Embedded Success dSPACE

Real and Virtual Development with SystemDesk

Joe Fairchild

Project Manager – Software Development and Validation

dSPACE, Inc.

Goals of AUTOSAR

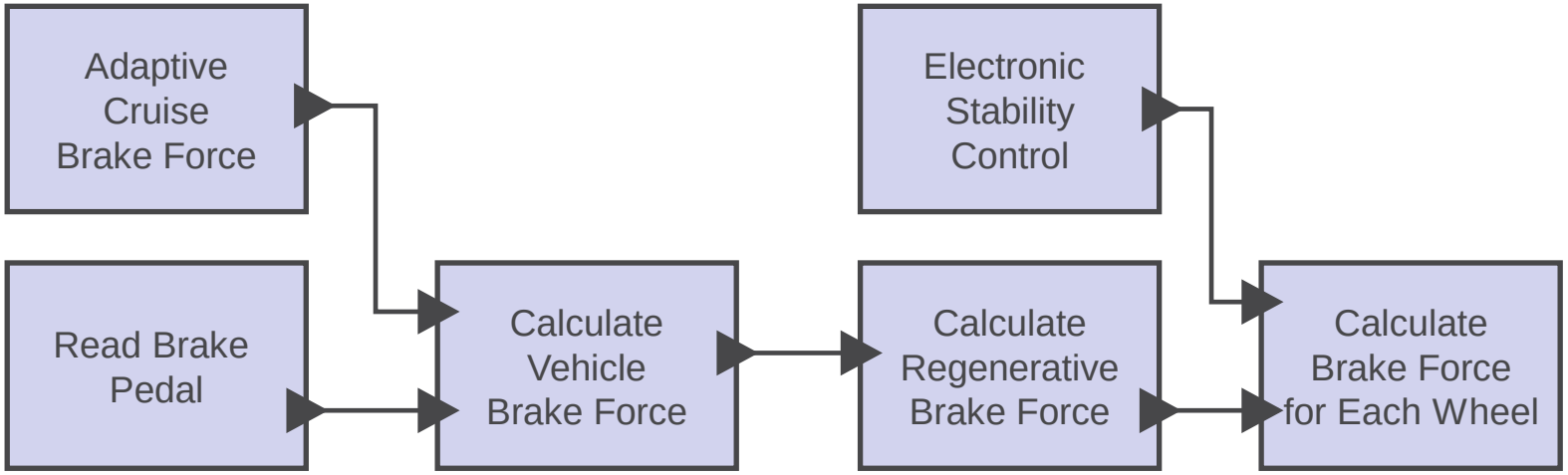
- Create libraries of software components
  - **Reusable**
  - **Hardware-independent**
- Be flexible
  - Easy to **exchange** hardware or software components
  - Easy to **change configurations**
- **Make tools interoperable and exchangeable** by
  - Standardized methodology
  - Standardized interfaces and exchange formats
- “Cooperate on standards, Compete on implementation”

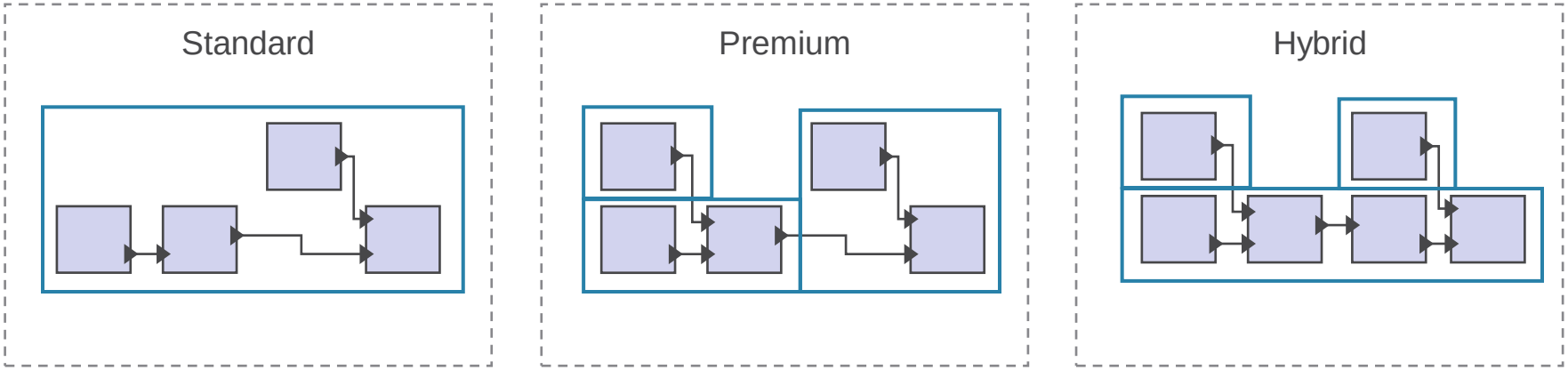
dSPACE‘s AUTOSAR Involvement

Joined the AUTOSAR partnership as a **Premium Member** in April 2004.

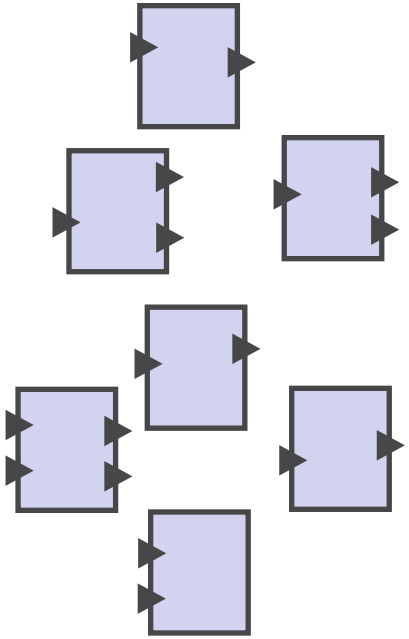
dSPACE is active in specifying parts of the architecture.

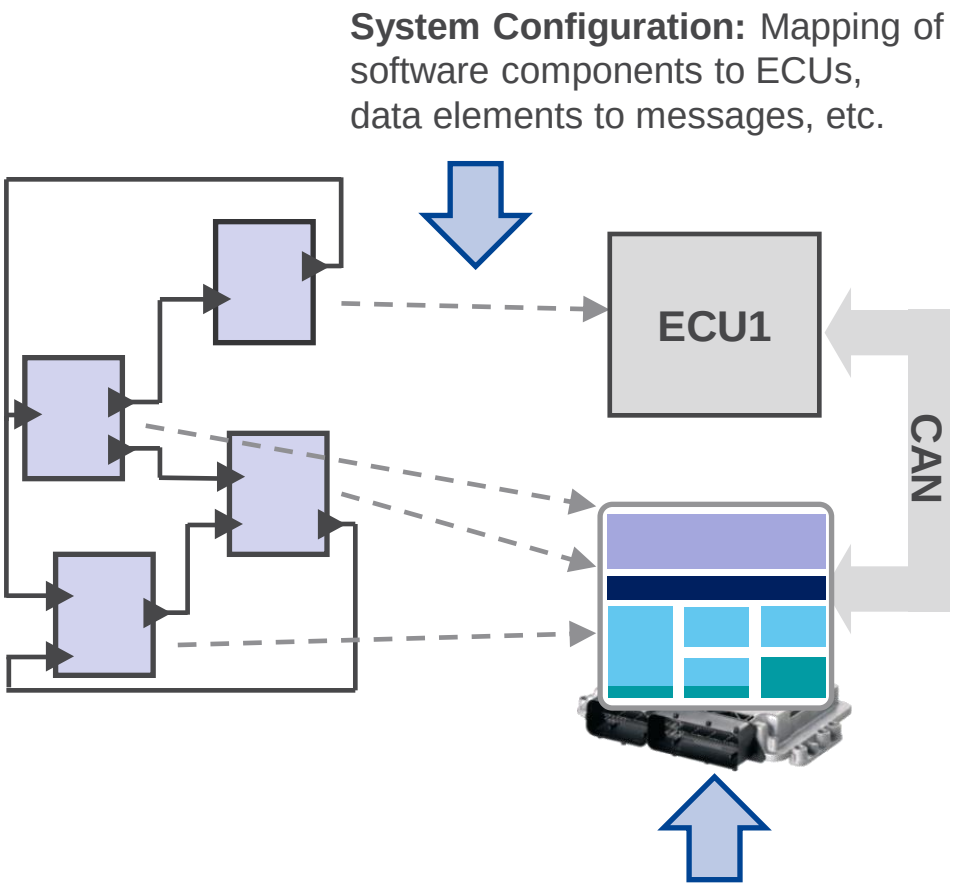
Logical vs. Physical Architecture





AUTOSAR Software Development Methodology





Library of software components (C code functions with XML describing interfaces)

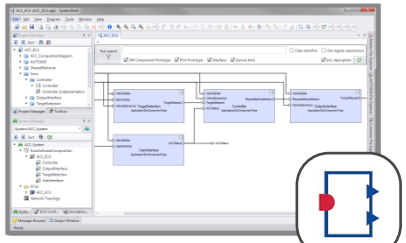
System (interconnection of software components)

ECU
Configure OS, and basic software; generate code for RTE; compile and link

AUTOSAR Software Development Process

dSPACE

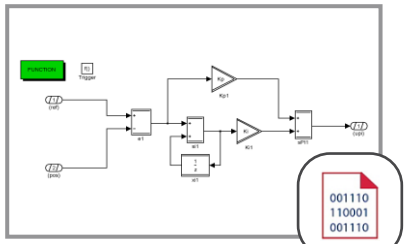
Design System Architecture



SystemDesk®

arxml

Develop Control Functions



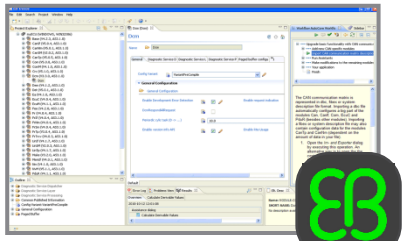
TargetLink®

.c

Application

Configure Basic Software

arxml



tresos Studio

.c

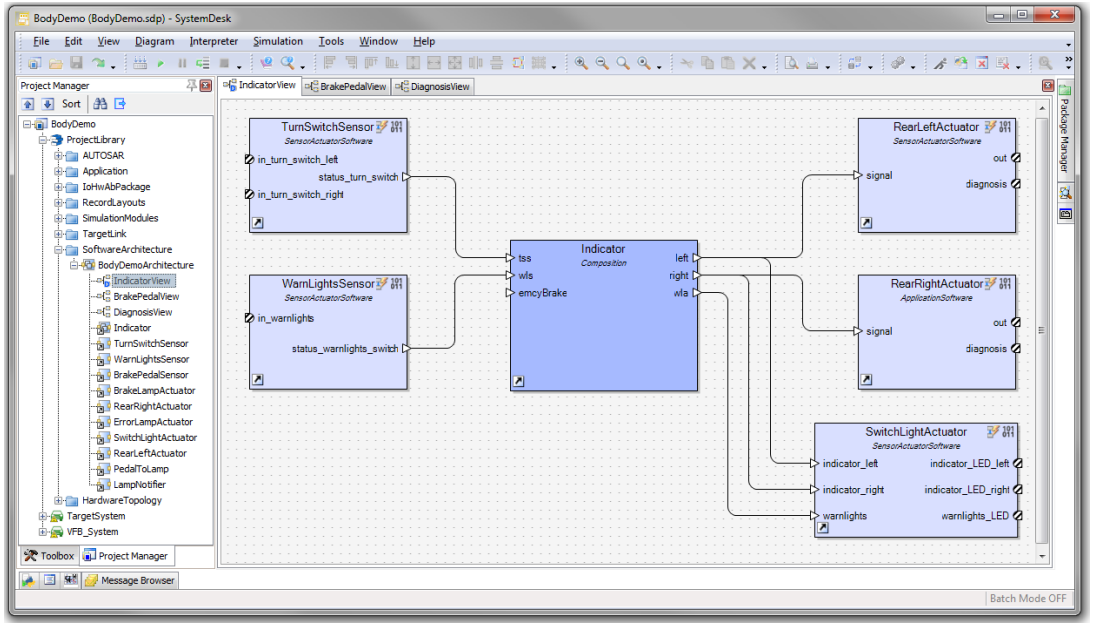
BSW Stack

ECU

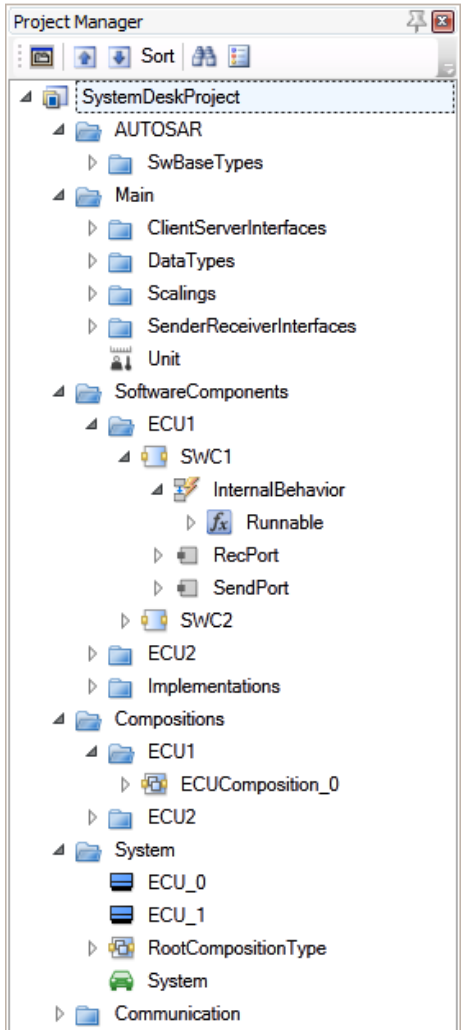
SystemDesk at a Glance

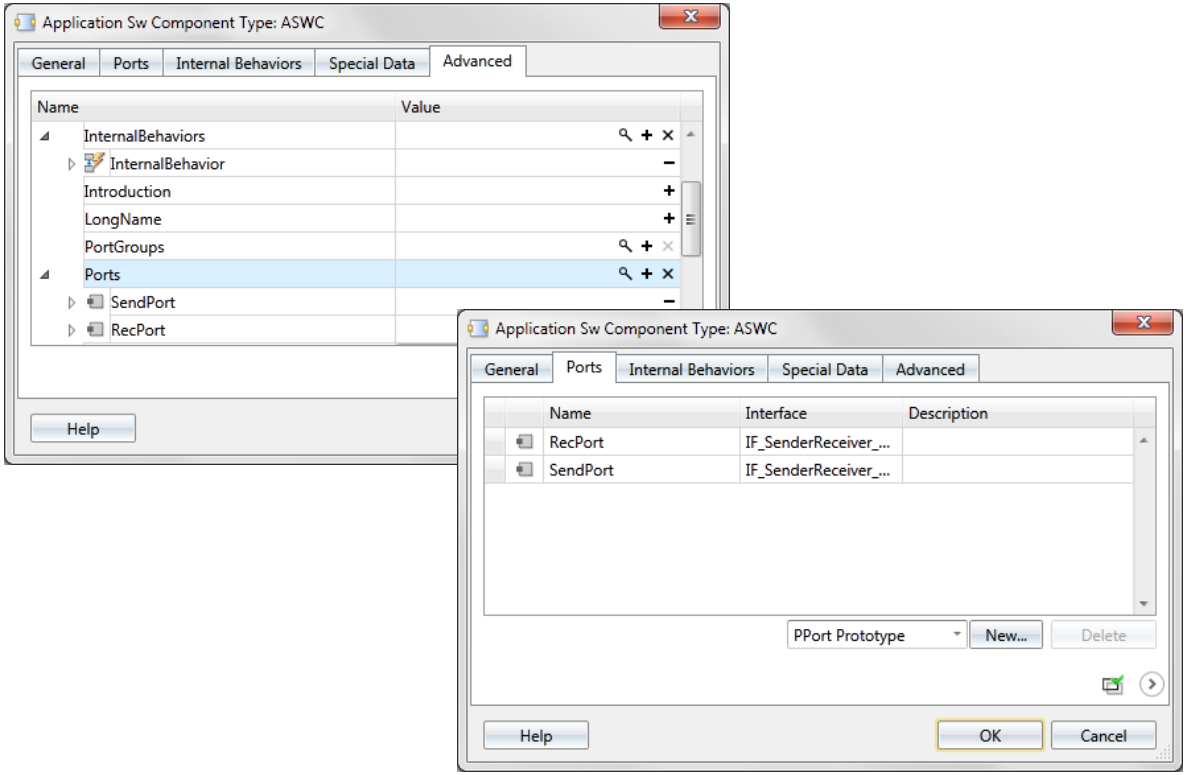
- Modeling tool for AUTOSAR software and system architectures
  - Support of multiple versions of AUTOSAR standard, including most current versions
  - Integration of software architecture onto an ECU
  - Virtual integration of architectures for PC-based SIL simulations

- Process support
  - Complete tool automation via Python or C# API
  - Special round-trip support with TargetLink



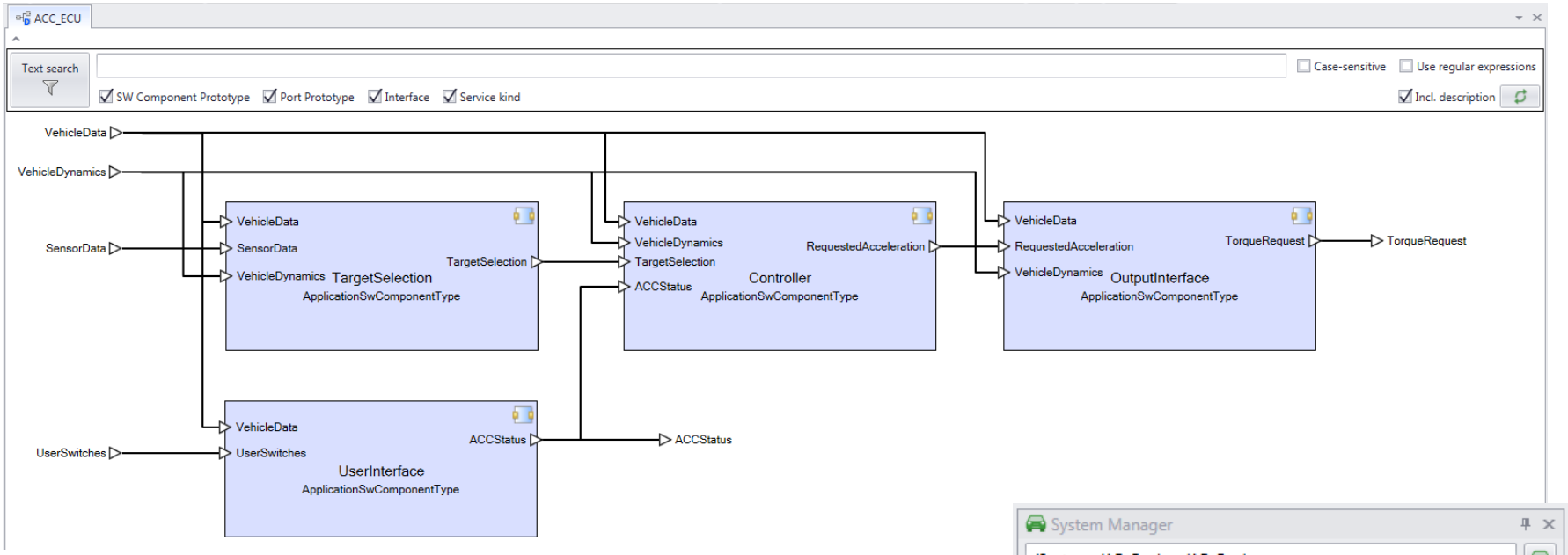
SystemDesk – Modeling of AUTOSAR Architectures

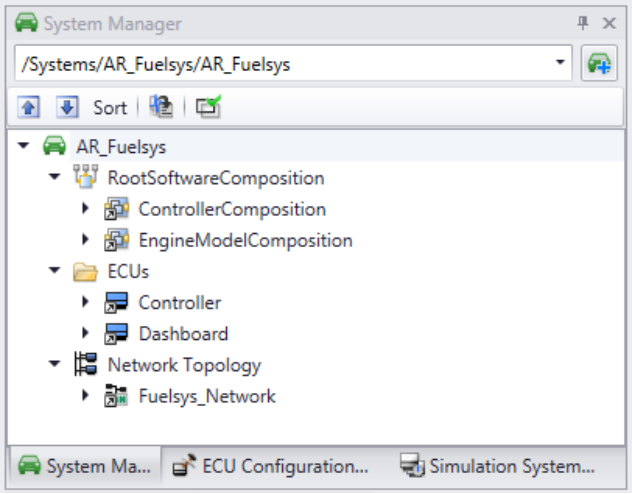




- Work with complex AUTOSAR objects in comfortable dialogs

SystemDesk – Modeling of AUTOSAR Architectures

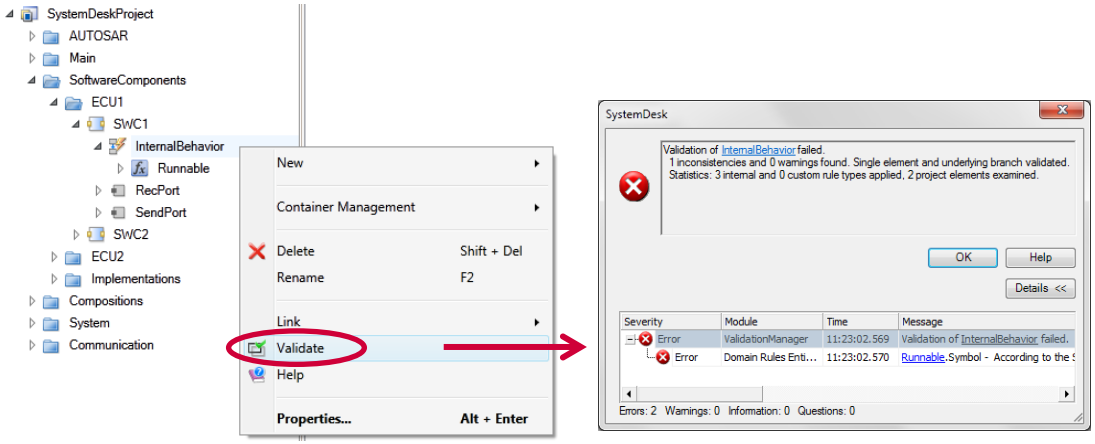




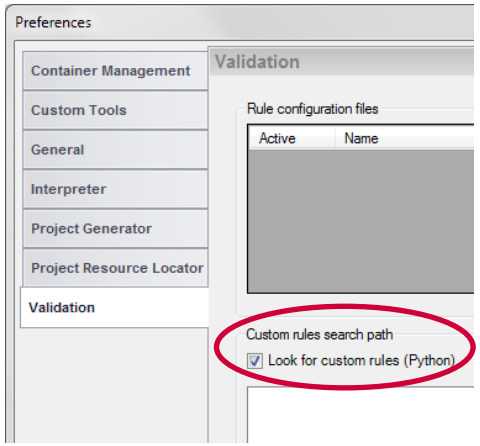
- Model architecture of software at a functional level
- Map functions to ECUs and Networks

Validation

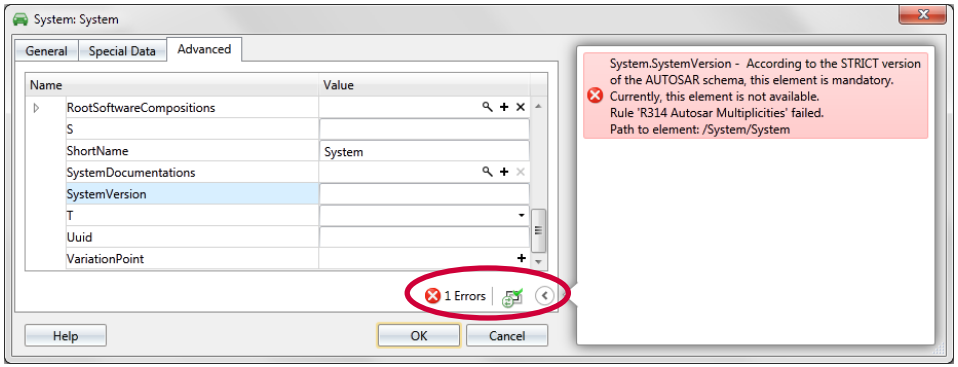
Complete validation of a branch of the project



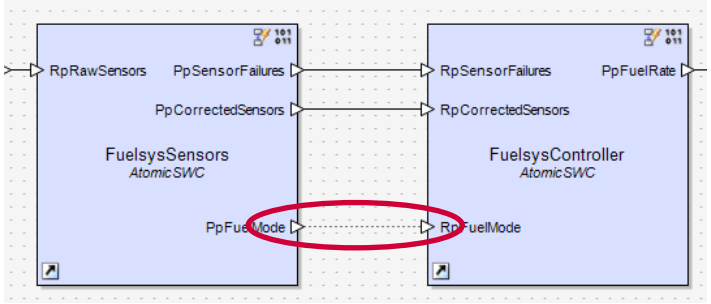
Definition of custom rules



Validation in dialogs

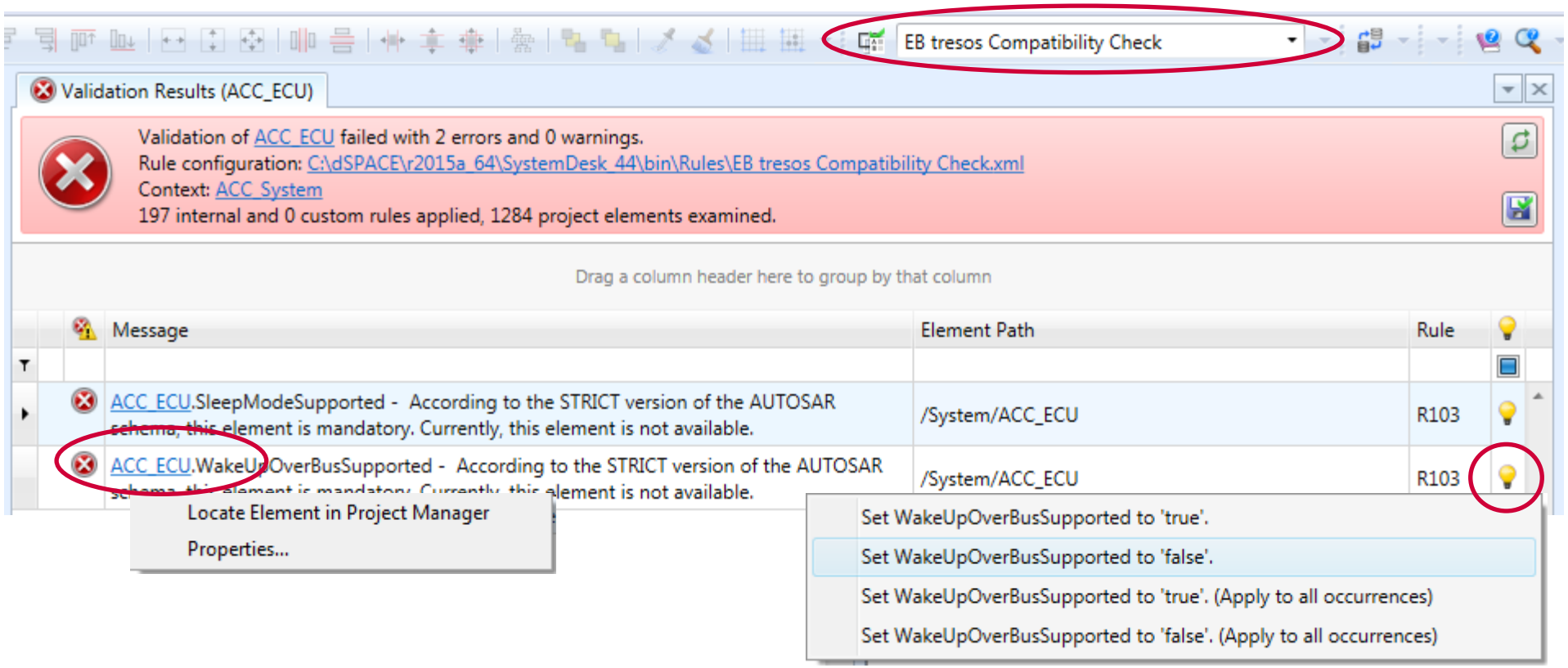


Visualization of incompatible interfaces

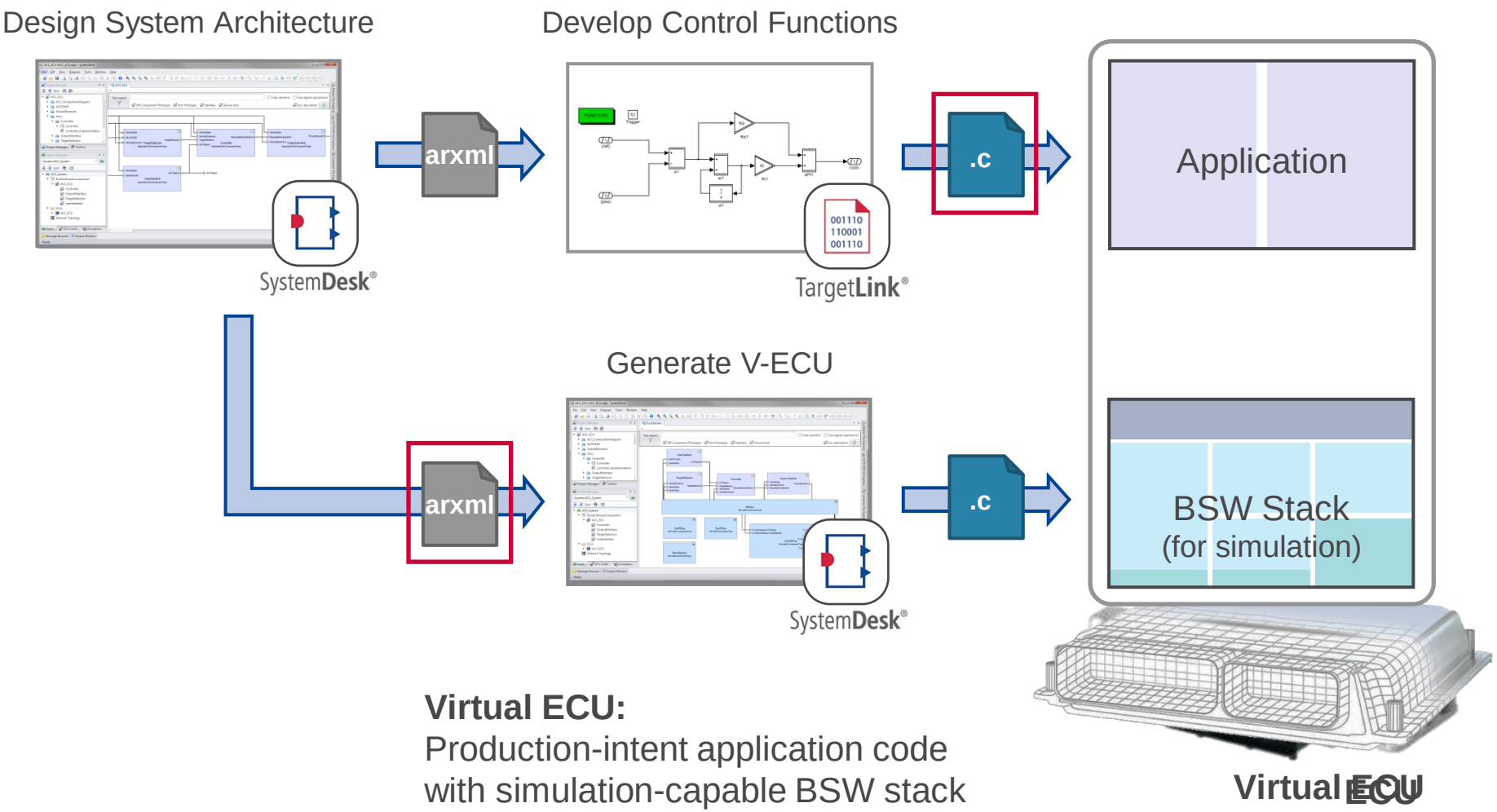


Validation – EB tresos Compatibility Check

Validate project for import into EB tresos



Virtual ECU Generation Process



Virtual ECU:
Production-intent application code
with simulation-capable BSW stack

Re-use V-ECU across dSPACE Platforms

Virtual ECU

MicroAutoBox II

Run production-intent code on a rapid prototyping platform

VEOS

Simulate realistic ECU on desktop environment without hardware

SCALEXIO

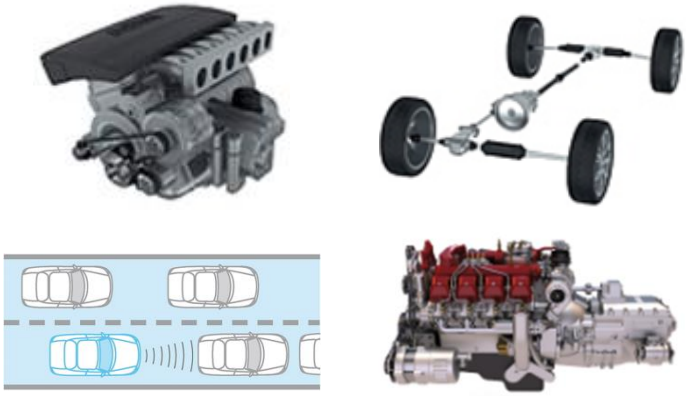
Connect Virtual ECU to physical signals for Hardware-In-Loop testing

Pieces Required for Simulation and Testing

dSPACE

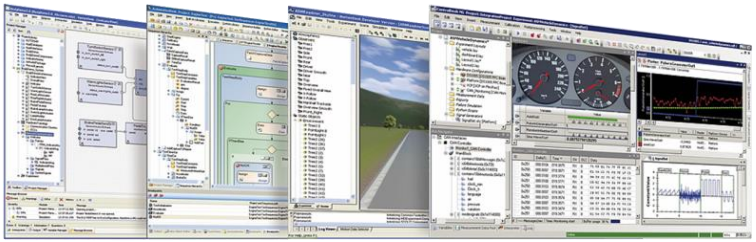
Device / System Under Test

Environment Models



Execution Platform

Test Tools

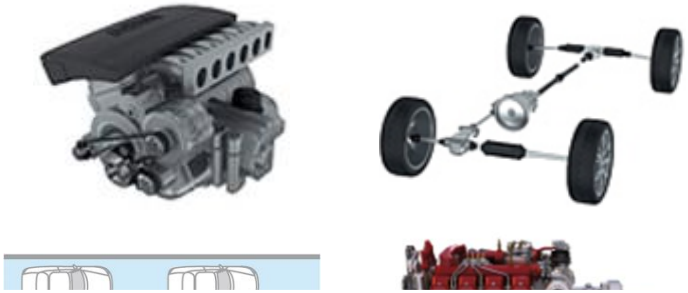


Pieces Required for Simulation and Testing

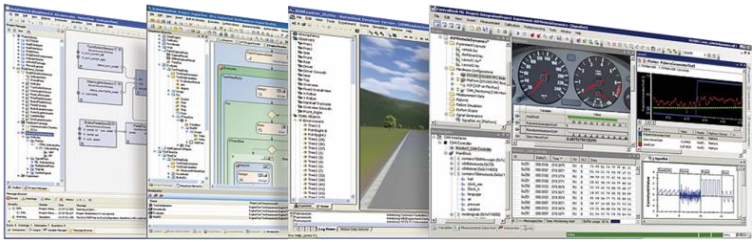
dSPACE

Device / System Under Test

Environment Models



dSPACE's Philosophy: Utilize industry **standards**
Create **scalable** solutions
Enable **seamless** transitions between phases of development

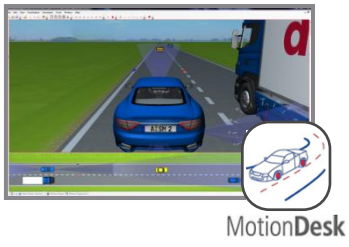


Integrated Tool Chain for Virtual ECU Development

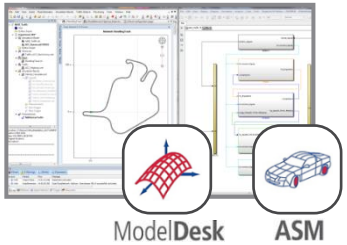
dSPACE

Instruments and Simulation Control

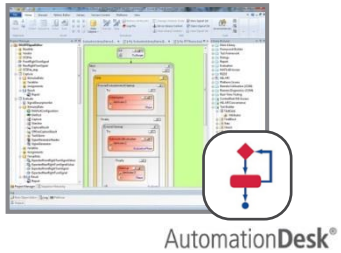
Visualization and Animation



Road and Maneuver Definition



Test Automation and Evaluation

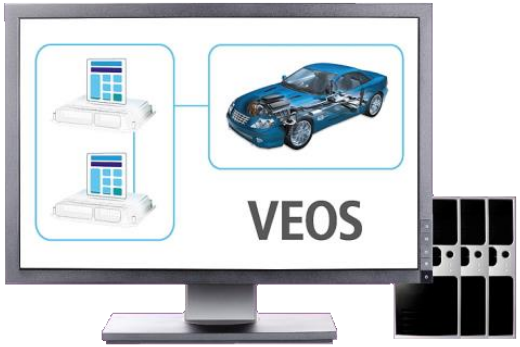


Third-Party Models and Test Tools

FMI

XCP

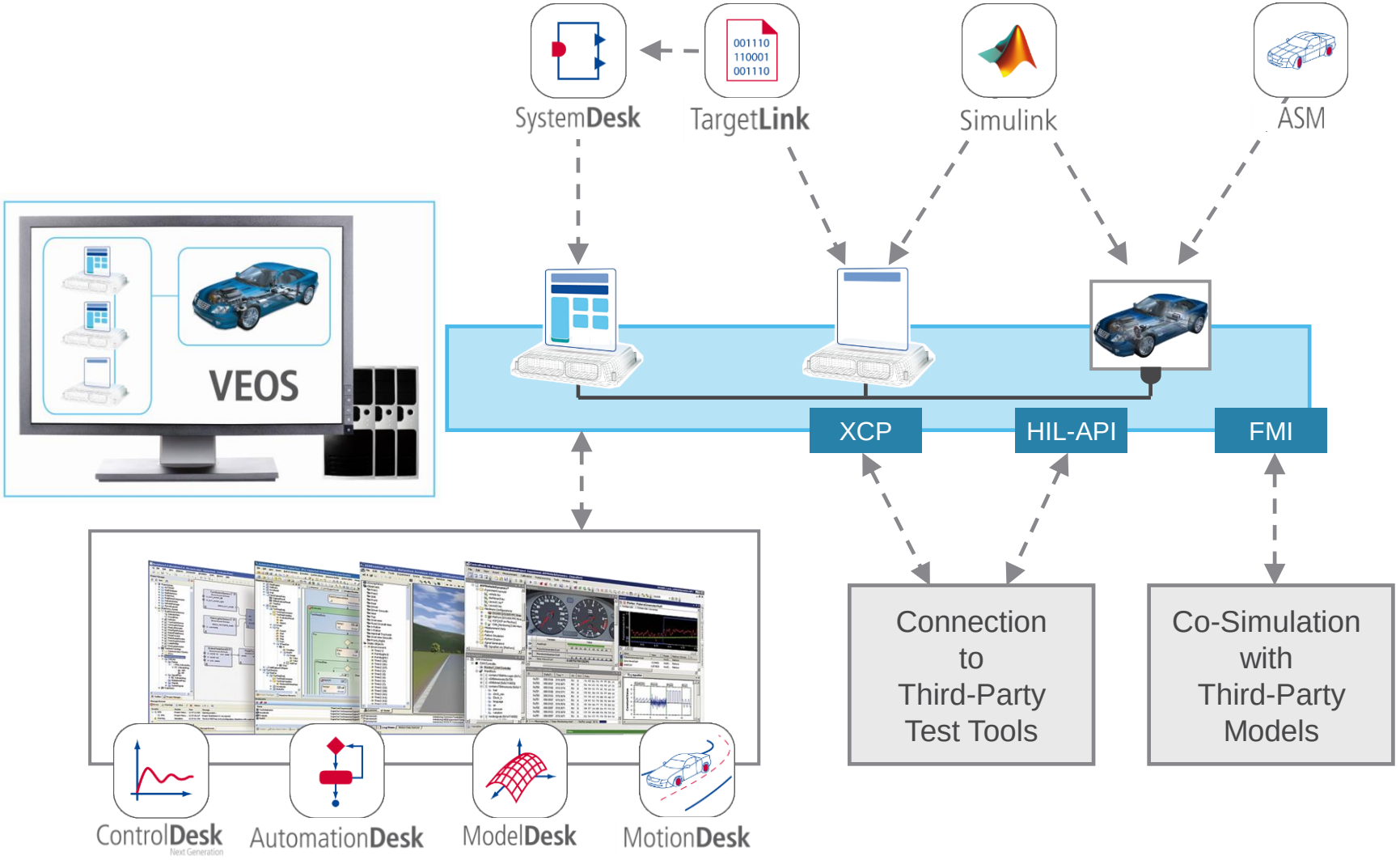
HIL-API



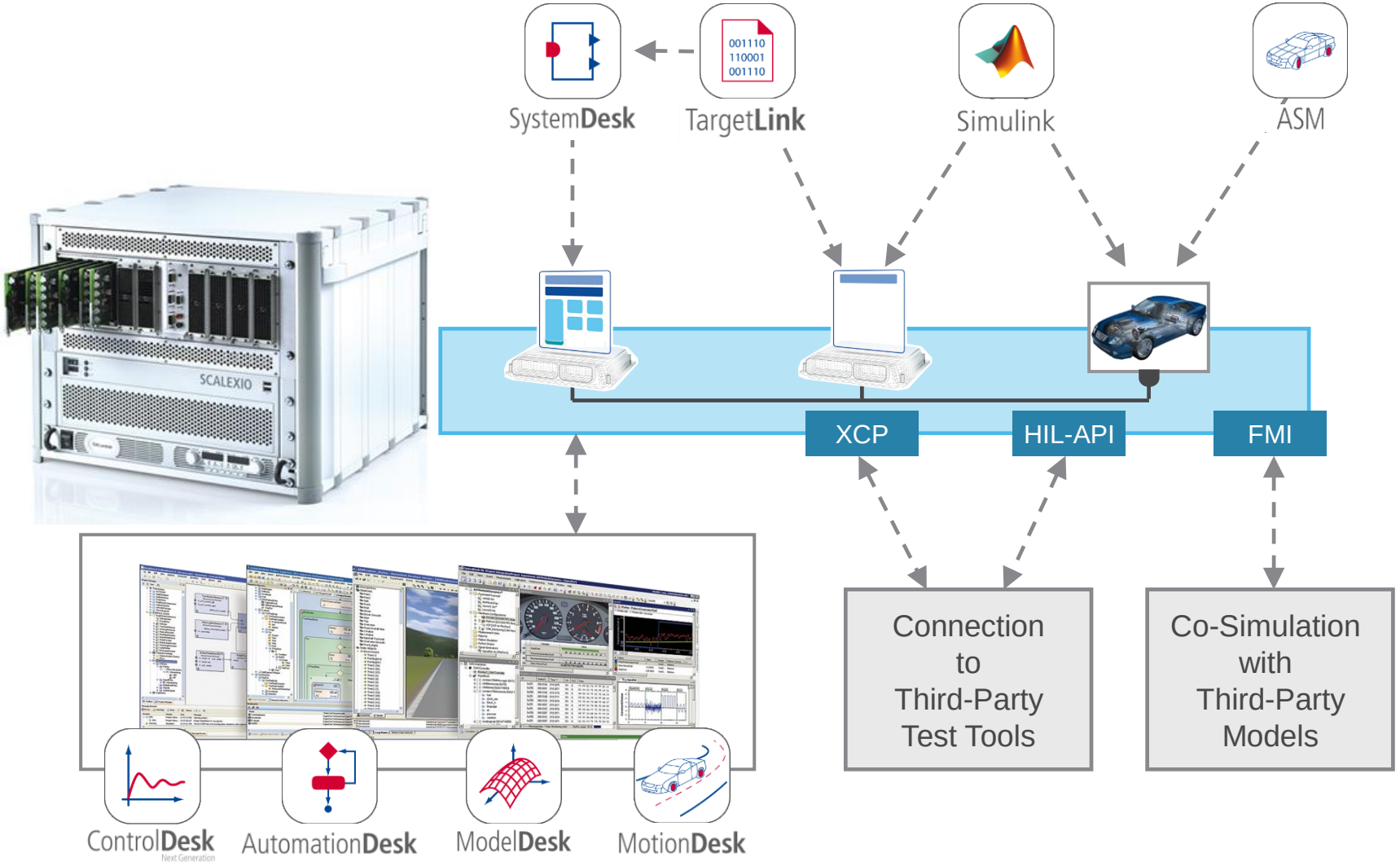
Early, PC-based validation of ECU software and functions

Seamless reuse of models, layouts, tests, data during real ECU testing

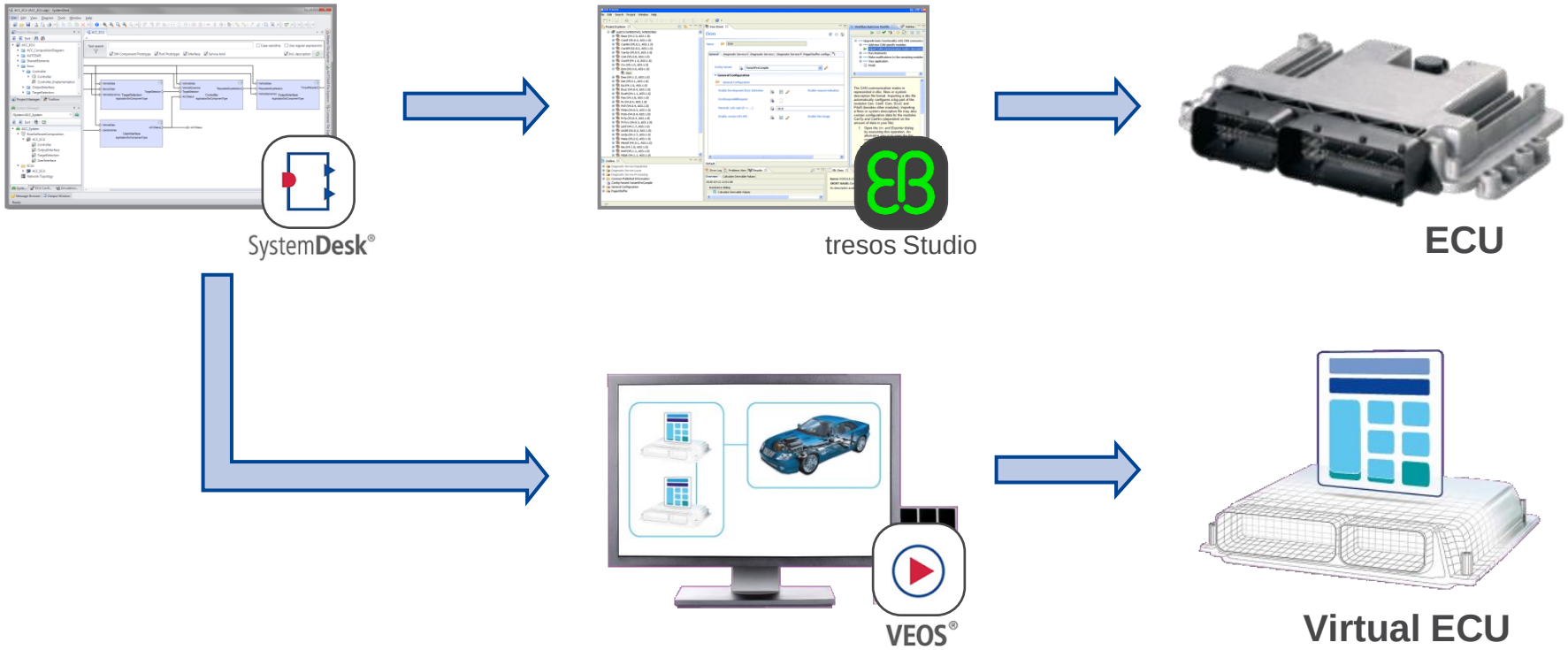
VEOS Player – Linking it all Together



Seamless transition from Offline Simulation to HIL



dSPACE and Elektrobit – Your AUTOSAR Partners



A **Standards-Based,**
Scalable,
Seamless solution for doing real and virtual testing across the entire software development cycle

Embedded Success dSPACE

Thank you! Any questions?