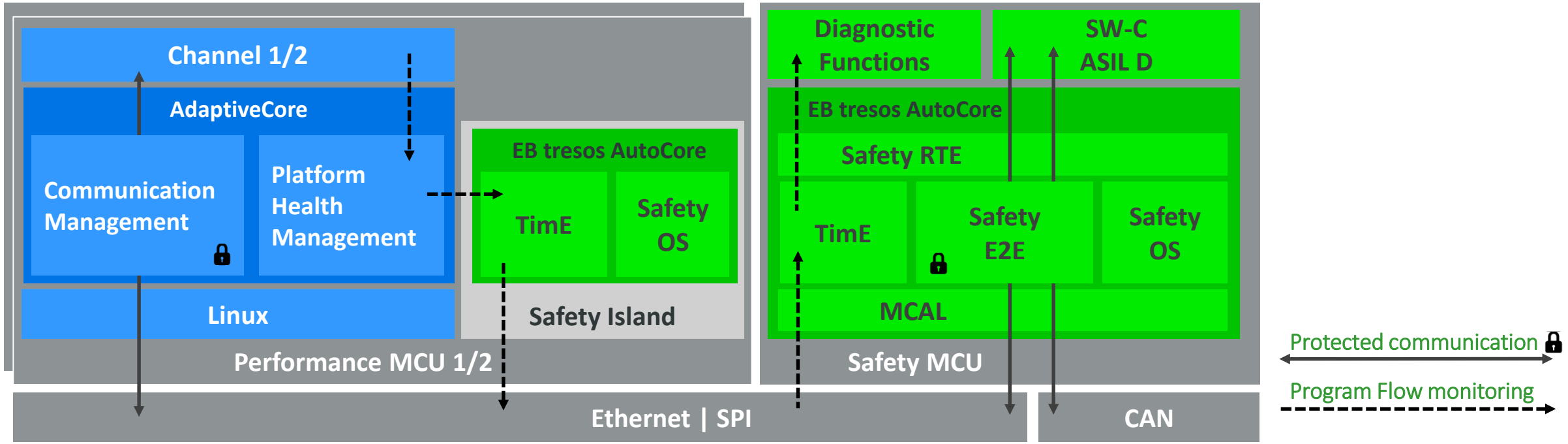
Controllability 3: Driver is sleeping.

ASIL = Severity x Exposure x **Controllability**

		C1	C2	C3
S1	E1	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	A
	E4	QM	A	B
S2	E1	QM	QM	QM
	E2	QM	QM	A
	E3	QM	A	B
	E4	A	B	C
S3	E1	QM	QM	A
	E2	QM	A	B
	E3	A	B	C
	E4	B	C	D

Strategies to achieve safety for highly automated driving

- Redundant channels (e.g.: 1oo2D)
- 3 Layer safety solution (E-Gas concept)
- Safety parts fully certified up to ASIL-D



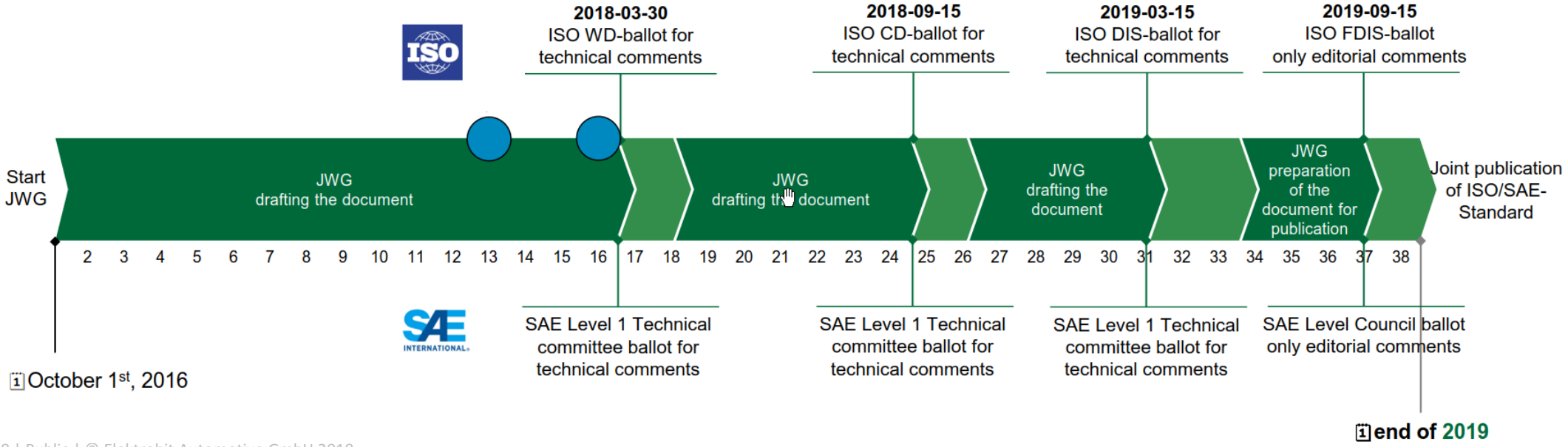
Cybersecurity engineering

- The standard is being worked on:
  ISO/SAE 21434 – Cybersecurity Engineering
- Contains general requirements, will probably include requirements for software updates
- Publication: end of 2019

→ Projects with an **SOP in 2019 / 2020** need to take it into account.

General Outline (Draft)

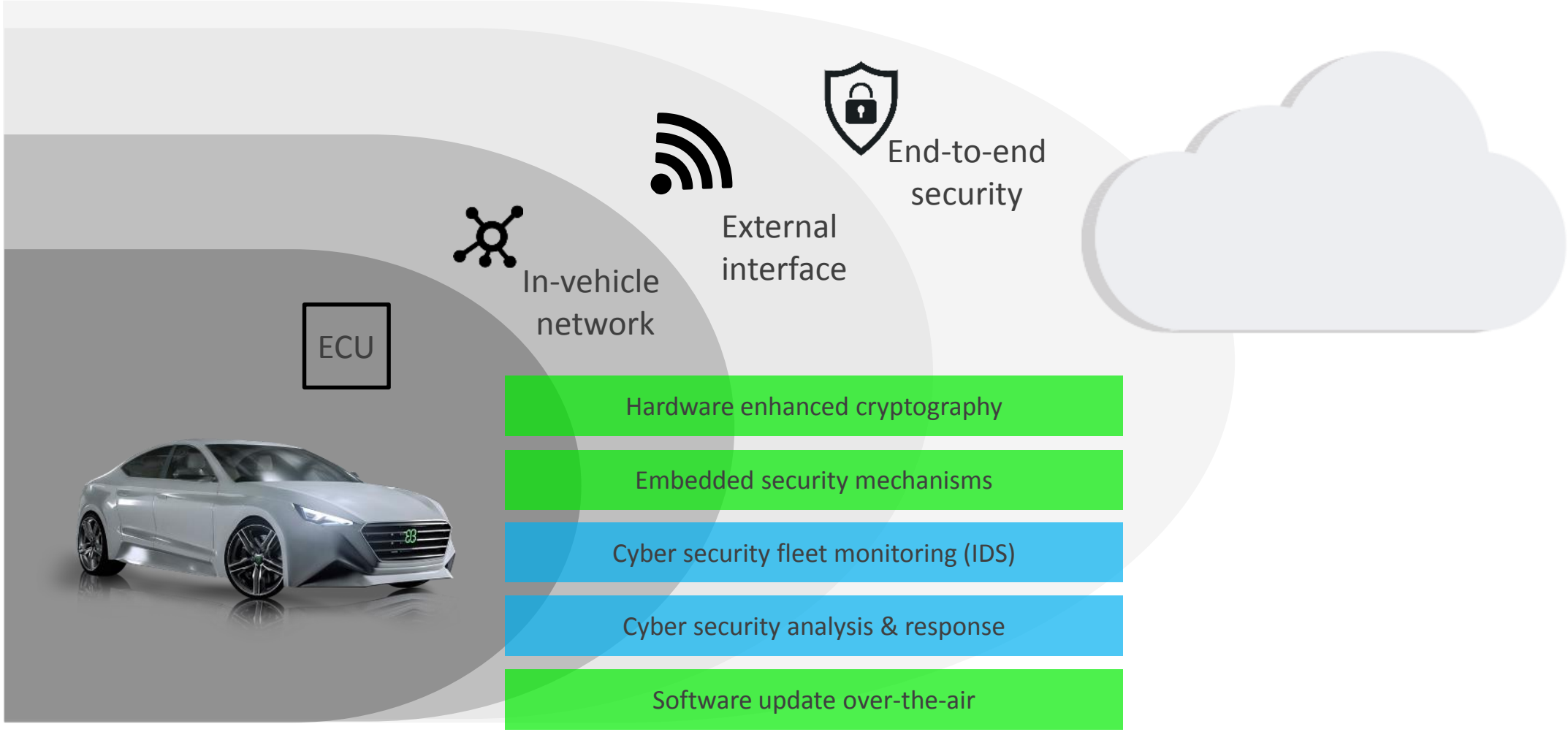
1 Scope
2 Normative references
3 Terms and Definitions
4 Introduction
5 Cybersecurity Engineering – General Requirements
6 Cybersecurity Management – Govern Process
7 Cybersecurity Management – Engage Process
8 Cybersecurity Engineering – Develop Process
9 Cybersecurity in Production & Post-production
10 Appendixes



Elektrobit

ARGUS

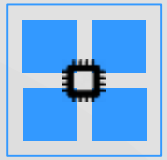
Security solutions



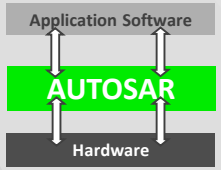
EB

Argus

Summary



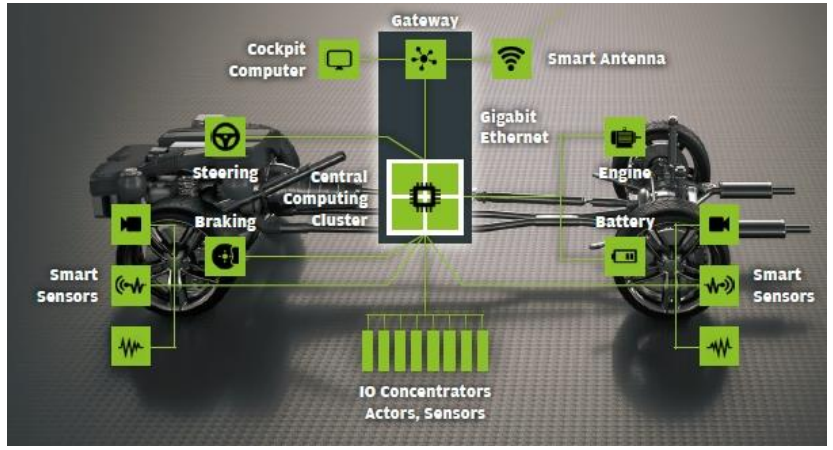
Standardization of Middleware to develop application independent for high performance ECUs



Enabling standardized GPU virtualization
Allows sharing hardware resources in a efficient way

EB extended the safety product portfolio to fulfill the requirements of high performance ECUs

EB & Argus are offering an end-to-end-security solution for the future vehicle architecture



EB tresos solution for NVIDIA DRIVE PX 2

Adaptive & Classic AUTOSAR for DrivePX planned to be available at October 2018!

- Integrating technologies from NVIDIA, Infineon (safety processor) and EB
- The EB software provides seamless integration capability of Linux and AUTOSAR applications

Visit our booth at the GTC

#532

Questions?

Elektrobit

sales@elektrobit.com
www.elektrobit.com