



AUTOSAR Ethernet Design and SoA

PREEvision Webinar, 2020

Agenda

- ▶ **PREEvision at a glance**

 - Service Oriented Architectures

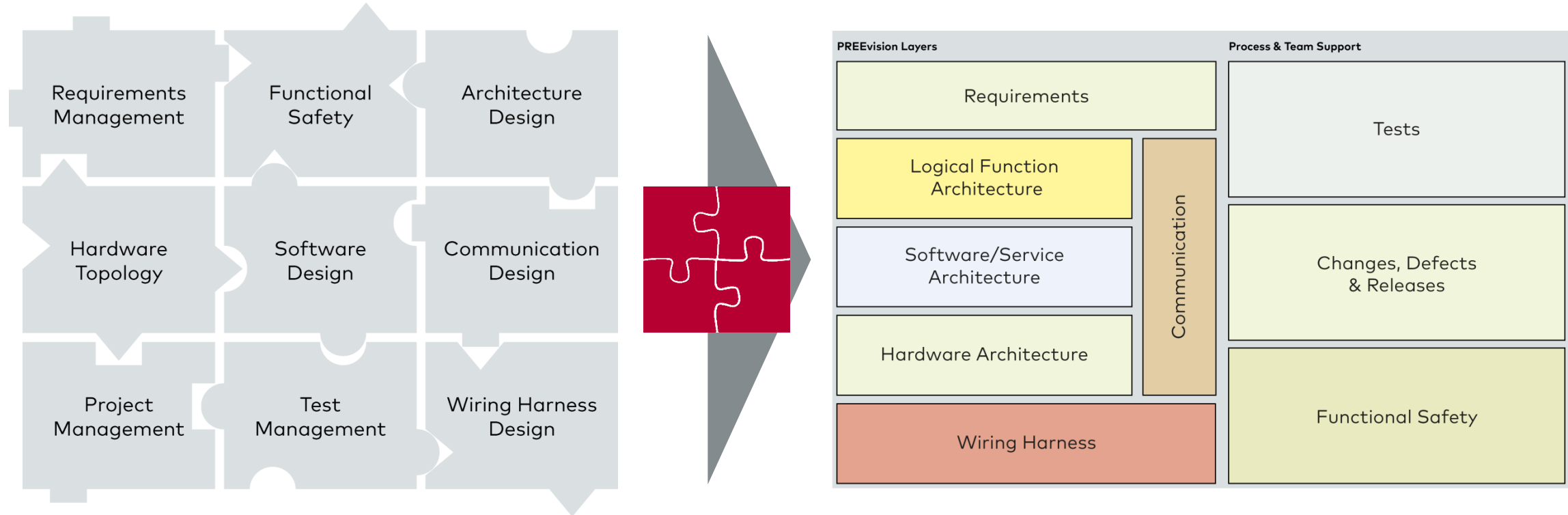
 - Tool Demo

 - What's next: SOA & Ethernet in PREEvision 9.5

 - Summary



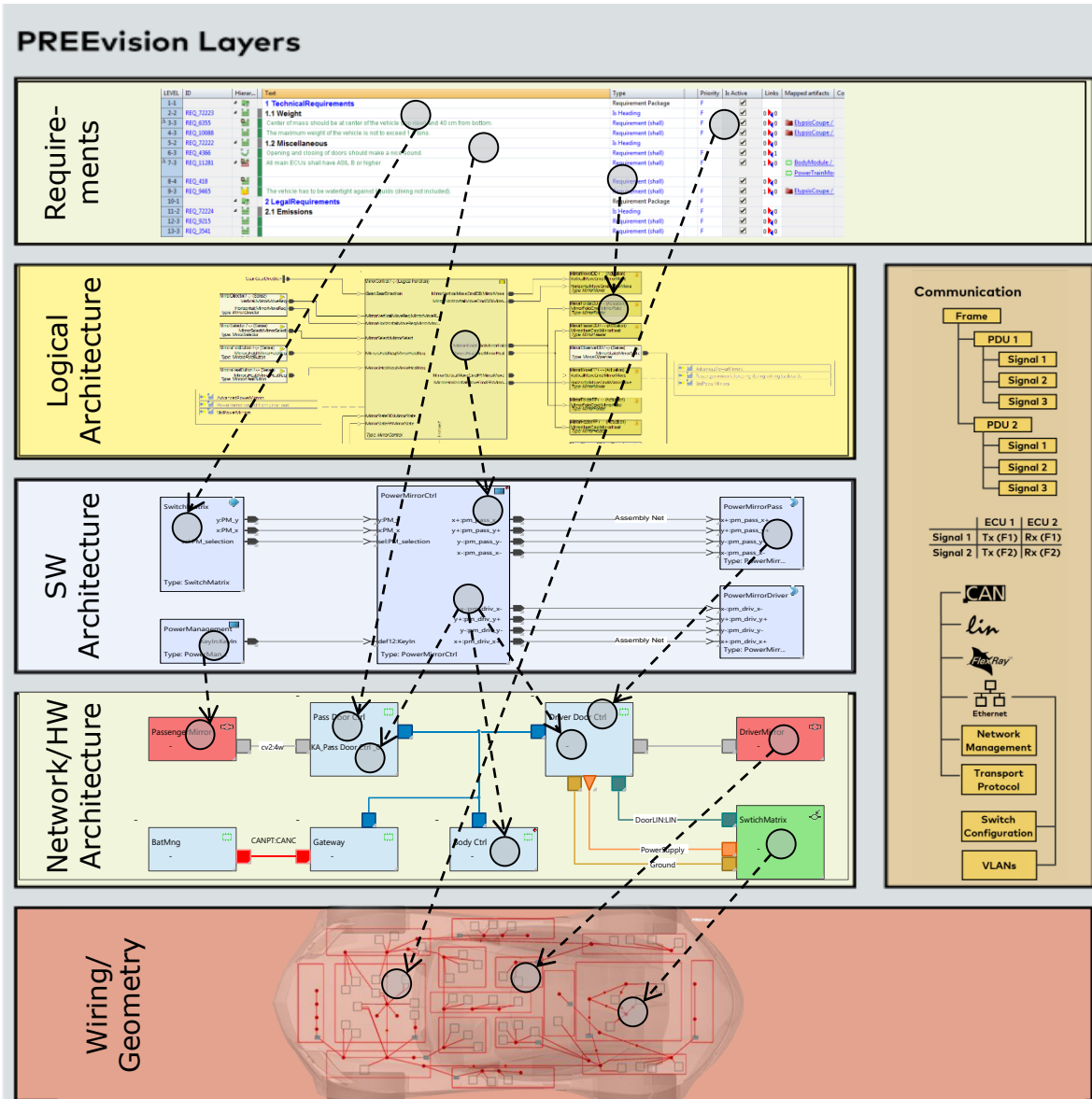
From a Fragmented Tool Landscape to an Integrated Solution



- ▶ **Tool Landscape Approach:**
Multiple Tools and Interfaces
 - > “Best in Class”
 - > Various, not connected data silos
 - > File based collaboration

- ▶ **Integrated Tool Approach:**
PREEvision E/E Engineering Environment
 - > Integrated and Model Based Design
 - > Central backbone as “Single Point of Truth”
 - > Data based collaboration

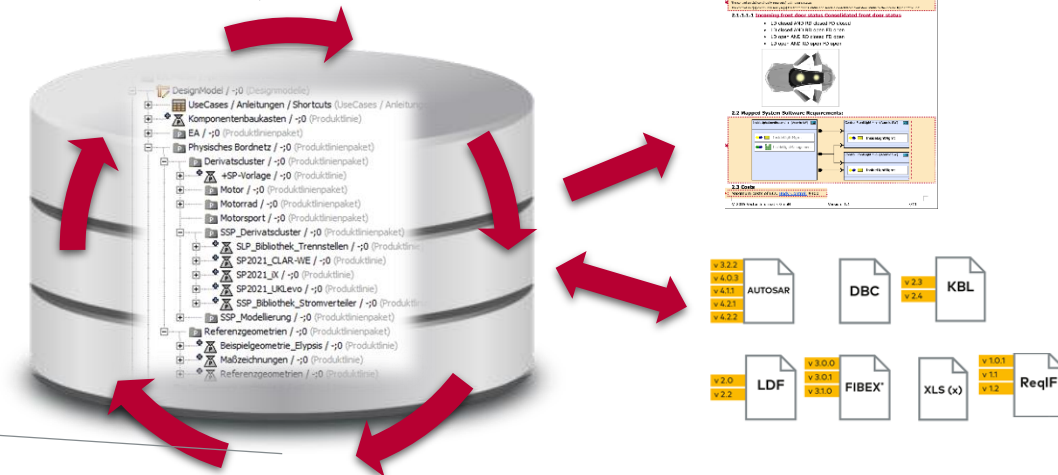
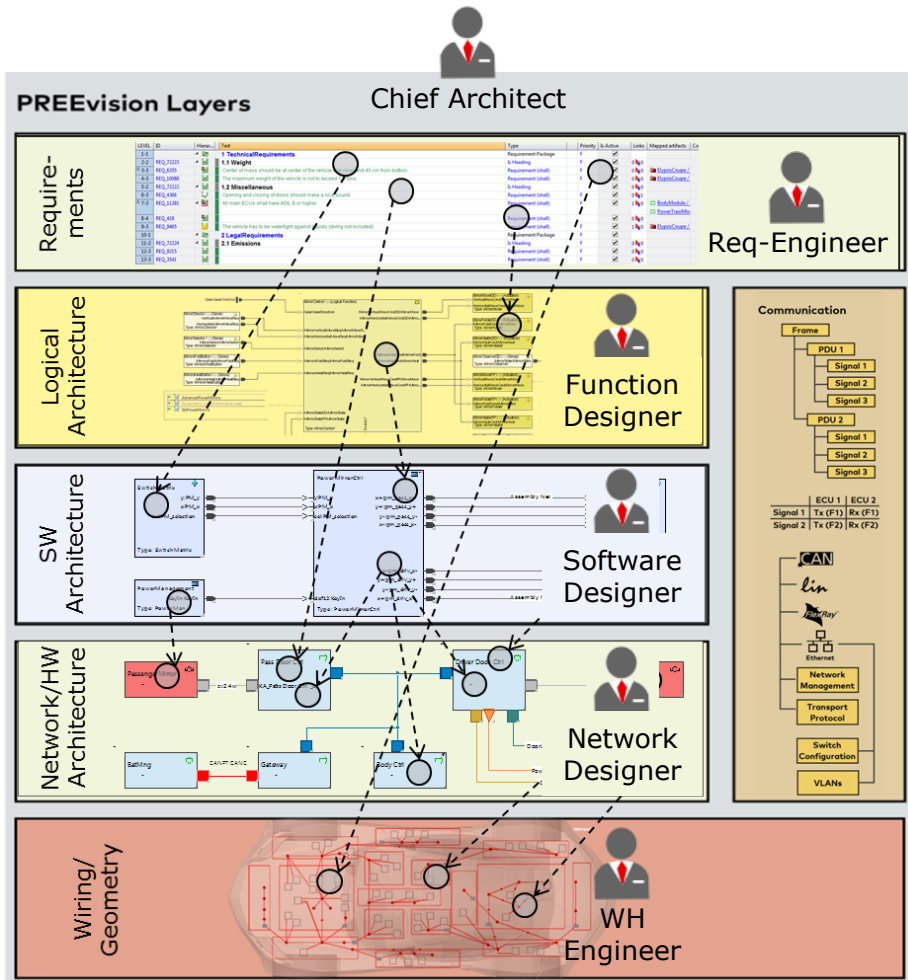
Model Based Development



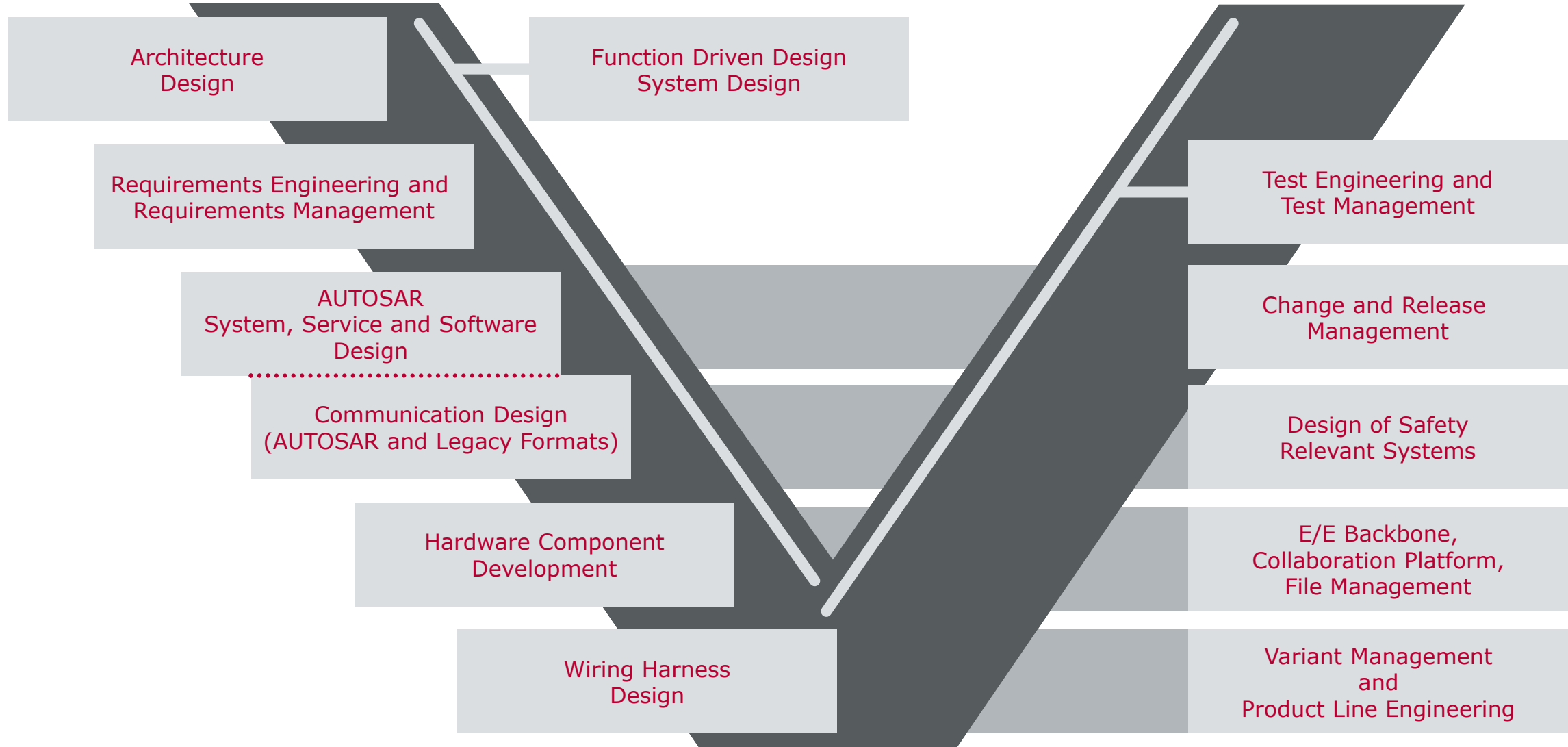
- ▶ Domain specific language and data model
- ▶ Single source model across all Automotive E/E development Use Cases:
 - The model is the **Single Point of Truth**
 - Mappings ensure full **traceability**
 - The model can be **analyzed by metrics**
- ▶ All data objects have a semantic meaning
 - Base for various **model checks** e.g. for **Correctness, Completeness, Consistency**
- ▶ **Automated algorithms** for synthetization, scheduling, signal routing, etc.

Model Based Development in large groups

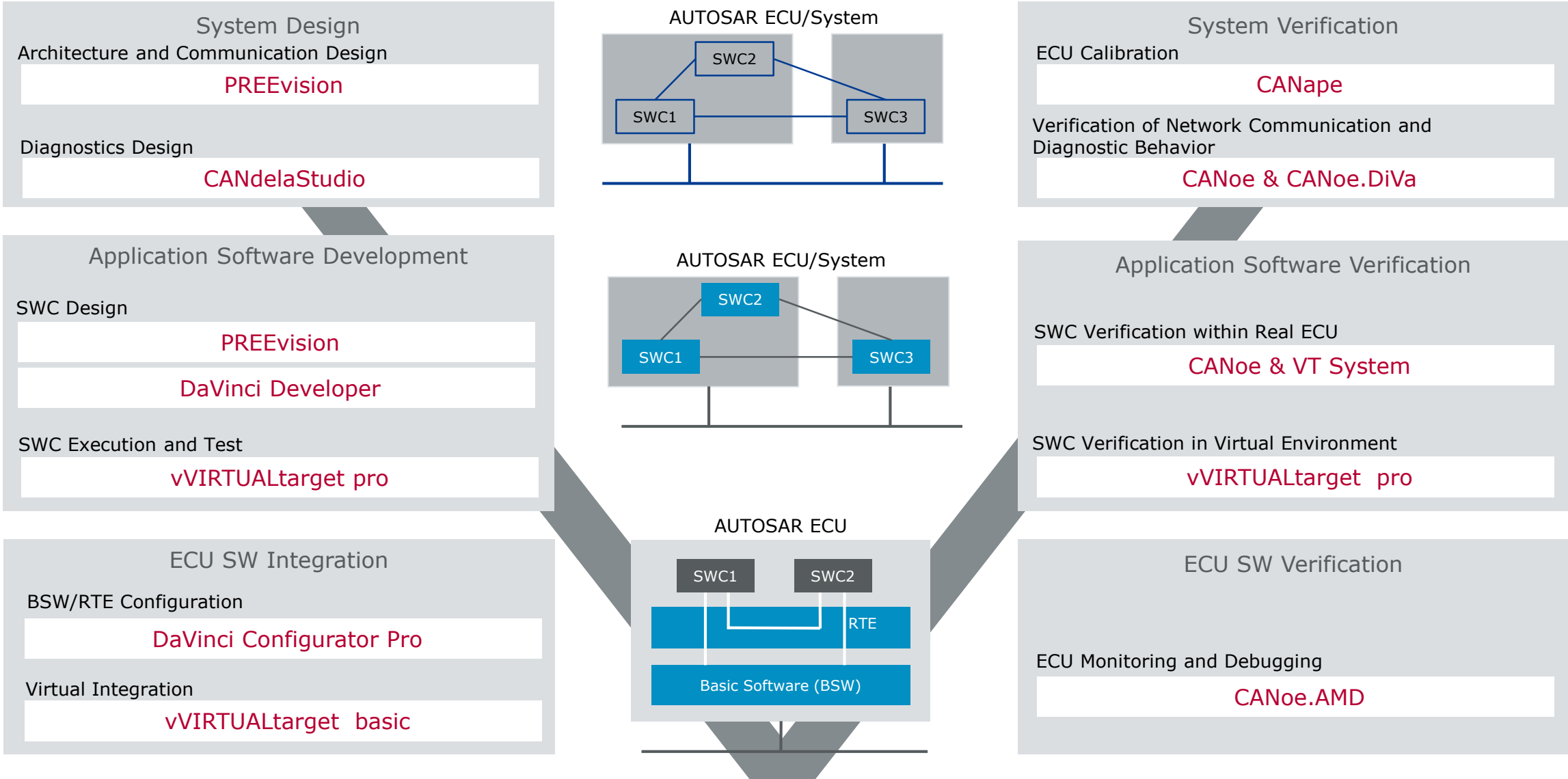
- ▶ Multi User – Single Source
→ Multiple users **work collaboratively**, supported by **Lock&Commit, Life Cycles, Rights&Role Management and Ticket System**
- ▶ Multiple projects are handled in one system, supported by **Productline Management, Variant Management, etc.**
- ▶ **Import and export** with industry standard exchange formats (AUTOSAR, DBC, LDF, FIBEX, RIF, ReqIF, KBL, CSV)
- ▶ The model is the **Single Point of Truth**



Supported Use Cases

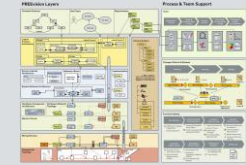


The complete Vector AUTOSAR Tool Chain

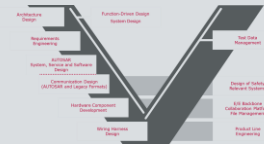


Bottom Line – The PREEvision Assets

Rich model-based **Automotive Data Model**.



Professional **Engineering Functionality** to work with this data model (MBSE).



Collaboration of many users at many sites on **one** Single Point of Truth.



Product Line Engineering to manage the complexity of many variants.



Customizable → Tailoring to customers process by **configuration**.



Agenda

PREEvision at a glance

▶ **Service Oriented Architectures**

Tool Demo

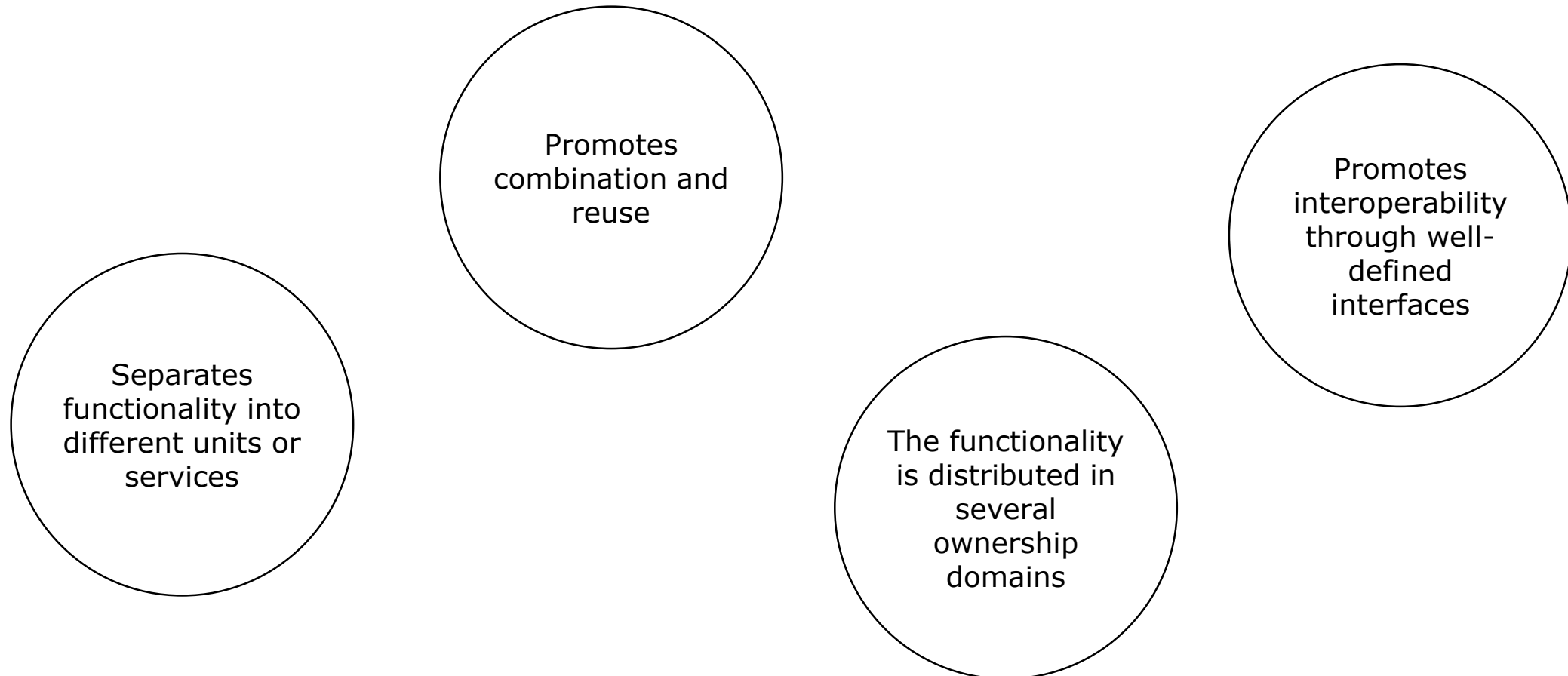
What's next: SOA & Ethernet in PREEvision 9.5

Summary



Service Oriented Architecture

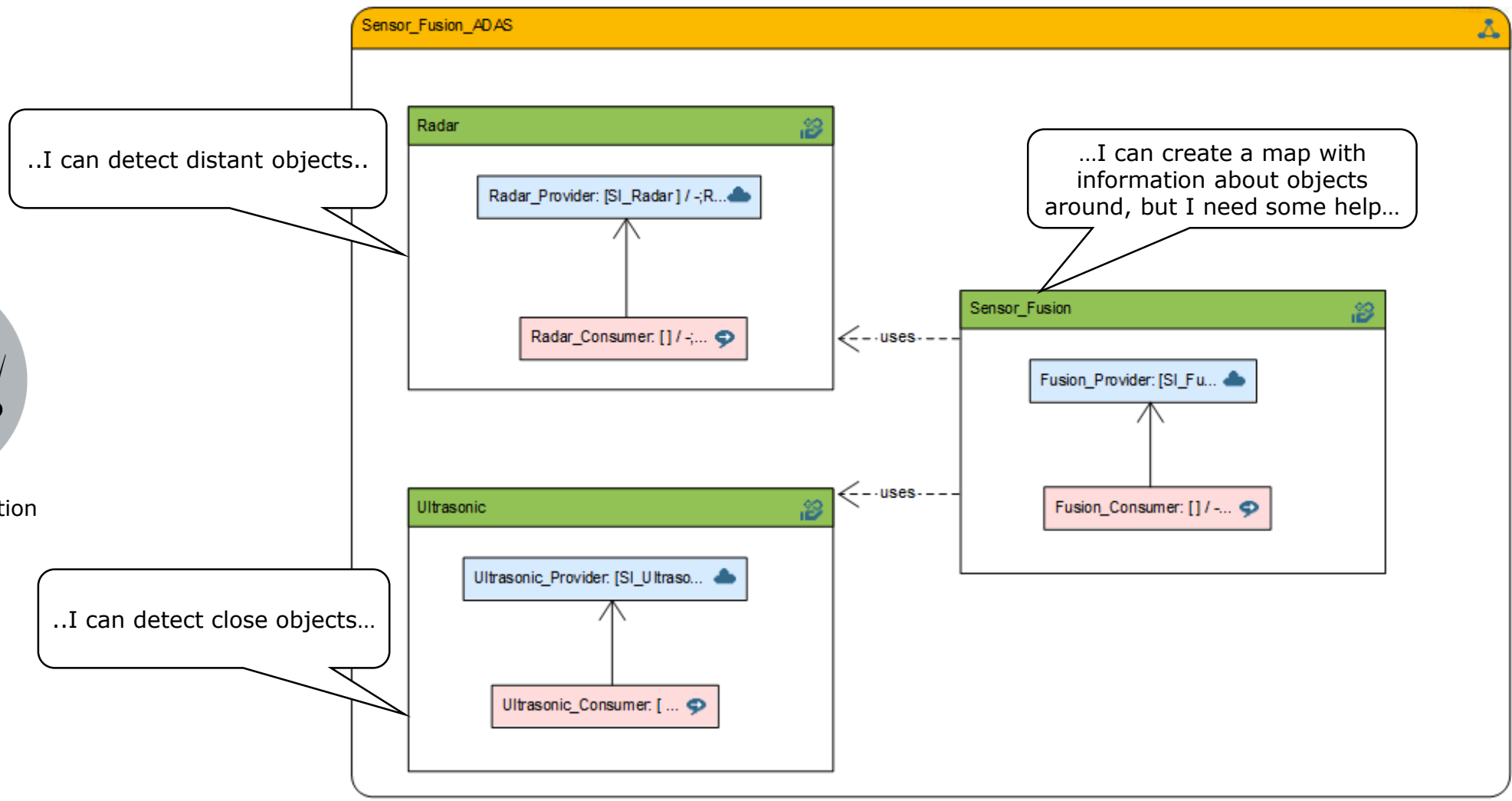
- ▶ Service Oriented Architecture is a way of designing software where the participating components provide and consume services through a defined protocol over a network



Example of a Service Oriented Architecture

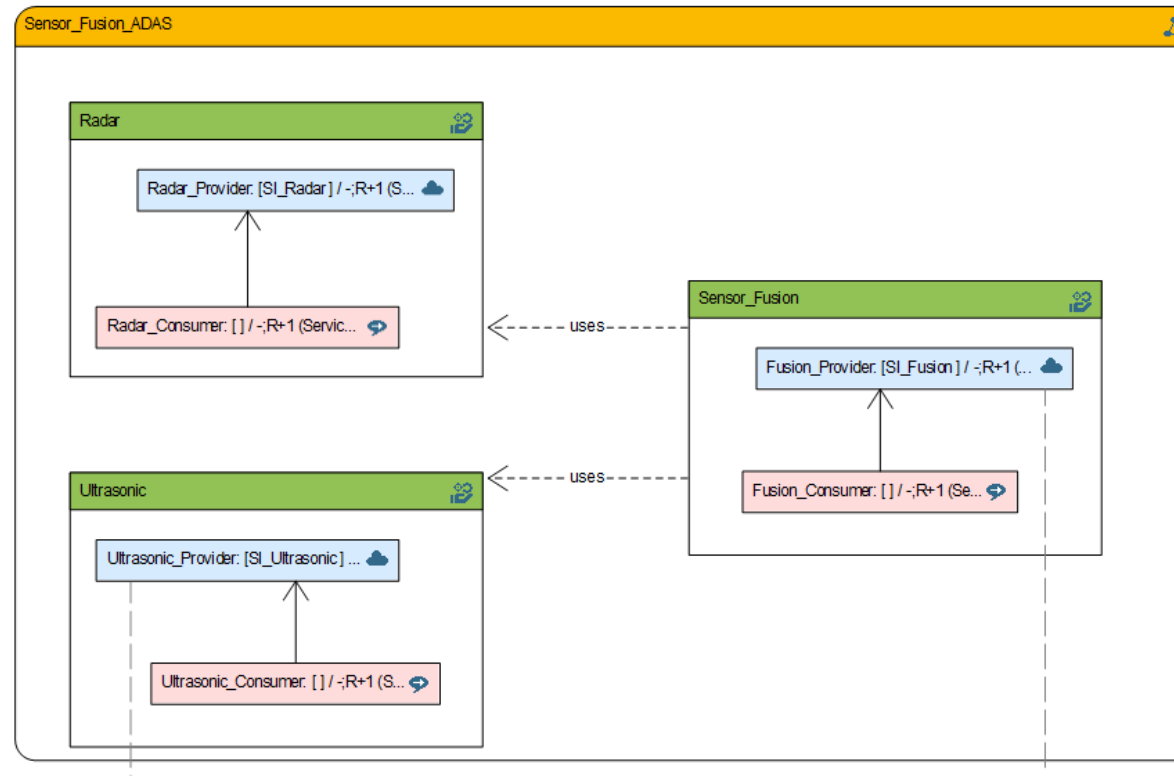


Service orchestration

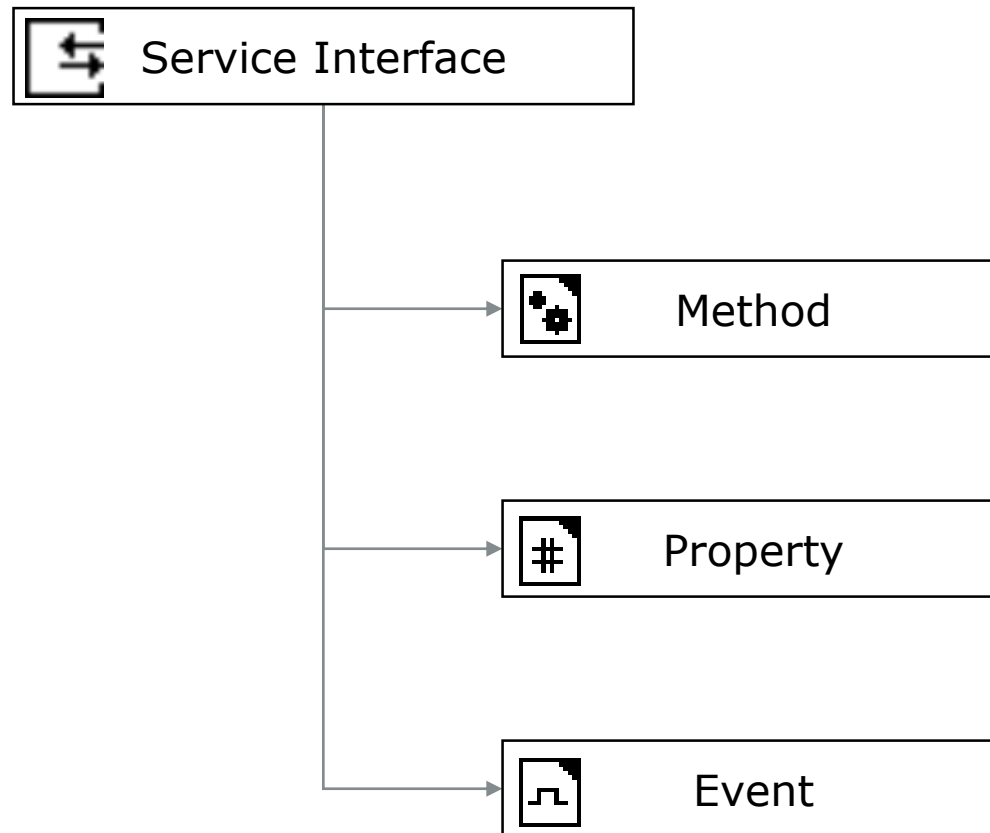


Service Interface (1/2)

- ▶ Service interface is the mean to access to the functionality of a service



Service Interface (2/2)

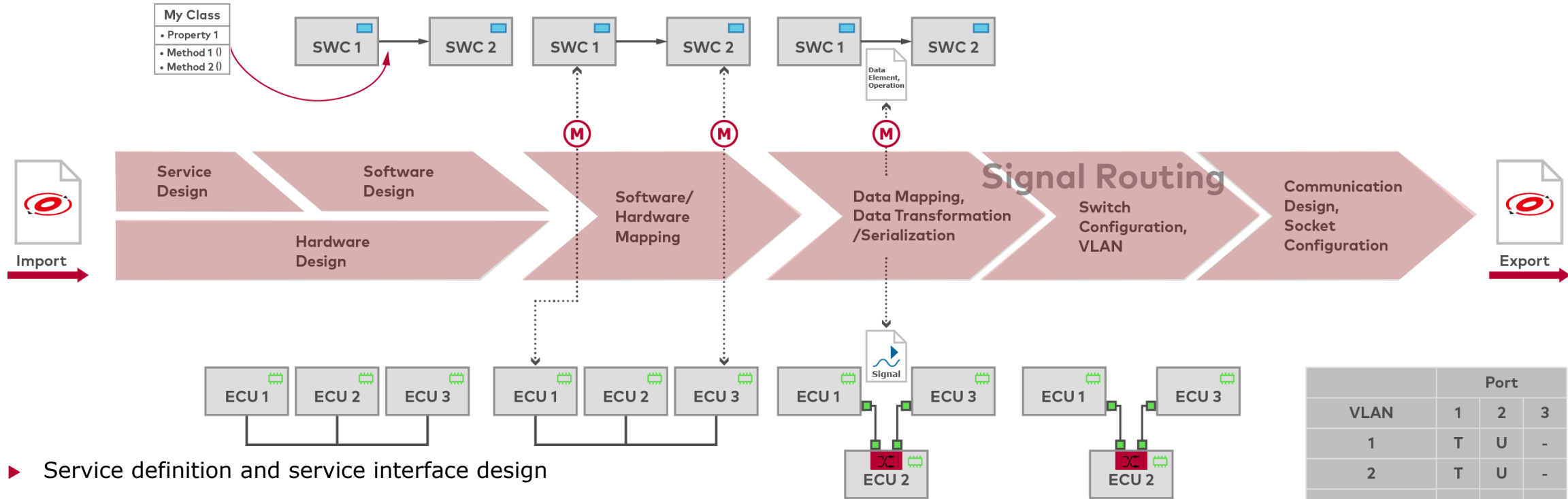


A **method** represents a function that is executed by a provider on request of one or more consumer(s)

A **property** (field, attribute) represents a piece of data hosted by a provider that exposes to one or more consumer(s) a get and/or a set method. Consumers can optionally receive notifications of changes of the field's value.

An **event** represents an update to a piece of data. The provider decides when to send this update and the occurrence of it is transmitted from a to one or more consumer(s).

Service Oriented System Design Workflow

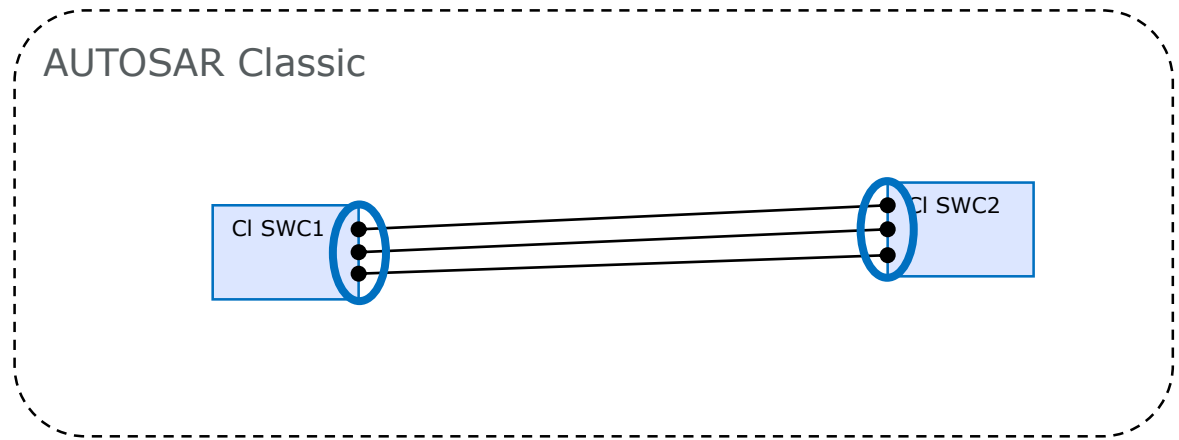
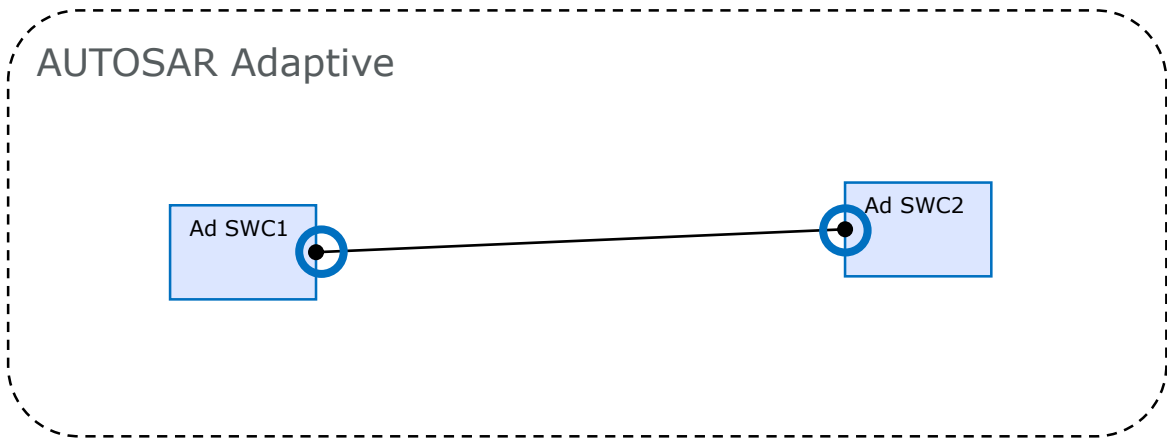
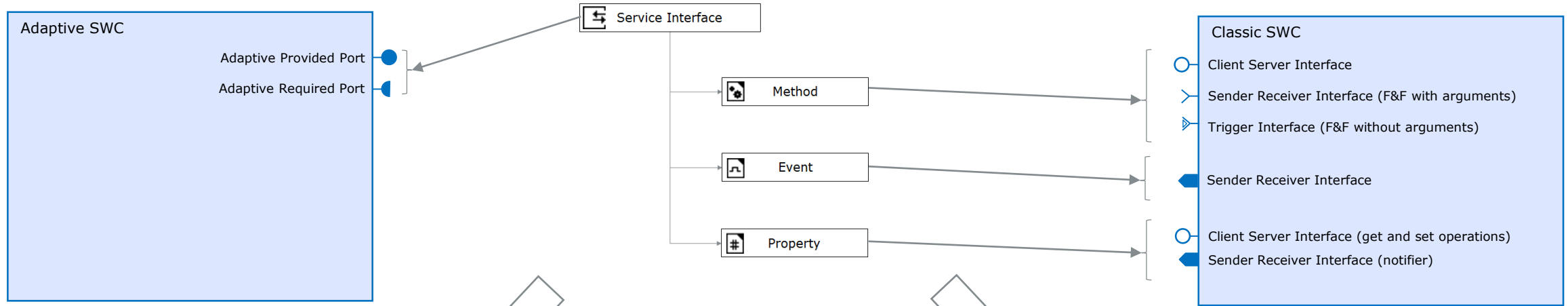


- ▶ Service definition and service interface design
- ▶ Virtual function bus (derived from service definition)
- ▶ Switched topology definition
- ▶ Deployment of service provider and service consumer
- ▶ Communication design
 - ▶ VLAN configuration
 - ▶ Switch configuration
 - ▶ Socket communication

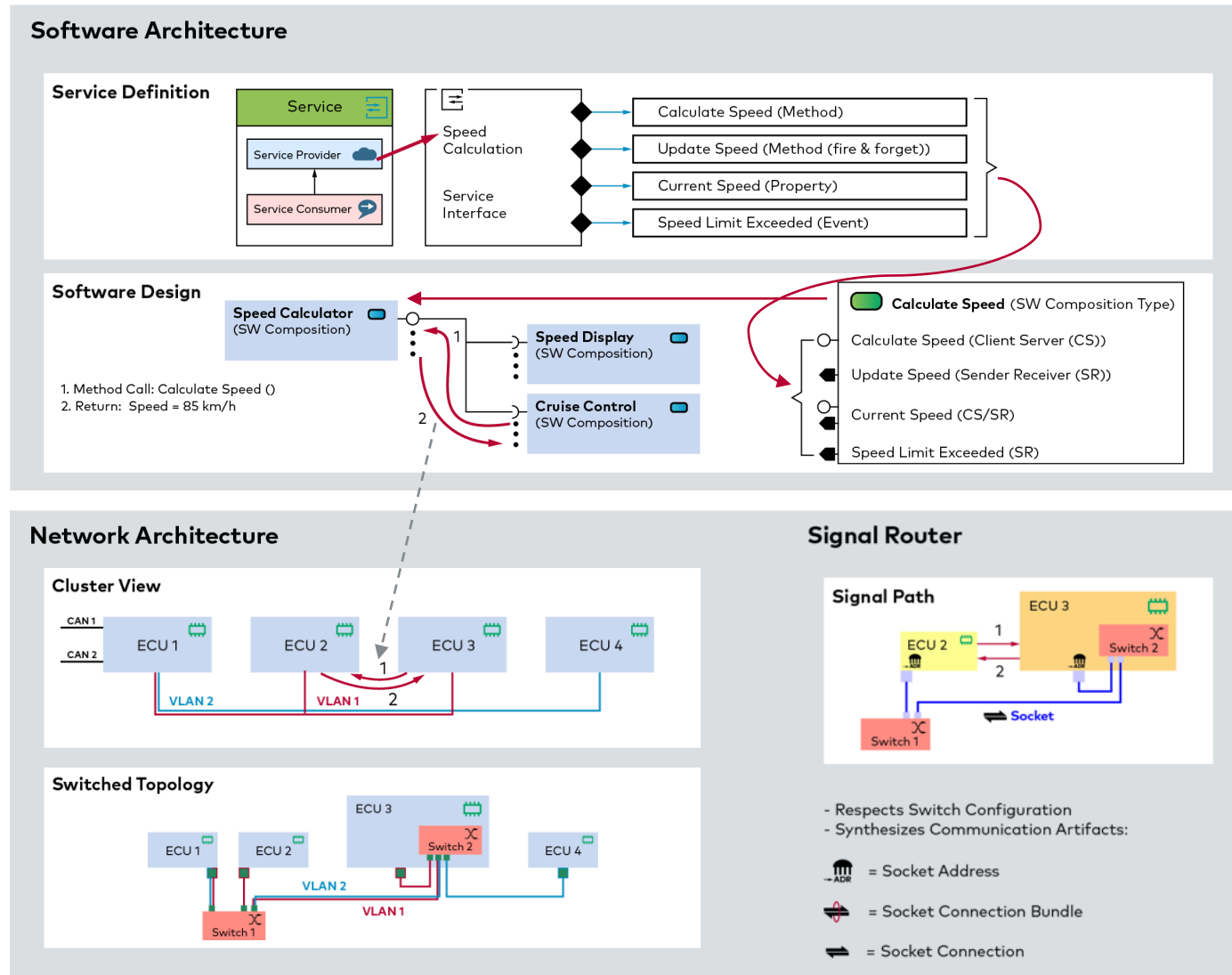
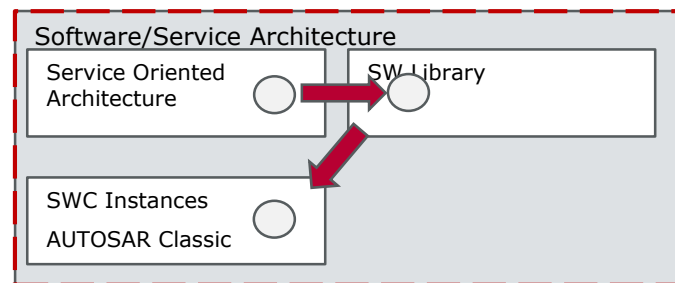
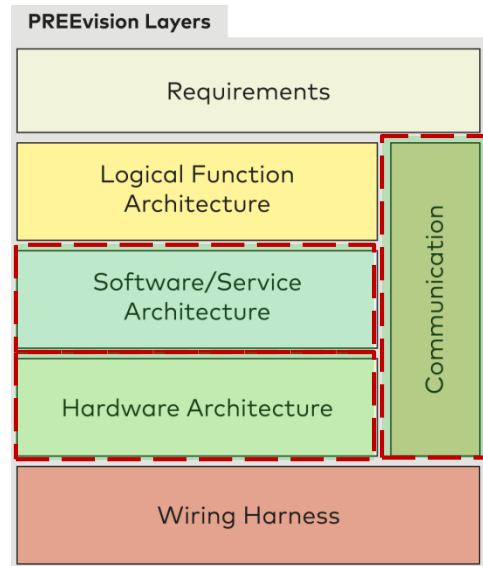
VLAN	Port		
	1	2	3
1	T	U	-
2	T	U	-
...			
4094			

T = send tagged, U = send untagged

Software synthesis



SOA Design Workflow (AUTOSAR Classic)



Agenda

PREEvision at a glance

Service Oriented Architectures

▶ **Tool Demo**

What's next: SOA & Ethernet in PREEvision 9.5

Summary



SOA and Ethernet in PREEvision - Demo

▶ **Workflow:**

- > Small SoA Design, Implementation in SOME/IP
- > Technology mapping to AUTOSAR Classic
- > Hardware Architecture
- > Software Design – Service instantiation
- > Signal Routing
- > Ethernet setup, Transformer assignment, Service Discovery

▶ **Based on PREEvision 9.5**

- > Perspective: “Service and Ethernet Design”
- > SoA & Ethernet Explorer

▶ **Content:**

- > Use Case: Sensor Fusion (Radar + Ultrasonic) and usage in ADAS environment
- > Fusion of Sensor Data
- > Ethernet network with three ECU’s (Two provider, one consumers)
- > Three service instances

Agenda

PREEvision at a glance

Service Oriented Architectures

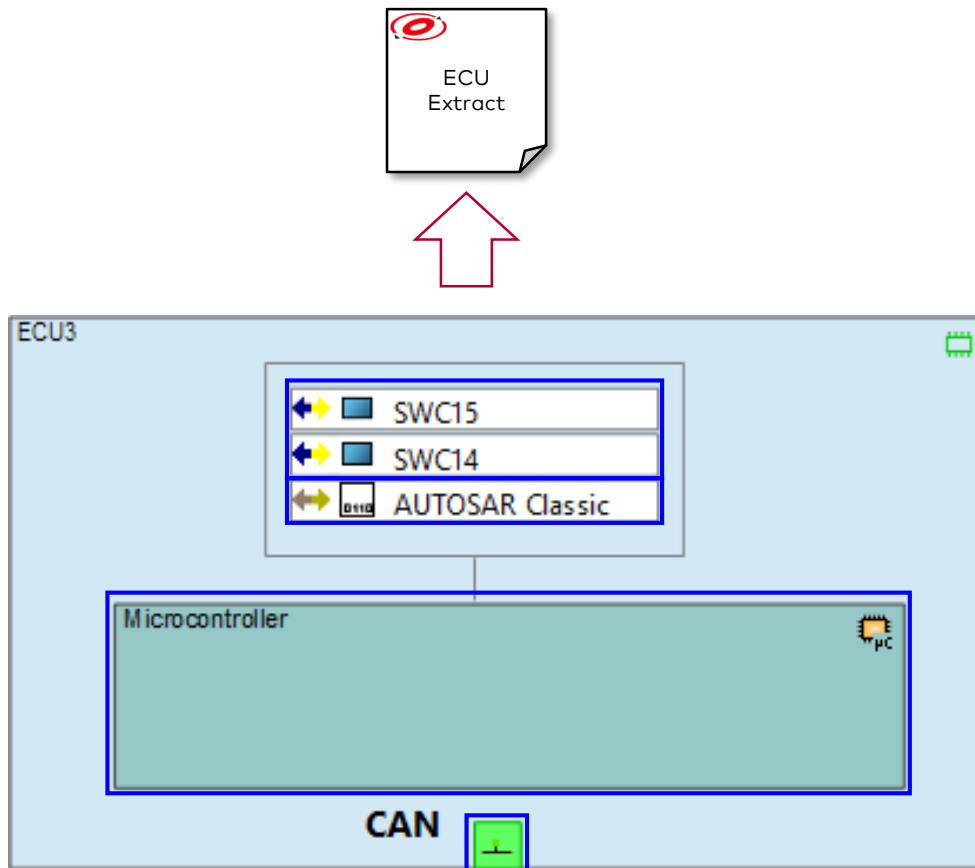
Tool Demo

▶ **What's next: SOA & Ethernet in PREEvision 9.5**

Summary



Common ECU



- ▶ Example: common ECU with one microcontrollers (μ C)
- ▶ Application Software runs on the ECU
- ▶ Basic Software runs on the ECU
- ▶ Some bus connector (e.g. CAN)
- ▶ For the ECU can be exported the an ECU Extract

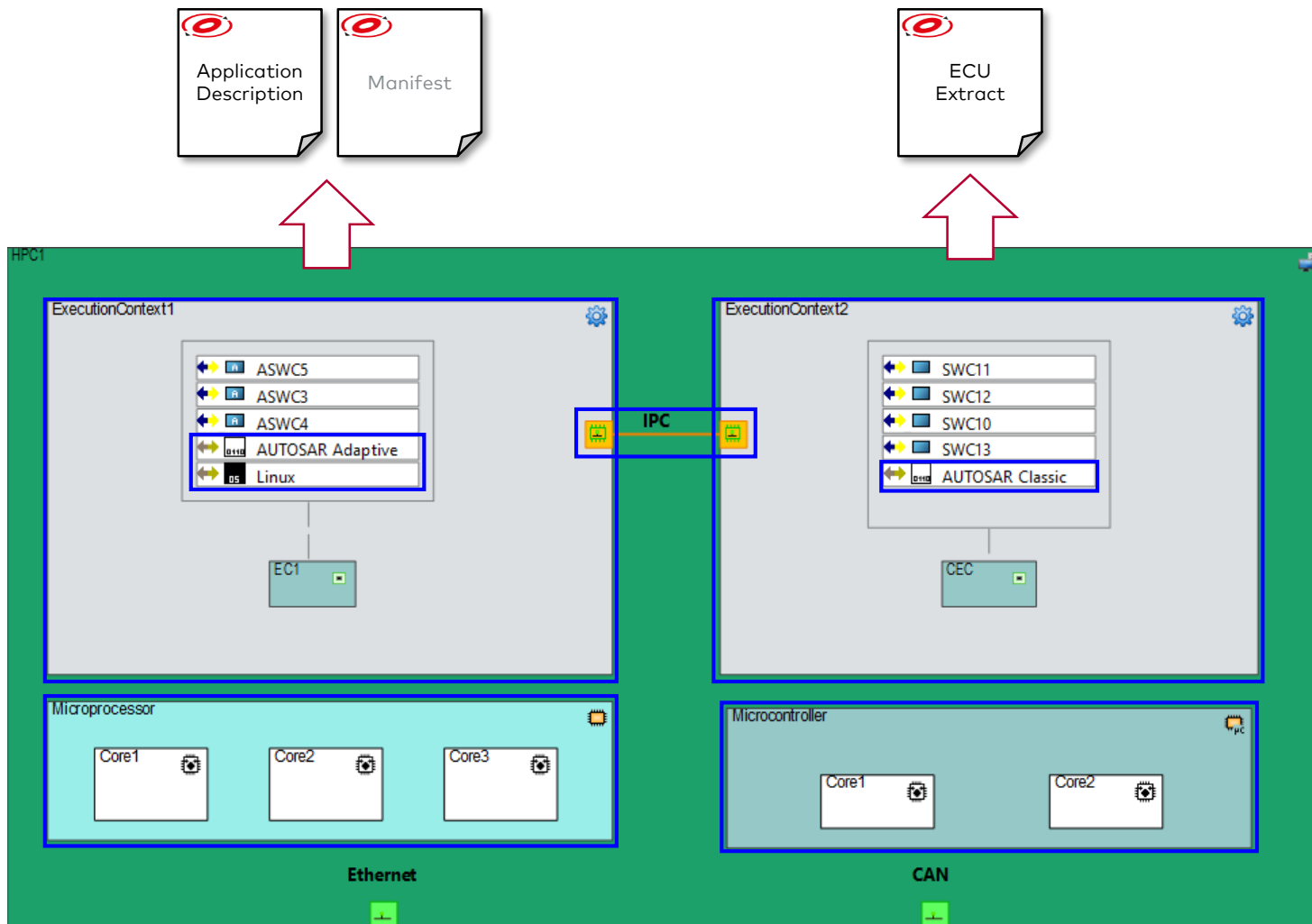
Trying a definition

*"High Performance Computing most generally refers to the practice of **aggregating** computing power in a way that delivers **much higher performance** than one could get out of a typical ... computer ... in order to solve large problems in science, engineering, or business" (1)*

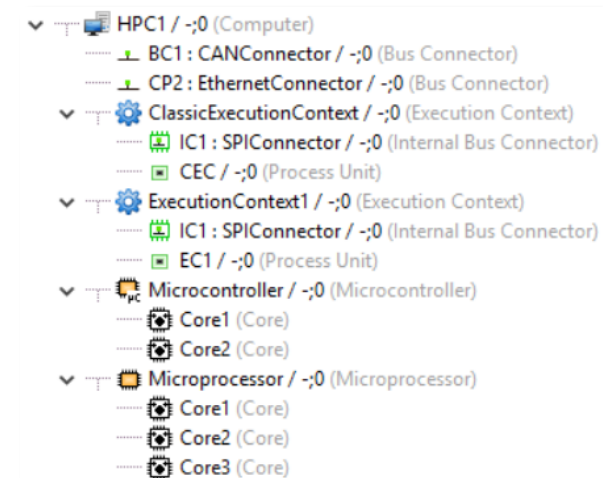
Modeling aspects of High Performance Computers

- ▶ Hardware aspects
 - ▶ Microprocessors, Microcontrollers
 - ▶ Cores
 - ▶ Hypervisor (type 1)
- ▶ Platform aspects
 - ▶ Execution context
 - ▶ Hypervisor (type 2)
 - ▶ Basic software
- ▶ Communication aspects
 - ▶ Internal Communication
 - > Internal bus (**S**erial **P**eripheral **I**nterface/**I**nter-**P**rocess **C**ommunication/Ethernet)
 - > Internal bus connection
 - ▶ External Communication
 - > External bus (CAN, LIN, FlexRay, Ethernet)
- ▶ Resource usage aspects
 - ▶ Core assignment
 - ▶ Application partition

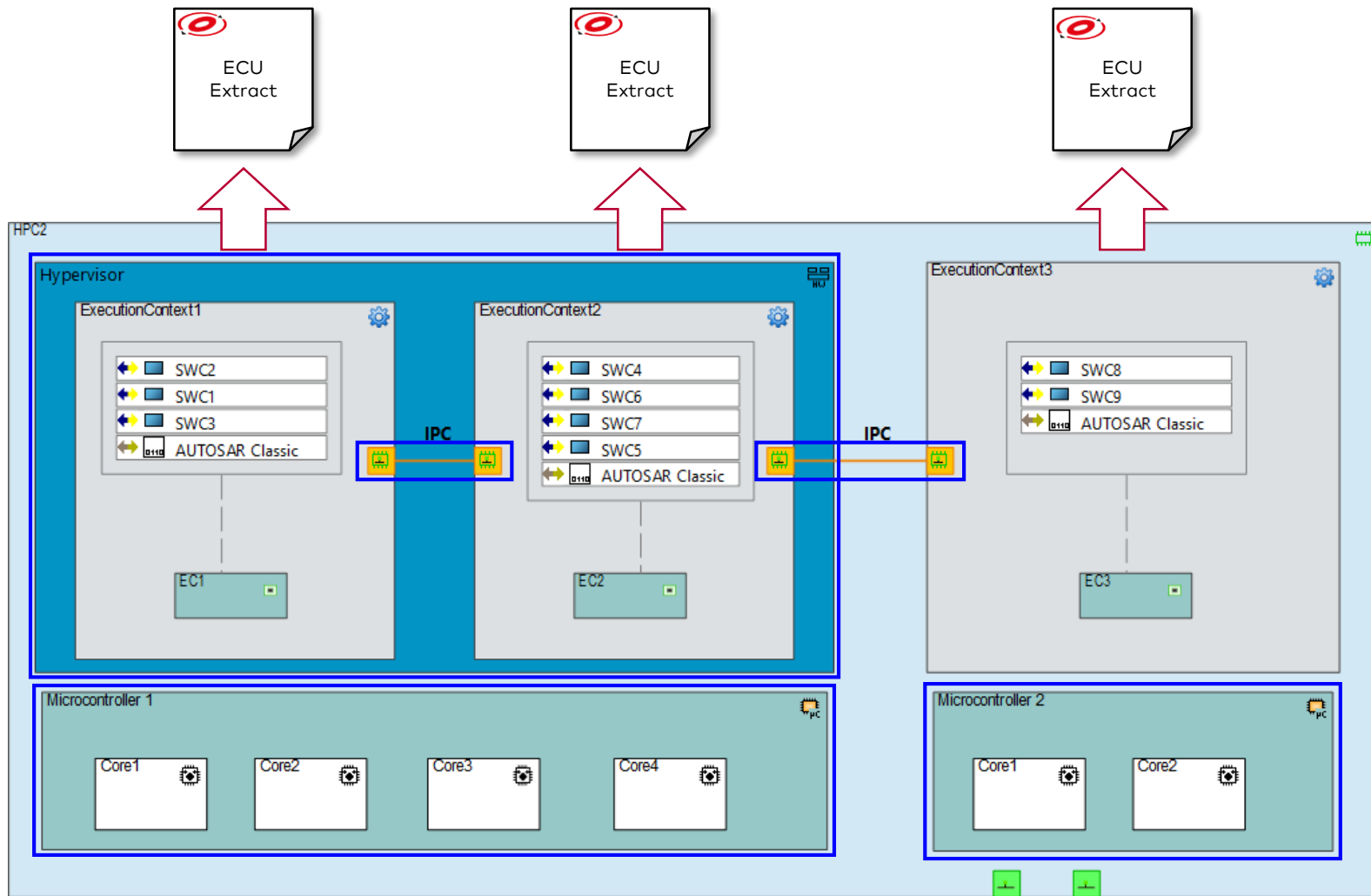
High Performance Computer – Example 1



- ▶ Example: One HPC with one microprocessor (μ P) and one microcontroller (μ C)
- ▶ In each execution context may run a **different basic software**
 - ▶ On the microprocessor runs AUTOSAR Adaptive
 - ▶ On the microcontroller runs AUTOSAR Classic
- ▶ Execution contexts may communicate via internal communication (e.g. **Inter Process Communication**)
- ▶ For every Execution Context can be exported the corresponding configuration file (ECU Extract, Adaptive Application or Manifest)

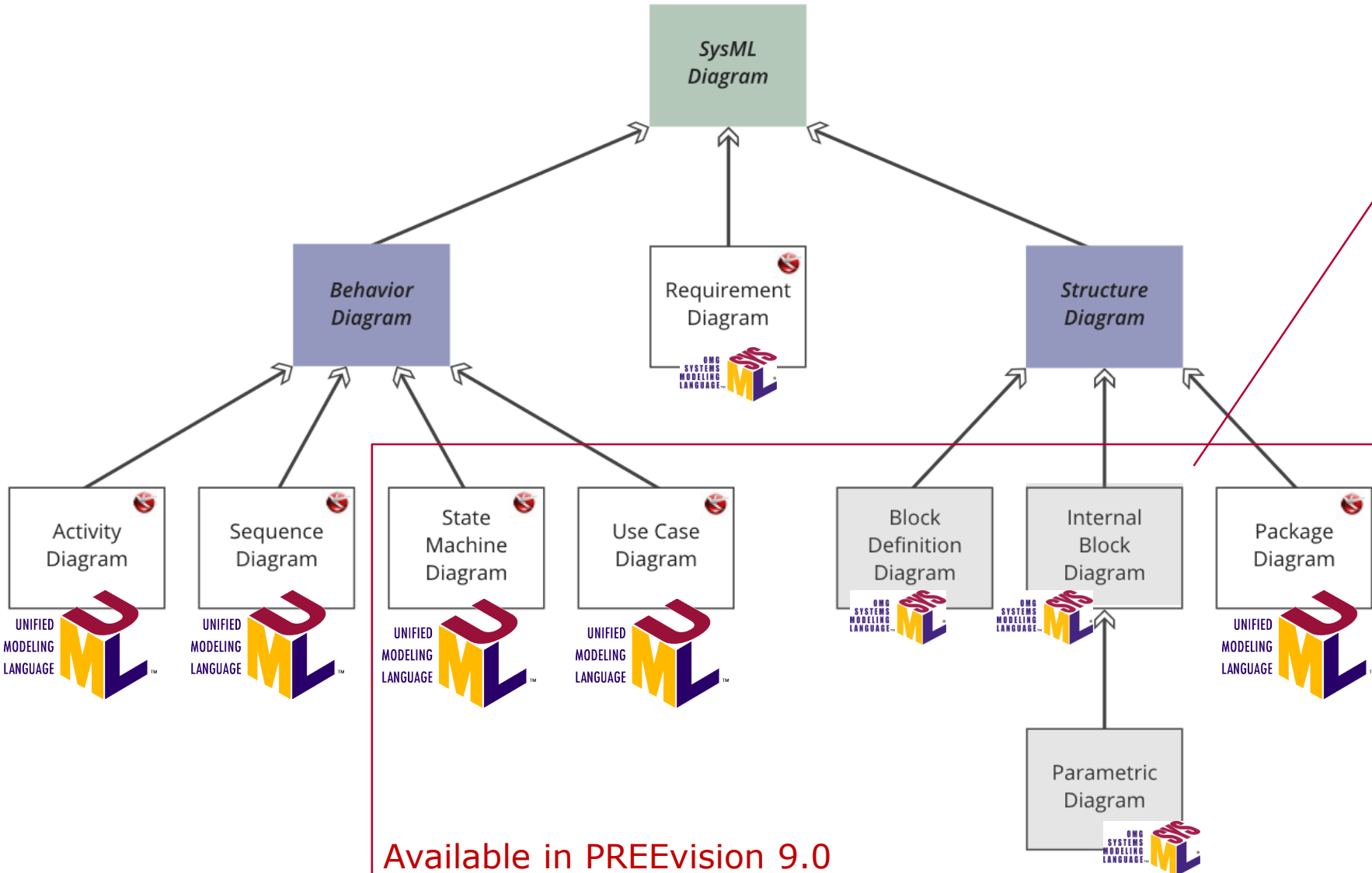


High Performance Computer – Example 2

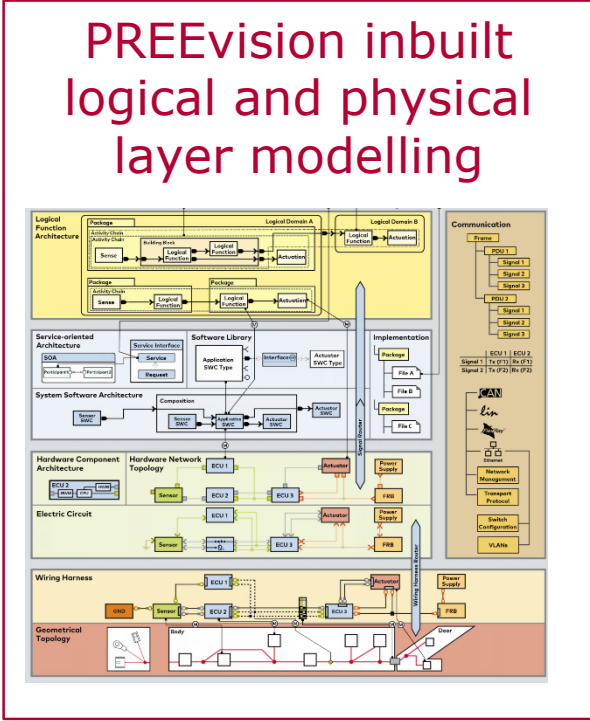


- ▶ Example: One HPC with two microcontrollers (μC)
- ▶ The **hypervisor** of type 1 defines **isolated execution contexts** where independent basic software stacks run (e.g. MICROSAR)
 - ▶ The hypervisor does not share peripheral resources (MICROSAR Lean hypervisor)
- ▶ Execution contexts may communicate via internal communication (e.g. **Inter Process Communication**)
- ▶ For every Execution Context can be exported an ECU Extract

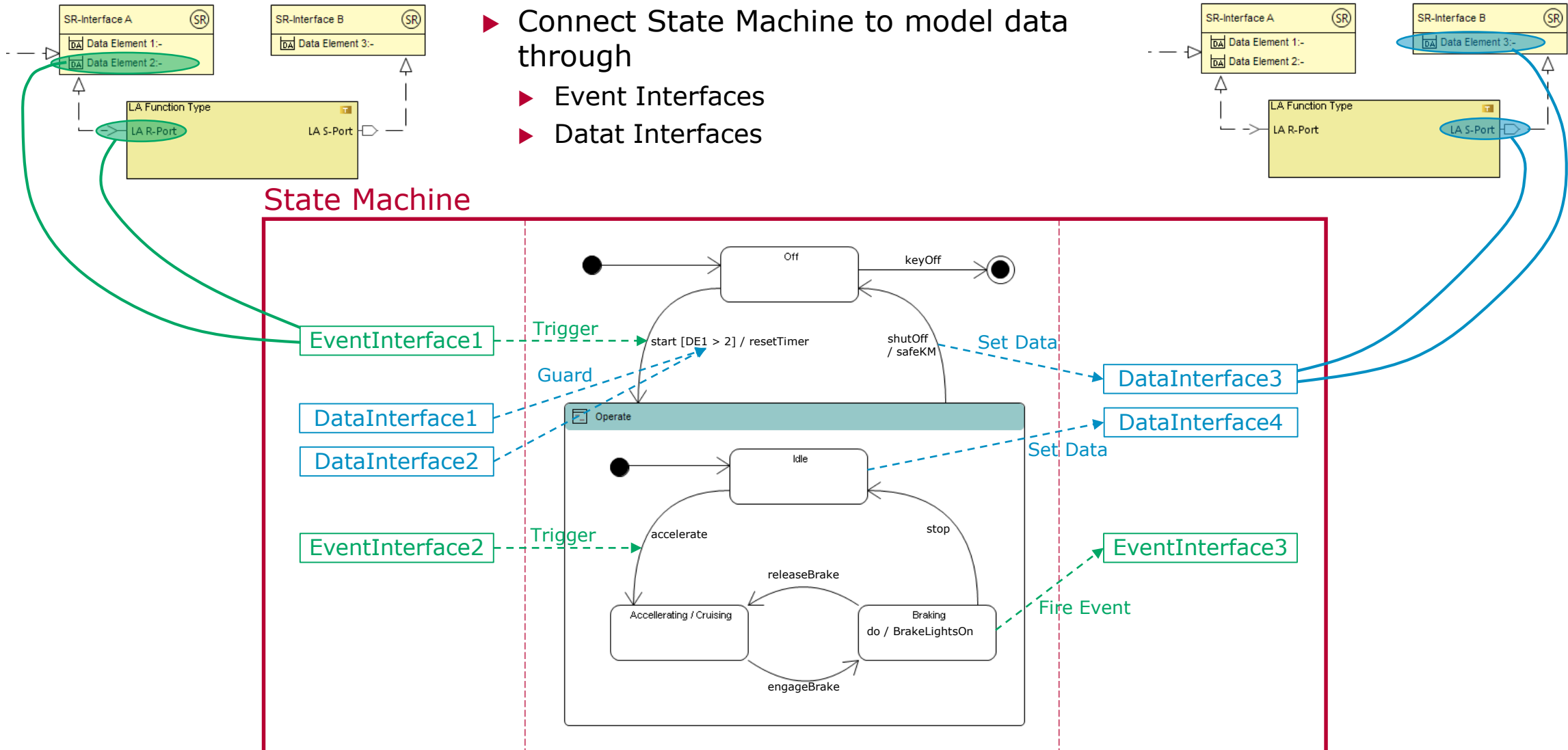
Focus SysML – PREvision as SysML-Tool for Automotive E/E Engineering



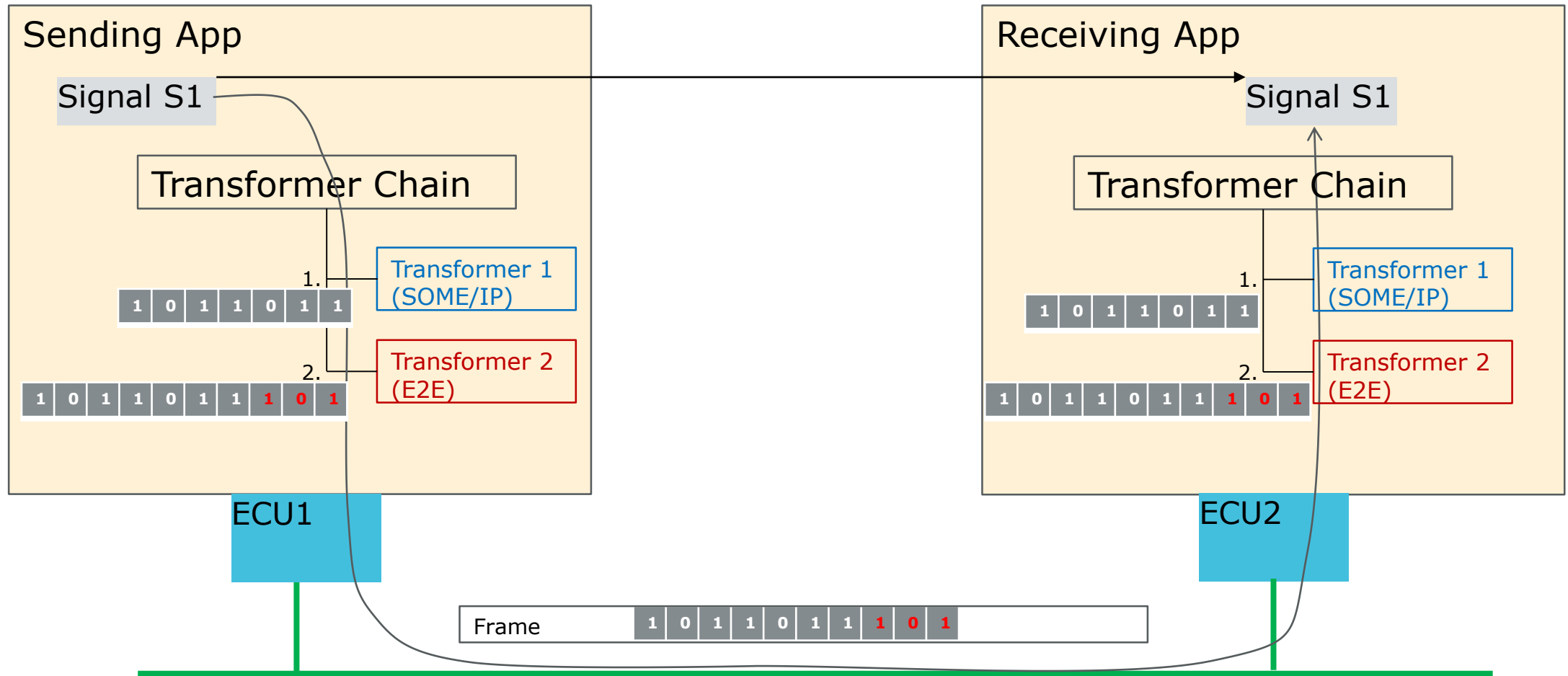
Available in PREvision 9.0



Data Interfaces



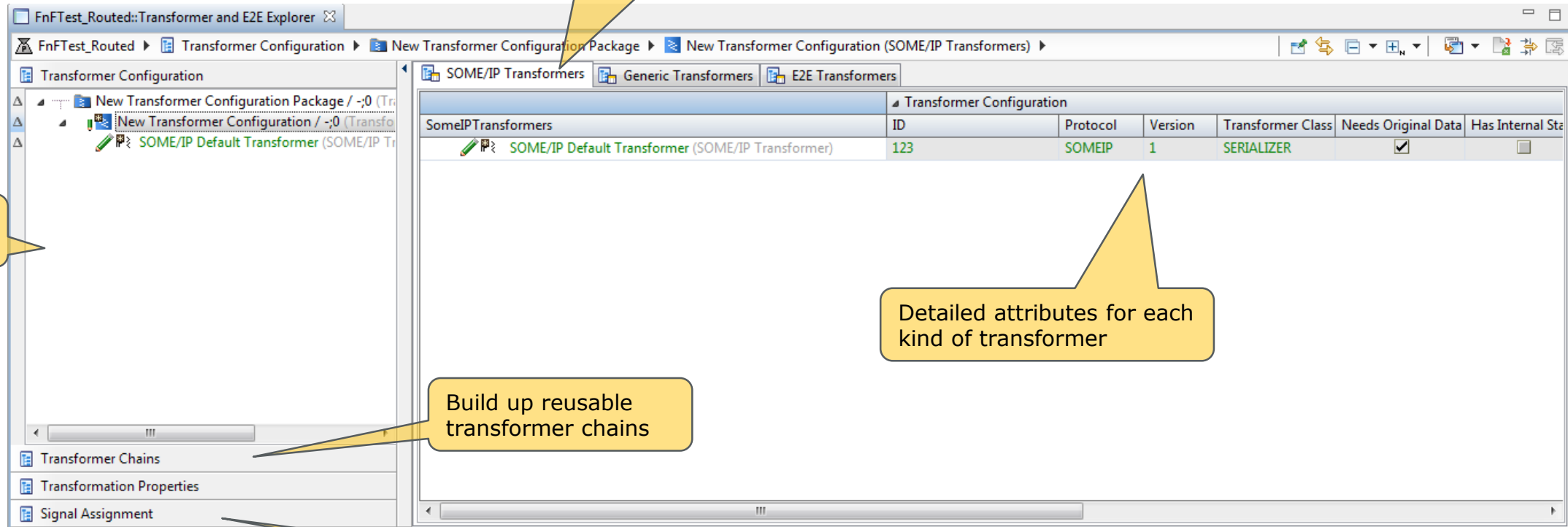
Transformer and End to End Protection



End to End Protection and Transformation

Comfortable Design Explorer

Support for SOME/IP, E2E and generic transformer

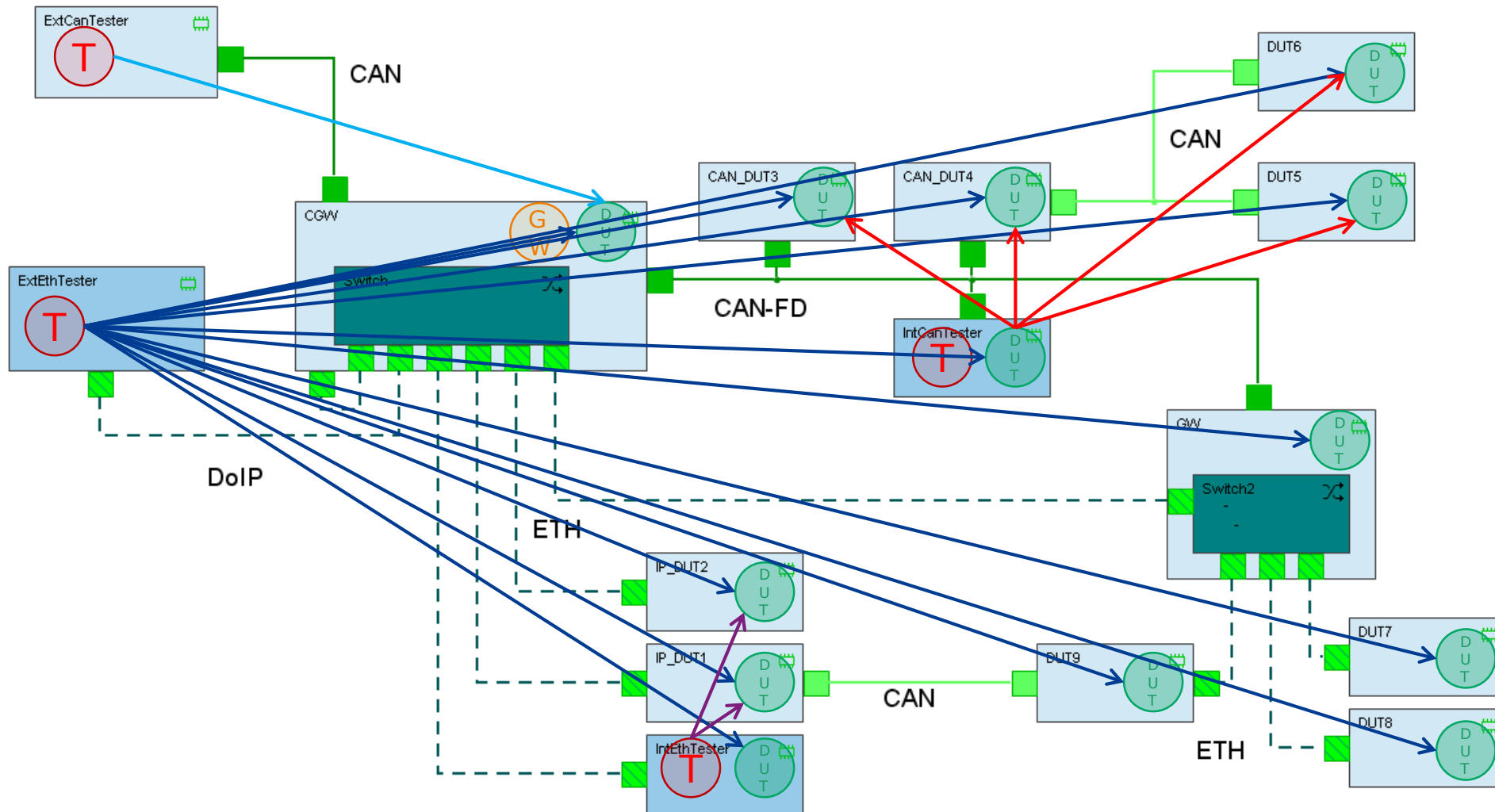


Build up reusable transformer chains

Detailed attributes for each kind of transformer

Assigning transformer chains to signals or signal groups

A Simple Network... to be diagnosable



Transport Protocol and Diagnostics communication

1. Find valid diagnosis paths automatically

Comfortable Design Explorer

The screenshot shows the 'Diagnostic Communication Explorer' window. The main table lists diagnostic paths between a Tester and a DUT. The 'Information View' at the bottom shows a tree structure of 'Found Diagnostic Paths' and a table of path details.

Tester	DUT	Detailed Path	Bus Type
Body Domain	B1	<No detailed path>	
Body Domain	B2	Seg: BA_Body Domain_1 → BA_ECU163_0	CAN : CAN / DIAG;0 (Bus Sy...
Body Domain	B3	Seg: BA_Body Domain_1 → BA_ECU164_0	CAN : CAN / DIAG;0 (Bus Sy...
DiagnosticsTester4	Body Domain	Seg: BA_DiagnosticsTester_0EEI → BA_Body Domain_0EEI	VLAN1 [100] / DIAG;0 (Vlar
DiagnosticsTester4	CGW	Seg: BA_DiagnosticsTester_0EEI → BA_CGW_0EEI	VLAN1 [100] / DIAG;0 (Vlar
DiagnosticsTester4	Comfort Domain	Seg: BA_DiagnosticsTester_0EEI → BA_Comfort Domain_0EEI	VLAN1 [100] / DIAG;0 (Vlar
DiagnosticsTester4	Driving Domain	Seg: BA_DiagnosticsTester_0EEI → BA_Driving Domain_0EEI	VLAN1 [100] / DIAG;0 (Vlar
DiagnosticsTester4	Infotainment Domain	Seg: BA_DiagnosticsTester_0EEI → BA_Infotainment Domain_0EEI	VLAN1 [100] / DIAG;0 (Vlar

Source	Bus	Target
Body Domain / DIAG;0 (ECU)	CAN : CAN / DIAG;0 (Bus Syste...	B1 / DIAG;0 (ECU)

3. Result posted to information view

2. Create unique path information automatically

4. Manual selection of ambiguous paths

Agenda

PREEvision at a glance

Service Oriented Architectures

Tool Demo

What's next: SOA & Ethernet in PREEvision 9.5

▶ **Summary**



Summary – SOA & Ethernet @ PREEvision

- ▶ Service oriented Architectures (SOAs) provide flexible, open and dynamic distributed systems.
- ▶ They are enablers for
 - > Connectivity and Autonomous Driving
 - > SW Update, SW Upgrade and Vehicle Service in the field
 - > New Building Set Strategies and Handling of Variants.
- ▶ Future E/E Architectures will combine the strengths of Signal Oriented and Service Oriented Architectures.
- ▶ AUTOSAR Adaptive and AUTOSAR Classic will be deployed in the same vehicle, even in the same ECU.
- ▶ PREEvision already supports the introduction of SOAs, the migration to SOAs and their implementation in
 - > AUTOSAR Classic
 - > AUTOSAR Adaptive
- ▶ Enhanced Ethernet and AUTOSAR capabilities coming with PREEvision 9.5
 - > Support of mixed System: AUTOSAR Classic and AUTOSAR Adaptive
 - > High Performance Computers
 - > DoIP & Diagnostic Communication Infrastructure
 - > TSN Features: Global Time Synchronization, Credit based Shaper

For more information about Vector
and our products please visit

www.vector.com

Author:
Alexander Mayr
Vector Germany