VECTOR

Extend Product Lineup using E-Mobility Platform

Vector Conference 2020

V1.0 | 2020-10-15

Agenda

- **Overview**

PREEvision Introduction

Model Based System Engineering

Product Line Management with Reuse Concept

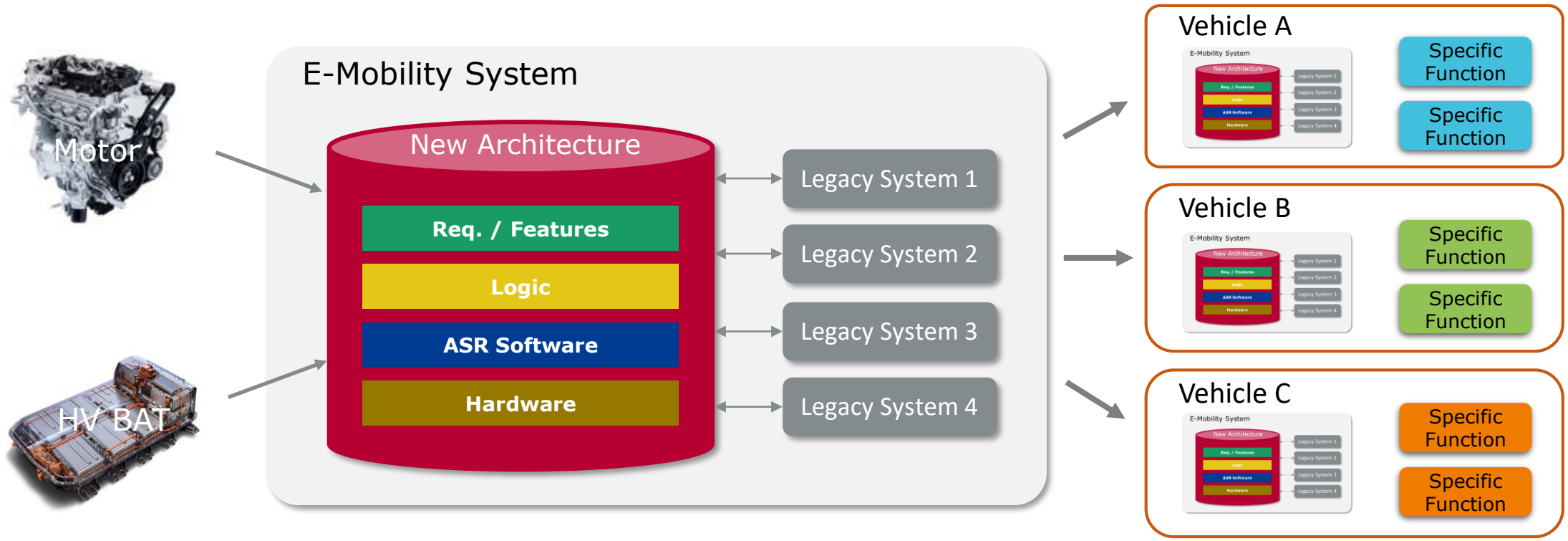
E-Mobility Architecture Design with Traceability and Consistency

System Requirement Specification Generation

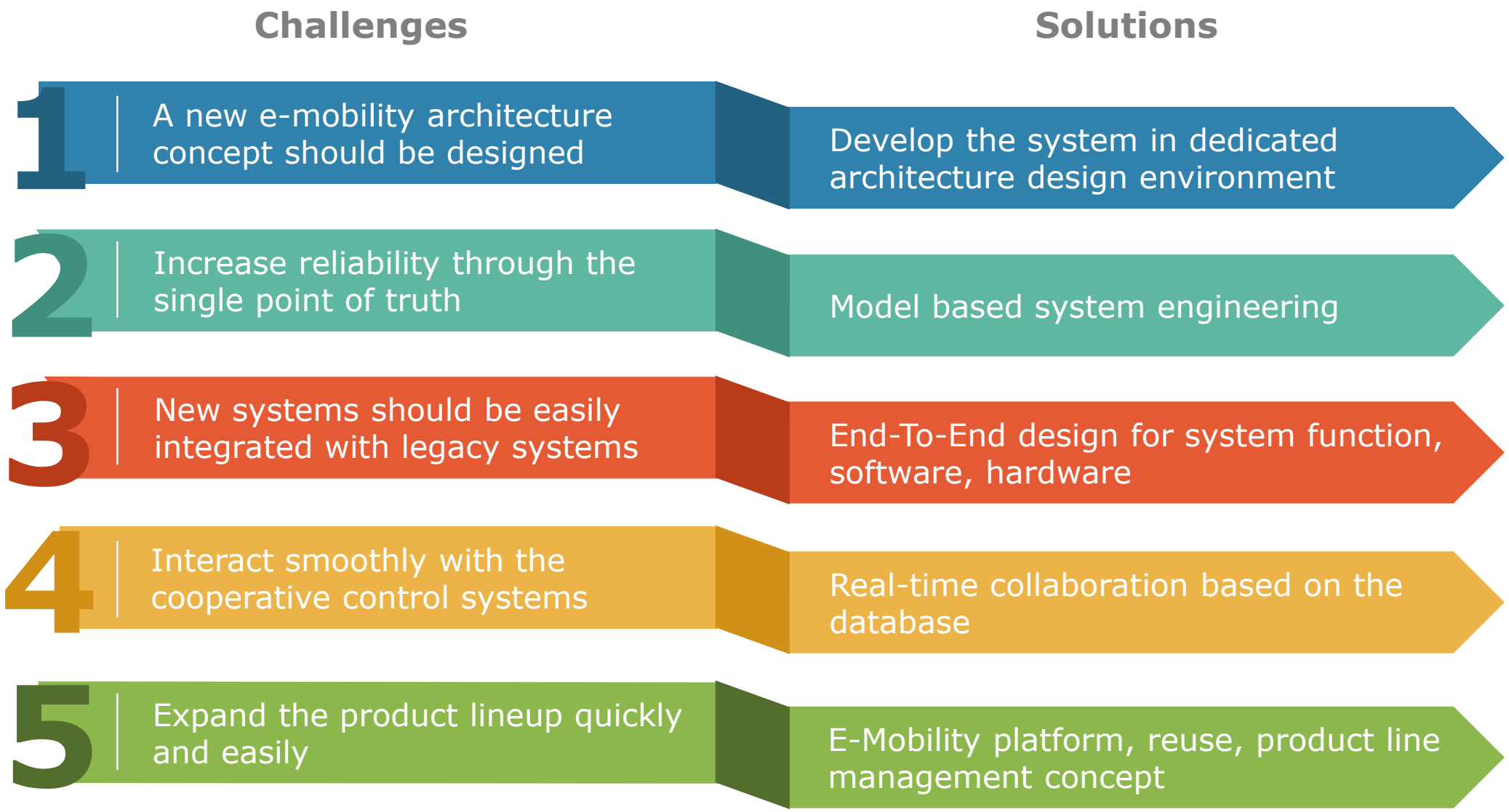
Summary

E-Mobility Development Objectives

- Motors and inverters take over the roles of engine and transmission in the vehicle driveline
- As the number of power consuming devices increases, a high-voltage power supply system is required
- The new architecture of the electrification system must be reliably integrated with the legacy systems
- E-Mobility product lineup needs to be expanded quickly to meet customer demand



Challenges & Solutions



Agenda

Overview

PREEvision Introduction

Model Based System Engineering

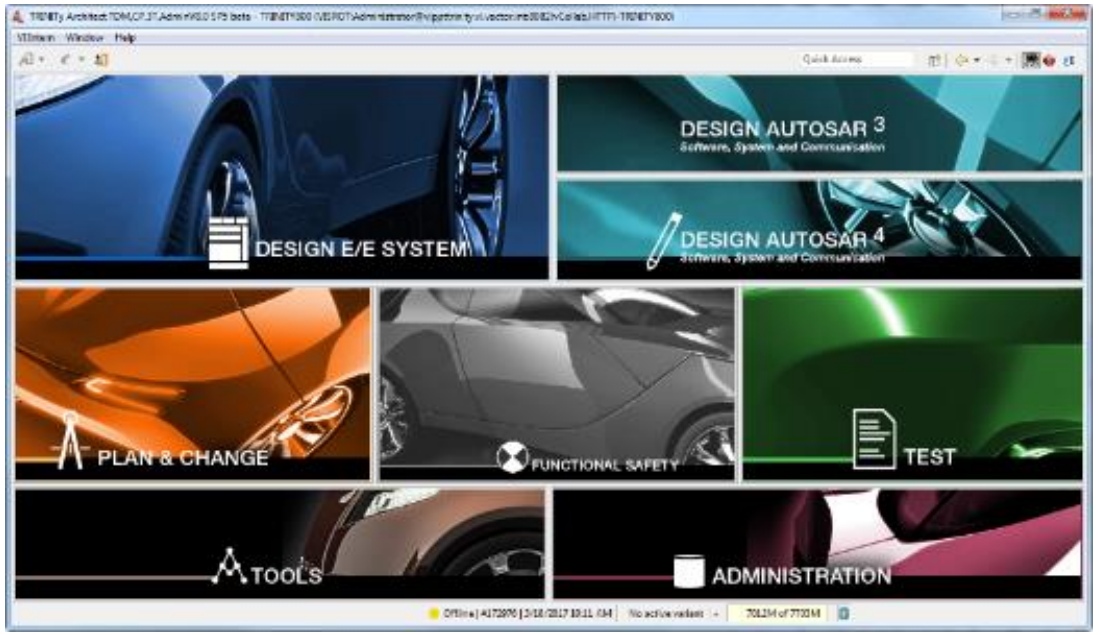
Product Line Management with Reuse Concept

E-Mobility Architecture Design with Traceability and Consistency

System Requirement Specification Generation

Summary

Enabler for today’s and tomorrow’s E/E platforms

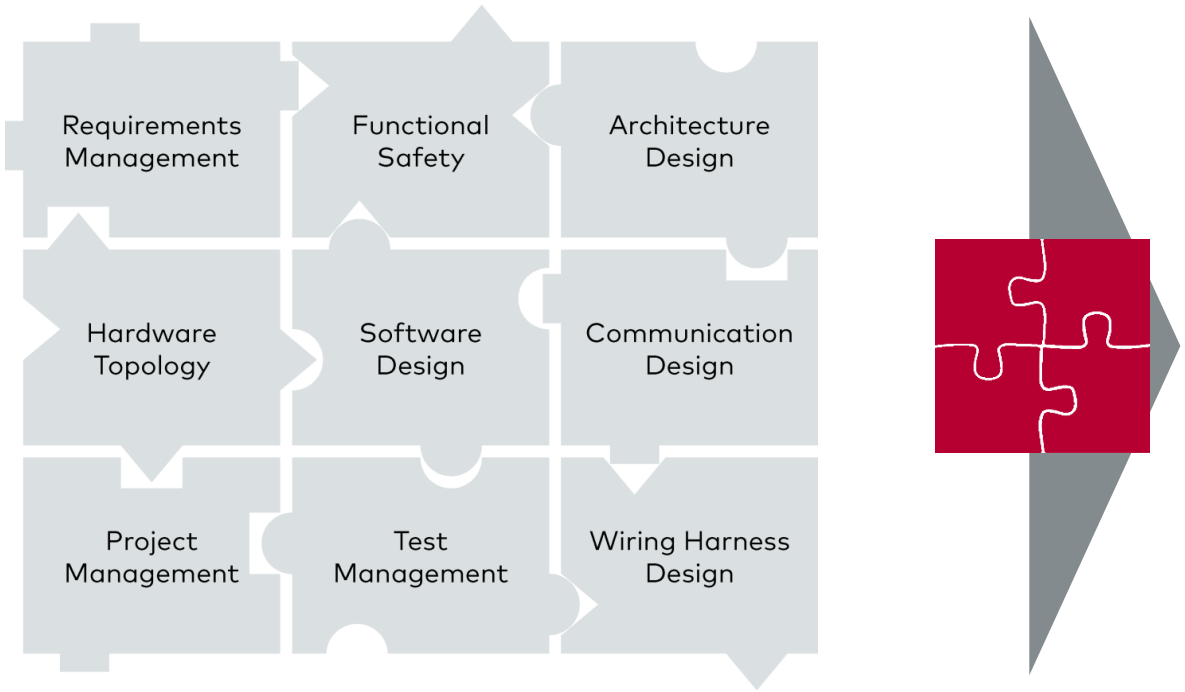


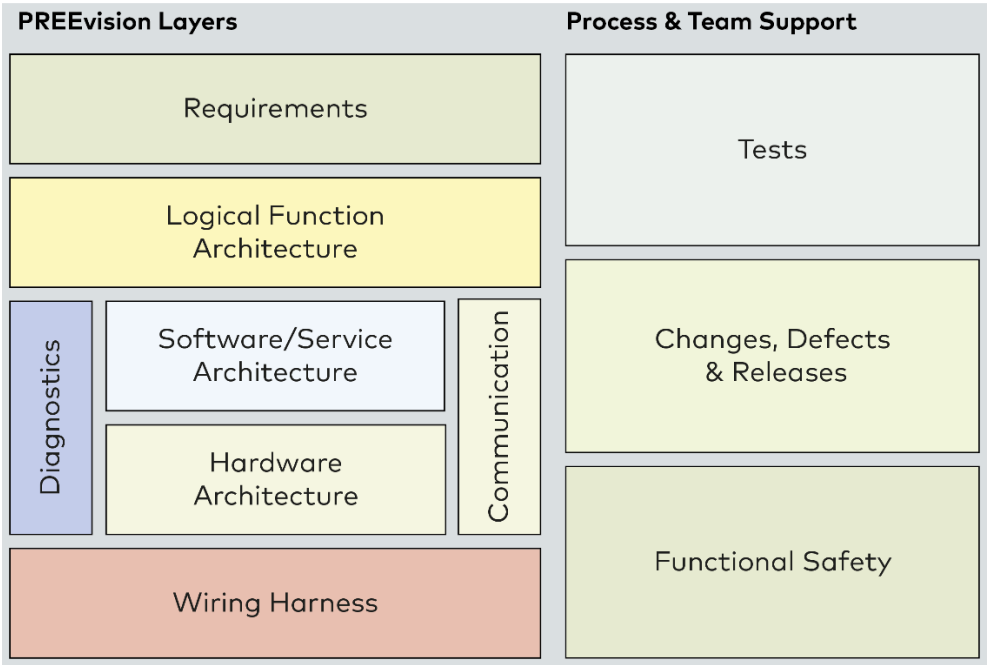
Digital engineering platform for automotive E/E systems engineering

> ONE platform for the design, management and documentation of complete E/E systems
> Integrated product line and variant management support
> Support of AUTOSAR (Classic and Adaptive) methodology
> Design of safety-relevant systems… and more!

Single-Point-of-Truth

- Documents have reached their limit – now it is time to put your data at the heart of everything you do!





Tool Landscape Approach (Best in Class)

- **Document** oriented development
- Various independent **data silos**
- **File based** collaboration

Integrated Tool Approach

- PREEvision **E/E Engineering** Environment
- **Data based** collaboration
- Central **backbone** as "Single Point of Truth"

The PREEvision Layer Model

PREEvision Layers

Chief Architect

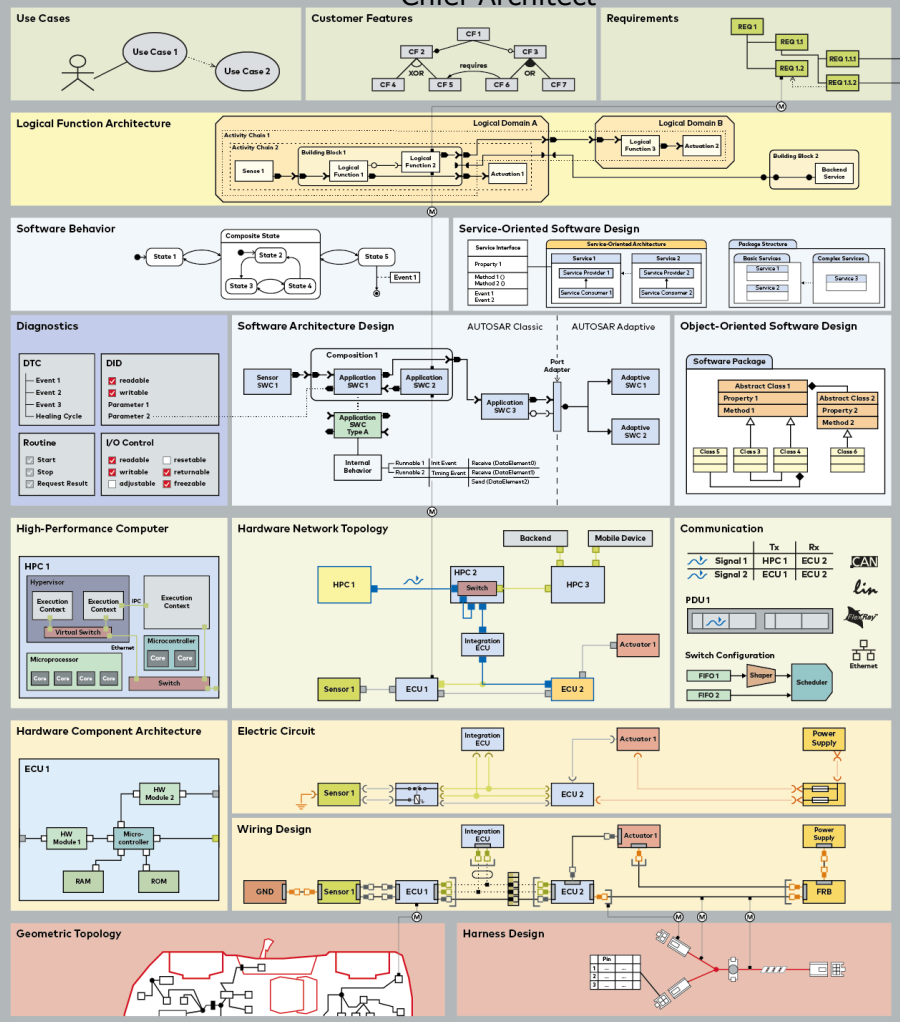
Req-Engineer

Function Designer

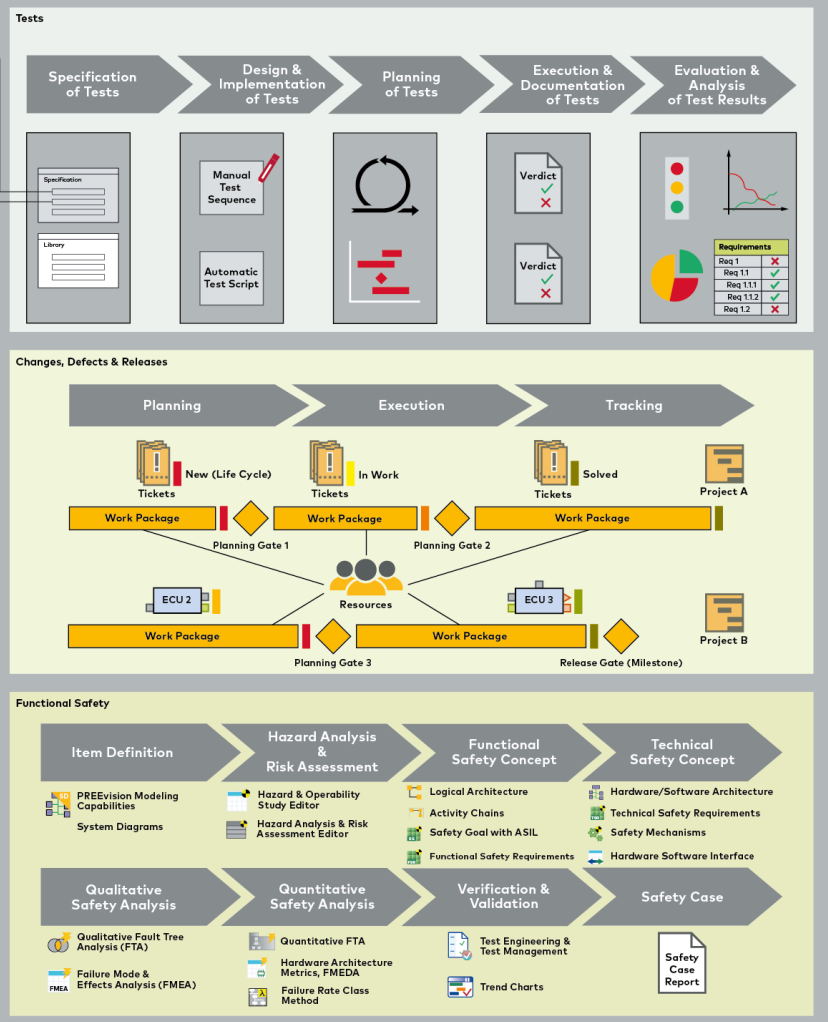
Function Designer
Network Designer
Diagnostic Designer

Component Designer

Wiring Harness Architect/Designer



Process & Team Support



Test Manager
Test Engineer

Project Manager

Safety Engineer
(Quality Manager)

4.4k Classes | 2.2k Attributes | 2.5k Relations

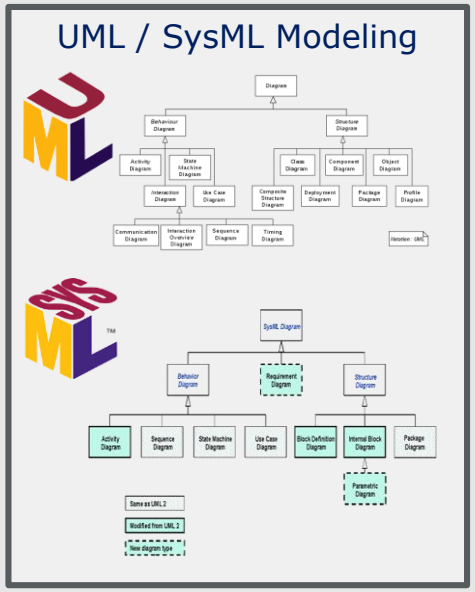
Agenda

- Overview
- PREEvision Introduction
- **Model Based System Engineering**
- Product Line Management with Reuse Concept
- E-Mobility Architecture Design with Traceability and Consistency
- System Requirement Specification Generation
- Summary

Two major Concepts

Data Model

SysML Modeling with Automotive Profile



Automotive Meta Model

Automotive E/E

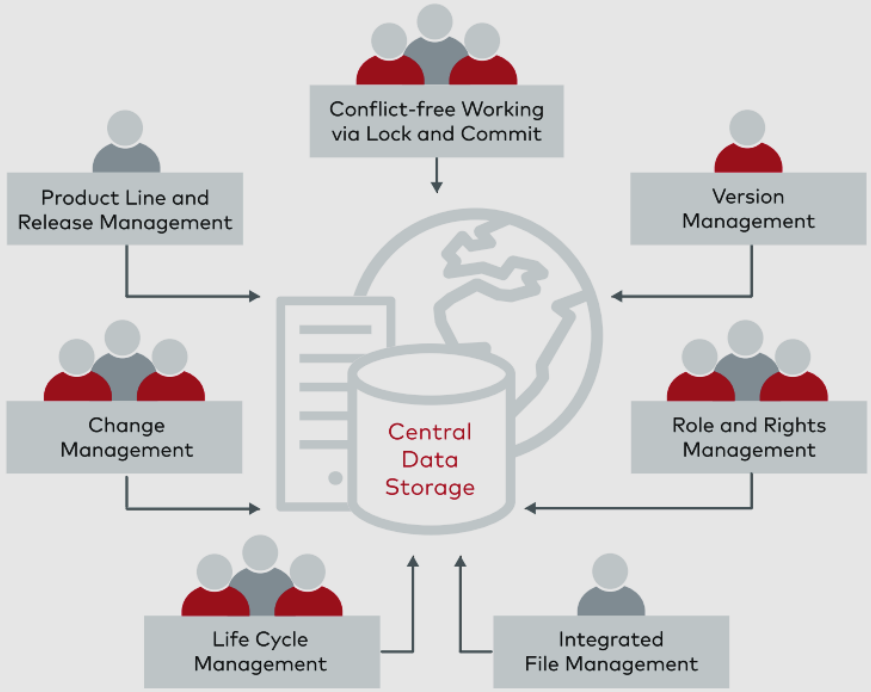
4400 Classes

2200 Attribute Types

2500 Relation Types

- Meta model is the de-facto **automotive industry standard**, developed with global OEMs and suppliers
- Support of the entire development process in a single **MBSE** tool

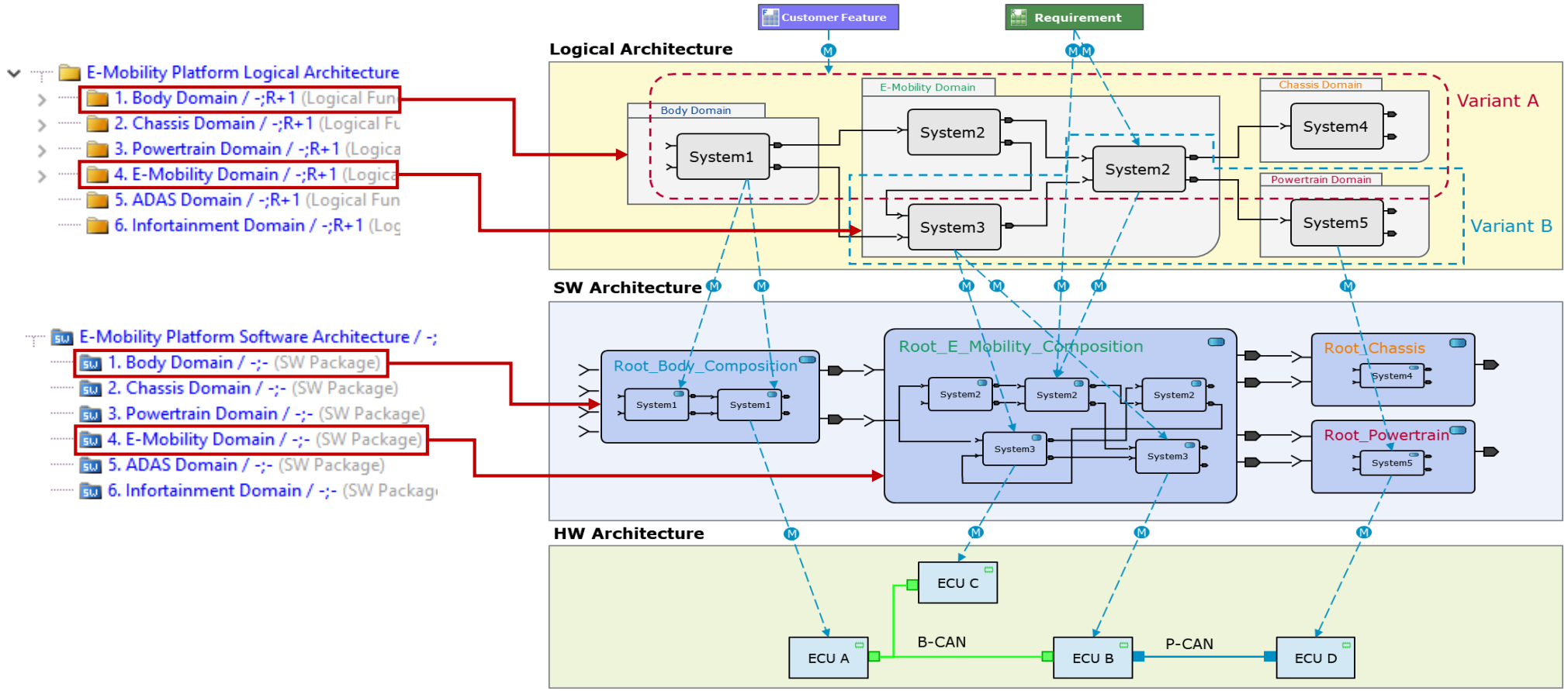
Collaboration Platform



- Powerful database for **real-time collaboration** including change-, release- and configuration management

System End-To-End Design

- End-to-end design from system requirements to function, software, hardware
- Relationships and dependencies can be visualized beyond the controller to the domain



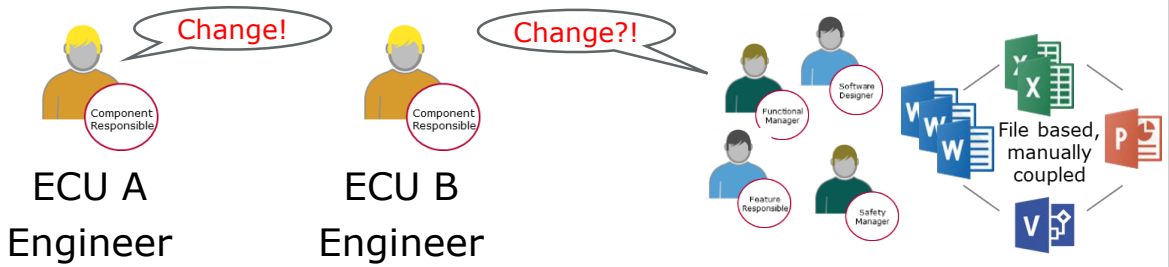
Traceability and Consistency

Design Change Scenario

- ECU A provides signal to ECU B
- ECU A interface to ECU B shall change
- Change control board needs to decide on change

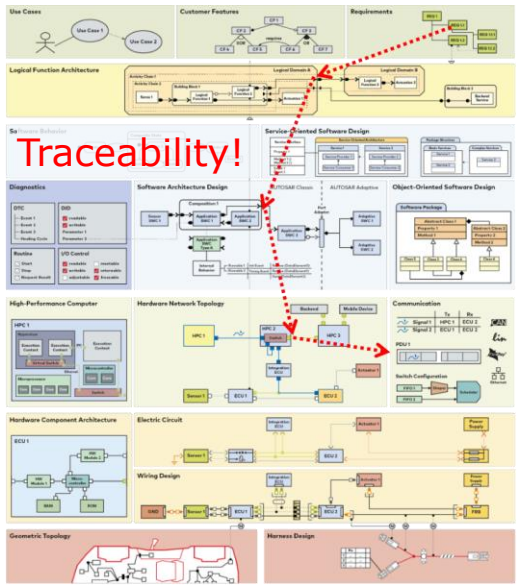
Document-based impact analysis

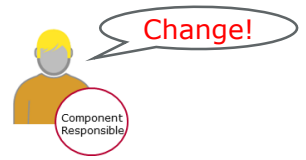
- High coordination and communication effort with related SW, COM, HW engineers
- High effort to manually trace dependencies in documents
- Application of change may impact other artifact / system



PREEvision Impact Analysis

- ECU A engineer follows IF change
  - > Efficient & systematic trace to SW, HW and COM
  - > Automatic consistency checks
- Efficient model-driven change analysis
- CCB decision is documented and change applied





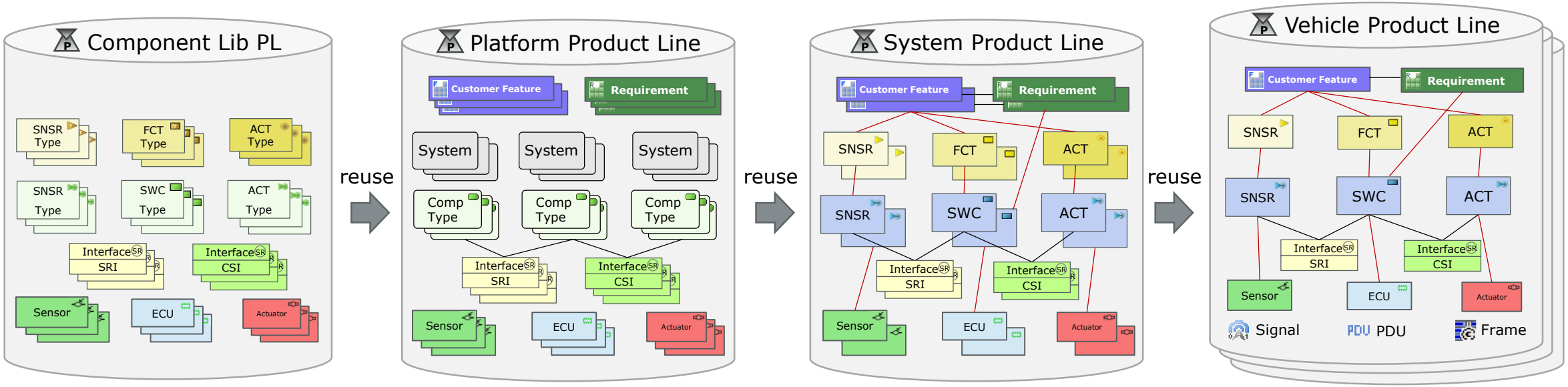
ECU A Engineer

Agenda

- Overview
- PREEvision Introduction
- Model Based System Engineering
- **Product Line Management with Reuse Concept**
- E-Mobility Architecture Design with Traceability and Consistency
- System Requirement Specification Generation
- Summary

From Platform to Vehicle

- Component Library Product Line : Component and type for standardization and reuse
- Platform Product Line : Modularization of the core technology of the E-Mobility system for reuse
- System Product Line : System configuration and data flow design including system variant
- Vehicle Product Line : Specific vehicle system function, software, hardware, communication for mass production



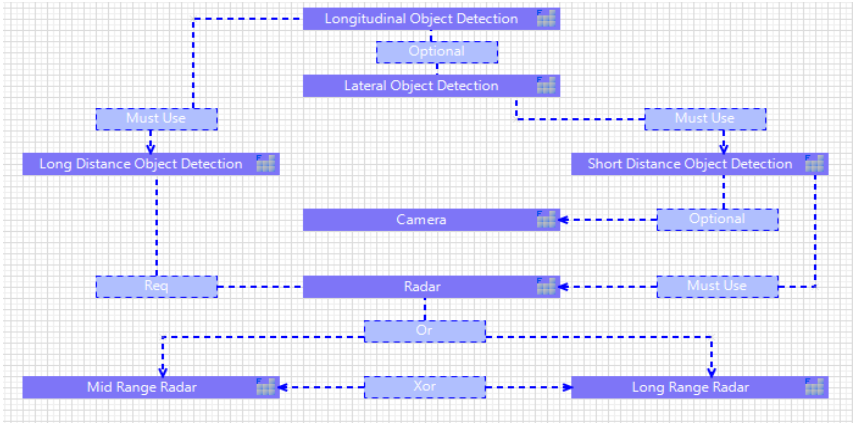
- Function block
- SW component type
- HW component
- Interface with data type

- Feature & Requirement
- Core technology function (Logic)
- Core technology SW module
- Core HW/Network topology
- Reused interface with data type

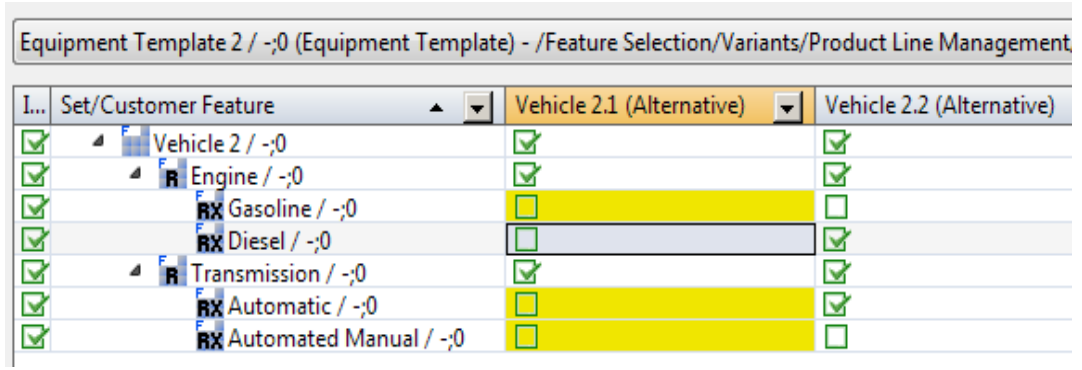
- System specific feature & req.
- System specific function
- System SW architecture
- System HW architecture
- Reused interface with data type

- Tailored vehicle feature & req.
- Tailored vehicle specific function
- Tailored vehicle SW architecture
- Tailored vehicle HW architecture
- Vehicle specific communication

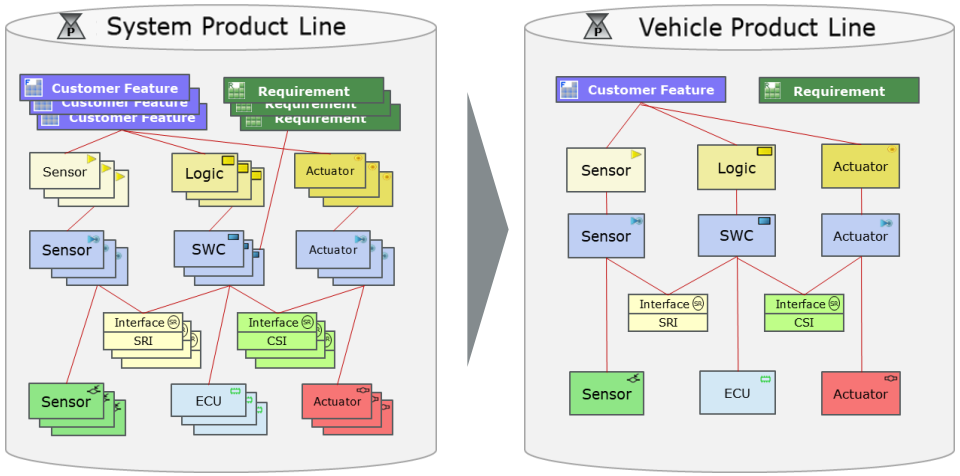
Variant Management



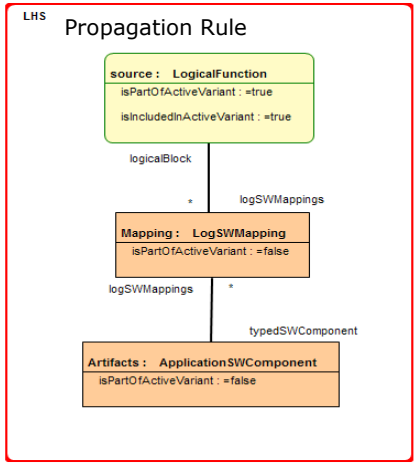
- Create Customer Feature relations using Variant Diagram

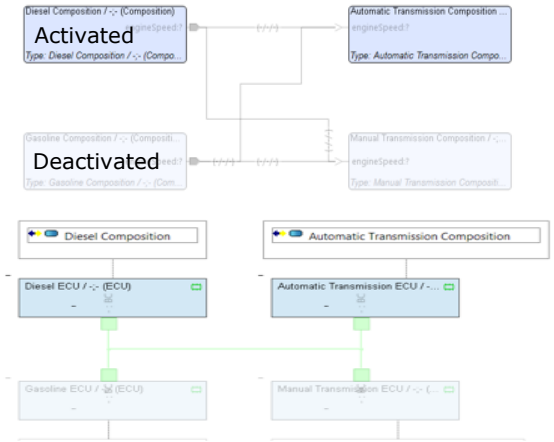


- Configure Alternatives by selecting Customer Features using variant templates



- Derive a new product line to create a vehicle-specific system using the propagated artifacts



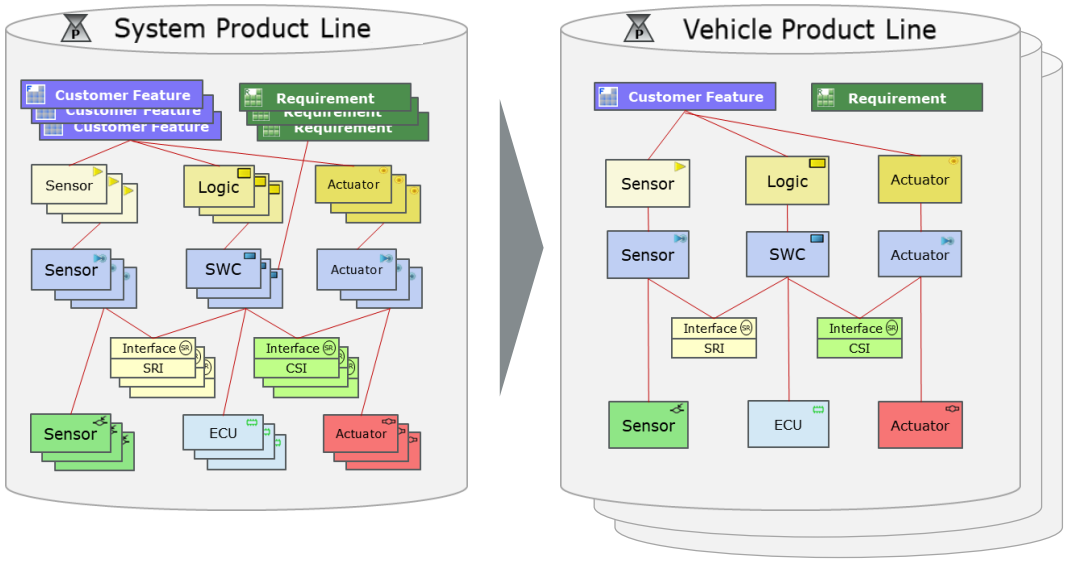


- Propagate artifacts that are mapped to the Customer Features into an Architecture Variant container using Propagation Rule

Product Line Derivation Method

- PREEvision supports three product line derivation methods (Clone & own / Reuse / Asset)
- Each of the three methods has a different mechanism

How to transfer the data from platform to system and system to individual vehicle product line?



1. Copy (Clone & own)

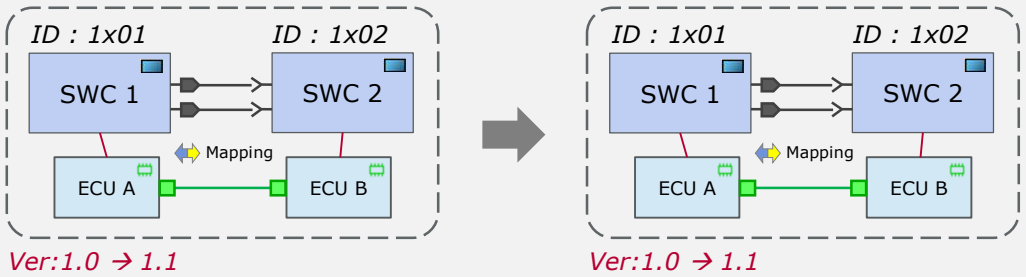
Relation is broken

2. Reuse

Synchronized

3. Asset

Synchronized Set

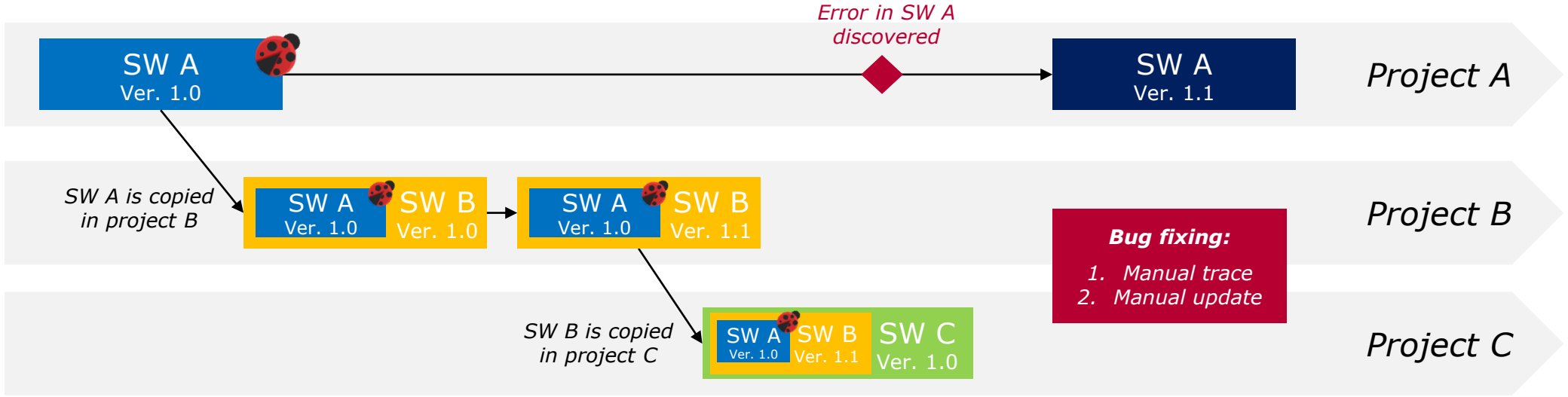


Clone & own approach

Reuse is nothing new to the automotive industry

- Goal: Taking advantage of commonality shared across vehicle families
- Typical approach: Clone & own approach

Example: Clone & own and bug fixing over time



Biggest disadvantage of a Clone & own approach:

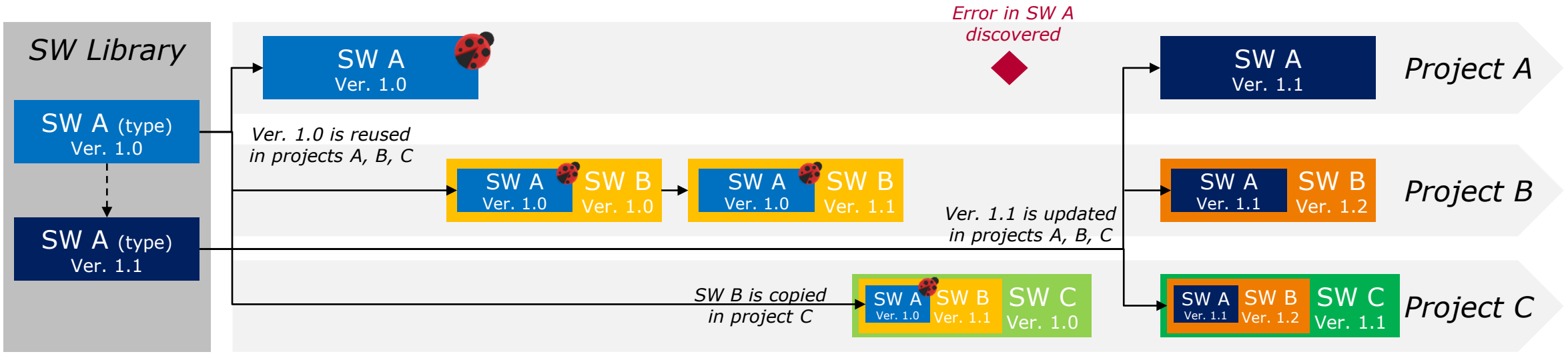
- Changes need to be updated manually over multiple projects!
- **High effort** and **error prone**!

Library-based reuse approach

Library-based standardization and reuse

- Library: Management of reusable components
- Projects: Instantiation and **update** of library components

Example: Library-based SW with reuse in projects and bug fixing over time



Biggest advantage of a library-based approach:
- Type – Instance Concept: **No copies, but reuses!**
- New revisions can be updated to projects very efficiently!

Agenda

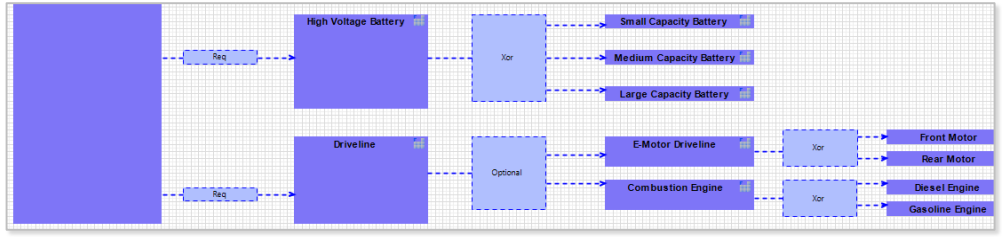
- Overview
- PREEvision Introduction
- Model Based System Engineering
- Product Line Management with Reuse Concept
- **E-Mobility Architecture Design with Traceability and Consistency**
- System Requirement Specification Generation
- Summary

Customer Feature and Requirement Design

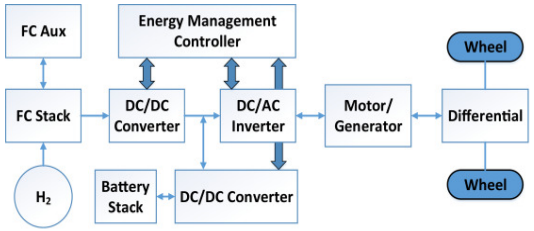
- E-Mobility Platform Feature List is created by separating common and variation point of each system
- Through Feature Condition Diagram, user can select effective feature combination and reduce human Error
- System Requirement List describes the technical requirements or the regulations of the system

Name	Description
E-Mobility System (Variant Diagram)	
4.1 E-Mobility Platform / -;- (Customer Feature)	E-Mobility Platform
4.1.1 Electric Power Source / -;- (Customer...	Electric Power Source
4.1.1.1 External Charge / -;- (Customer...	Charging function from external power
4.1.1.1.1 Boosting Charge / -;- (Cu...	Rapid charging function through quick charging terminal(~380V)
4.1.1.1.2 Slow Charge / -;- (Custo...	Charging function from general household voltage terminals(220V~)
4.1.1.2 Fuel Cell Stack / -;- (Customer ...	Fuel Cell Stack
4.1.1.2.1 Capacity / -;- (Customer ...	The capacity of the fuel cell stack is expressed in kW, and when it is 60kW, it is referred to as a large capacit
4.1.1.2.1.1 80 kW / -;- (Custom...	Fuel Cell Stack with 80kW capacity
4.1.1.2.1.2 100 kW / -;- (Cust...	Fuel Cell Stack with 100kW capacity
4.1.1.2.2 Feature / -;- (Customer Fe...	System Feature
4.1.1.2.2.1 H2 Supply / -;- (Cus...	Hydrogen supply function
4.1.1.2.2.2 FC temperature Con...	Fuel cell temperature status check function
4.1.1.2.2.3 O2 Supply / -;- (Cus...	Oxygen flow and quantity measurement function
4.1.1.2.2.4 Cooling Control / -;...	Cooling cell stack function
4.1.1.2.2.5 H2 Tank Manageme...	Condition and leak of hydrogen tank check function
4.1.1.2.2.6 Battery Management...	Battery status and voltage and current status measurement Function
4.1.1.2.2.7 FC Stack Protection ...	Fuel Cell Stack protection from high temperature
4.1.1.2.2.8 TQ Generation and R...	Motor torque and regenerative braking control Function
4.1.1.2.2.9 Power Distribution ...	Power distribution for high voltage
4.1.1.2.2.10 FC Monitoring / -;-...	Cell Stack monitoring function
4.1.2 High Voltage Battery / -;- (Customer...	High Voltage Battery
4.1.2.1 Small Capacity Battery / -;- (C...	Generally, Small Capacity Battery is used for fuel cell electric vehicle, and its capacity is around 1kWh.
4.1.2.2 Medium Capacity Battery / -;- (...	Generally, Medium Capacity Battery is used for Hybrid electric vehicle, and the capacity is more than 1kWh.
4.1.2.3 Large Capacity Battery / -;- (C...	Large Capacity Battery
4.1.2.3.1 60 kWh / -;- (Customer Fe...	1st Grade of large capacity battery. 60 kWh means power generation for hour in the battery cell.
4.1.2.3.2 80 kWh / -;- (Customer Fe...	Large Capacity Battery is for long range distance driving. In general Electric Vehicle choices these battery

<E-Mobility Platform Feature List>



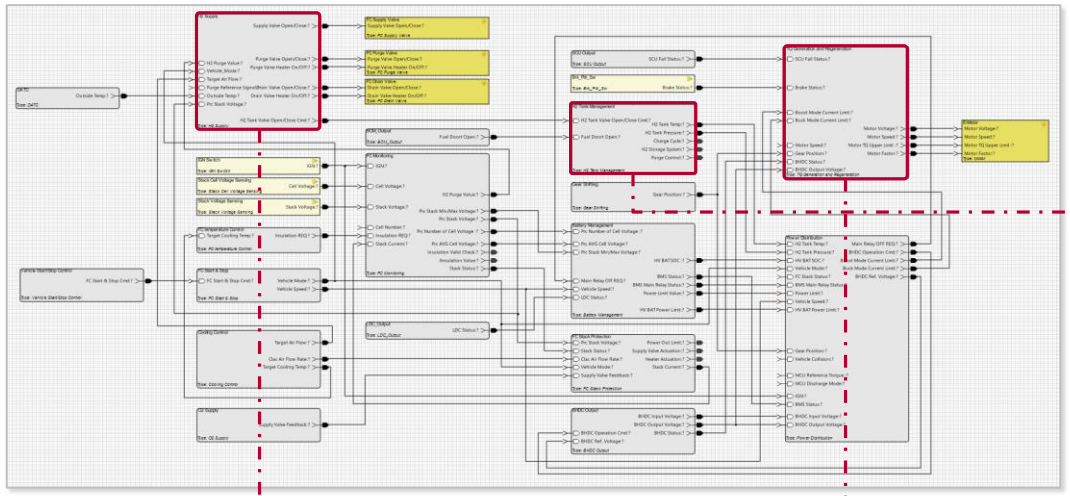
<Feature Condition Diagram>

LEVEL	ID	Hierar...	Text	Description
20-3	REQ_41534		**4.1.2.2 System Structure**	
21-4	REQ_41559		**--- HW Structure**	The structure of FCEV system is divided into FC Stack part, Inveter and Converter part and Motor/ Generation part as shown below.  Variant of FECVsystem can be expressed as follows, and the product line for each variant can be mass produced as a product.
22-3	REQ_41535		**4.1.2.3 System Feature**	
23-4	CF_01		**4.1.2.3.1 H2 Supply**	
24-5	CF_01_001		**--- H2 Supply Target Pressure Calculation**	H2 Supply Target Pressure Calculation • H2 Supply acquires object information and sends it to the FCEV system as input. • H2 Tank is divided into Variant according to Range, and the range and type to ranges are different. • In general, Target Pressure is calculated by FCU for acknowledging exact pressure calculation
25-5	CF_01_002		**--- H2 Supply Pressure Fluctuation Control**	H2 Supply Pressure Fluctuation Control • H2 Supply Pressure Fluctuation Control can detect the fluctuation of H2 Tank. • Final Fluctuation Value determining control mode.
26-5	REQ_41563		**--- Current Calculation**	Current Calculation • Current Calculation is the most accurate means to detect current in min/max range. • Current Calculation is divided into Mono camera and Stereo camera. • Detection angle and resolution vary depending on the variant.
27-5	CF_01_001		**--- H2 Purge Control**	H2 Purge Control

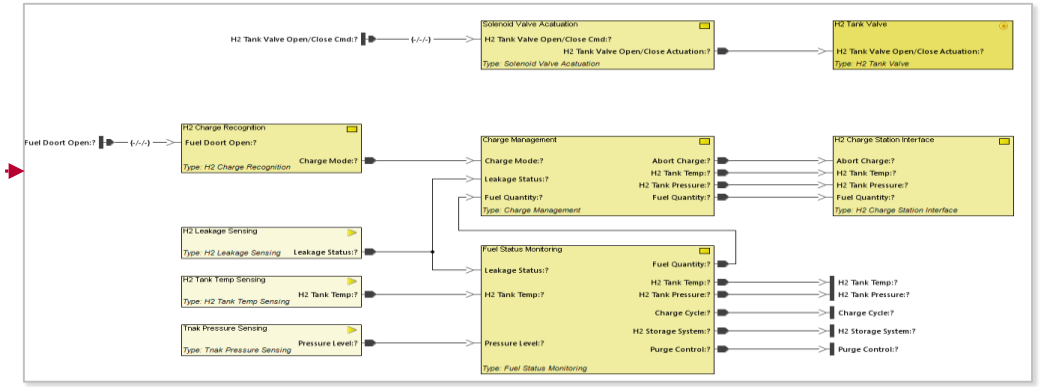
<System Requirement List>

Logical Architecture Design

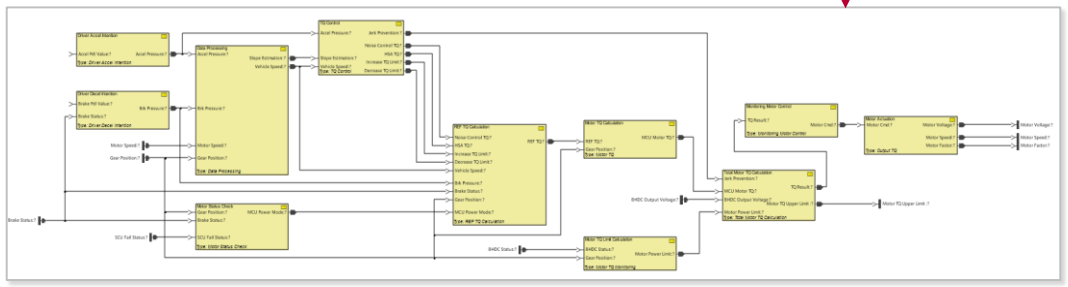
- Logical Architecture describes the abstracted e-mobility functions of the entire system
- End-to-end design including the system to be developed and the functions of the cooperative control system
- Before technical implementation, define the system scope and role by using abstract functions and interfaces



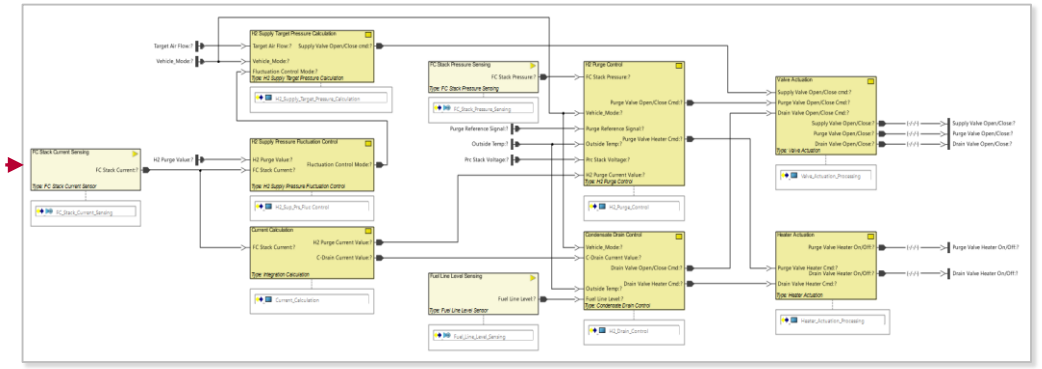
H2 Tank Management Function



Motor Torque Generation Function

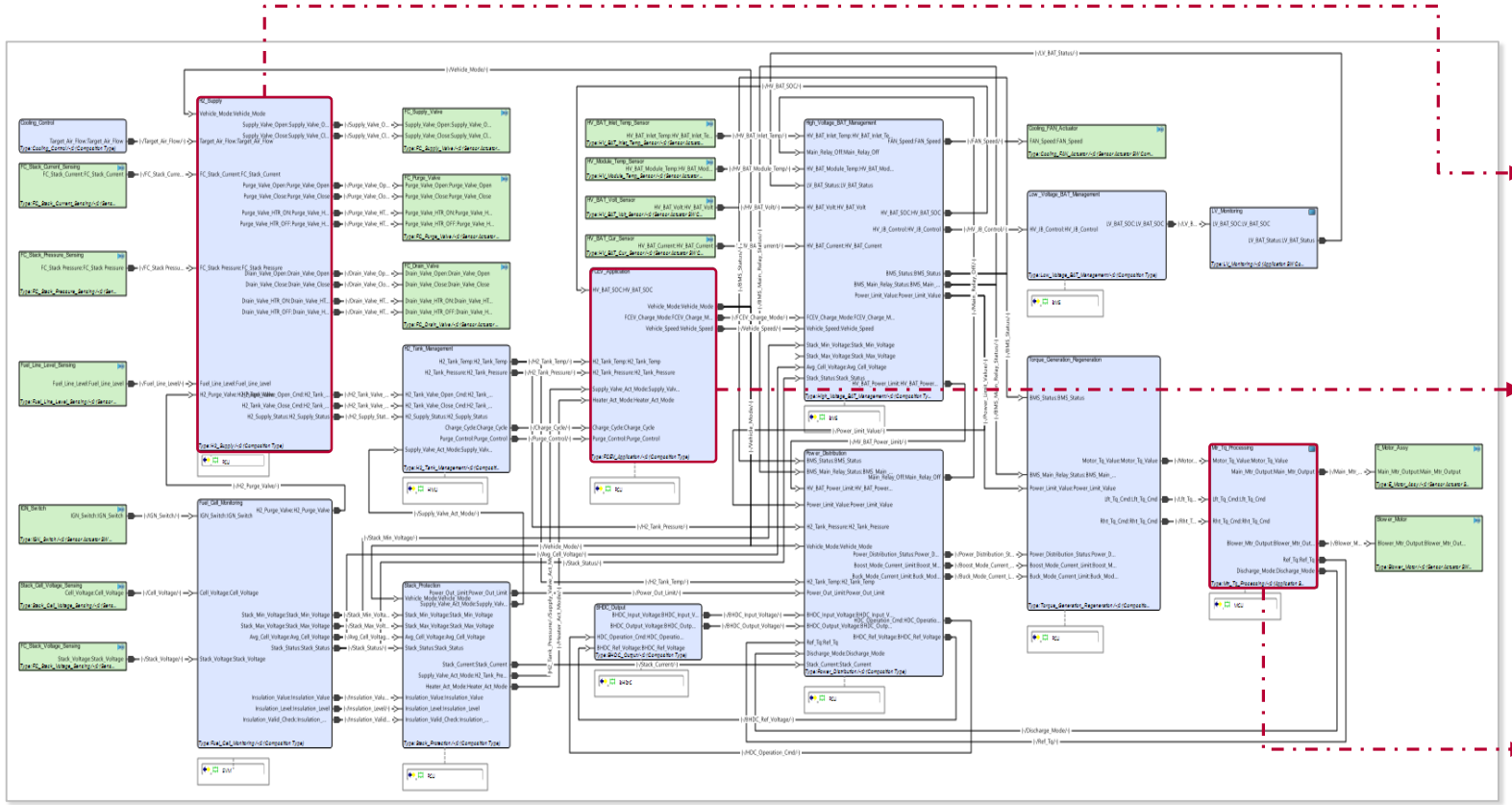


H2 Supplement Function

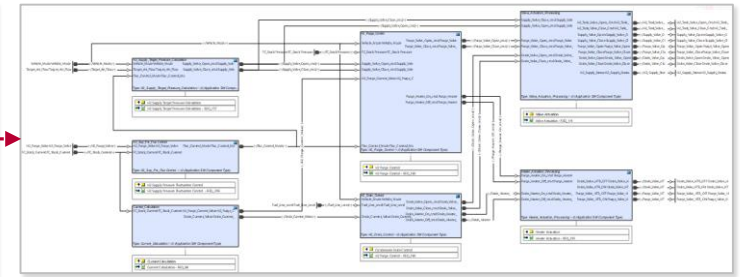


Software Architecture Design

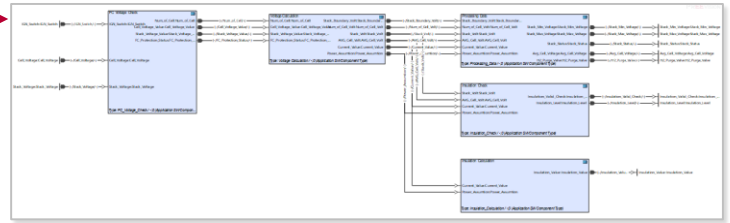
- Software Architecture defines system functions and interfaces at the technical implementation level
- Software components are instantiated and reused based on the types of E-platform library
- Each software component or composition is mapped to the ECU to be implemented



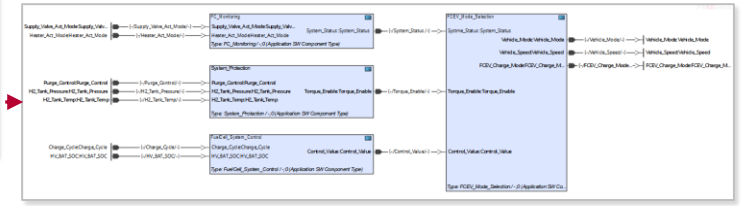
Mtr_Torque_Gen_Regen



High_Voltage_Monitoring

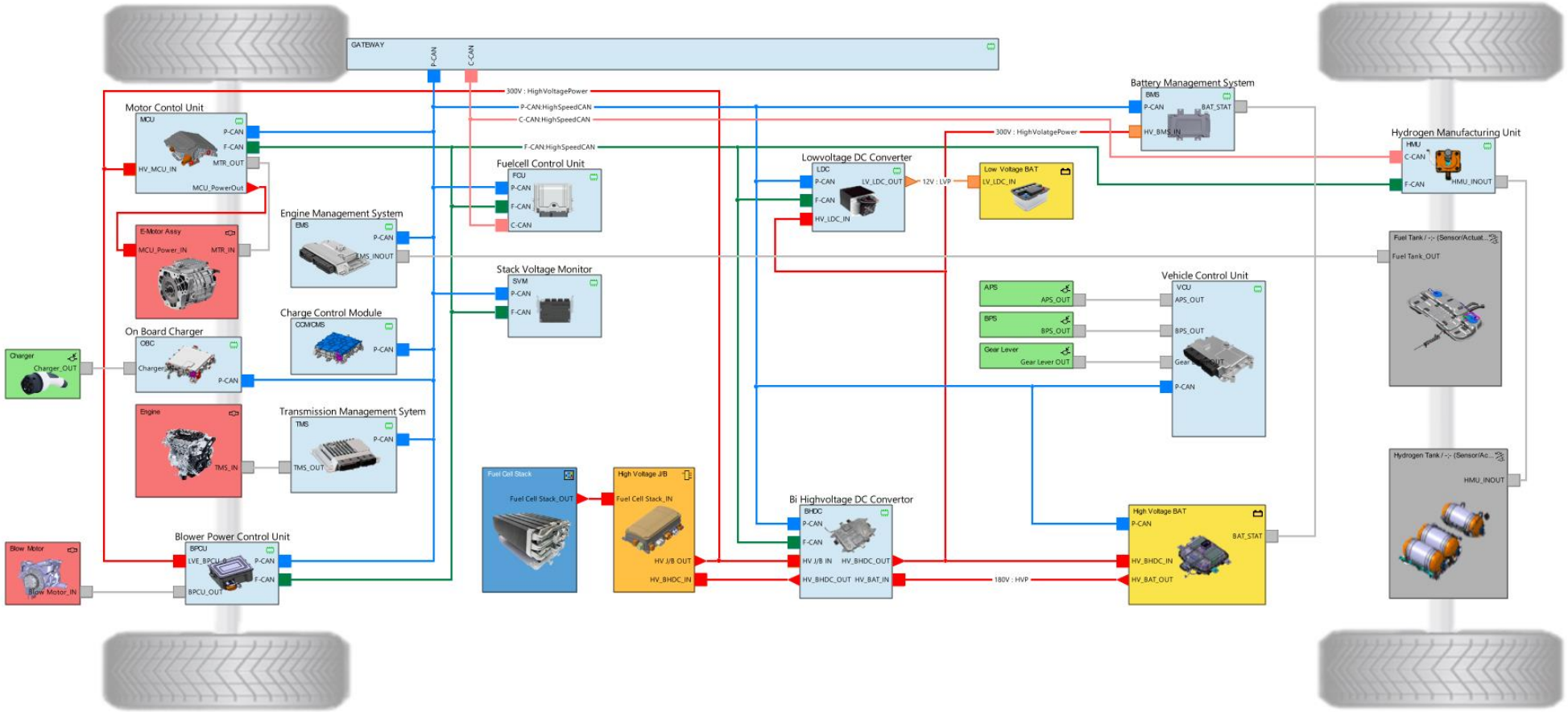


Mtr_Tq_Processing



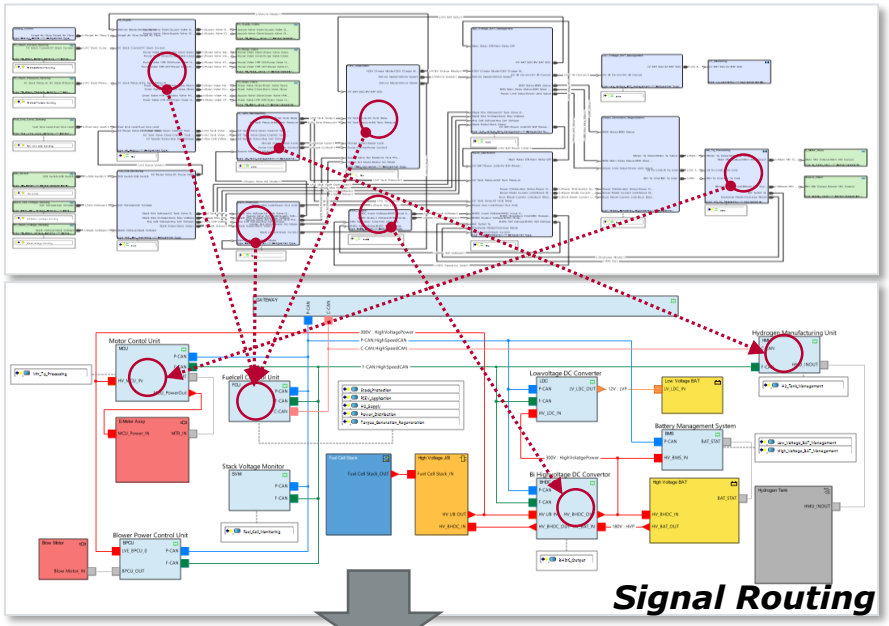
Hardware Architecture and Network Topology Design

- In hardware architecture, communication and hardwire connections between sensors, ECUs, and actuators are described
- Network topology describes high voltage power distribution lines, converters and batteries of E-Mobility system
- The connection between cooperative ECUs are describes on the topology to enable hardware End-to-End design

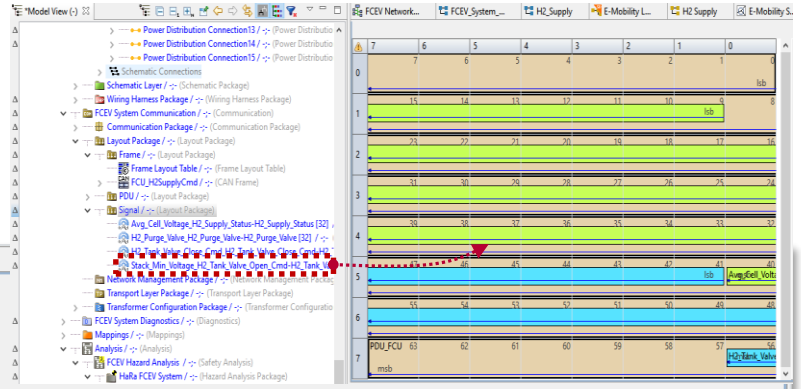


Communication Design

AUTOSAR or DBC Import | SW Design / HW Design | SW/HW Mapping | Data Mapping | Signal Routing | **Communication Design Layout, Routing, Bus Access** | AUTOSAR Export



Signal Assignment to Frame(Message)



*Layout Package::Signal Definition Editor

Mapping State	Count
UnMapped Signals	0
UnAssigned Signals	0
Signals never sent	0
Signals never received	0
Signals total	9

System Signal	Bit Length	Base Type	Base Type Encoding	Computation Method	Unit	Offset	Min	Max	Factor	Raw Initial Value	Physical
FadeInTiming_FadingMode	4	sInt4	Unkonwn (UNKOWN)								
FadeOutTiming_FadingMode	4	sInt4	Unkonwn (UNKOWN)								
Fading_ControlCmdFront	4	sInt4	Unkonwn (UNKOWN)								
Fading_ControlCmdRear	4	sInt4	Unkonwn (UNKOWN)								
FadingState_FrontFadingState	4	sInt4	Unkonwn (UNKOWN)								
FadingState_RearFadingState	4	sInt4	Double (IEEE-DOUBLE)								
OnOffOnDoorOpen_InsideLig...	2	sInt2	Float (IEEE-FLOAT)								
OpenClosed_DoorStatus	1	Bool	Unkonwn (UNKOWN)								
SDS_DD_State	1	Bool	Binary Coded Decimal (BCD)								
			Signed Integer (SIGNED)								

*Layout Package::Signal Editor (Packaging and Routing)

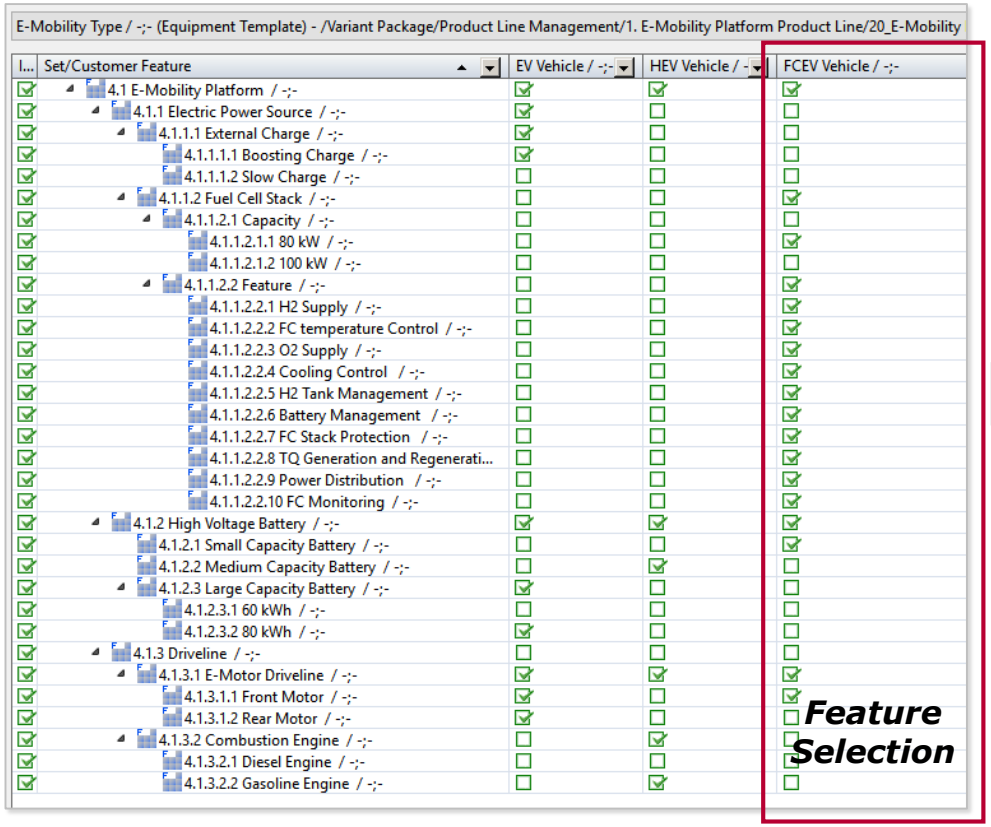
Signals	SPA	PDUs	PFA	Frames	Frame Tx	PDU Tx	SignalTx	Routing of STx
FadeInTiming_FadingMode [4] / -...	[0]			FadingMode / -;0 (CAN Fram	FadingMode	...	FadeInTiming_FadingM	Roof / -;0 (R
					FadingMode	...	FadeInTiming_FadingM	Body / -;0 (R
FadeOutTiming_FadingMode [4] /...	[1]			FadingMode / -;0 (CAN Fram	FadingMode	...	FadeOutTiming_Fadi	Body / -;0 (R
					FadingMode	...	FadeOutTiming_Fadi	Roof / -;0 (R
Fading_ControlCmdFront [4] / -;0 ...	[0]			FadingControl / -;0 (CAN F	FadingControl	...	Fading_ControlCmdFron	Body / -;0 (R
					FadingControl	...	Fading_ControlCmdFron	Roof / -;0 (R

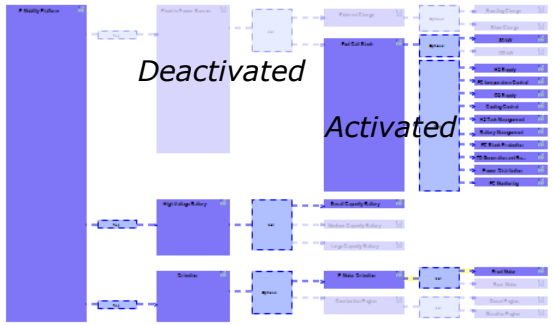
ESP::System Signal Mapping Editor (ECU Based)

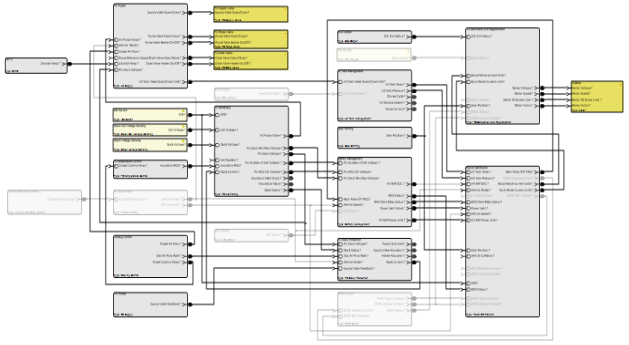
ECU	Mapped SW Component	Port	Data Element	System Signal Mapping	Signal / Signal Group	Transmittable Signal
ESP	FrontLeft	FLPPVehicleSpeed:WheelSpeed	WheelSpeed	COM:System Signal Mapping96	WheelSpeed_FLPPVehicleSpeed	WheelSpeed_FLPPVehicleSpeed
	RearLeft	RLPPVehicleSpeed:WheelSpeed	WheelSpeed	COM:System Signal Mapping97	WheelSpeed_RLPPVehicleSpeed	WheelSpeed_RLPPVehicleSpeed
	FrontRight	FRPPVehicleSpeed:WheelSpeed	WheelSpeed	COM:System Signal Mapping95	WheelSpeed_FRPPVehicleSpeed	WheelSpeed_FRPPVehicleSpeed
	RearRight	RFPPVehicleSpeed:WheelSpeed	WheelSpeed	COM:System Signal Mapping98	WheelSpeed_RRPPVehicleSpeed	WheelSpeed_RRPPVehicleSpeed

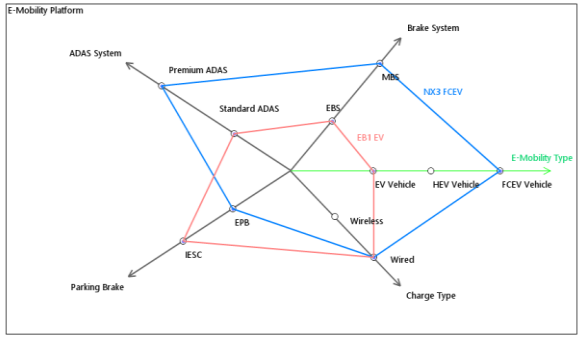
Architecture Variant Design

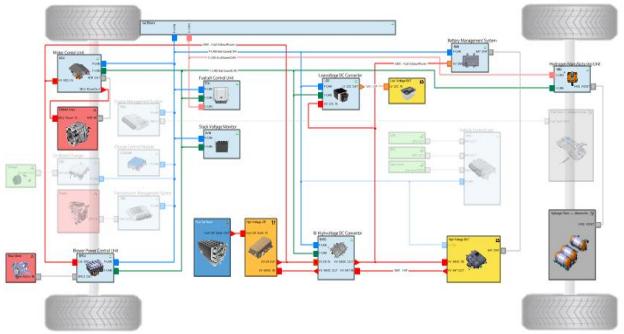
- Customer feature selection enables deployment from E-Mobility Platform to each EV, HEV, and FCEV System Product Line
- Corresponding artifacts mapped to the selected feature can be activated or deactivated through variant activation
- This can be applied not only to the system but also to the entire vehicle, and the engineer in charge of each vehicle system can continue the detailed design within the deployed product line





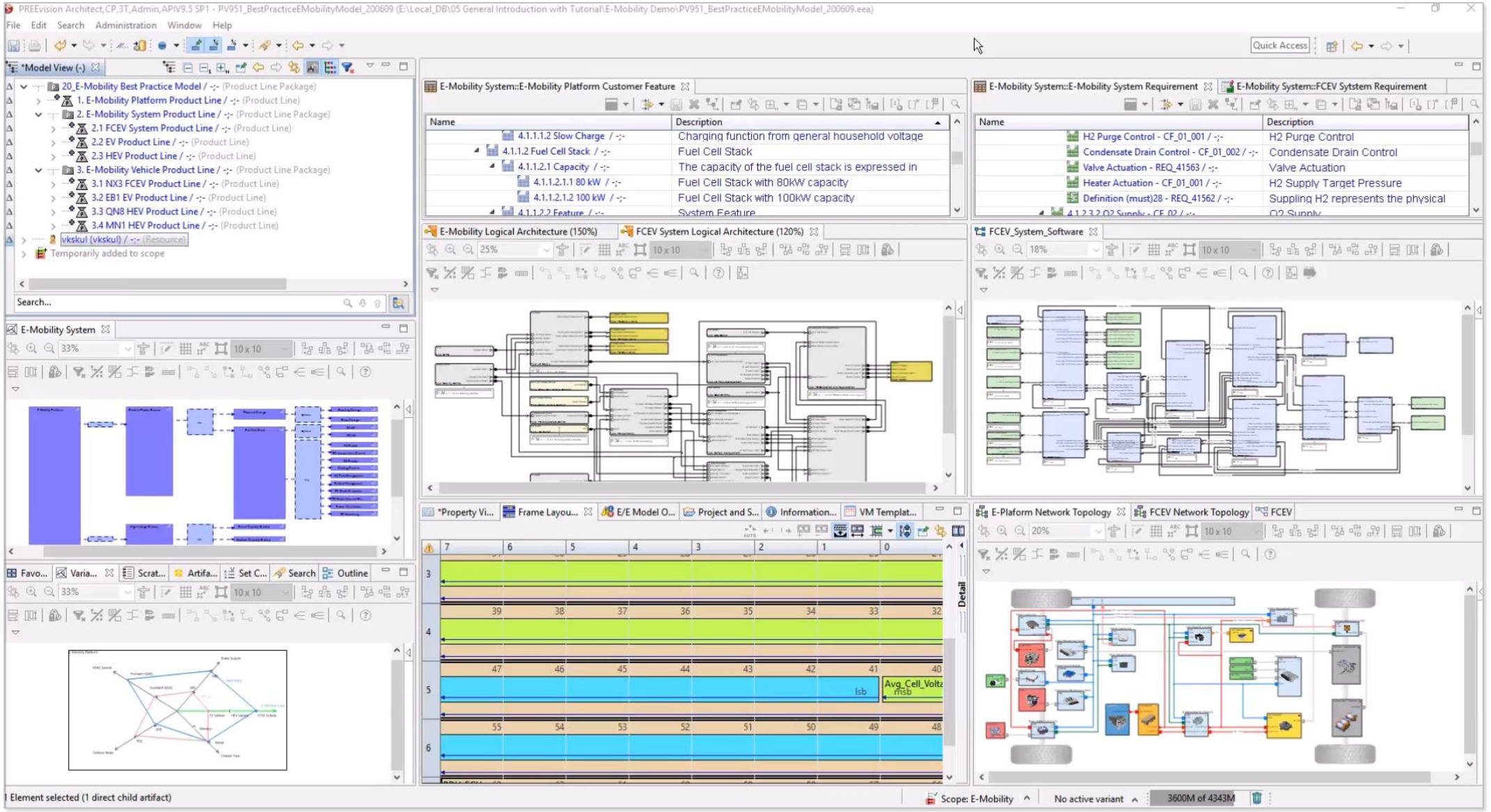






Activate and deactivate artifacts

PREEvision Tool Demo

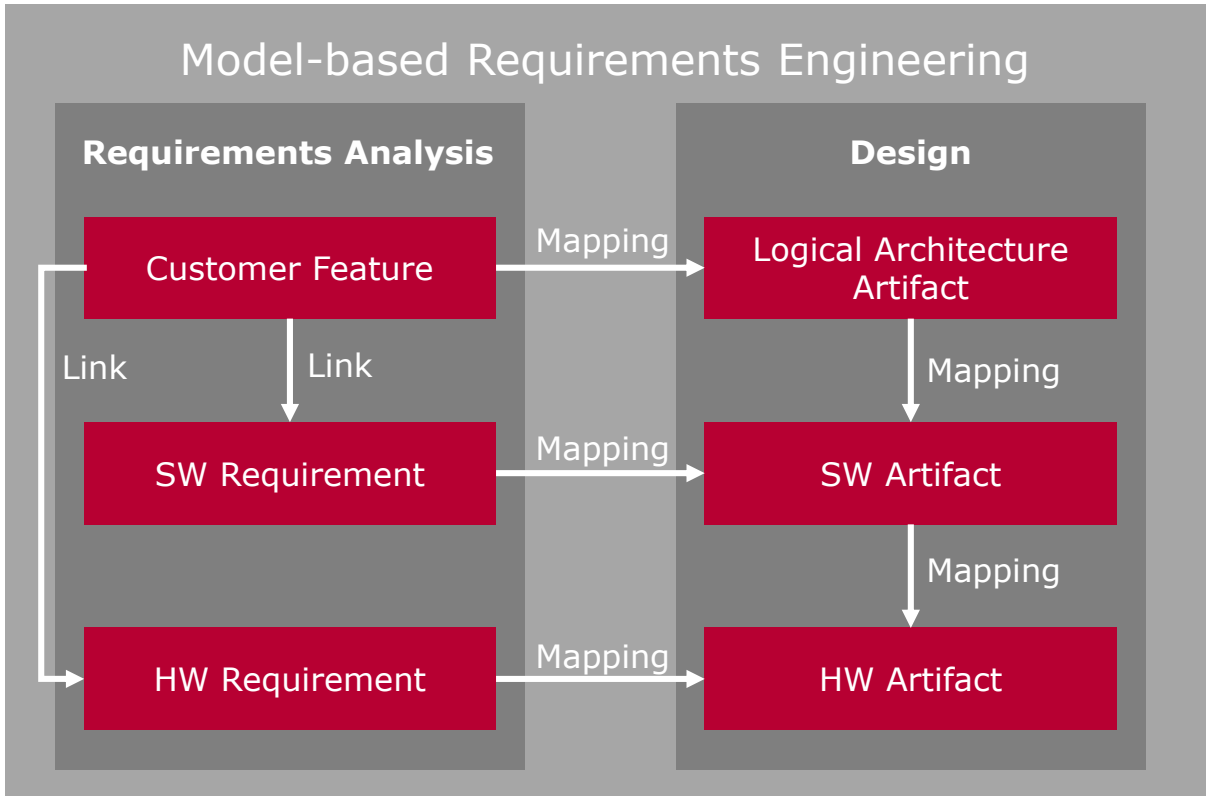


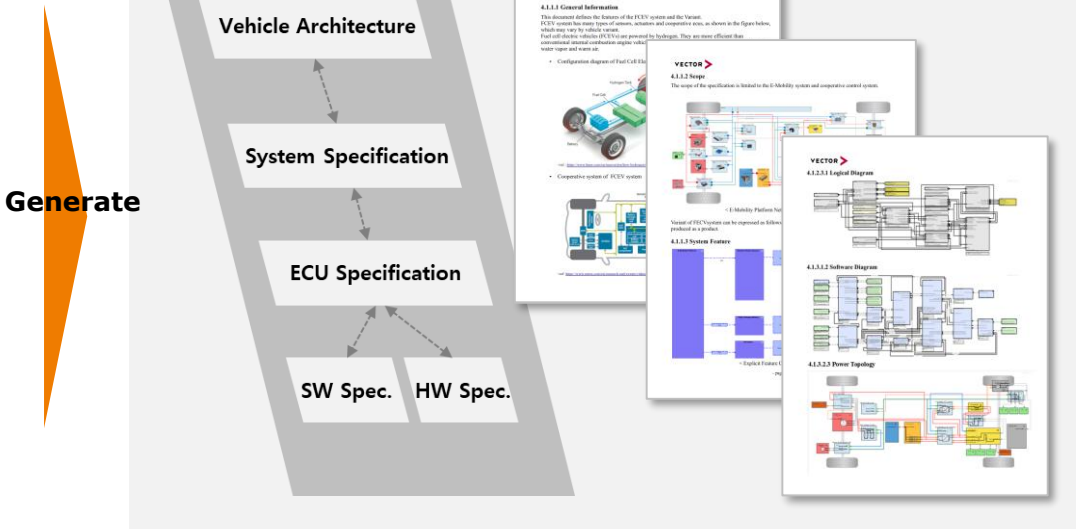
Agenda

- Overview
- PREEvision Introduction
- Model Based System Engineering
- Product Line Management with Reuse Concept
- E-Mobility Architecture Design with Traceability and Consistency
- **Report Generation**
- Summary

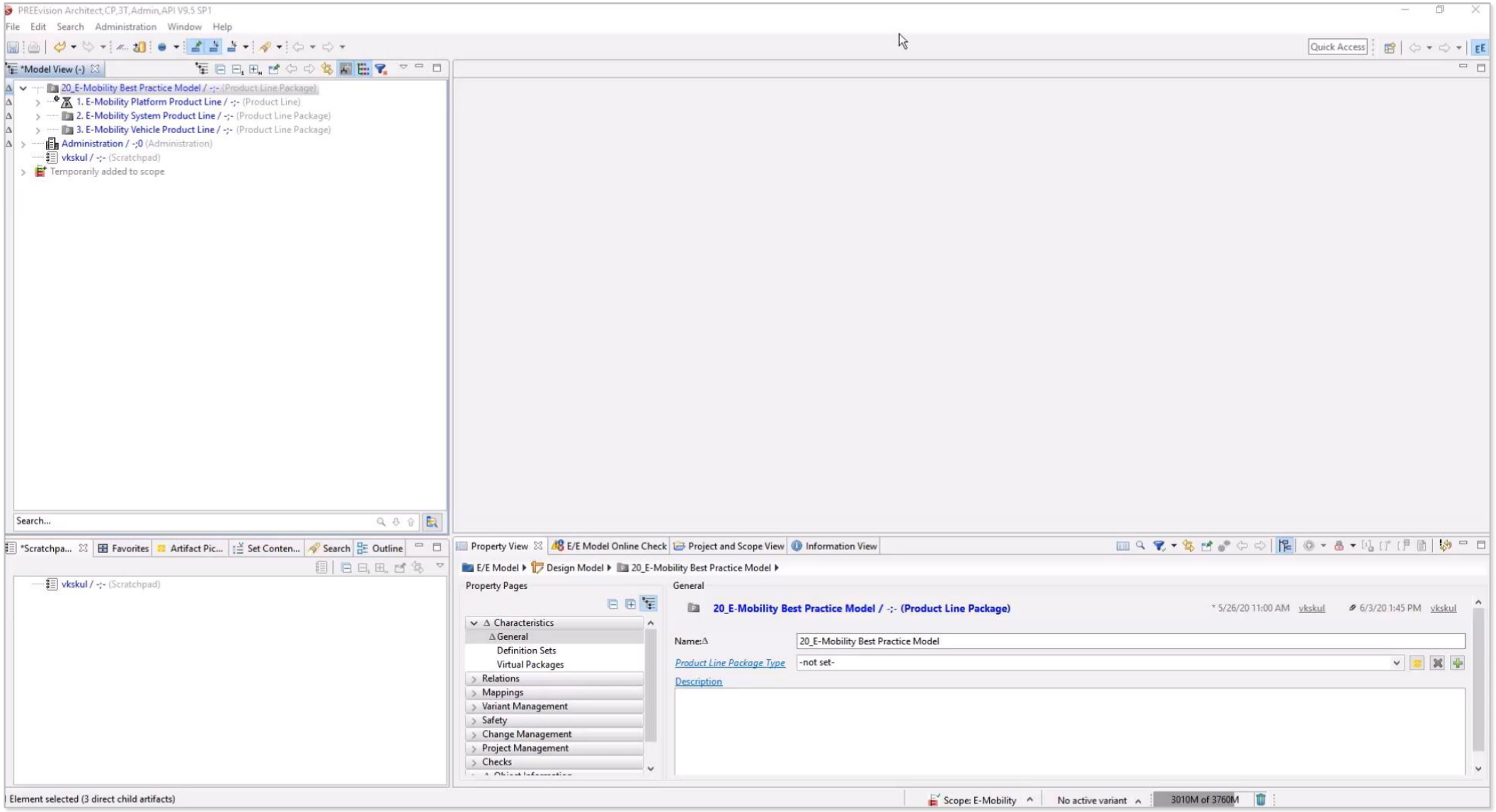
Requirement Specification

- PREEvision supports an integrated requirements-engineering and management
- This allows a bidirectional traceability starting from requirements, going over design artifacts and up to test data
- Users can apply the company's own document template





Video Tutorial



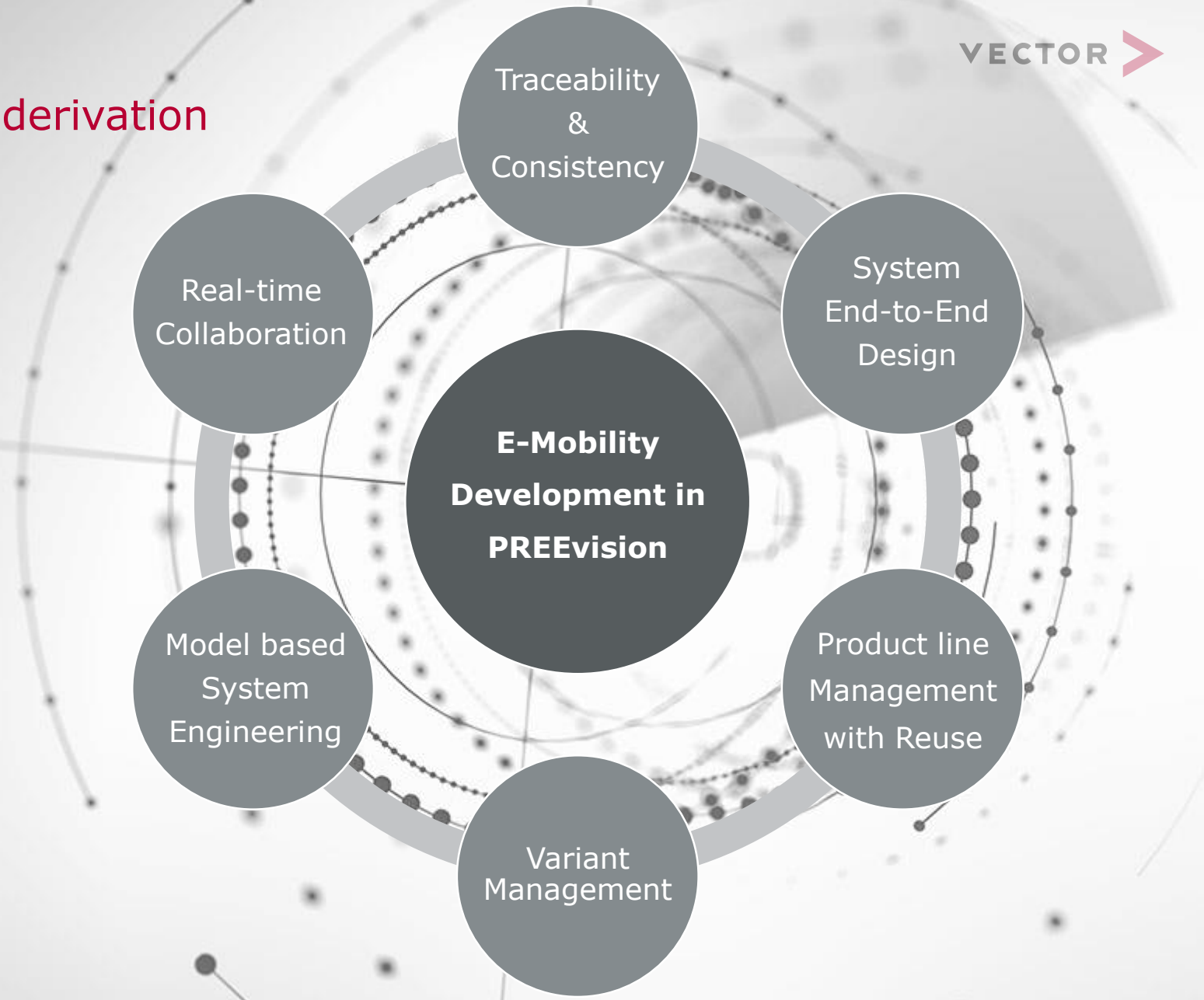
Agenda

- Overview
- PREEvision Introduction
- Model Based System Engineering
- Product Line Management with Reuse Concept
- E-Mobility Architecture Design with Traceability and Consistency
- Report Generation
- **Summary**

Summary

Efficient and reliable product derivation

- Using the model-based design tool, it is possible to develop the E-Mobility system from requirements to test
- The product lineup can be extended in an efficient way by reusing the platform
- It is possible to collaborate in real time with the developer of the cooperative control system based on the database server
- Change management and version control for all data sets can be performed using assets
- Since full traceability and consistency can be secured, a reliable and robust system can be designed



PREEvision – This is Model-Based E/E Engineering

www.vector.com/preevision

Author:
Kim, Eui-yeul
Vector Korea