Software-Defined Vehicle Realization with Model-Based Design and FEV’s Gateway

Dr. Hamzeh Alzubi, FEV North America

- Intelligent mobility and software manager.
- Leads the design, development, and implementation of software programs and applications.
- Over15 years of professional experience in software development for aerial and ground vehicles.
- Ph.D. from Oakland University. His Ph.D. research project was the first-place winner at the UAE Drones for Good competition in 2016.

FEV.io

Software-Defined Vehicle Realization with Model-Based Design and FEV’s Gateway

Hamzeh Alzubi, FEV.io

MathWorks
AUTOMOTIVE CONFERENCE 2024

AGENDA

- **Introduction**
- Software Defined Vehicle (SDV) Motivation
- FEV.io’s SDV HPC Software Development
- Applications development for SDV with MathWorks tools
- SDV HPC testing on an existing EE architecture using FEV.io’s gateway
- FEV.io: full service SDV development

An overview of growing challenges within the automotive industry

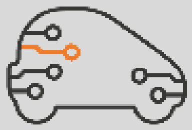
WIDE RANGE OF MOBILITY

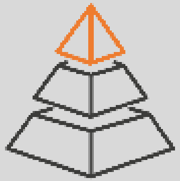
Passenger	Commercial

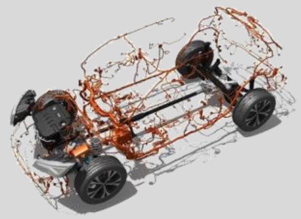
High | Mid | Low (Variant)

GROWING INFLUENCERS

EE Architecture	Complexity	Wiring

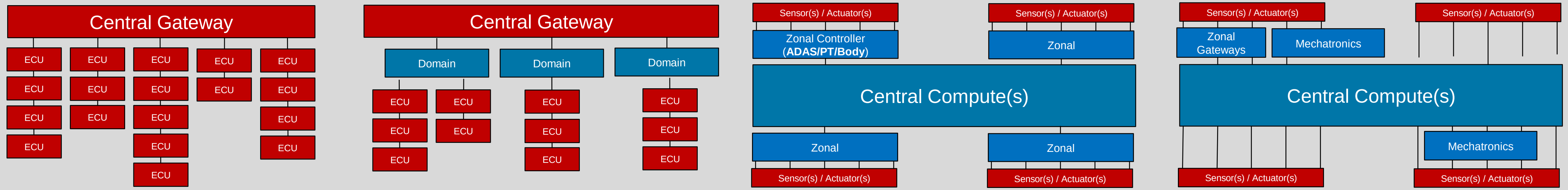






EVOLVING EE ARCHITECTURE

Distributed	Domain	Zonal	Central

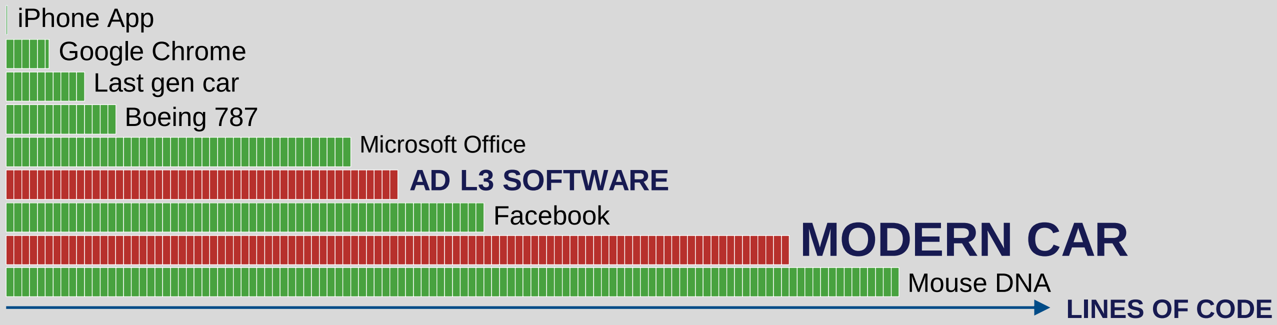


One SOLUTION for all multi-dimensional requirements **SOFTWARE DEFINED VEHICLE**

CAV related services are disrupting product development and are strongly pushing towards redefined E/E and SW architectures

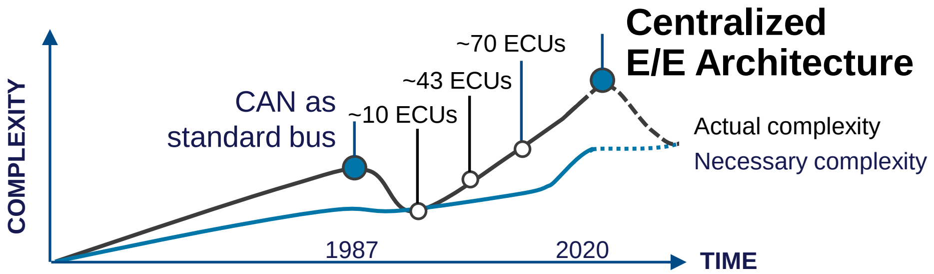
150 million LINES OF CODE

TYPICALLY CONTAINED BY A MODERN CAR
(50 MILLION LINES REQUIRED FOR AD L3 AD[1] SYSTEM)



100 ELECTRONIC CONTROL UNITS (ECU)

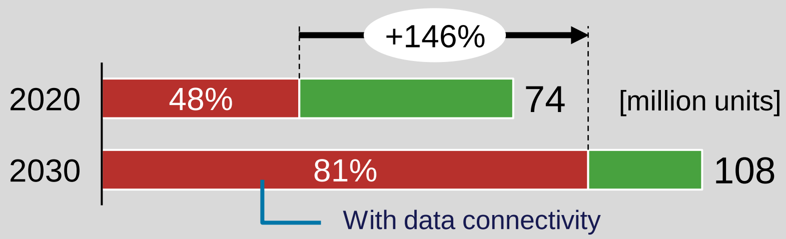
IN CONVENTIONAL CARS ARE INCREASING COMPLEXITY
OF CURRENT E/E ARCHITECTURES



>1,000 GB OF DATA

PER VEHICLE AND DAY EXPECTED
TO BE GENERATED BY 2030

GLOBAL **DATA CONNECTED**[2] VEHICLE SALES



1) L3 AD = SAE level 3 Automated Driving; 2) Direct connection to OEM backend via at least 2G/2G/4G connectivity; CAV: connected, automated vehicle

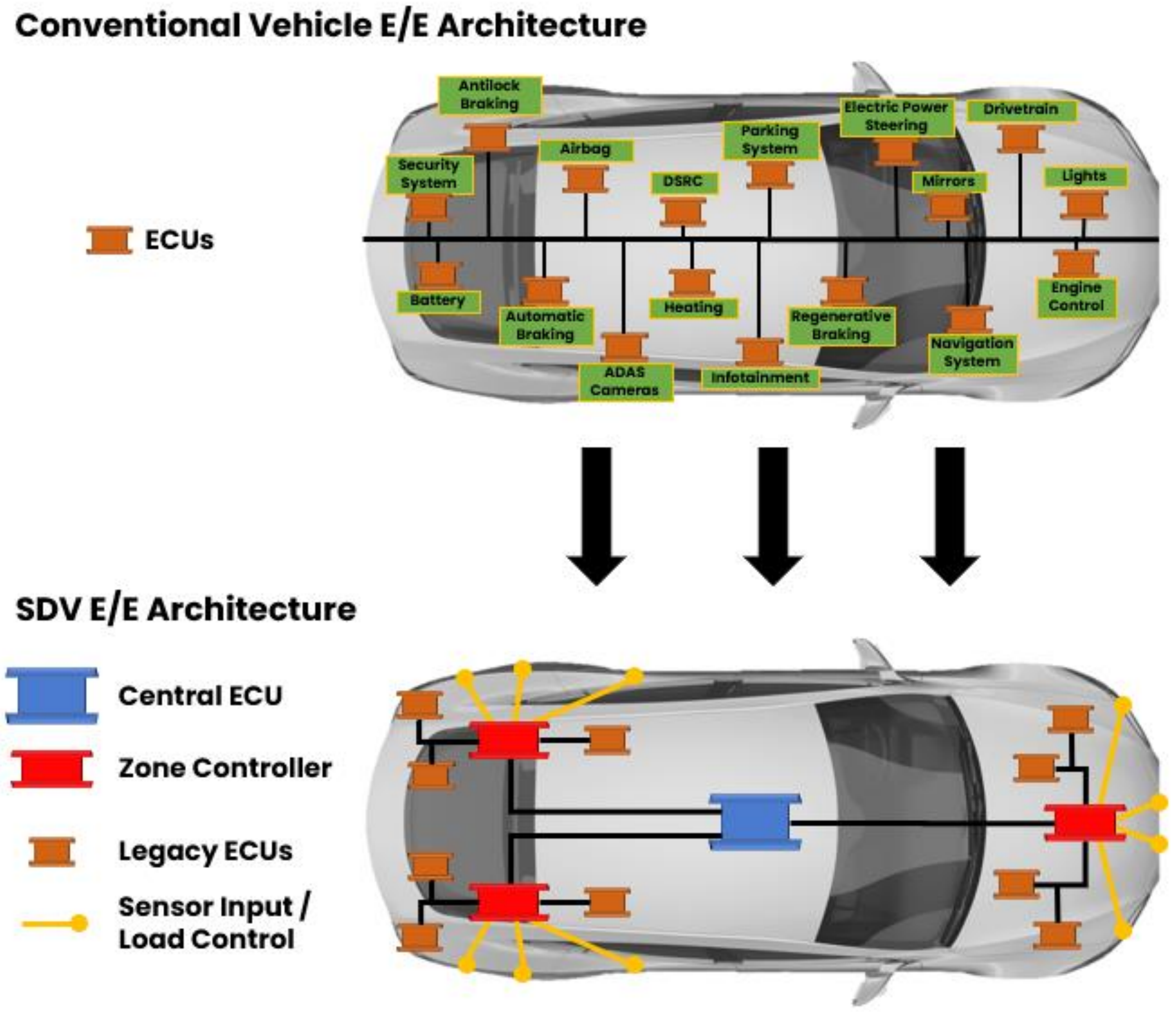
AGENDA

- Introduction
- **Software Defined Vehicle (SDV) Motivation**
- FEV.io’s SDV HPC Software Development
- Applications development for SDV with MathWorks tools
- SDV HPC testing on an existing EE architecture using FEV.io’s gateway
- FEV.io: full service SDV development

Software Defined Vehicle (SDV) Motivation

Example highlights of SDV motivation:

- Increasing software complexity
- New trends in the automotive market such as electrification, autonomous driving and connectivity to meet customers' expectations
- Reduced number of ECUs
- Decoupling of hardware from software
- Improved vehicle life cycle management
- Over-the-Air (OTA) software updates
- Enabling new ways to interact with vehicle users
- SDV provides the following benefits:
  - Lower costs
  - Less weight
  - Faster time to market
  - Increased security



AGENDA

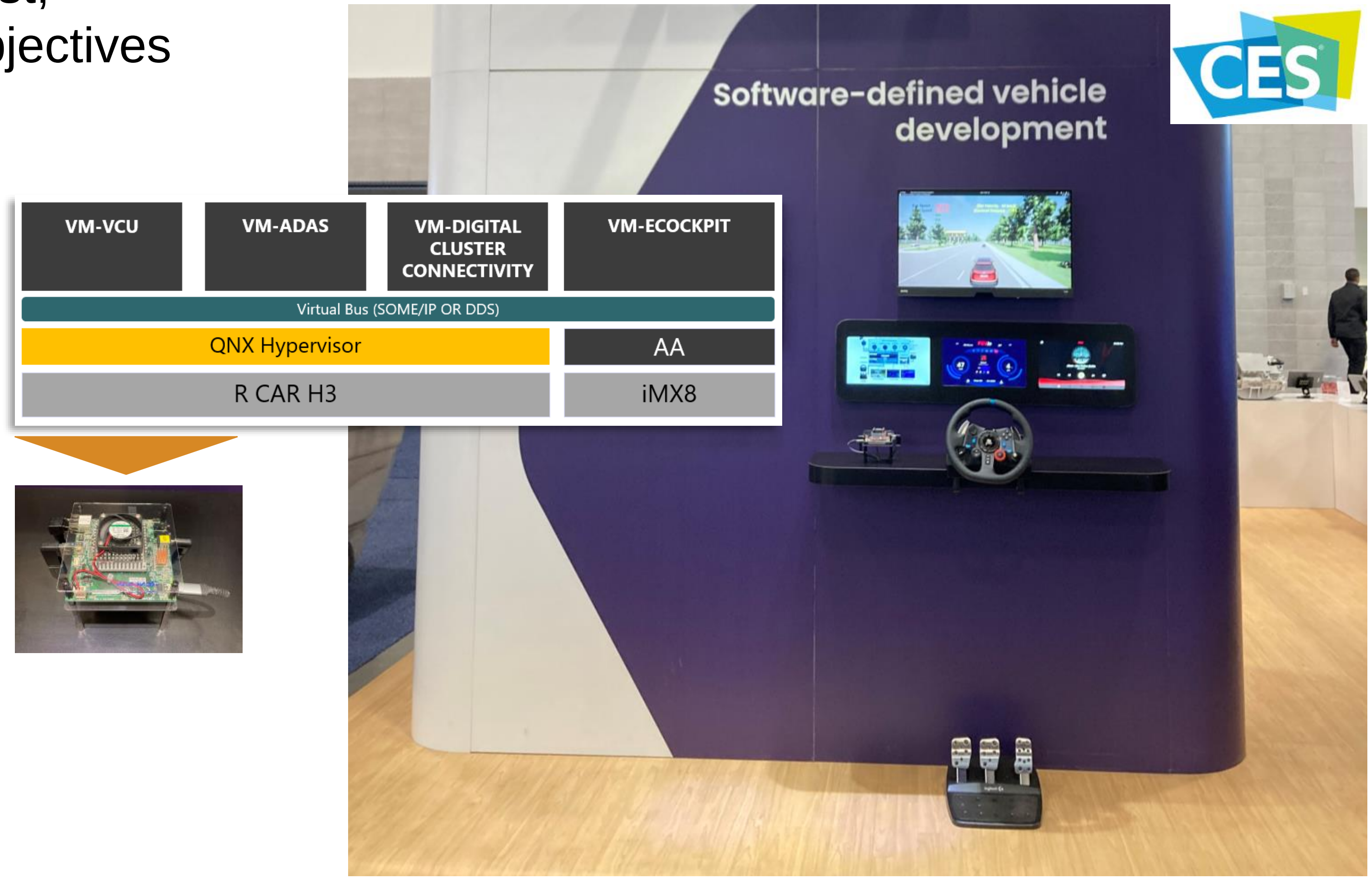
- Introduction
- Software Defined Vehicle (SDV) Motivation
- **FEV.io’s SDV HPC Software Development**
- Applications development for SDV with MathWorks tools
- SDV HPC testing on an existing EE architecture using FEV.io’s gateway
- FEV.io: full service SDV development

FEV.io provides cutting edge EE architecture and software solutions to address the challenges of transitioning to SDV

FEV.io demonstrator of cross-domain Central Controller for Software Defined Vehicles

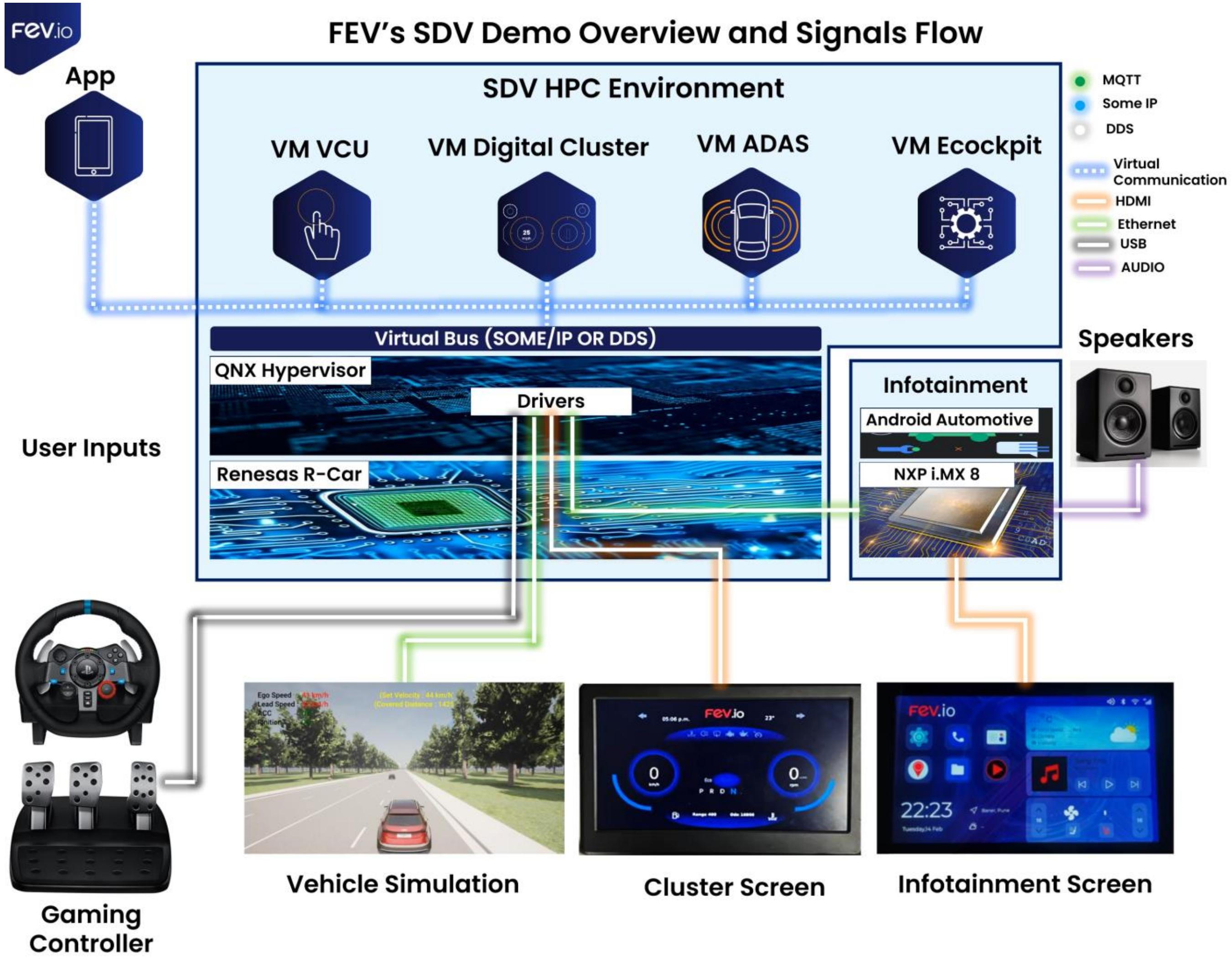
Scalable, modular demonstrator to test, improve and showcase the project objectives

- Demonstrate mix criticality load on HPC
- Redesign legacy functions into SOA Architecture
- Create virtual bus for maximal abstraction of SW



FEV.io SDV demo architecture with DDS, SOME/IP, and MQTT

SDV demo video



AGENDA

- Introduction
- Software Defined Vehicle (SDV) Motivation
- FEV.io’s SDV HPC Software Development
- **Applications development for SDV with MathWorks tools**
- SDV HPC testing on an existing EE architecture using FEV.io’s gateway
- FEV.io: full service SDV development

Applications development for SDV with MathWorks tools

MATLAB & SIMULINK

Legacy ECU
- ECU application (C)
- Classic AUTOSAR
- Real-time OS
- Microcontroller

VM - VCU
- VCU application (C++)
- Adaptive AUTOSAR
- QNX Neutrino OS

VM - ADAS
- ACC application
- Posix OS

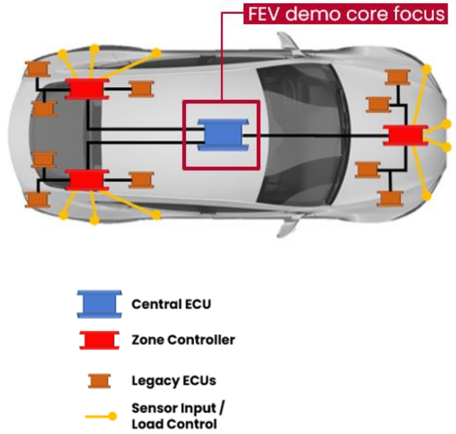
VM - Cluster
- Digital Cluster application
- Posix OS

VM – Ecockpit
- Infotainment application
- Posix OS

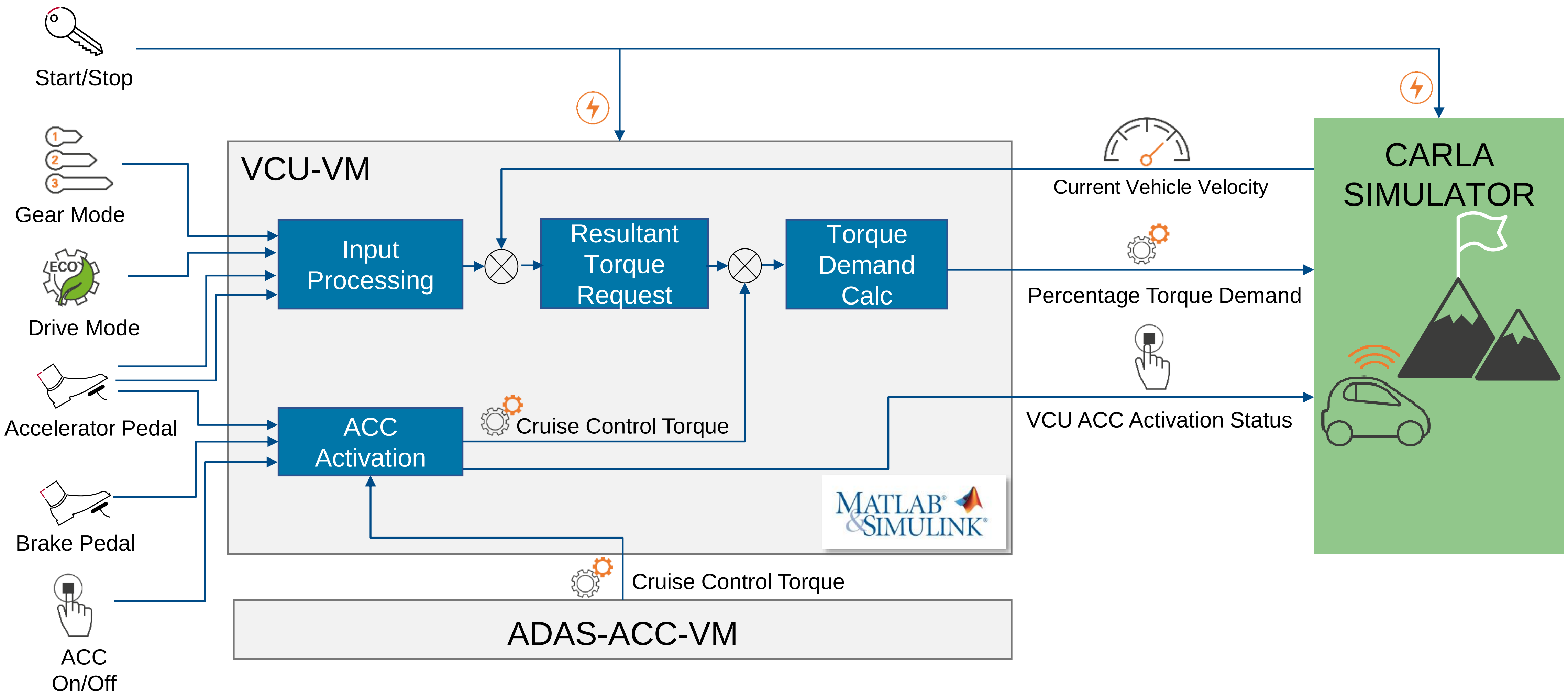
Virtual Bus

QNX Hypervisor | Android Automotive

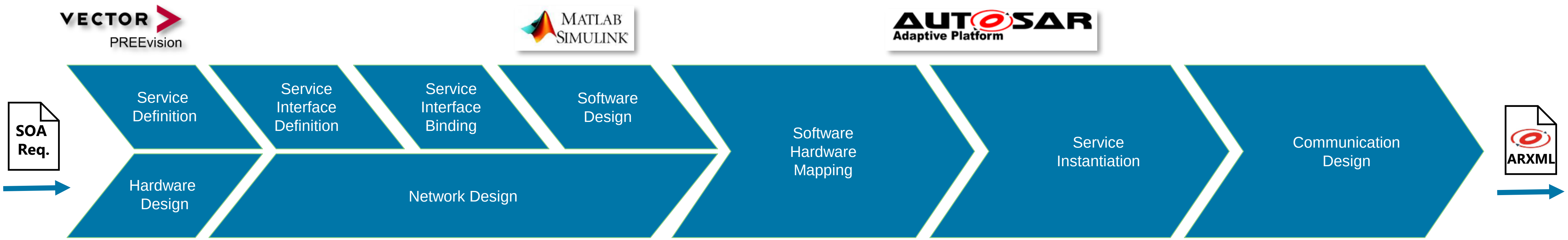
Renesas R CAR HPC | NXP iMX8



Migration of Legacy Functions to Service Oriented Architecture (SOA): E-Drive Torque Path Arbitration



Integration of vehicle function in VCU Cluster on HPC with Autosar Adaptive for SOA architectural design



Legacy to SOA architecture approach:

SYSTEM DESIGN

- Use Cases elaboration
- Features definition
- Requirements derivation
- System design

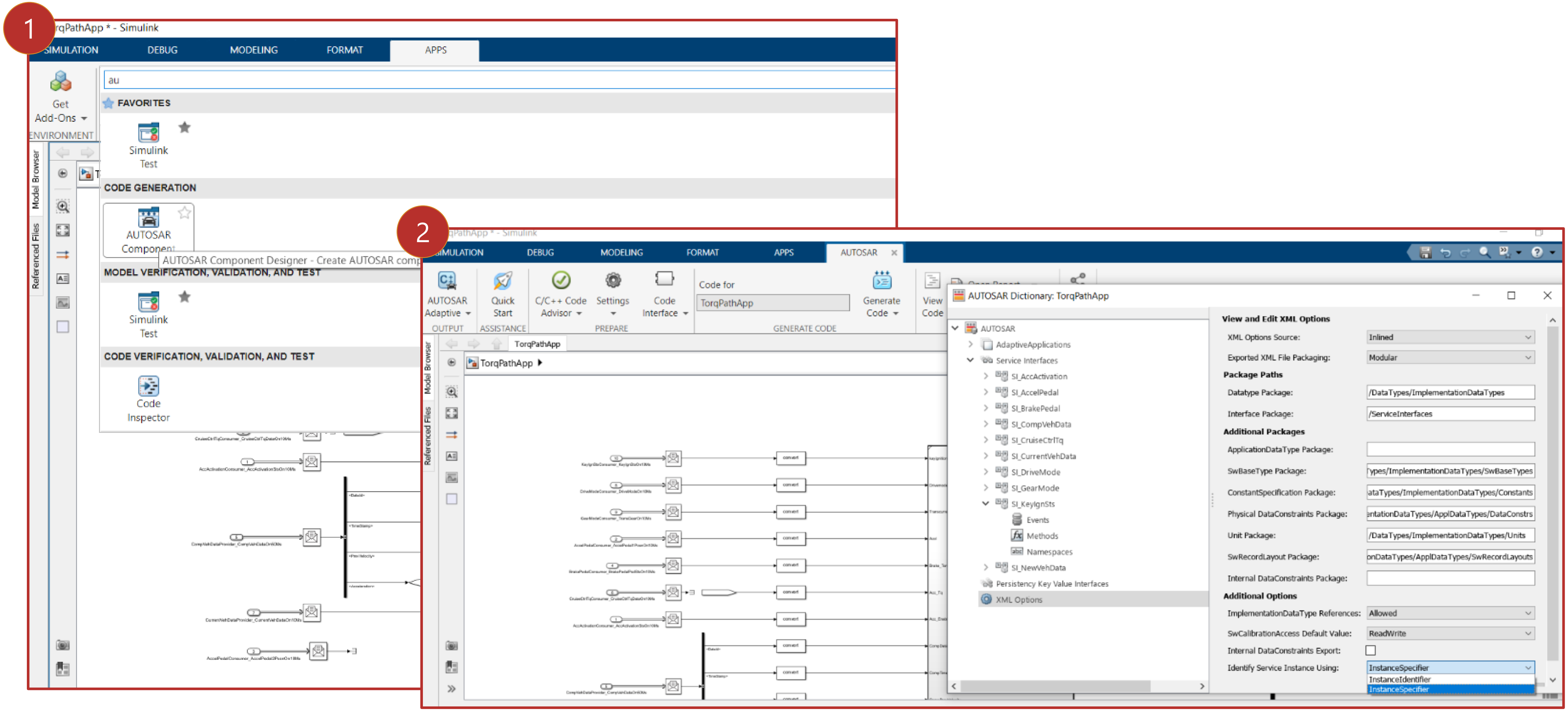
SOFTWARE DESIGN

- Service Architecture Definition
- Application Design in MATLAB/Simulink
- Signal Modeling including timing req. analysis
- Deployment configuration of Autosar Adaptive stack

INTEGRATION:

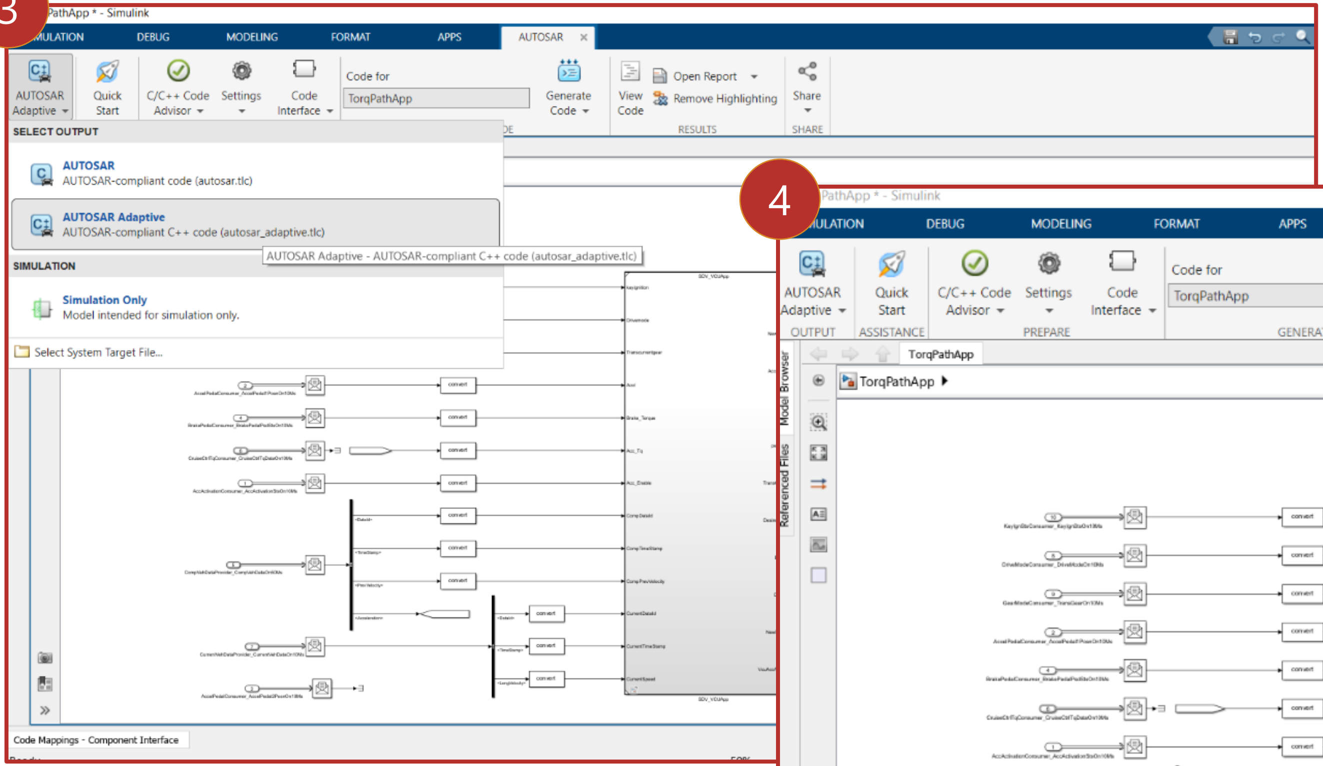
- Application implementation
- Integration with Vector Adaptive Stack
- Deployment of QNX OS and Hypervisor
- Set up of HIL Demo

Realizing the Legacy Functions as Adaptive SWC in Simulink

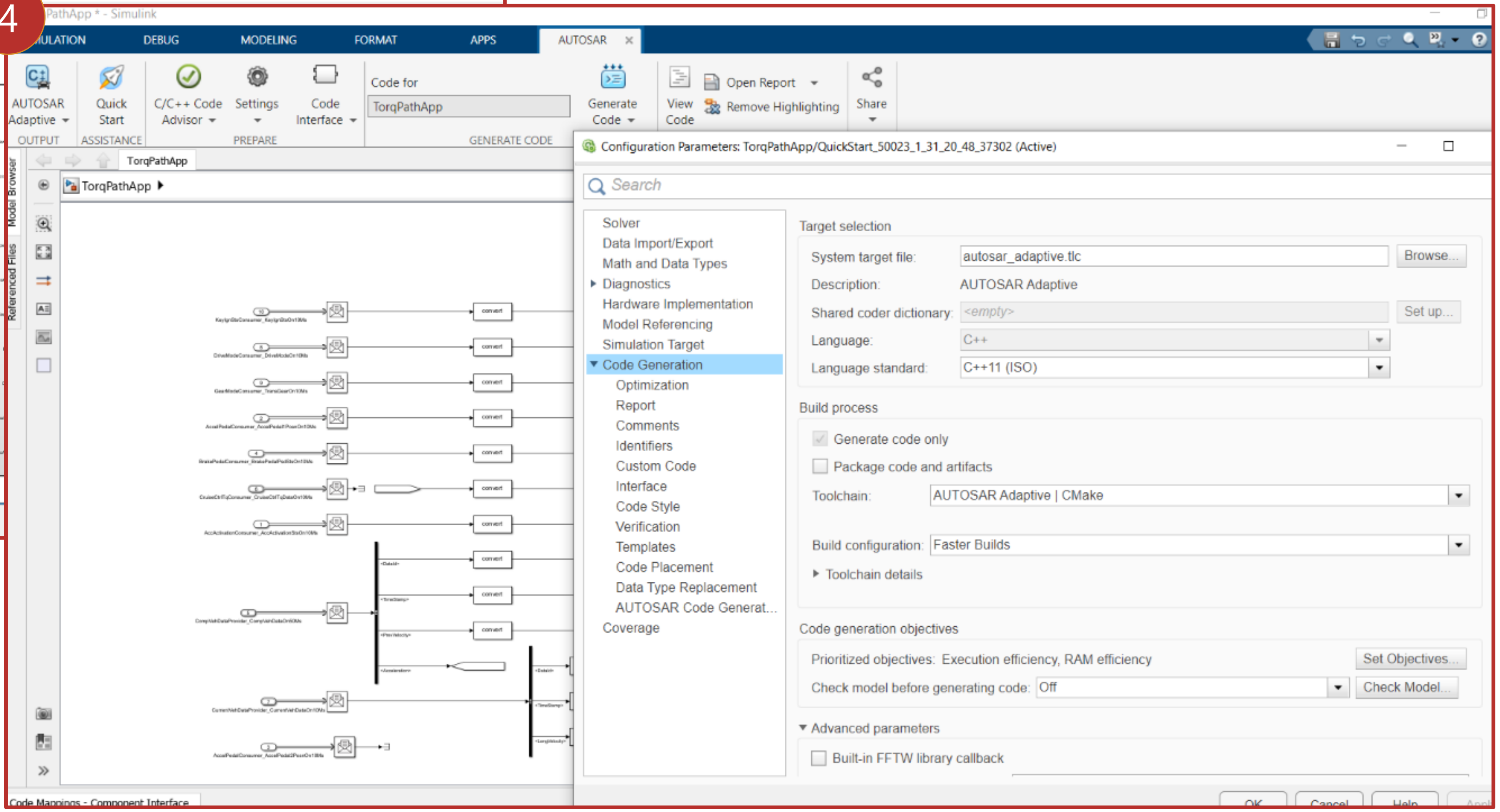


Adaptive AUTOSAR compliant C++ Code Generation with Embedded Coder

3



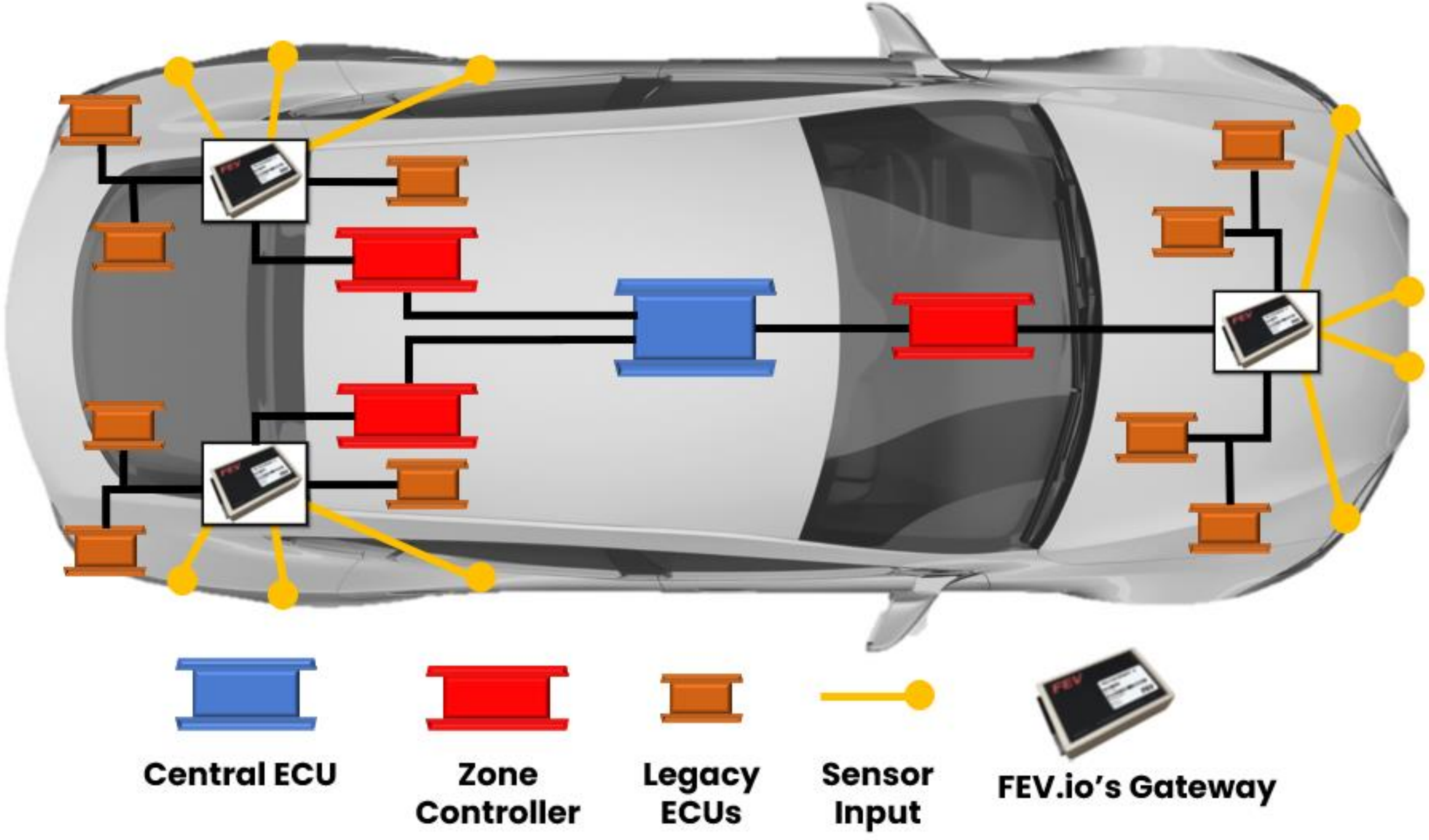
4



AGENDA

- Introduction
- Software Defined Vehicle (SDV) Motivation
- FEV.io’s SDV HPC Software Development
- Applications development for SDV with MathWorks tools
- **SDV HPC testing on an existing EE architecture using FEV.io’s gateway**
- FEV.io: full service SDV development

FEV.io's gateway enables SDV HPC and Zonal controller integration and testing on an existing OEM EE architecture

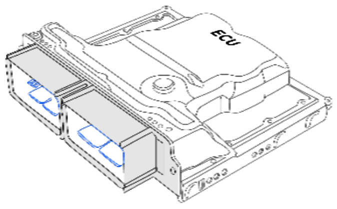


AGENDA

- Introduction
- Software Defined Vehicle (SDV) Motivation
- FEV.io’s SDV HPC Software Development
- Applications development for SDV with MathWorks tools
- SDV HPC testing on an existing EE architecture using FEV.io’s gateway
- **FEV.io: full service SDV development**

FEV.io helps customers to transform traditional signal-based ECUs to Service Oriented implementation

SIGNAL BASED ECU

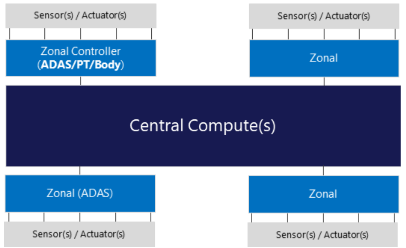


FUNCTION SOA

DEPLOY DECISION

- Central
- Zonal
- Mechatronics

DEPLOYMENT



VECTOR

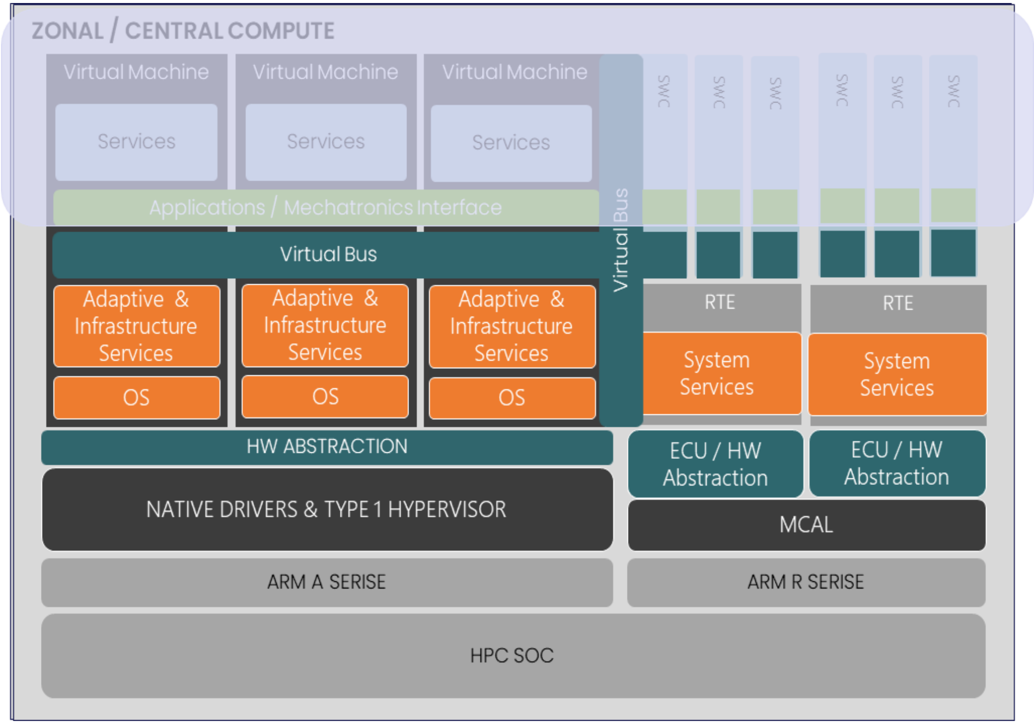
FEV VALUE PROPOSITION

- Function IP & Experience
- Doman Expertise
- SOA Experience
- FUSA & Cysec Experience

Domain Experts + SOA Architects + System Architects

- Multiple HPC Experience
- FEV IP
- Classic & Adaptive AUTOSAR
- Zonal & Ethernet Experience

FEV.io is building a SDV platform utilizing Model-Based tools and supporting customers on SDV specific topics below



- HPC Board bring up
- HAL & BSP Porting
- Hypervisor porting
- FEV Virtual Bus IP (Porting services)
- Platform Partitioning for Zonal / Central compute
- Classic & Adaptive Platform bring-up
- SDV Platform Development & Maintenance
- Functional Safety & Cybersecurity Development

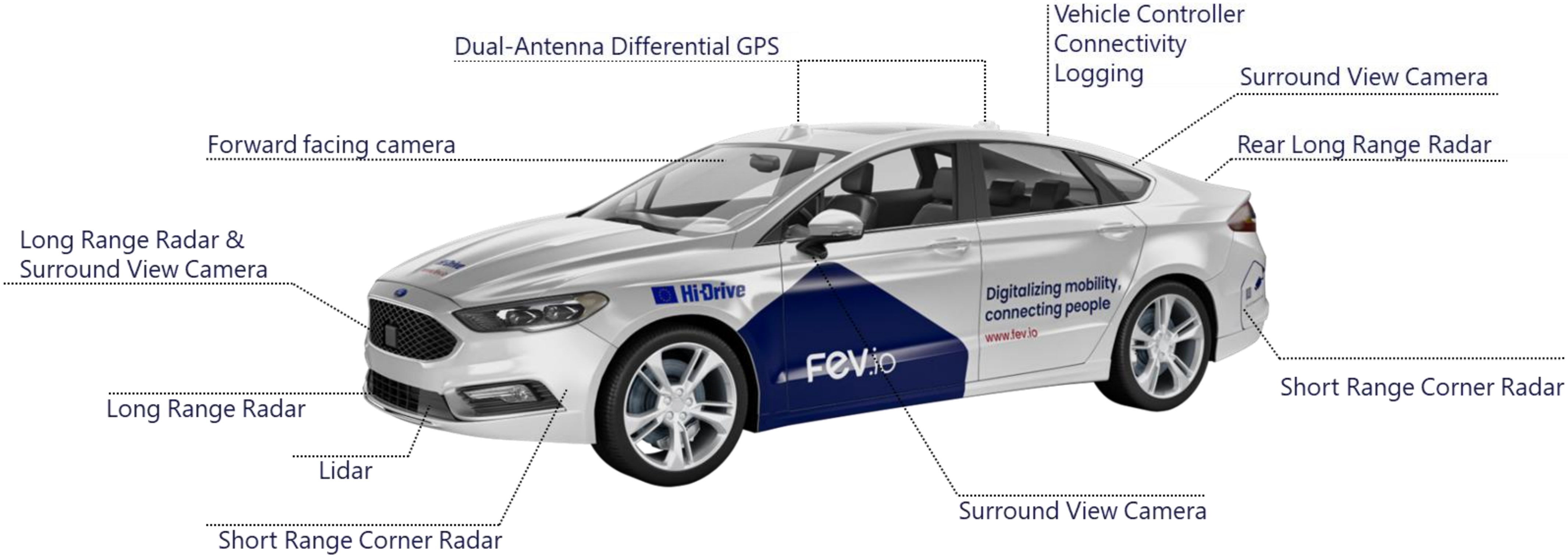
FEV VALUE PROPOSITION

- Hypervisor Experience
- Linux & QNX Experience
- System Development

Platform Experts + **System Architects**

- FEV IP (Virtual Bus)
- Classic & Adaptive AUTOSAR
- Zonal & Ethernet Experience

FEV.io Smart Vehicle Demonstrator Becoming a SDV Development Platform



Additional Cameras (8x, for logging purposes only)
(Surround View, IC, Driver/Co-Driver, Steering Wheel, Driving Pedals)

Conclusion

- Automotive EE architecture is evolving to accommodate the continuous addition of complex features
- SDV supports the successful deployment of emerging technologies such as autonomous driving, connectivity, and electrification
- Transitioning to a SDV environment can be done efficiently with maximum use of legacy software
- FEV.io’s gateway has proven to be a key enabler for SDV development and testing
- MathWorks provides the needed tools and support to migrate MATLAB/Simulink based legacy software to application layer of SDV environment
- FEV.io can be your development partner in emerging technologies like SDV

MathWorks
AUTOMOTIVE CONFERENCE 2024
North America

Thank you

FEV.io