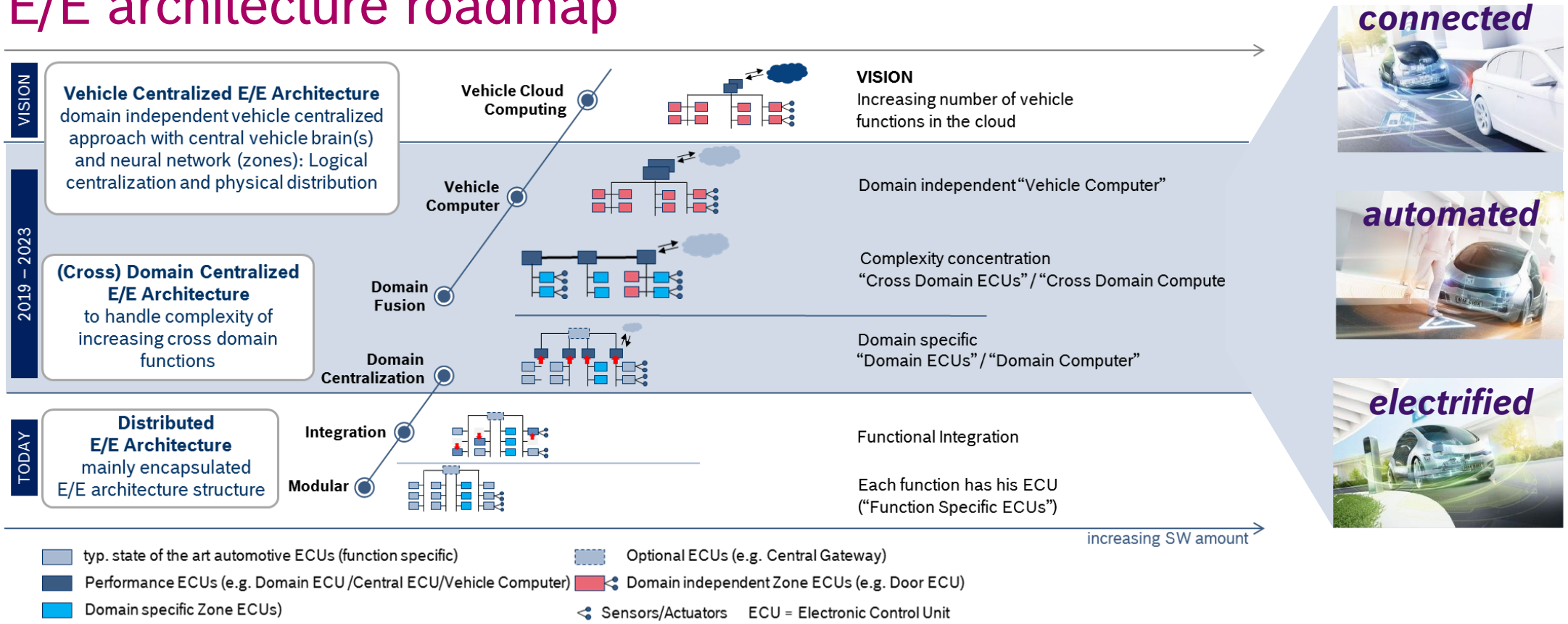
THE FUTURE AUTOMOTIVE OPERATING SYSTEM

ROBERT BOSCH GMBH – GUNNAR PIEL

BOSCH

VRTE - Vehicle Runtime Environment

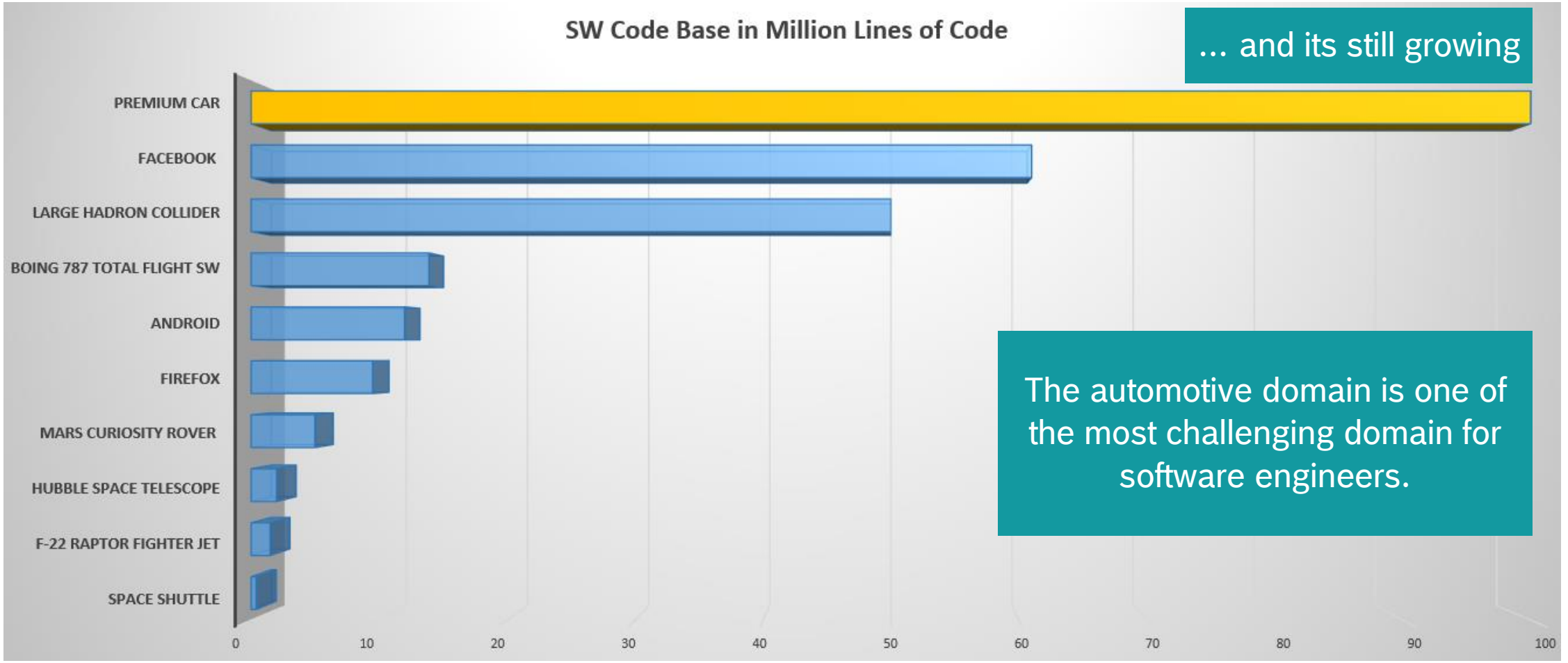
E/E architecture roadmap



Vehicle Computer as integral part in 2019ff. New architectures vary with legacy constraints.

VRTE - Vehicle Runtime Environment

Total Amount of Software in Modern Cars



... and its still growing

The automotive domain is one of the most challenging domain for software engineers.

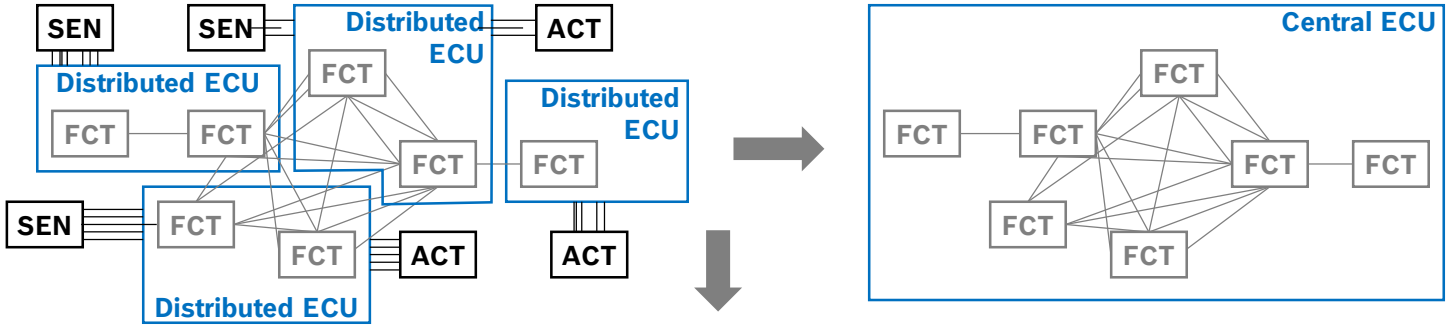
Source of data: http://www.informationisbeautiful.net/visualizations/million-lines-of-code/

BOSCH

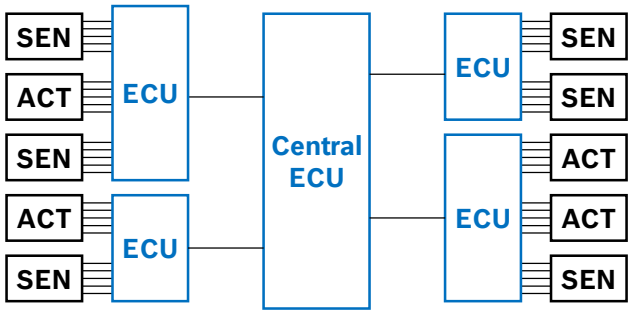
VRTE - Vehicle Runtime Environment
Software in centralized EE-architecture

EE-architecture

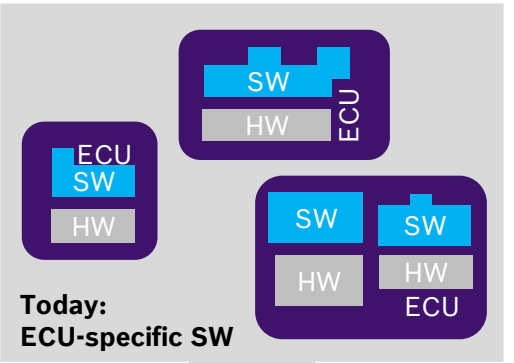
- Further **logical centralization** due to increasing interconnection of functions (highly interconnected, distributed functions more complex than integration)



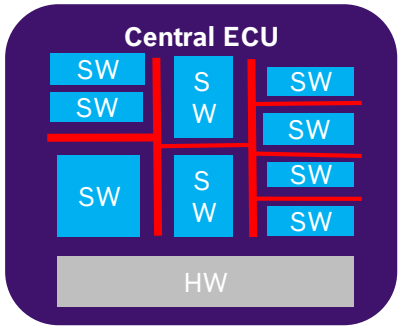
- **Physical distribution** into Central ECU & deeply embedded ECUs



SW-architecture

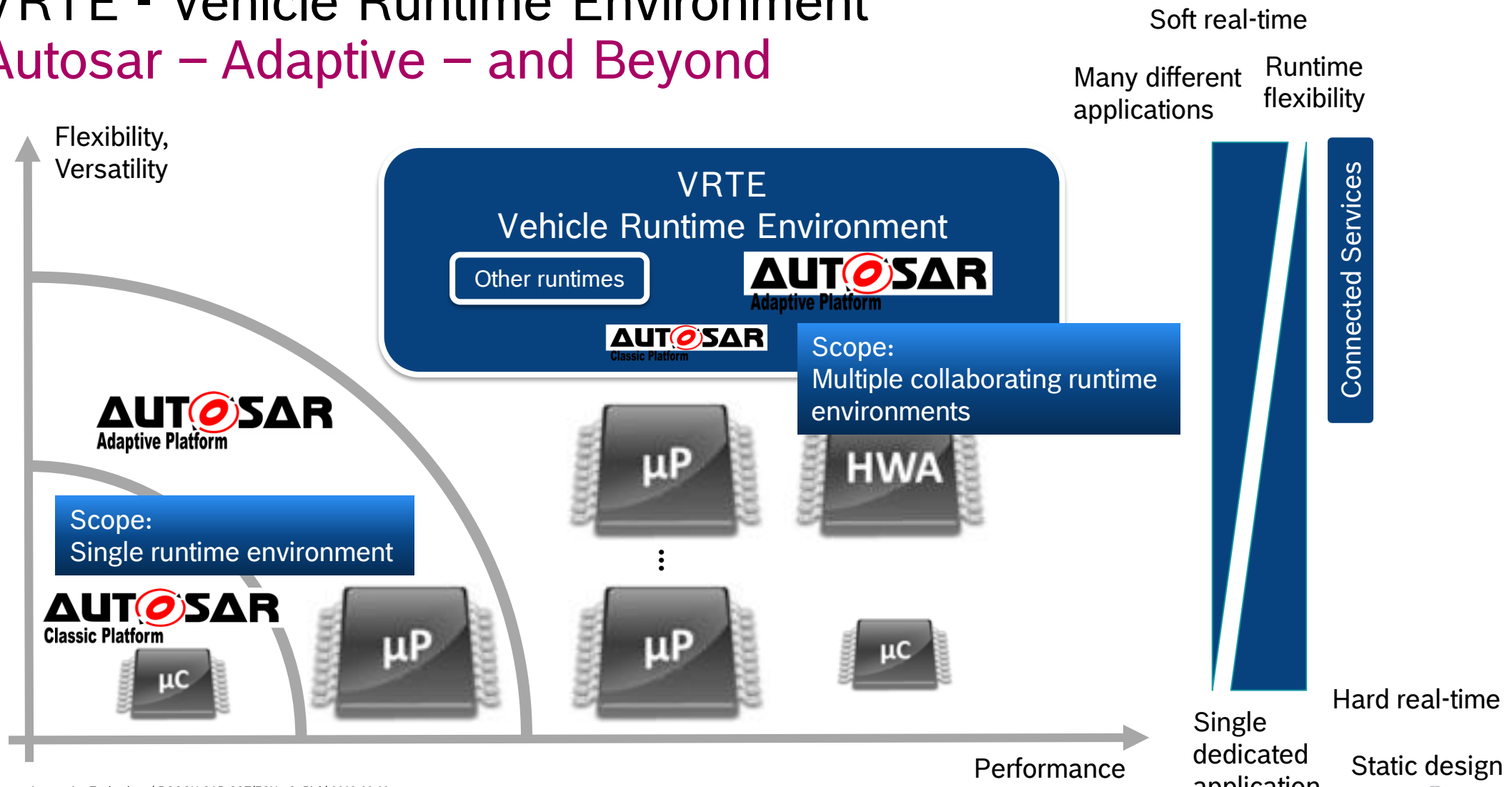


EE-architecture drives SW-architecture



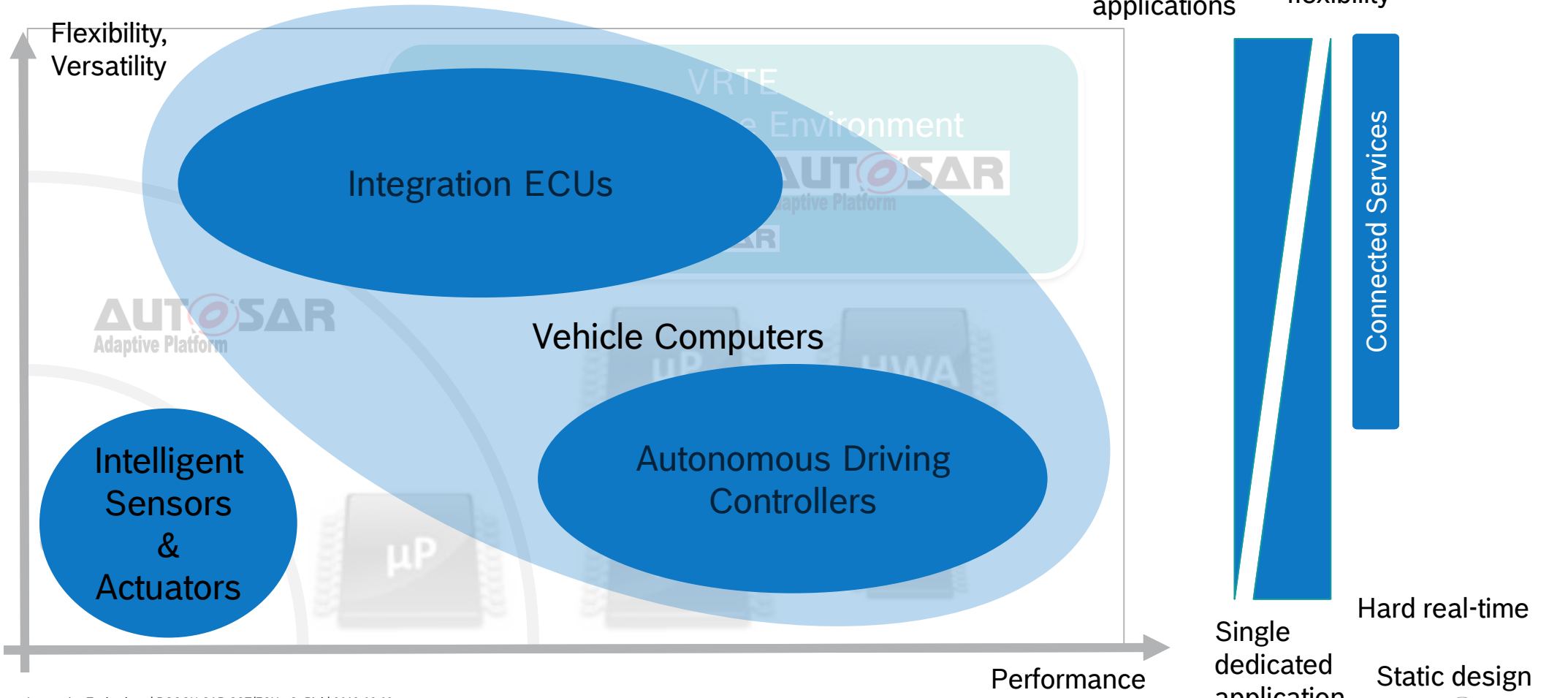
BOSCH

VRTE - Vehicle Runtime Environment
Autosar – Adaptive – and Beyond



VRTE - Vehicle Runtime Environment

Vehicle Computers



VRTE - Vehicle Runtime Environm

Most important features

HW – SW separation
- ❑ Standard HW
- ❑ HW sourced separately from SW
- ❑ SW business

SW integration
- ❑ High computing performance, µP
- ❑ Many different applications
- ❑ Heterogeneous real-time, safety & security properties
- ❑ Many different SW suppliers
- ❑ Different SW development processes, tools & SW lifecycles
- ❑ Service oriented architecture
- ❑ Delta V&V

SW customization
- ❑ Lifetime SW modifications & extensions e.g. security patches
- ❑ Dynamic SW composition

Connectivity
- ❑ FOTA, SOTA
- ❑ Cloud services
- ❑ V2X
- ❑ Security

Tooling & SDK
- ❑ High UX
- ❑ Efficient development → TTM
- ❑ Modular & integratable tool chain

System of systems
- ❑ Reuse SW systems
- ❑ Cross-domain/BU
- ❑ Strong collaboration
- ❑ Demands organizational changes

Dependability
- ❑ Availability
- ❑ Safety
- ❑ Security
- ❑ Real-time
- ❑ Fault tolerant SW systems

HW acceleration
- ❑ AI, Neural networks
- ❑ Graphics sharing

Portability
- ❑ HW & VMM & OS independence
- ❑ Service orientation architecture
- ❑ Communication channel independence

Networking
- ❑ Gbit Ethernet
- ❑ Deterministic communication (TSN)
- ❑ Time synchronization

IT-like SW
- ❑ OSS & COTS SW
- ❑ Heterogeneous processes
- ❑ Continuous delivery
- ❑ Service oriented architecture
- ❑ Tools

Migration
- ❑ Integration of legacy SW
- ❑ Porting of legacy SW

VRTE - Vehicle Runtime Environm

Most important features

HW – SW separation
- ❑ Standard HW
- ❑ HW sourced separately from SW
- ❑ SW business

SW integration
- ❑ High computing performance, µP
- ❑ Many different applications
- ❑ Heterogeneous real-time, safety & security properties
- ❑ Many different SW suppliers
- ❑ Different SW development processes, tools & SW lifecycles
- ❑ Service oriented architecture
- ❑ Delta V&V

SW customization
- ❑ Lifetime SW modifications & extensions e.g. security patches
- ❑ Dynamic SW composition

Connectivity
- ❑ FOTA, SOTA
- ❑ Cloud services
- ❑ V2X
- ❑ Security

Tooling & SDK
- ❑ High UX
- ❑ Efficient development → TTM
- ❑ Modular & integratable tool chain

System of systems
- ❑ Reuse SW systems
- ❑ Cross-domain/BU
- ❑ Strong collaboration
- ❑ Demands organizational changes

Dependability
- ❑ Availability
- ❑ Safety
- ❑ Security
- ❑ Real-time
- ❑ Fault tolerant SW systems

HW acceleration
- ❑ AI, Neural networks
- ❑ Graphics sharing

Portability
- ❑ HW & VMM & OS independence
- ❑ Service orientation architecture
- ❑ Communication channel independence

Networking
- ❑ Gbit Ethernet
- ❑ Deterministic communication (TSN)
- ❑ Time synchronization

IT-like SW
- ❑ OSS & COTS SW
- ❑ Heterogeneous processes
- ❑ Continuous delivery
- ❑ Service oriented architecture
- ❑ Tools

Migration
- ❑ Integration of legacy SW
- ❑ Porting of legacy SW

BOSCH

VRTE - Vehicle Runtime Environment

What drives VRTE architecture

Freedom from risk

Health and safety risk mitigation (Functional safety)
VRTE enables SW and HW development to support safety goals up to ISO26262 ASIL D.

Security

Integrity
VRTE contributes to maintain the integrity of the SW with respect to growing security risks from connectivity and complex SW systems by tightly controlling individual privileges.

Maintainability

Reusability
VRTE is designed to be used in many different ECU products by various Bosch divisions. It is offered as infrastructure SW product to the open market.

Reliability

Fault tolerance
VRTE provides freedom from interference by preventing individual SW faults from compromising the whole SW system.

Compatibility

Interoperability
VRTE contributes to the interoperability of SW components from different internal and external suppliers as well as interoperability between established SW de facto-standards such as Autosar Classic, Autosar Adaptive, Genivi etc.

Portability

Adaptability
VRTE is designed to be easily adaptable to evolving HW platforms, usage scenarios, products and new SW components.

Product driven

Business & Technology driven

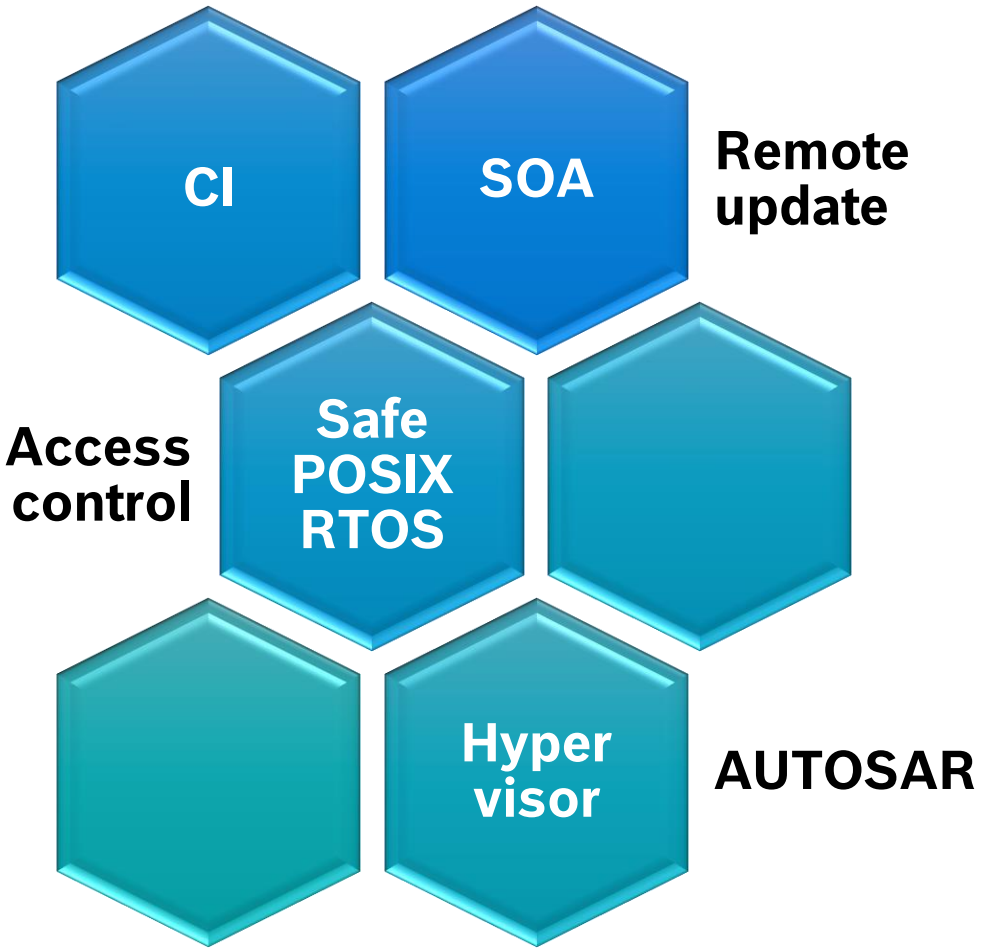
Priority / Precedence

Terminology according to ISO25010 software quality.

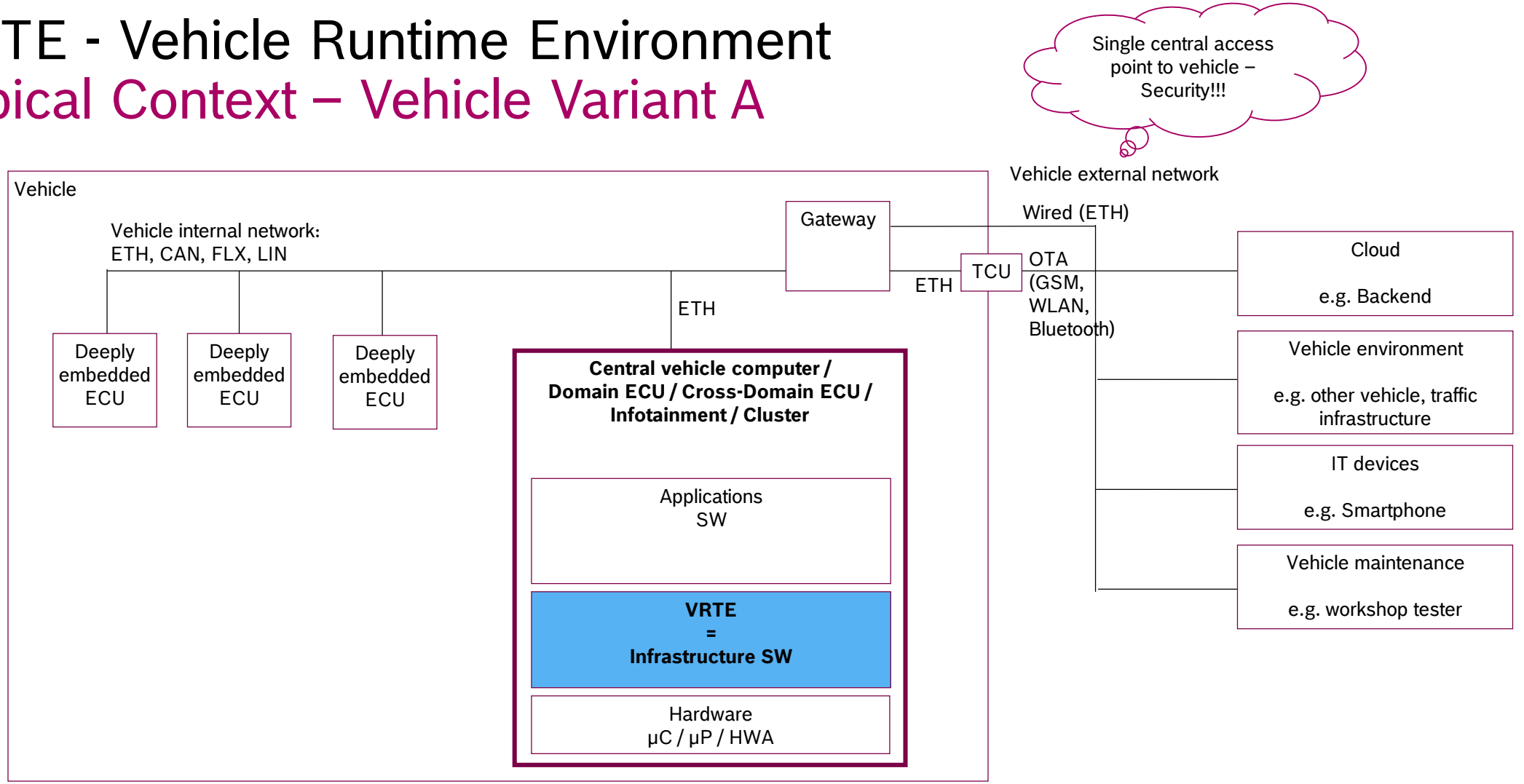
BOSCH

VRTE - Vehicle Runtime Environment

Key SW technologies

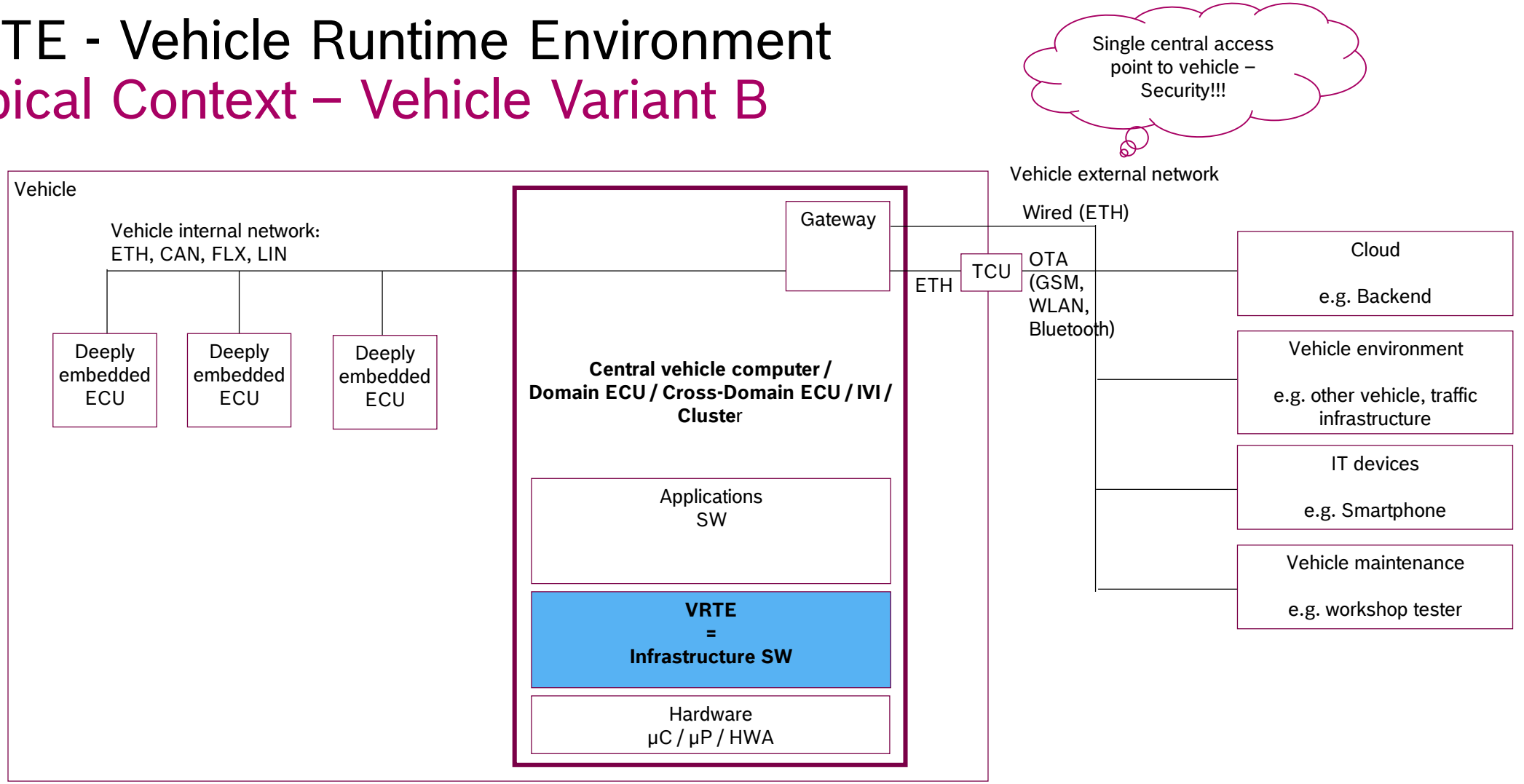


VRTE - Vehicle Runtime Environment
Typical Context – Vehicle Variant A



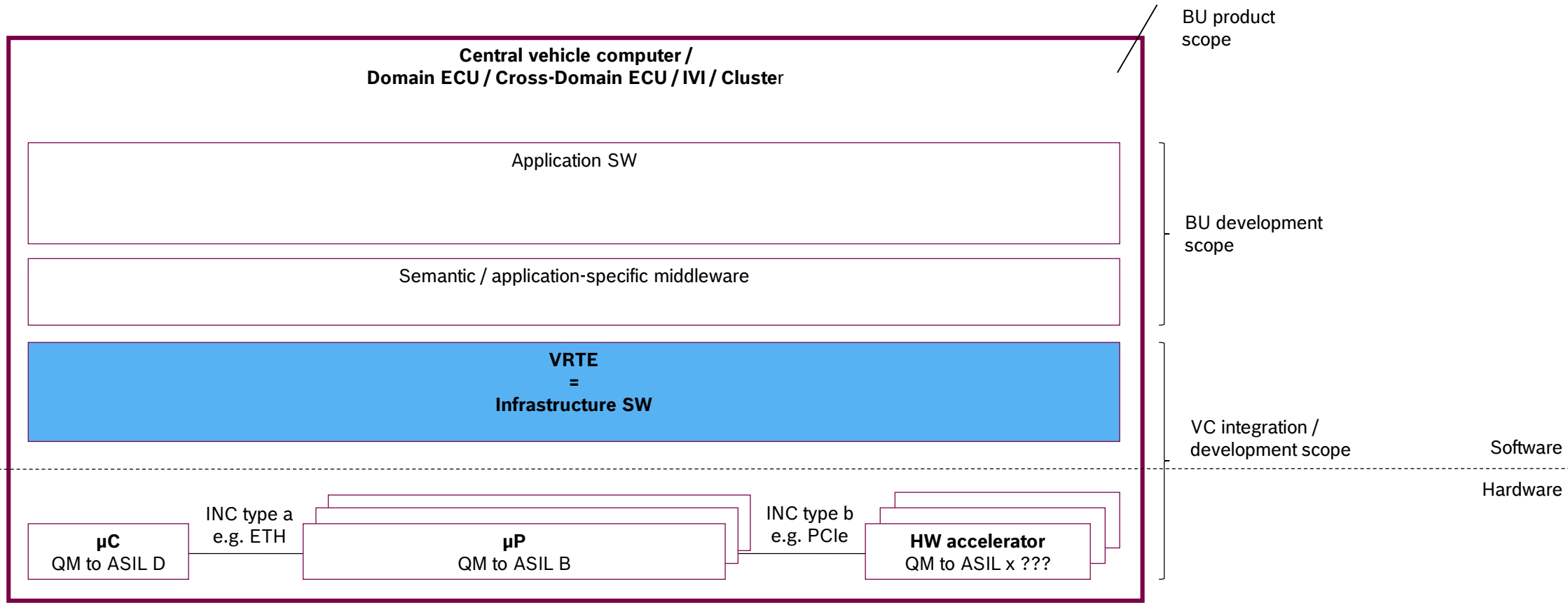
TCU: Telematics control unit **OTA**: Over the air **ETH**: Ethernet **µC**: Microcontroller **µP**: Microprocessor **HWA**: Hardware accelerator

VRTE - Vehicle Runtime Environment
Typical Context – Vehicle Variant B



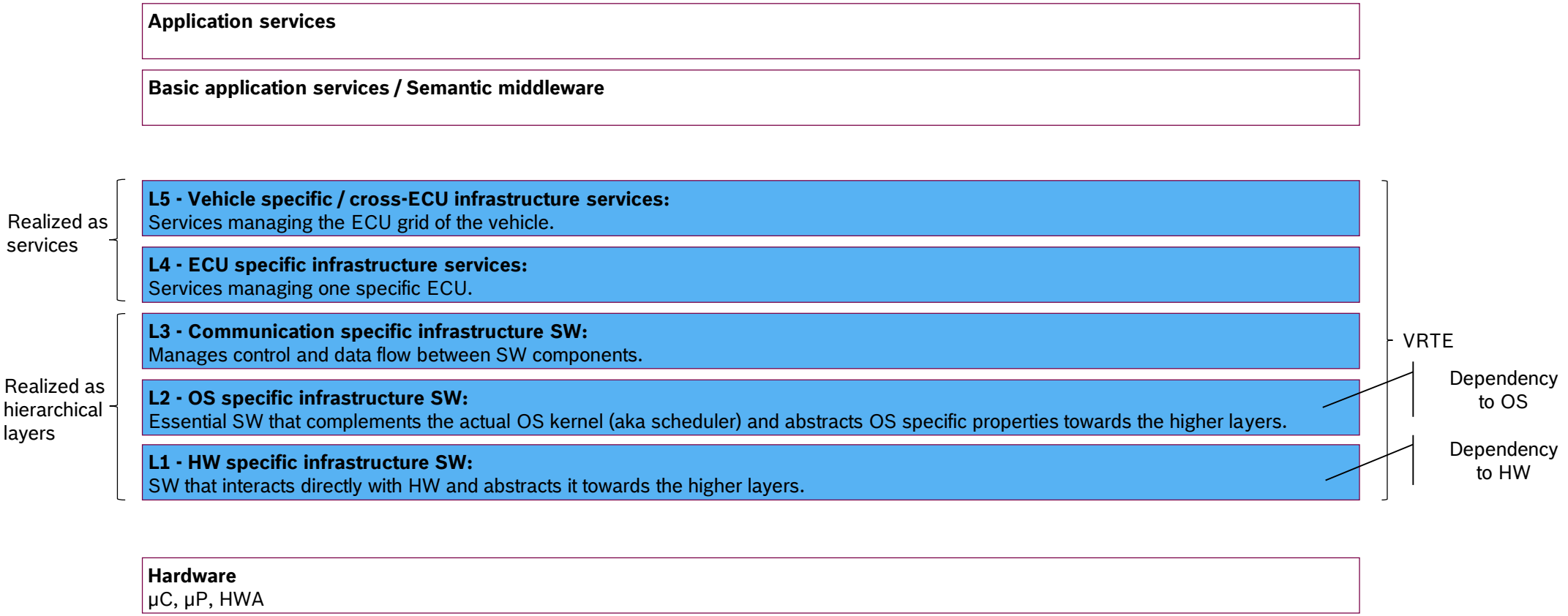
TCU: Telematics control unit **OTA**: Over the air **ETH**: Ethernet **µC**: Microcontroller **µP**: Microprocessor **HWA**: Hardware accelerator

VRTE - Vehicle Runtime Environment
Typical Context – SW & HW



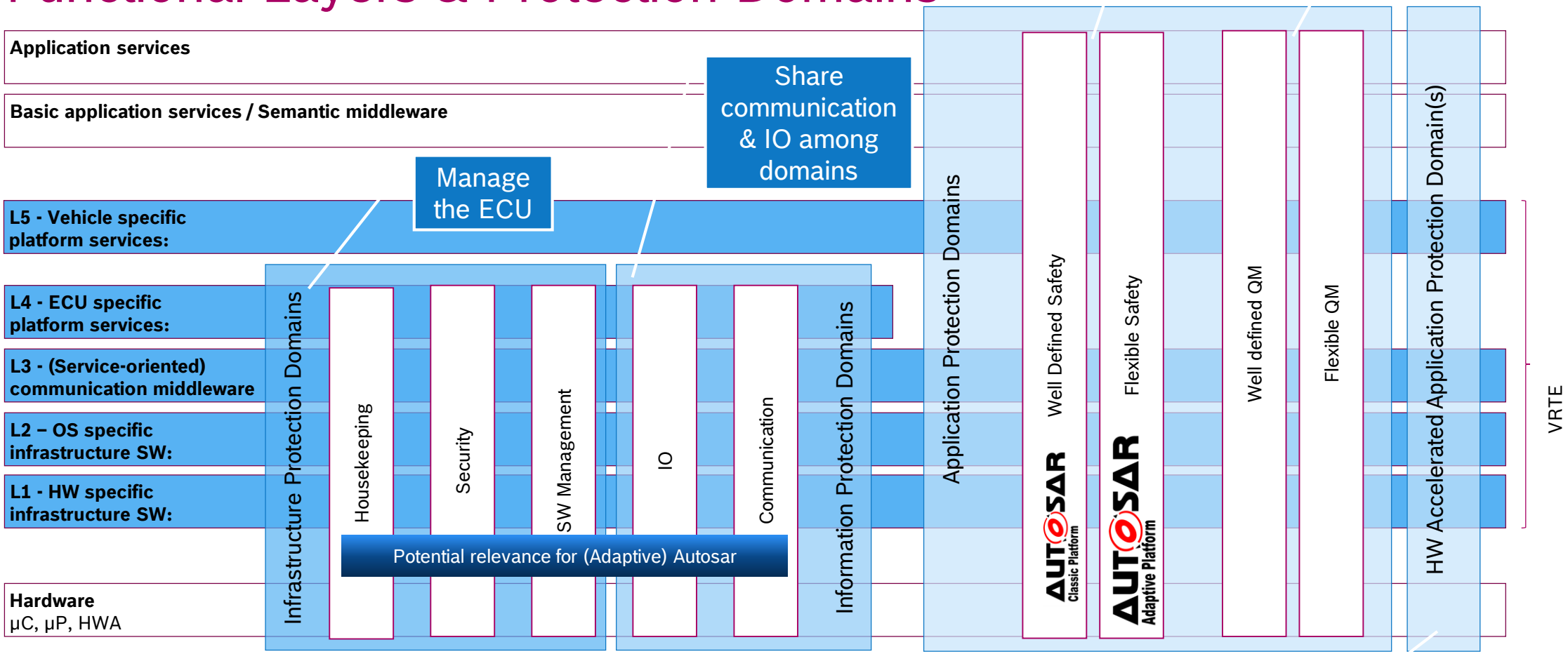
BU: RB business unit **VC:** RB Vehicle computer campus **µC:** Microcontroller **µP:** Microprocessor **INC:** Inter-node communication **ETH:** Ethernet **PCIe:** PCI express

VRTE - Vehicle Runtime Environment
Functional Layers



µC: Microcontroller **µP**: Microprocessor **HWA**: Hardware accelerator

VRTE - Vehicle Runtime Environment
Functional Layers & Protection Domains

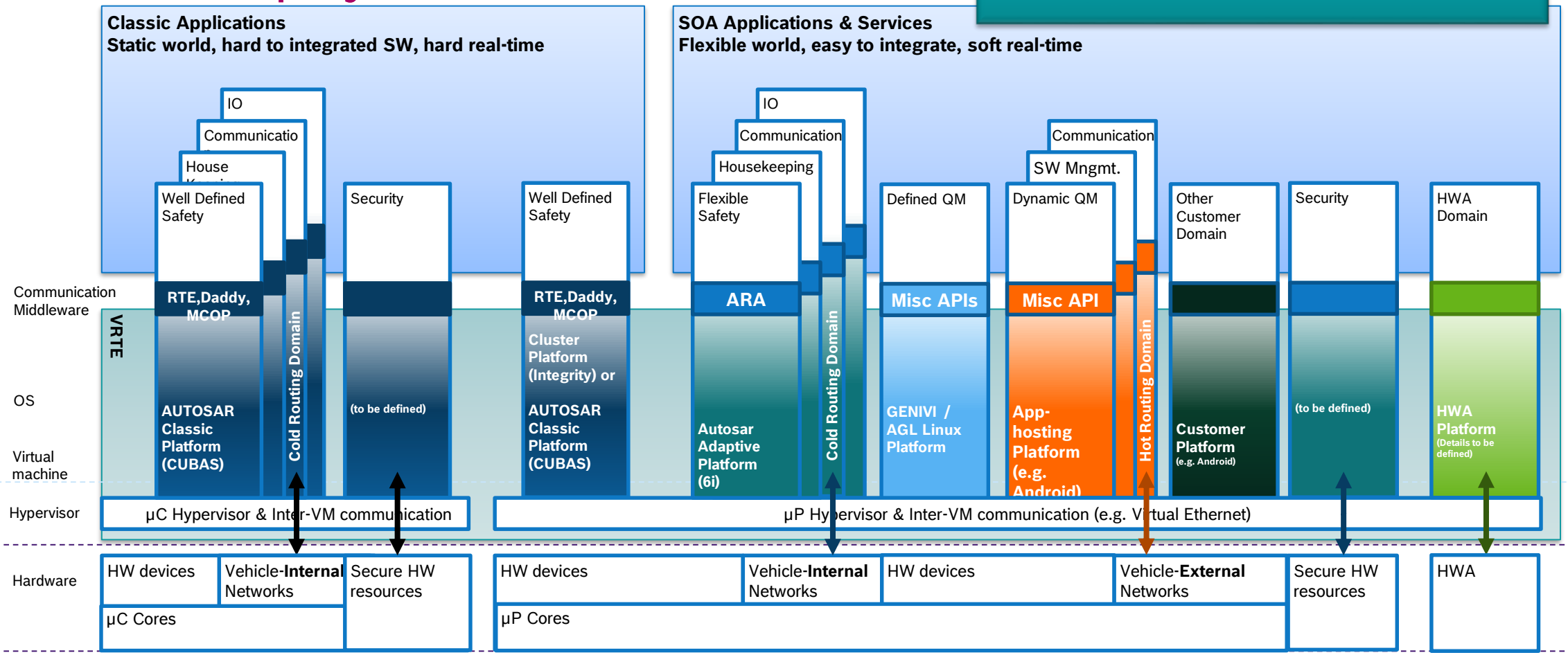


µC: Microcontroller **µP**: Microprocessor **HWA**: Hardware accelerator **IO**: Input/output

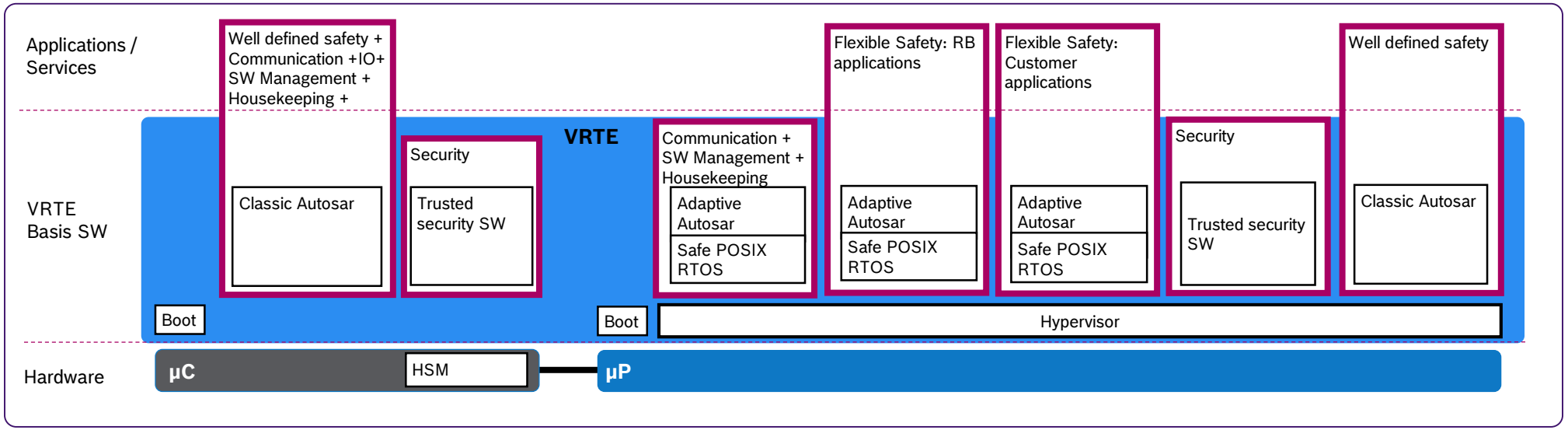
BOSCH

VRTE - Vehicle Runtime Environment
Domain deployment overview

This is a 150% architecture.
It is tailored for specific products.



Autosar – Adaptive – and Beyond
First deployment scenarios



Legend: Protection domain, + indicates domain merger

Thank You!