Vehicle Computer as integral part in 2019ff. New architectures vary with legacy constraints.
Total Amount of Software in Modern Cars

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/
VRTE - Vehicle Runtime Environment
Software in centralized 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

EE-architecture drives SW-architecture
VRTE - Vehicle Runtime Environment

Autosar – Adaptive – and Beyond

Flexibility, Versatility

VRTE
Vehicle Runtime Environment

Soft real-time
Many different applications
Runtime flexibility

Connected Services

Performance

Scope: Single runtime environment

Scope: Multiple collaborating runtime environments

Hard real-time
Single dedicated application
Static design

Disconnected services

Classic Platform

Adaptive Platform

Other runtimes

μC

μP

HWA

© Robert Bosch GmbH 2018. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.
VRTE - Vehicle Runtime Environment

Vehicle Computers

Integration ECUs

Intelligent Sensors & Actuators

Autonomous Driving Controllers

Soft real-time

Hard real-time

Static design

Connected Services

Many different applications

Runtime flexibility

Several runtime environments

Flexibility, Versatility

Performance
VRTE - Vehicle Runtime Environment

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

**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 Environment

Most important features

**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

**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
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 ISO 26262 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.

**Priority / Precedence**
Product driven
Business & Technology driven

Terminology according to ISO25010 software quality.
VRTE - Vehicle Runtime Environment

Key SW technologies

- CI
- SOA
- Remote update
- Access control
- Safe POSIX RTOS
- Hypervisor
- AUTOSAR
**VRTE - Vehicle Runtime Environment**

**Typical Context – Vehicle Variant A**

![Diagram of vehicle runtime environment]

**Vehicle**
- Vehicle internal network: ETH, CAN, FLX, LIN
- Deeply embedded ECU

**Gateway**
- ETH

**Central vehicle computer / Domain ECU / Cross-Domain ECU / Infotainment / Cluster**
- Applications SW
- **VRTE** = Infrastructure SW
- Hardware µC / µP / HWA

**Vehicle external network**
- Wired (ETH)
- TCU (GSM, WLAN, Bluetooth)

**Cloud**
- e.g. Backend

**Vehicle environment**
- e.g. other vehicle, traffic infrastructure

**IT devices**
- e.g. Smartphone

**Vehicle maintenance**
- e.g. workshop tester

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

© Robert Bosch GmbH 2018. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.
Vehicle Runtime Environment (VRTE)

Typical Context – Vehicle Variant B

Vehicle internal network:
- ETH
- CAN
- FLX
- LIN

Deeply embedded ECUs

Vehicle external network:
- Wired (ETH)
- OTA (GSM, WLAN, Bluetooth)

Central vehicle computer / Domain ECU / Cross-Domain ECU / IVI / Cluster

Applications
- SW

VRTE = Infrastructure SW

Hardware
- µC / µP / HWA

Cloud
- e.g. Backend

Vehicle environment
- e.g. other vehicle, traffic infrastructure

IT devices
- e.g. Smartphone

Vehicle maintenance
- e.g. workshop tester

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

TCU: Single central access point to vehicle – Security!!!

Deeply embedded ECUs

© Robert Bosch GmbH 2018. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.
VRTE - Vehicle Runtime Environment

Typical Context – SW & HW

Central vehicle computer / Domain ECU / Cross-Domain ECU / IVI / Cluster

Application SW

Semantic / application-specific middleware

VRTE = Infrastructure SW

μC QM to ASIL D

INC type a e.g. ETH

μP QM to ASIL B

INC type b e.g. PCIe

HW accelerator QM to ASIL x ???

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

© Robert Bosch GmbH 2018. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.
## Functional Layers

**L1 - HW specific infrastructure SW:**
- SW that interacts directly with HW and abstracts it towards the higher layers.

**L2 - OS specific infrastructure SW:**
- Essential SW that complements the actual OS kernel (aka scheduler) and abstracts OS specific properties towards the higher layers.

**L3 - Communication specific infrastructure SW:**
- Manages control and data flow between SW components.

**L4 - ECU specific infrastructure services:**
- Services managing one specific ECU.

**L5 - Vehicle specific / cross-ECU infrastructure services:**
- Services managing the ECU grid of the vehicle.

---

**Hardware**

- µC, µP, HWA

- µC: Microcontroller
- µP: Microprocessor
- HWA: Hardware accelerator
VRTE - Vehicle Runtime Environment

Functional Layers & Protection Domains

Application services

Basic application services / Semantic middleware

L5 - Vehicle specific platform services:

L4 - ECU specific platform services:

L3 - (Service-oriented) communication middleware

L2 - OS specific infrastructure SW:

L1 - HW specific infrastructure SW:

Hardware

µC, µP, HWA

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

Potential relevance for (Adaptive) Autosar

Run safe applications

Run QM applications

Run applications on HWA

Run QM applications

 {-}
VRTE - Vehicle Runtime Environment

Domain deployment overview

Classic Applications
Static world, hard to integrated SW, hard real-time

SOA Applications & Services
Flexible world, easy to integrate, soft real-time

This is a 150% architecture.
It is tailored for specific products.
Autosar – Adaptive – and Beyond
First deployment scenarios
Thank You!