Navigating the Shift to Centralized E/E Architectures in SDVs

07.05.2024

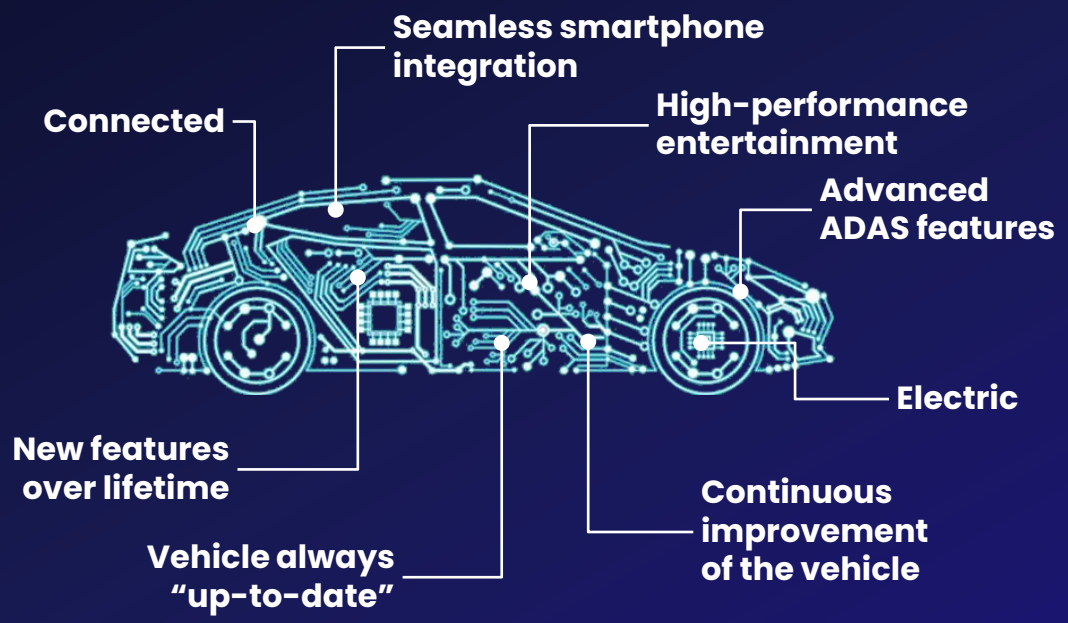
Thomas Hülshorst, Akshay Bujone, Christian Kirchhof, Marius Wegener, Jan Hake

Prepared for

MATHWORKS AUTOMOTIVE CONFERENCE

MathWorks AUTOMOTIVE CONFERENCE 2024 Europe

The industry is transforming towards software-centric products



Market Demand

- Advanced Features
- Updateability
- Upgradeability

Cost Reduction

- Simplified, modular hardware
- Cost efficient SW platforms

Competition

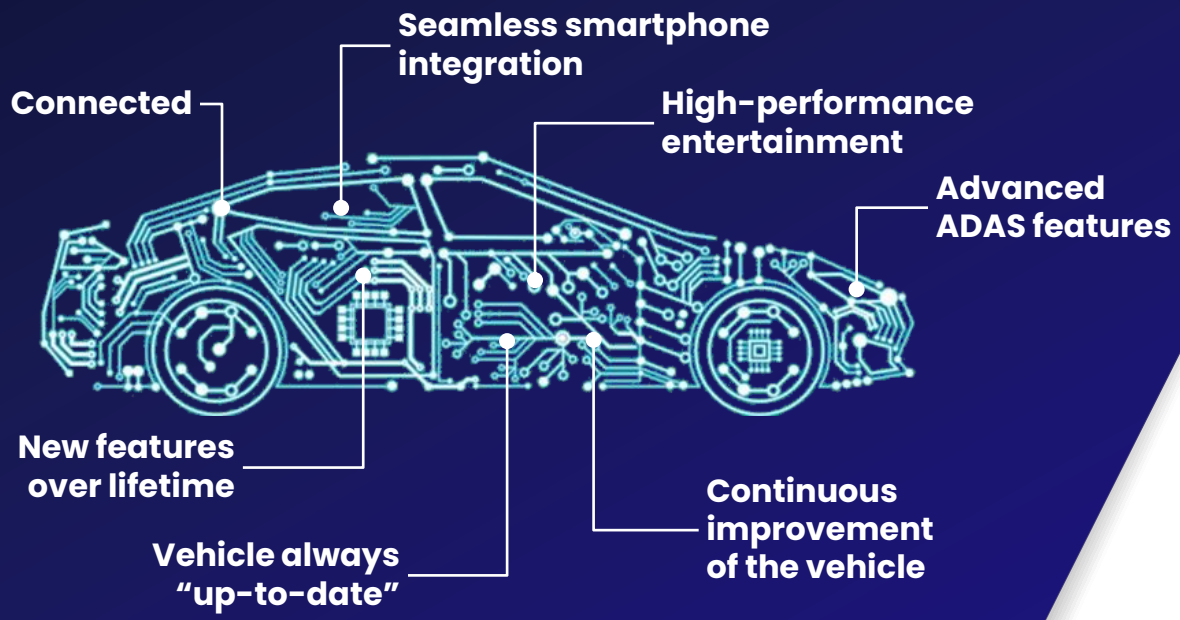
- Development efficiency by DevOps, automation, virtualization
- Reduced time-to-market and; new way of working
- Reduced recalls (by OTA and data analytics)

Software-Defined Vehicles (SDVs)

VALUE FOR CUSTOMERS IS CLEAR…

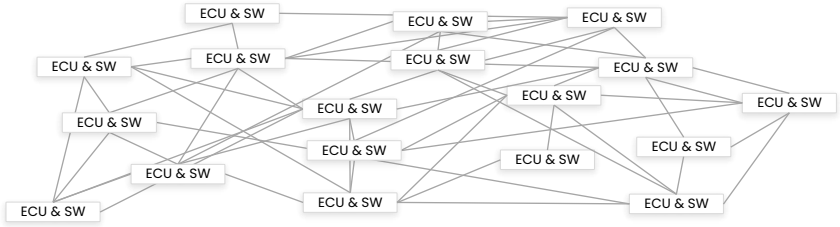
"When I bought a car in the past, the day I got it was the best the car would ever be. With a software-defined car, the day you buy it is the worst it will ever be.
From the point forward, however, it's going to be like magic."

Jensen Huang
CEO NVIDIA



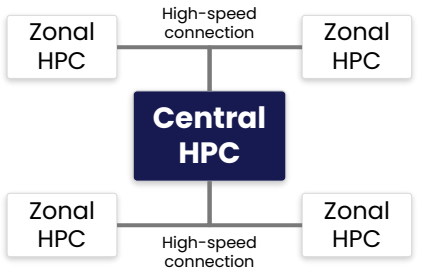
…BUT TECHNICAL BACKBONE IS NOT READY

Traditional E/E & SW Architecture:
Decentralized & signal-oriented (monolithic)

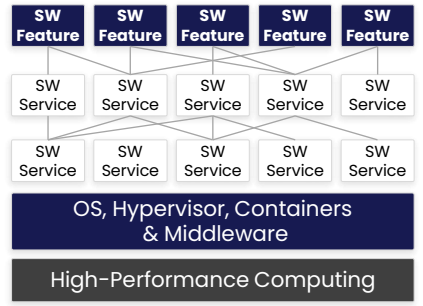


VISION
(IN-VEHICLE)

Centralized E/E Architecture w/ **high-performance** computing cluster

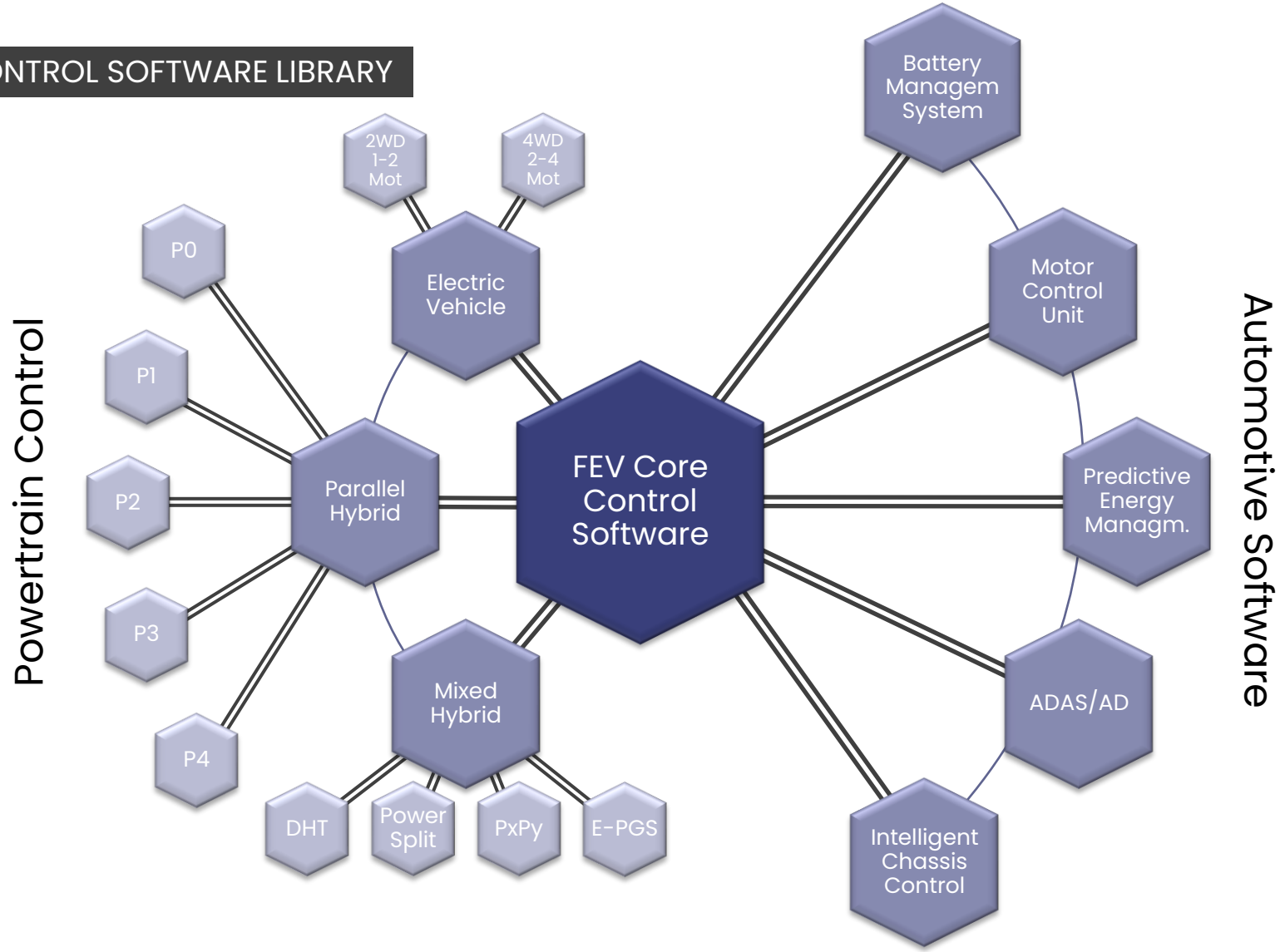


Mainly **service-oriented** software architecture



Our legacy: FEV has a wide range of cross-domain control features and well established automotive-grade processes, methods and toolchains

FEV AUTOMOTIVE CONTROL SOFTWARE LIBRARY



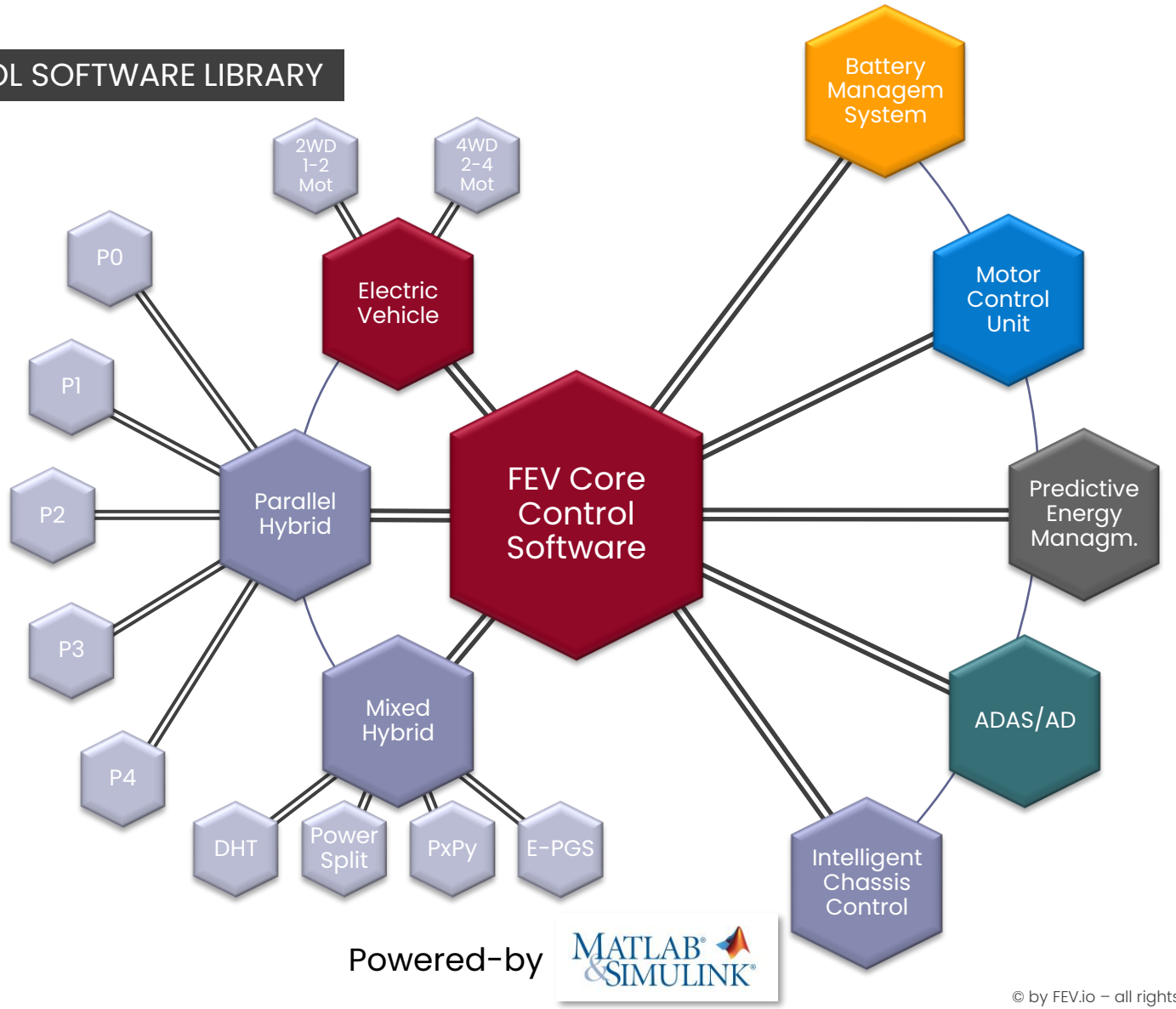
Our legacy: FEV has a wide range of cross-domain control features and well established automotive-grade processes, methods and toolchains

FEV AUTOMOTIVE CONTROL SOFTWARE LIBRARY

Feature Examples:

- Thermal Management
- Energy & Charge Management
- Torque Management
- Vehicle Coordination
- Diagnostics

Powertrain Control



Automotive Software

Feature Examples:

- State of Charge / State of Health
- Field-oriented Control
- Range Prediction
- Adaptive Cruise Control
- Traffic Jam Chauffeur

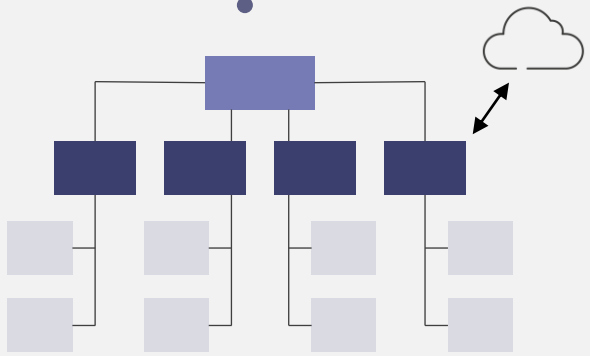
Powered-by MATLAB & SIMULINK

What about the relocation of features/functions? Major impact on function development

DEVELOPMENT FROM EMBEDDED CONTROLLER TO VIRTUALIZED PLATFORMS

Conventional E/E System

Functions hosted at **dedicated embedded controllers**

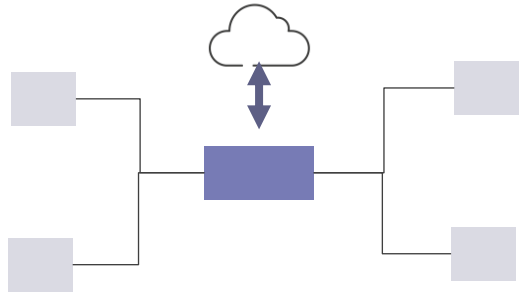


Signal-oriented Software architectures

Verification/Validation mainly at **embedded controller level**

Software-defined E/E System

Functions independent from dedicated controllers, hosted on **Cloud/Central/Zonal or Embedded target**



Software organized in **Service-oriented** architectures (SOA)

Virtualized platforms demand strong shift-left and **virtual verification and validation**

SOA = Service Oriented Architecture, SOP = Start of Production
Source: FEV

Feature reallocation: What is the benefit?

FROM TRADITIONAL ECU TO THE CLOUD

ON-BOARD ECU

- **PERFORMANCE: LIMITED**
  - ECU is not suited for highly complex or data-driven algorithms
- **DATA-ACCESS: LOW**
  - ECU is limited to data received via bus communication
- **AVAILABILITY: HIGH**
  - ECU is always available for calculation

ON-BOARD HPC

- **PERFORMANCE: IMPROVED**
  - Computational power can be allocated to algorithm when needed
- **DATA-ACCESS: MEDIUM**
  - Access to almost all on-board data (navigation, HVAC, powertrain) is available
- **AVAILABILITY: HIGH**
  - HPC is permanently available
  - Can be used as fallback solution when cloud is not available

CLOUD ENVIRONMENT

- **PERFORMANCE: UNLIMITED**
  - Performance can be used whenever requested
- **DATA-ACCESS: HIGH**
  - Access to large sets of data
  - Data-driven models can be adapted fleet wide
- **AVAILABILITY: MEDIUM**
  - Cloud is widely available
  - Resources can be scaled whenever they are needed based on actual customer request
  - In case of in-availability the local HPC compute can take over

Feature reallocation: What is important to remain automotive-grade?

FROM TRADITIONAL ECU TO THE CLOUD

ON-BOARD ECU

- **Hard Real-Time: HIGH**
  - Hard Real-Time requirements are Standard and can be met with existing solutions
- **Functional Safety: HIGH**
  - Functional Safety requirements are standard and can be met with existing solutions

ON-BOARD HPC

- **Hard Real-Time: MEDIUM**
  - Cross-domain applications with mixed criticality regarding hard real-time
- **Functional Safety: MEDIUM**
  - Cross-domain applications with mixed criticality regarding Functional safety
  - Up to ASIL-D possible, but higher effort

CLOUD ENVIRONMENT

- **Hard Real-Time: LOW**
  - Not automotive standard
- **Functional Safety: LOW**
  - Not automotive standard
  - Cloud infrastructure not available for ASIL-B/C/D today

Which functions are candidates for re-allocation?

Vehicle Motion, e.g. electric motor control

Field-oriented Control

Field-oriented Control

- Strict real-time requirements
- Functional safety relevant
- Implementation required to be adapted to controller hardware
- Fixed functionality that is not expected to change

Keep it on local ECU!

Predictive Energy Management, e.g.

Range Prediction

Range Estimation

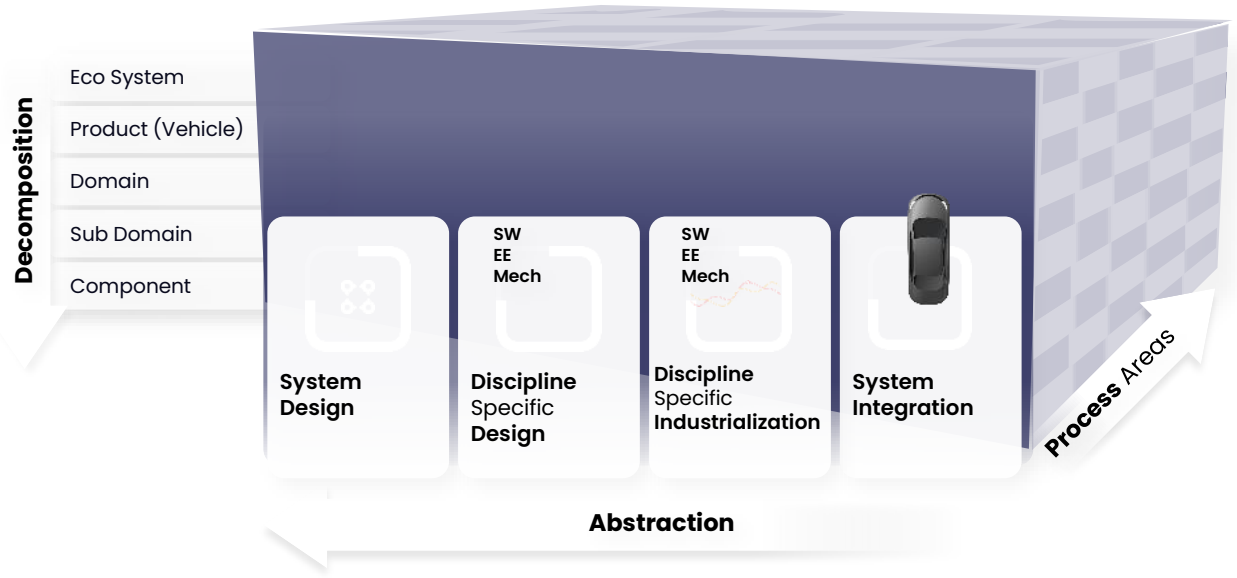
- No strict real-time requirements
- No functional safety requirements
- Implementation is independent of controller hardware
- Functionality can be improved and updated in the field
- Strong benefit of access to large sets of data
- Data-driven models can be adapted fleet wide

Reallocation recommended!

CUBE –The 360° Philosophy of FEV represents a mindset and a Systems Engineering process model to realize complex SDV enabled systems

CUBE UNITES ALL DISCIPLINES INTO A STRUCTURED APPROACH TO TACKLE HOLISTIC PRODUCT DEVELOPMENT

CUBE



- **Unites all development disciplines** under one common approach to realize complex, software driven products
- Creates a comprehensive perspective on products, with **clear technical responsibilities**
- Focuses on **collaboration** to manage the work efficiently

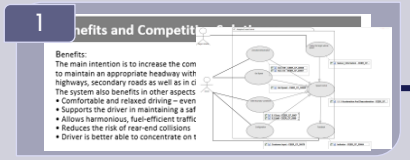
BENEFIT FOR SDV

- CONSISTENT SYSTEMS ENGINEERING
- FOCUS ON NON-FUNCTIONAL REQ. FOR SDV DECISION
- EFFICIENT MANAGEMENT OF LEGACY & OTS ELEMENTS
- SEAMLESS HANDOVER TO SW ARCHITECTURE DESIGN

High level Service Oriented Architecture definition methodology

SYSTEM DESIGN

PRODUCT SPECIFIC DESIGN

	A: FEATURE ANALYSIS	B: SYSTEM ARCHITECTURE	C: TOPOLOGY MAPPING	D: N/W BINDING	E: N/W DESIGN
	REQUIREMENTS	FUNCTION BEHAVIOR	DECOMPOSITION	PROTOCOLS	EEA
DOMAIN ANALYSIS	1 	5 Reusability & Sub-System Dependency Analysis	9 Service or Signal / S2s Decision	13 Client/Server or Sender/Receiver	
PERFORMANCE	2	6 Real Time or High performance	10 Methods, Events, Fields or Signals		
FUSA / CYSEC	3	7 ASIL and Secure Assent Data Definition	11 E2E Profile, SecOC		
NETWORK / HW CONSIDERATIONS	4	8 Network Topology	12 Datatypes, Periodicity		

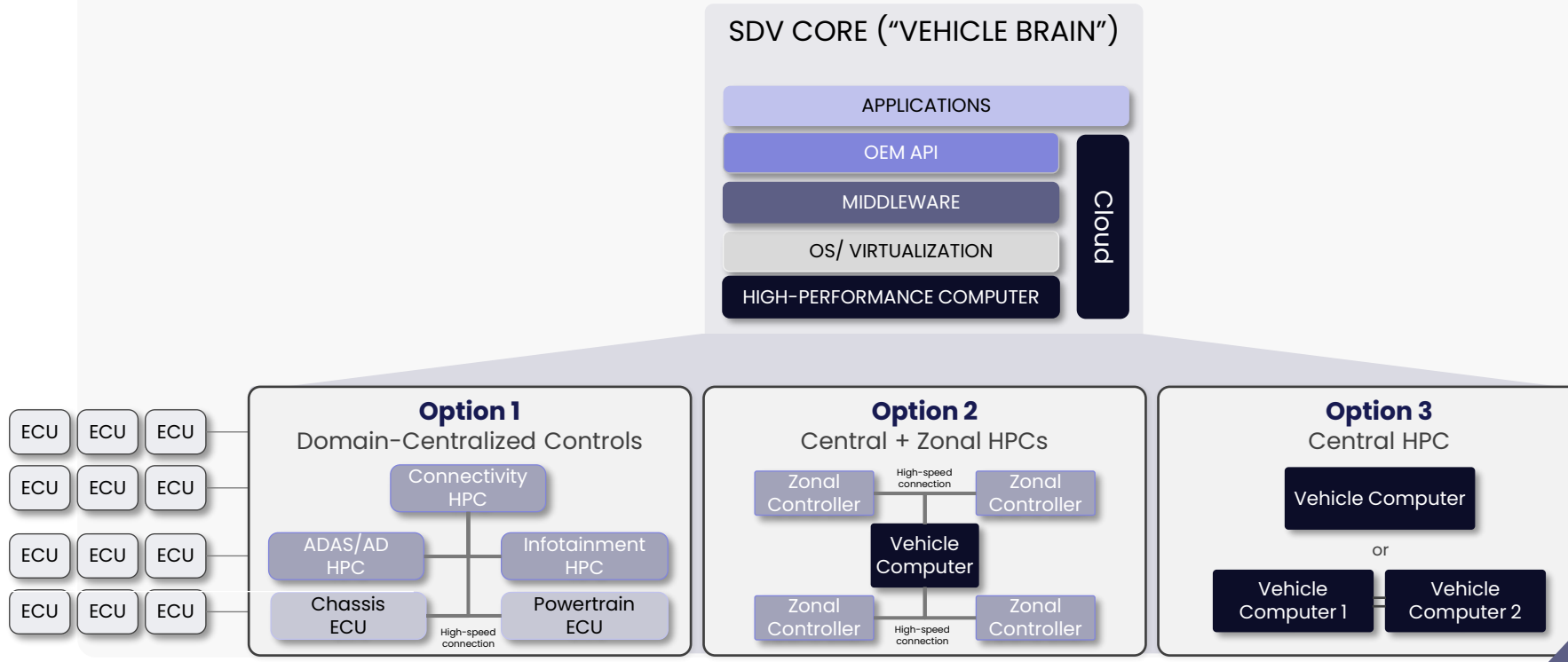
14 SOME/IP, DDS, MACsec, VLAN, AVB/TSN, HTTP/REST, MQTT and CAN, LIN

15 Service ID, Instance ID, Method / Field Event ID CAN-ID, LIN ID Message-Signal Layout

A high-performing E/E architecture builds the backbone for future product and development excellence – different levels of centralization are possible

E/E ARCHITECTURE STRATEGY

› Illustrative



Consideration of legacy through
stepwise evolution towards desired centralization level

CENTRALIZATION APPROACH

- Top-level control through “Vehicle brain”
- High performance and classical controllers
- Zonal controllers to minimize wiring

What is the right architecture for my use-cases?
Software architecture classification (simplified for illustration)

FEV.io

I VIRTUAL MACHINE ARCHITECTURE

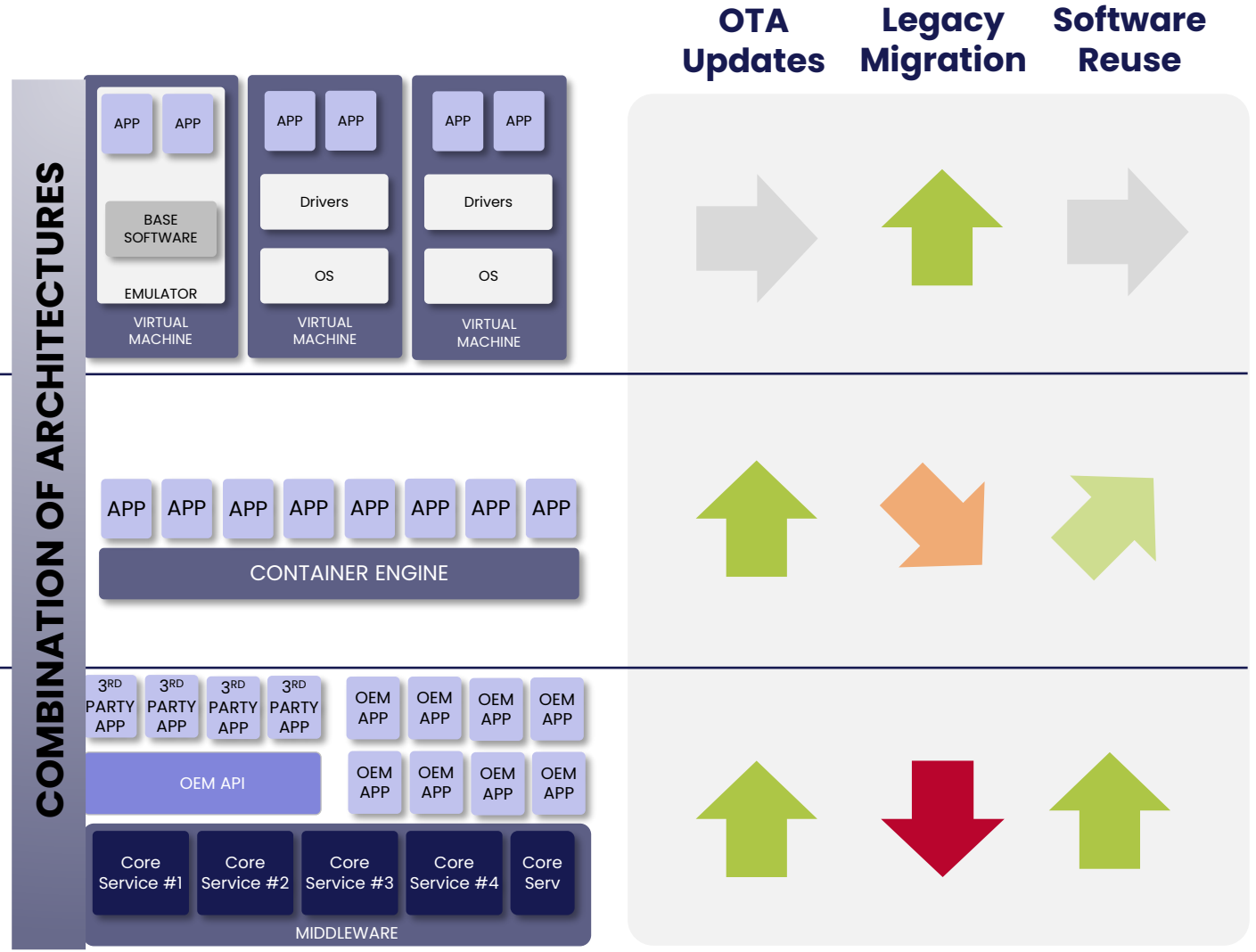
- Execute virtual machines on HPC
- Emulate legacy software “as is”
- Best for legacy migration and safety

II CONTAINERIZED ARCHITECTURE

- Facilitate over-the-air updates better
- Supports up-date of single features
- Leaner admin approach

III SERVICE-ORIENTED ARCHITECTURE

- Approach maximize Software re-use
- Supports best 3rd party app ecosystem



FEV SDV - Core platform - Current demo setup based on SDV Reference Architecture Type

Existing ADAS and Powertrain control features ported via VM

Flexible communication

DISPLAY

CORE PLATFORM

VM-VCU (MATLAB & SIMULINK)

VCU

Torque Management

VM ADAS (MATLAB & SIMULINK)

ADAS

Adaptive Cruise Control

VM-Digital Clus (MATLAB & SIMULINK)

CLUSTER

Virtual Bus (Some.IP / DDS)

Adaptive Autosar

QNX OS

Adaptive Autosar

AGL

AGL

QNX Hypervisor (customized)

HDMI

USB

R-CAR H3 HPC SOC

IVI CONTROLLER

VM-Infotainment

IVI

ANDROID

Fixed 3rd Party OS

i.MX8

USB

HDMI

JOYSTICK

Testing in Simulation

VEHICLE SIMULATION

Ethernet Switch

Internet in

DISPLAY

AA: Autosar Adaptive; IVI: In vehicle Infotainment

Overview global FEV SDV initiative for a software-driven product development including a collaborative engineering framework

FEV SDV CORE PLATFORM BUILD UP

FEV SDV platform demo / CES 2024

- Cross-domain applications
- HPC (Renesas, NXP, Qualcomm)
- Hypervisor (QNX, GHS, ..)
- FEV Virtual Bus
- OTA Enabled
- Vehicle API
- Classic & Adaptive enabled

FEATURE DEVELOPMENT BASED ON FEV IP

BODY

ADAS/AD

INFOTAIN-MENT

VEHICLE MOTION CONTROL

ENERGY / BATTERY

DEVELOPMENT WORKBENCH

COLLABORATION

PROCESSES

SW MGMT

CI/CD

SDV CLOUD & DATA CENTER BUILD UP

DIGITAL TWIN

DATA MGMT. AND ANALYSIS

CLOUD ORCHESTRATION

VIRTUAL ENGINEERING FRAMEWORK

VIRTUAL ➡

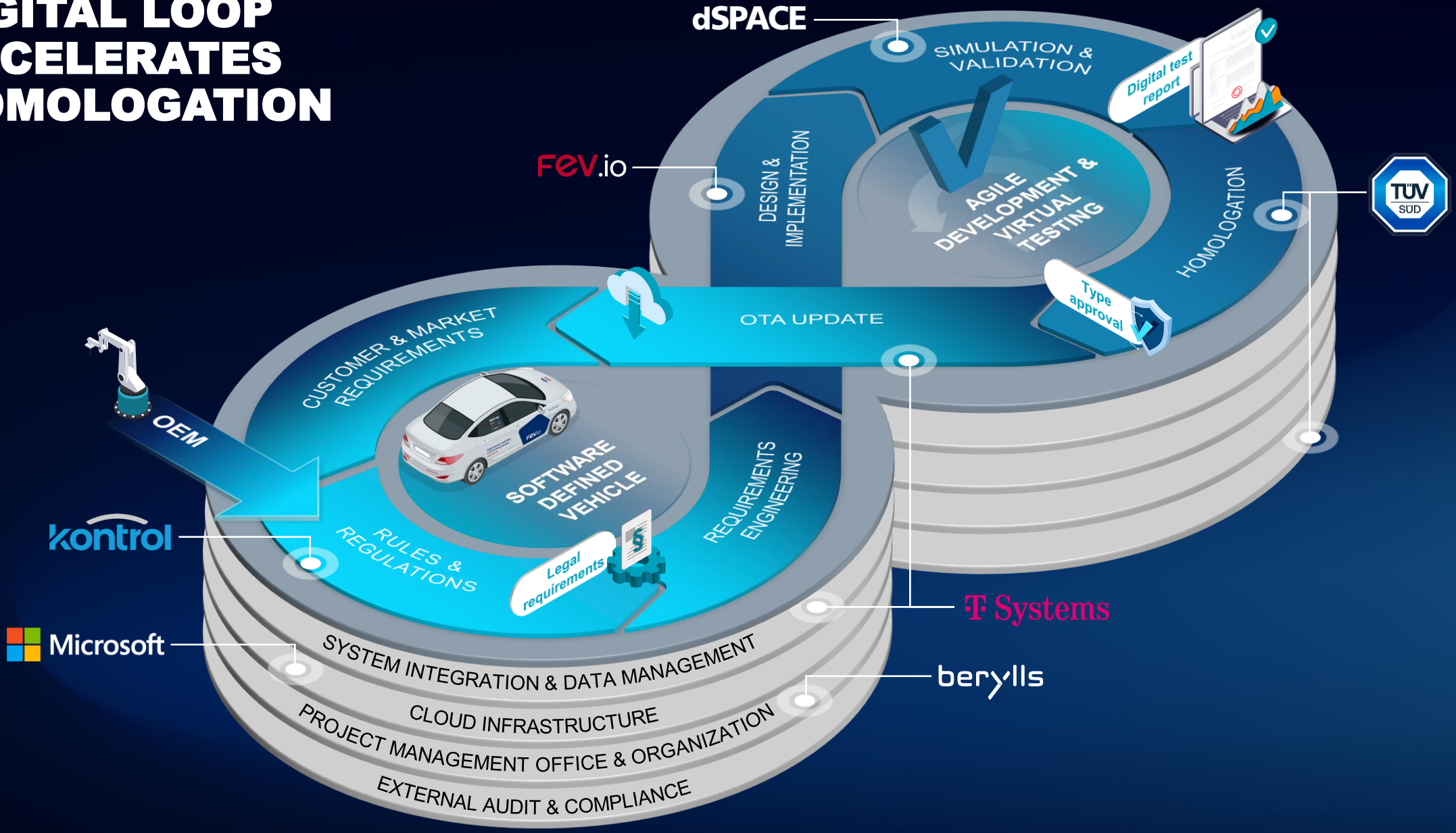
DESIGN

DEVELOP

INTEGRATE

VALIDATE

DIGITAL LOOP ACCELERATES HOMOLOGATION



FEV follows a joint technology initiative for core platform development meeting the SDV requirements

PROJECT VALUE PREPOSITONS

System & component excellence

Established agile cooperation scheme

Proven SW Development Framework