

AUTOSAR Method

Webinar 2011-10-19

> Introduction

AUTOSAR Method

Exchange Formats

Workflows OEM-TIER1

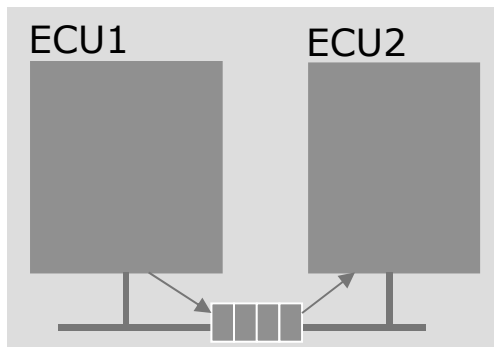
Workflows TIER1

Webinar Series

Introduction

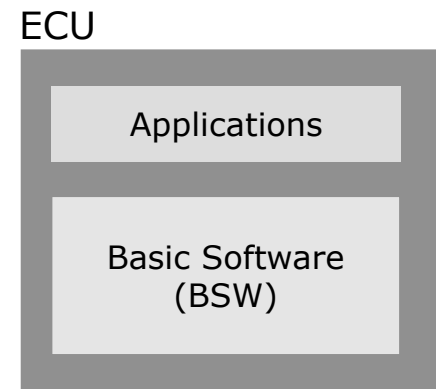
Current Workflow (non-AUTOSAR)

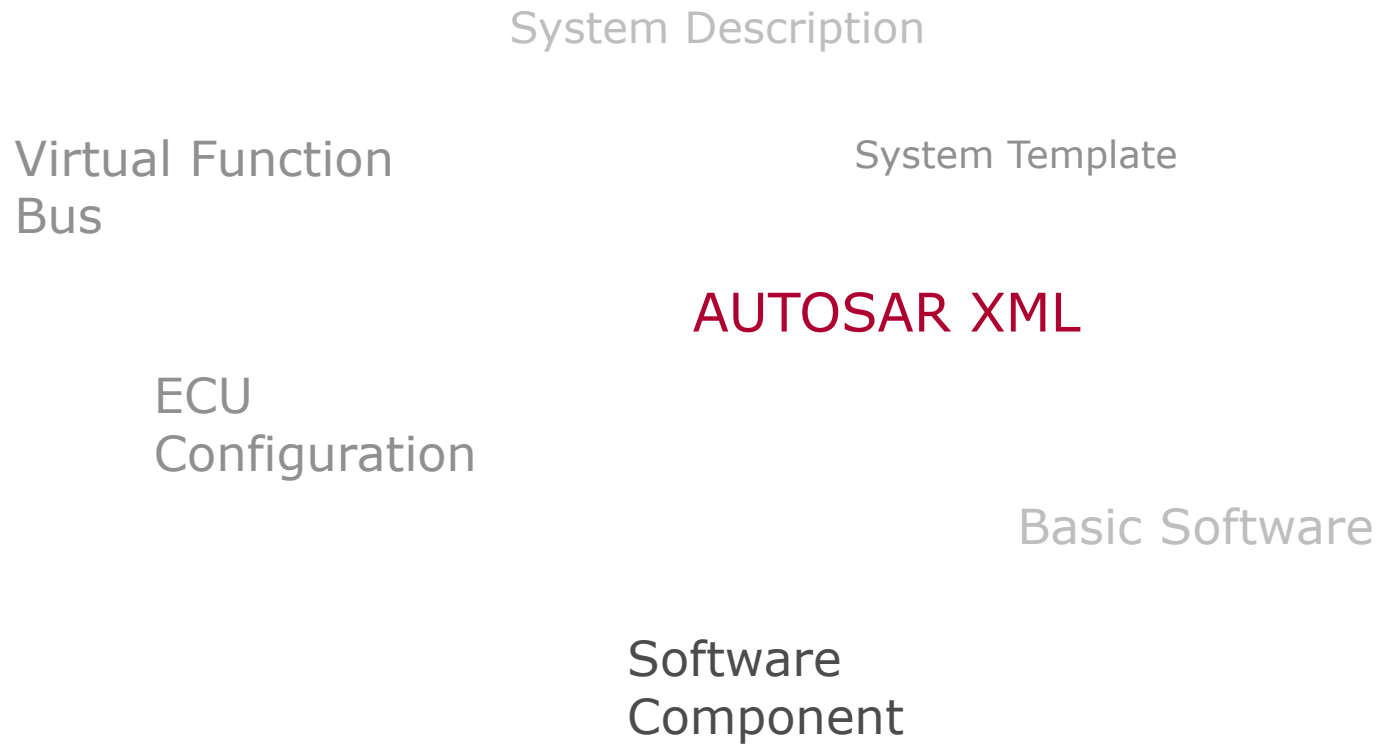
Network communication of the vehicle is specified



DBC, FIBEX, LDF

The ECU is configured based on the DBC, FIBEX, LDF file.





Introduction

> **AUTOSAR Method**

Exchange Formats

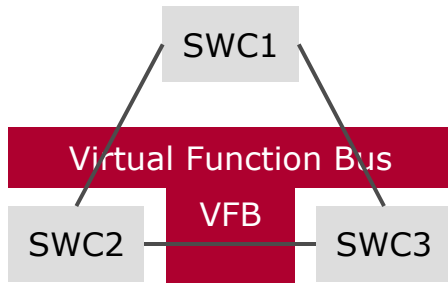
Workflows OEM-TIER1

Workflows TIER1

Webinar Series

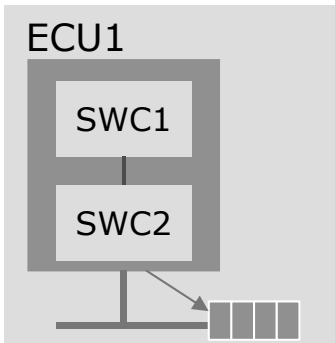
AUTOSAR Method

Complete SW functionality of the vehicle is defined as a system of SWCs...



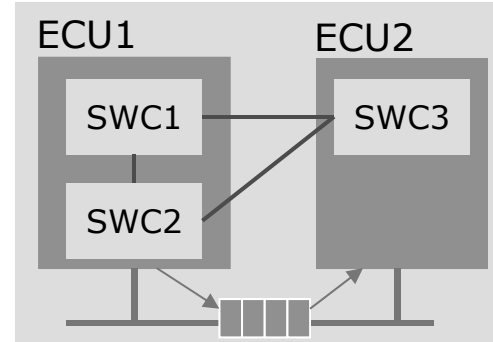
Software Component Description*

An extract is created for each ECU...



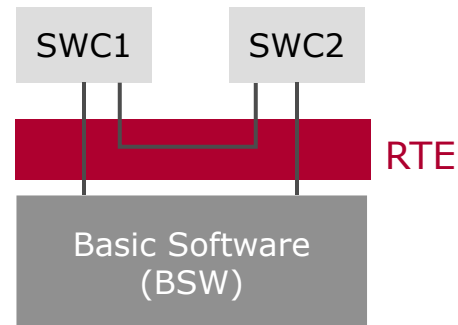
Extract of System Description*

..and distributed to ECUs



System Description*

The ECU is configured based on the ECU Extract.

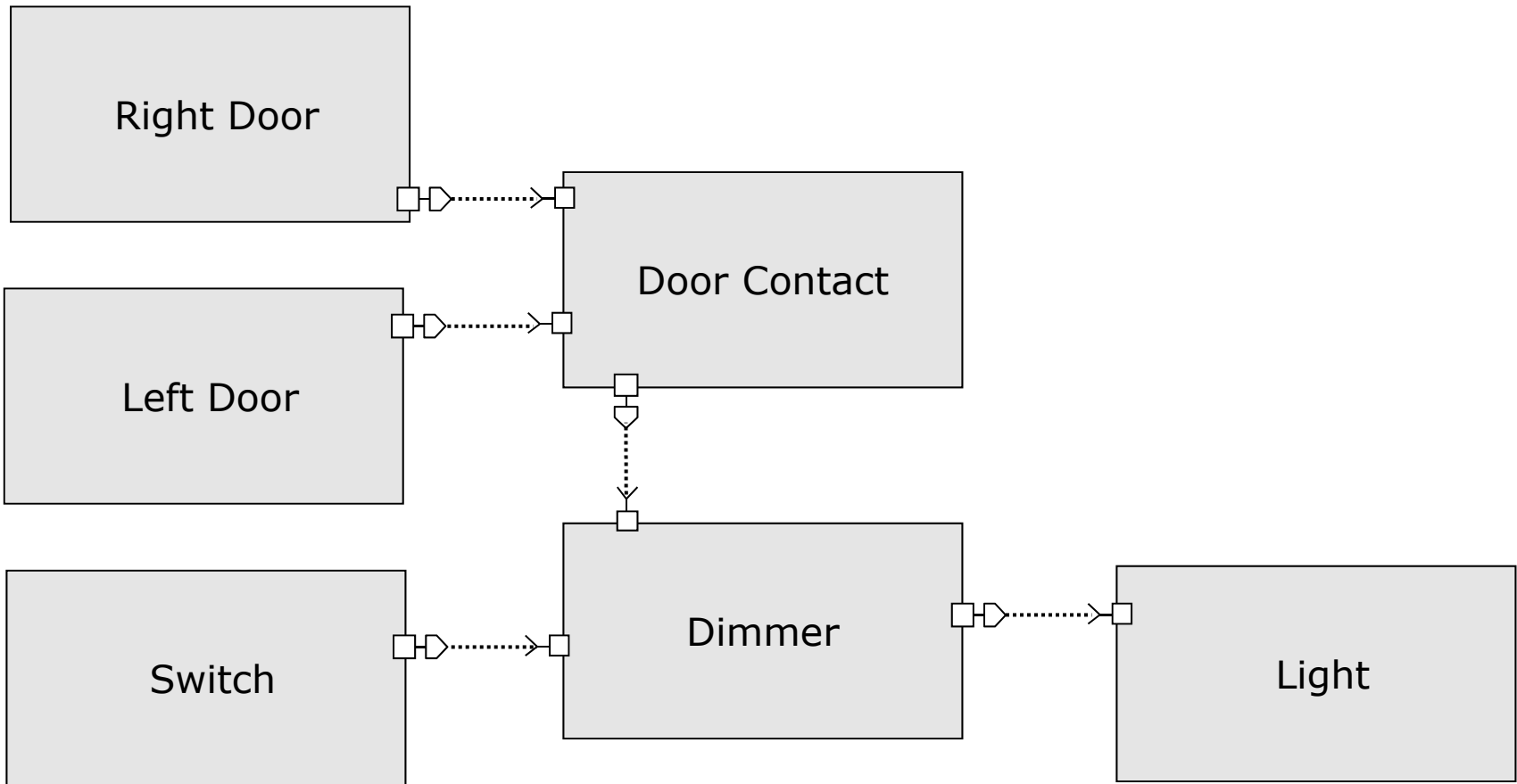


ECU Configuration Description (ECUC)*

* **AUTOSAR**

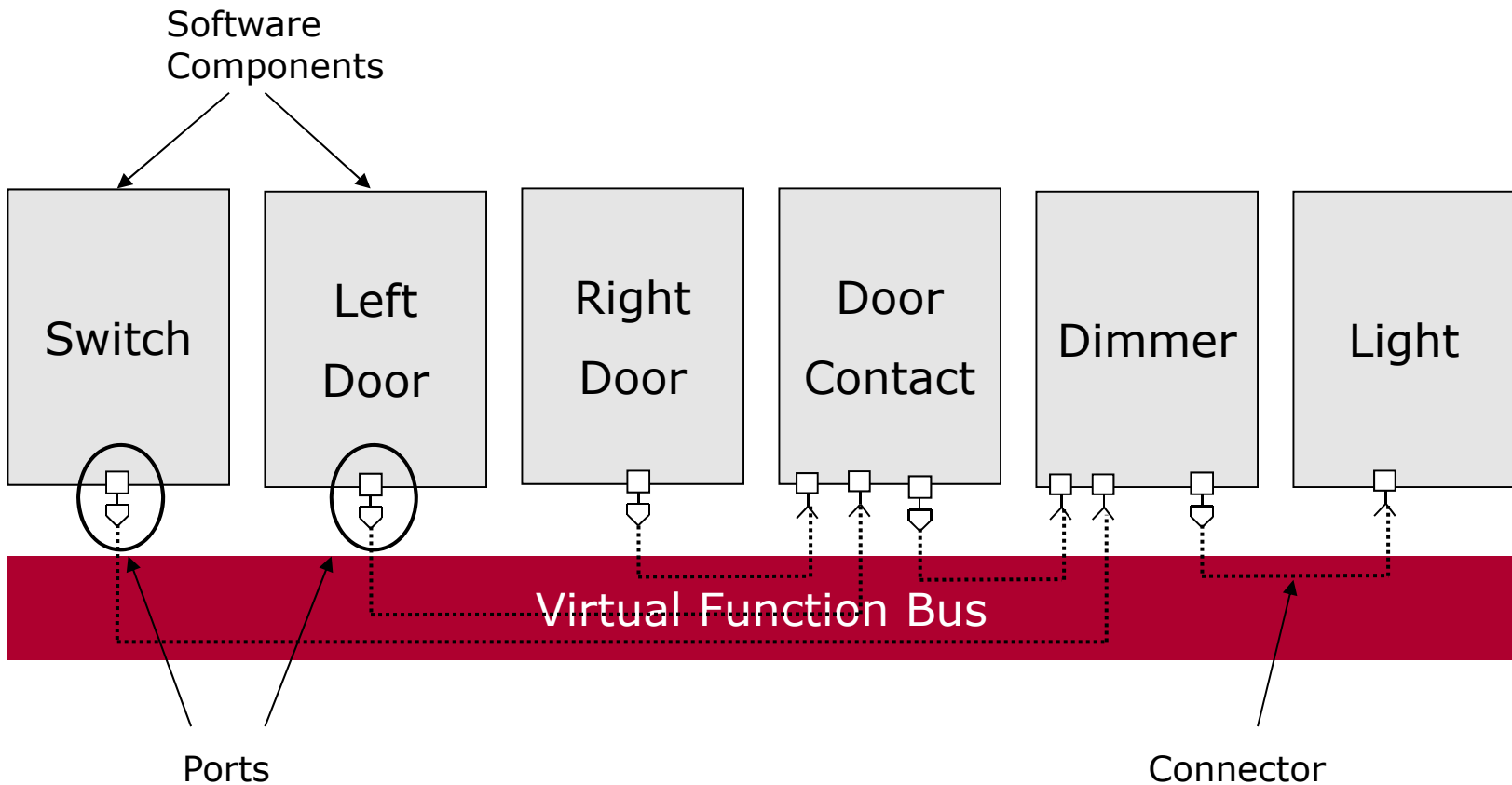
AUTOSAR Method

Example: Components view for interior lighting control



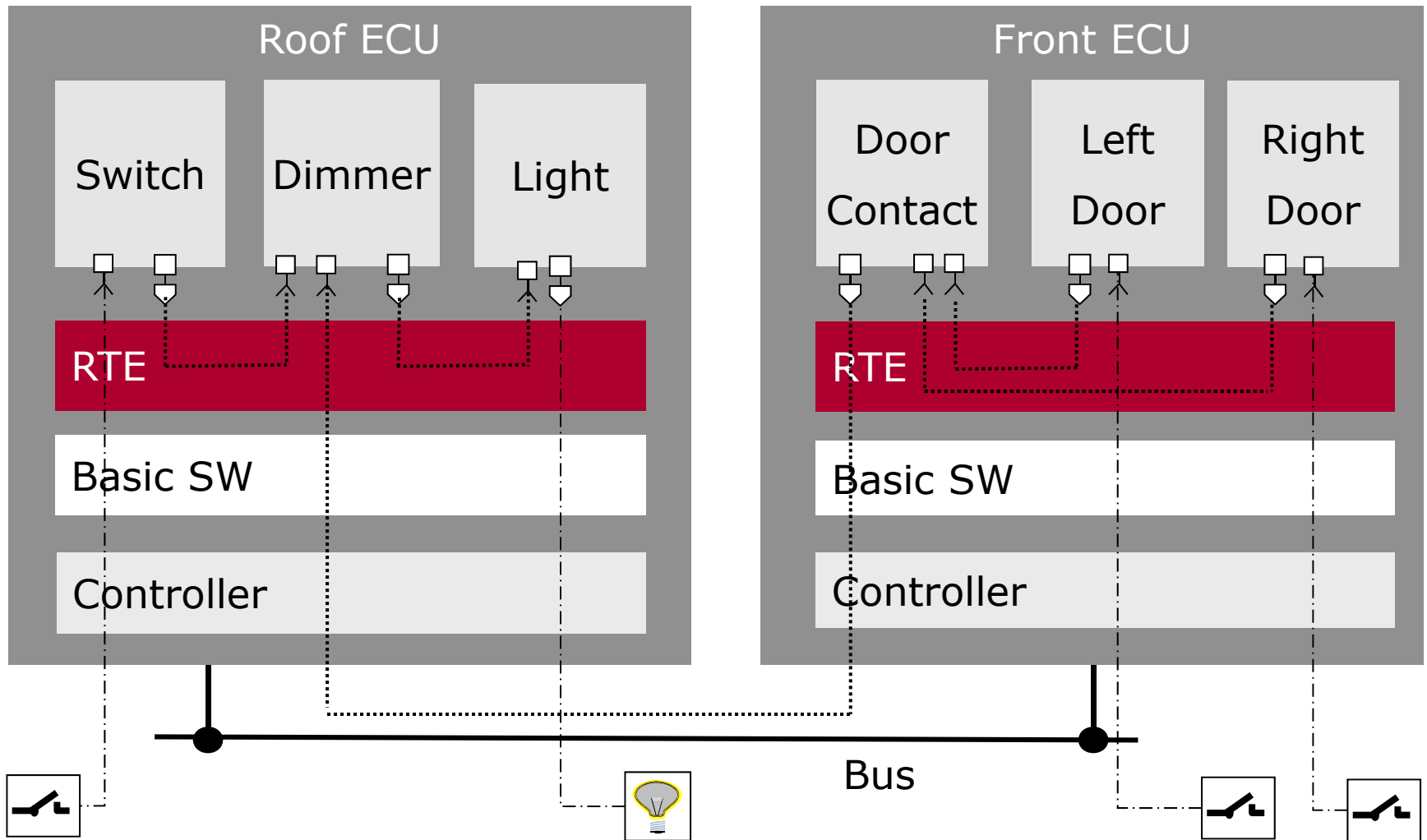
AUTOSAR Method

Example: Communication



AUTOSAR Method

Example: Distribution of components



Introduction

AUTOSAR Method

> Exchange Formats

Workflows OEM-TIER1

Workflows TIER1

Webinar Series

Templates

SW Component
Template

ECU Resource
Template

System
Template

Descriptions

SW Component
Description

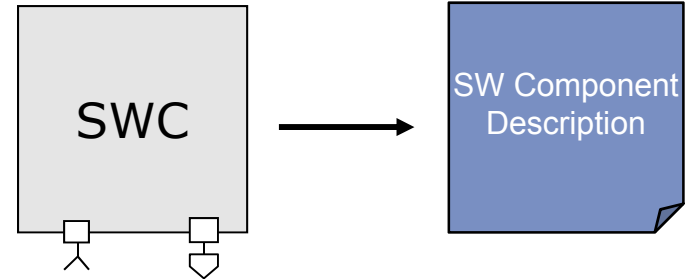
ECU Resource
Description

System
Description

- ▶ Standardized set of design artifacts
- ▶ Associated XML Schema
- ▶ Defined by **AUTOSAR**

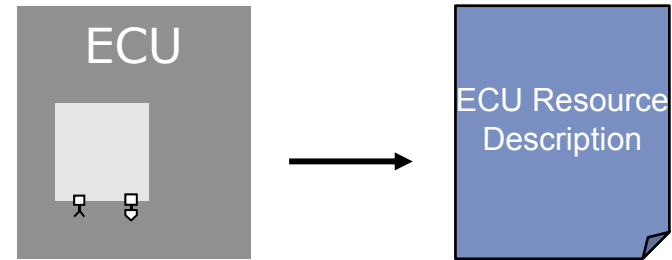
- ▶ Concrete specification based on the Template
- ▶ Represented as XML file
- ▶ Defined by the OEM or TIER1

The AUTOSAR “Software Component Description” contains the following information



- ▶ Ports and Interfaces (sender/receiver, client/server)
- ▶ Runnable Entities with trigger events, port access, etc
- ▶ Resource needs of the component (memory, CPU time, etc.)
- ▶ Structure and format are described by the “Software Component Template”.

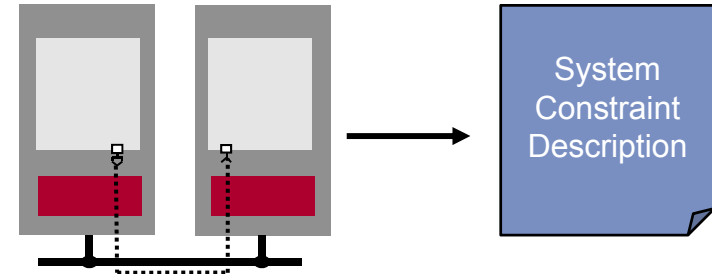
The AUTOSAR “ECU Resource Description” contains the following information



- ▶ Description of the hardware being used
 - > Sensors, actuators
 - > Memory
 - > Processor
 - > Communications periphery
 - > Pin assignments

- ▶ The structure and format are described by the “ECU Resource Template”.

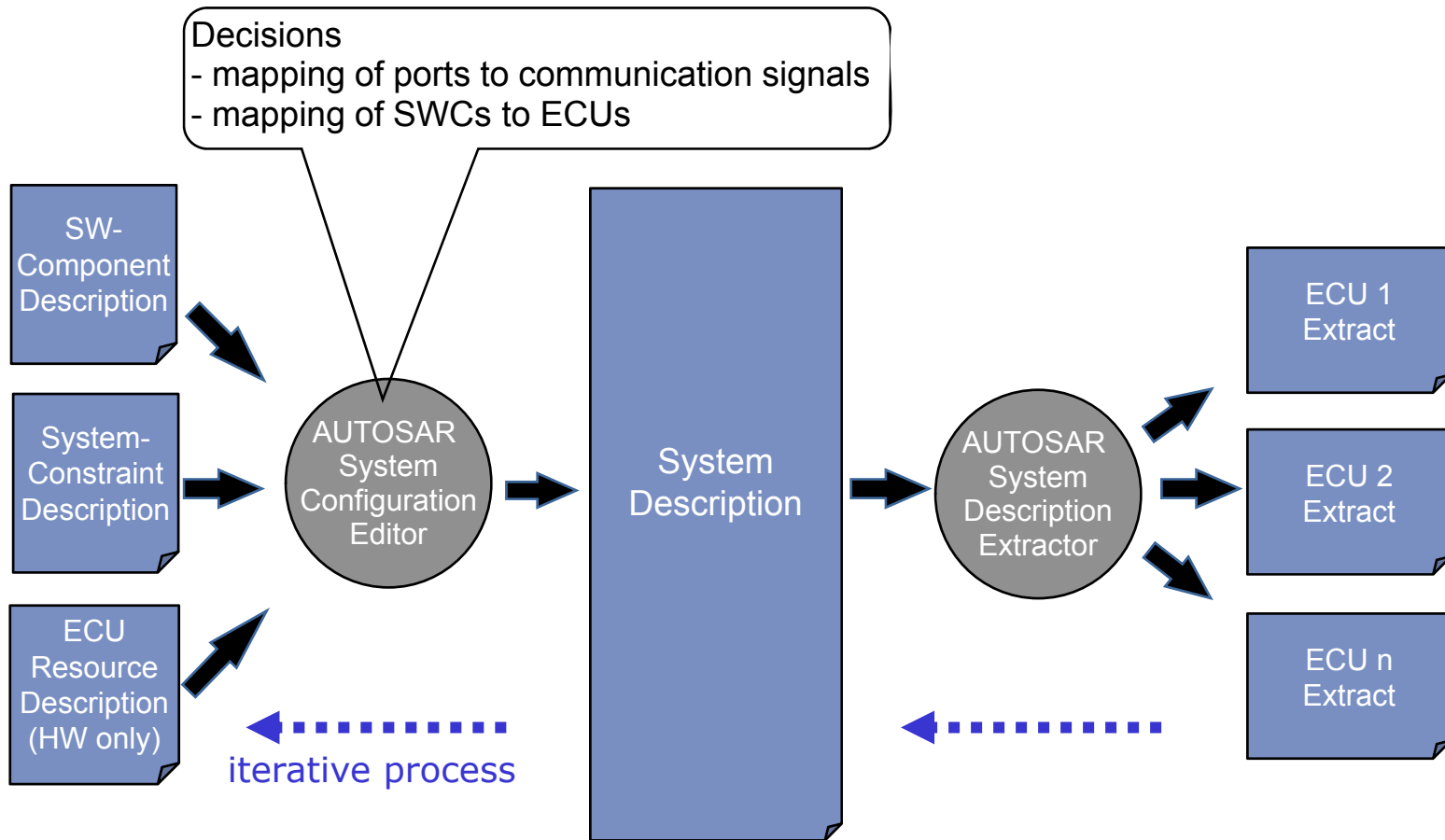
The AUTOSAR “System Constraint Description” contains the following information:



- ▶ Information of network topology
 - ▶ Limitations (“Constraints”)
 - ▶ Protocol
 - ▶ Given communication matrix (PDUs, signals, ...)
 - ▶ Baud rate, timing
-
- ▶ Structure and format are described by the “System Template”.

Exchange Formats

System Description



The AUTOSAR "System Description" contains the actual configuration of the overall system

The AUTOSAR "ECU Extract of System Description" contains the subset for one ECU

Introduction

AUTOSAR Method

Exchange Formats

> Workflows OEM-TIER1

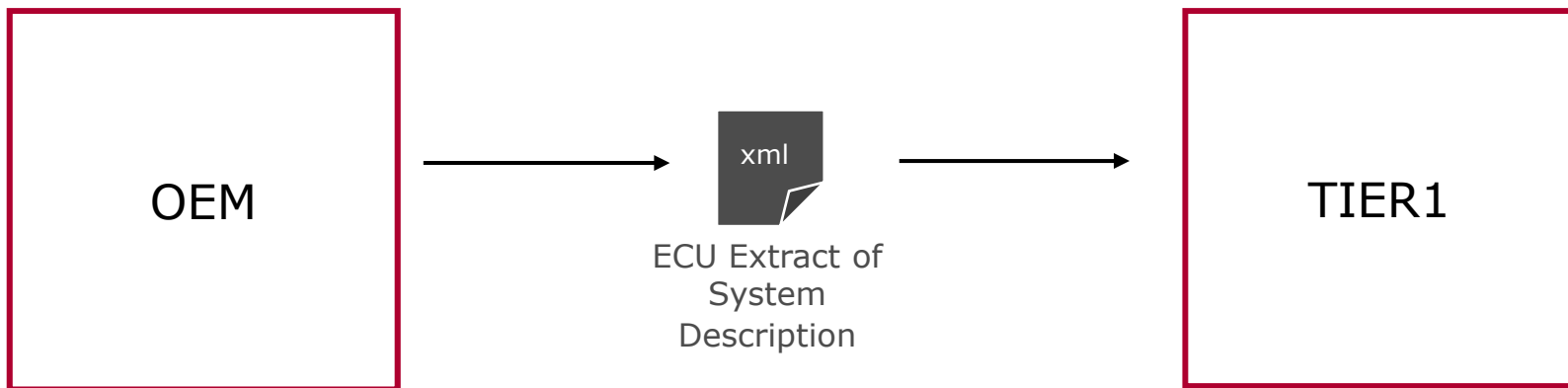
Workflows TIER1

Webinar Series

Workflows OEM-TIER1

AUTOSAR Workflow with OEM and TIER1

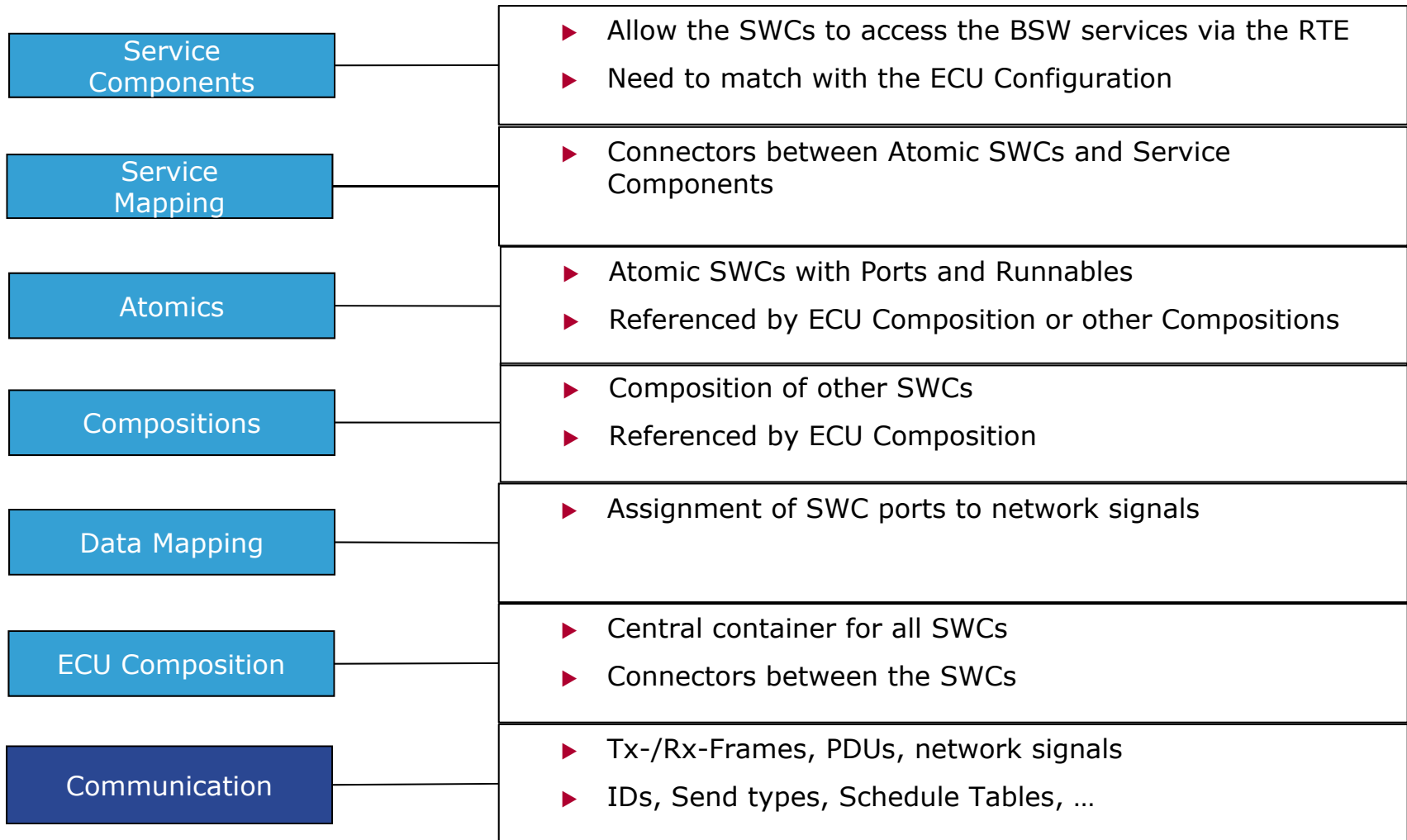
- ▶ Interface between OEM* and TIER1*: ECU Extract of System Description (ECUEX)
- ▶ OEM creates ECUEX based on vehicle system design
- ▶ TIER1 configures AUTOSAR ECU based on ECUEX



*Note: "OEM" and "TIER1" may also be organizational units within one company ("System Responsible" vs. "ECU responsible")

Workflows OEM-TIER1

Content of ECUEX



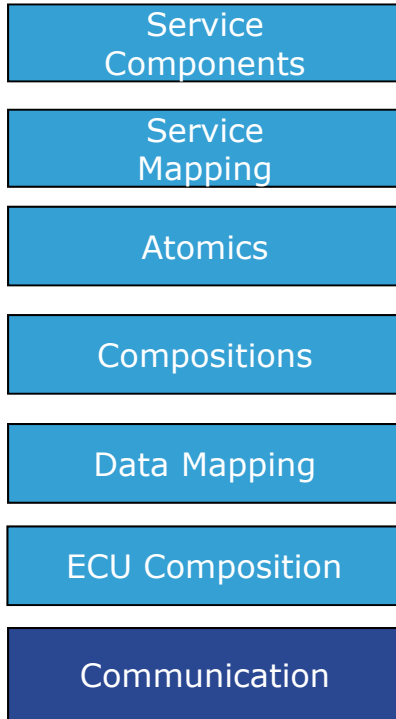
AUTOSAR XML format allows also ECUEX files with only a subset of this content!

- ▶ Typically, an OEM cannot provide the full content of the ECUEX because
 - ▶ Some content is only known after ECU integration (e.g. Service Components)
 - ▶ Some content is not in the responsibility of the OEM (e.g. SW implementation details)
- ▶ TIER1 must complete the ECUEX
- ▶ Iterations must be supported
 - ▶ TIER1 work must not be lost when getting update of ECUEX from OEM
- ▶ Several use cases possible
 - ▶ Some examples are shown in the next slides

Workflows OEM-TIER1

Content of ECUEX - Examples

Use Case 1: Network-oriented approach (same process as today, only with new AUTOSAR XML format instead of DBC/FIBEX/LDF)



- ▶ OEM defines network communication
- ▶ TIER1 implements all ECU functions by
 - ▶ ECU Composition
 - ▶ Data Mapping
 - ▶ Compositions
 - ▶ Atomics
 - ▶ Service Components/Mapping

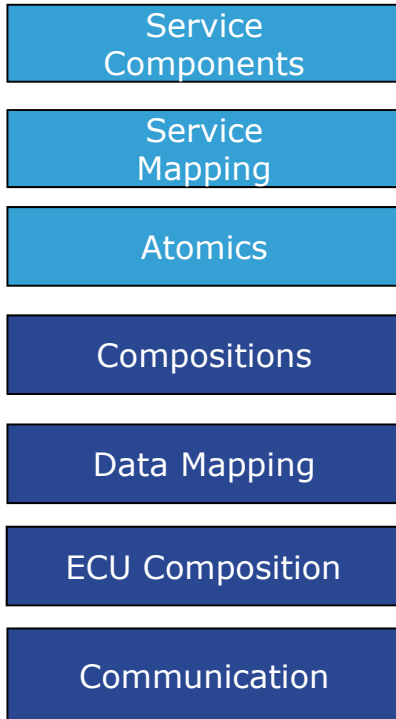
■ TIER1

■ OEM

Workflows OEM-TIER1

Content of ECUEX - Examples

Use Case 2: Functional system design approach, top-level architecture of ECU functional SW specified by OEM



- ▶ OEM defines network communication
- ▶ OEM defines ECU function interfaces within the ECU by
 - ▶ ECU Composition
 - ▶ 1 Composition for each ECU function (“empty” composition – only ports, no substructure)
 - ▶ Connectors between the Compositions
 - ▶ Data Mapping
- ▶ TIER1 implements the ECU functions by
 - ▶ Atomics
 - ▶ Service Components/Mapping

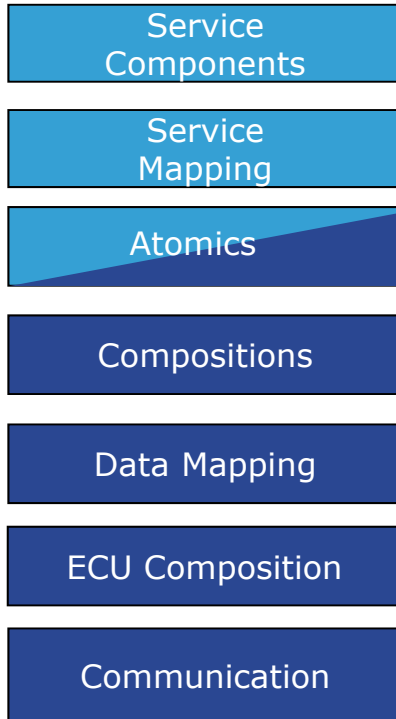
■ TIER1

■ OEM

Workflows OEM-TIER1

Content of ECUEX - Examples

Use Case 3: Functional system design approach, top-level architecture of ECU functional SW specified by OEM, SW partially implemented by OEM



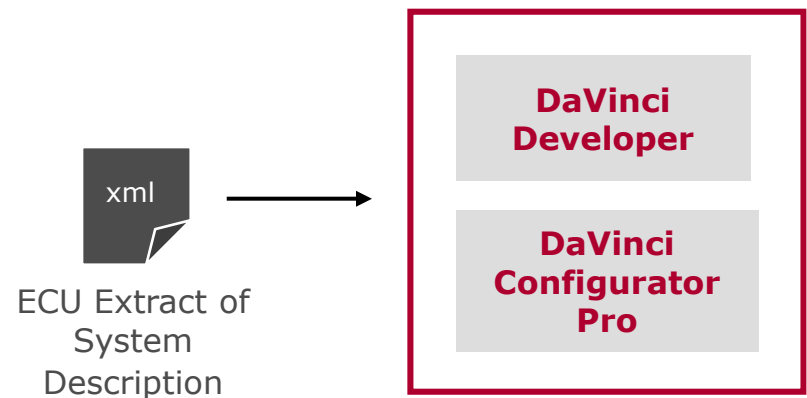
■ TIER1

■ OEM

- ▶ OEM defines network communication
- ▶ OEM defines ECU function interfaces within the ECU by
 - ▶ ECU Composition
 - ▶ 1 Composition for each ECU function (Ports only, no substructure)
 - ▶ Connectors between the Compositions
 - ▶ Data Mapping
- ▶ OEM provides implementation of some of the ECU functions by
 - ▶ Atomics
- ▶ TIER1 implements the remaining ECU functions by
 - ▶ Atomics
 - ▶ Service Components/Mapping

- ▶ Ideal situation: clearly separated responsibility regarding the content of the ECUEX
 - ▶ No conflicts due to concurrent modification of same content by OEM and TIER1
- ▶ Realistic: TIER1 requires flexibility to make modifications within content created by OEM
 - ▶ Add TIER1 standard SW, which is not in responsibility of OEM
 - > Additional ports (Service Ports, Application Ports)
 - > Additional top-level components
 - ▶ Define additional networks, which are not in responsibility of OEM
 - > e.g. private CAN within the ECU, LIN to sensor cluster
 - ▶ Fix issues in the OEM content to avoid delays in development

- ▶ Tool support for iterations in the workflow (supports the mentioned use cases)
 - ▶ Set up a project based on an ECUEX from OEM
 - ▶ Allow modification of design data by TIER1
 - ▶ Incorporate changes in the ECUEX via automatic update/merge
 - > Update the design data, which were originally imported via ECUEX
 - > Keep the design data added by TIER1 unchanged
- ▶ Convenience
 - ▶ Diff viewer to analyze the modifications
 - ▶ Automation of update workflow



Introduction

AUTOSAR Method

Exchange Formats

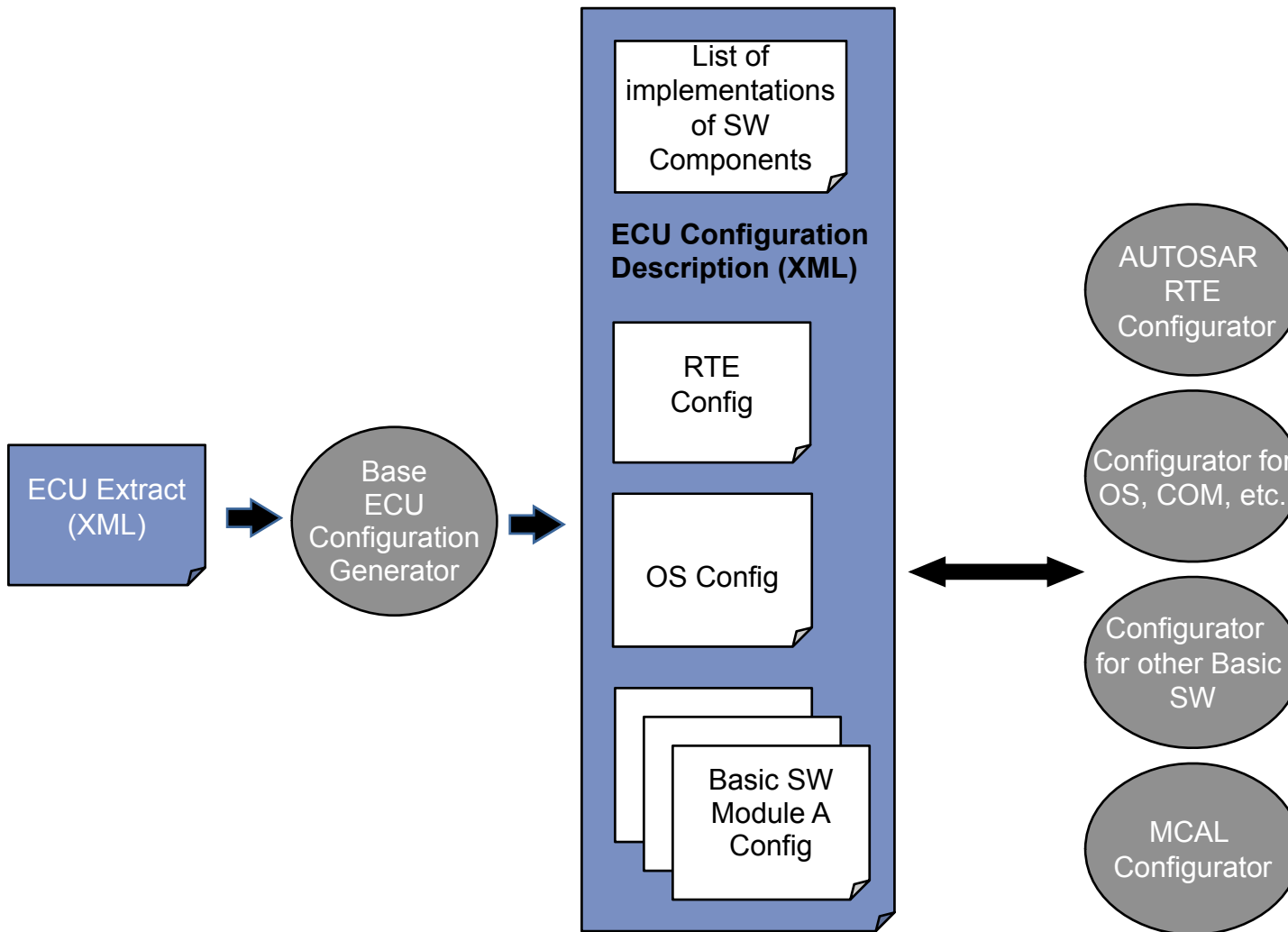
Workflows OEM-TIER1

> Workflows TIER1

Webinar Series

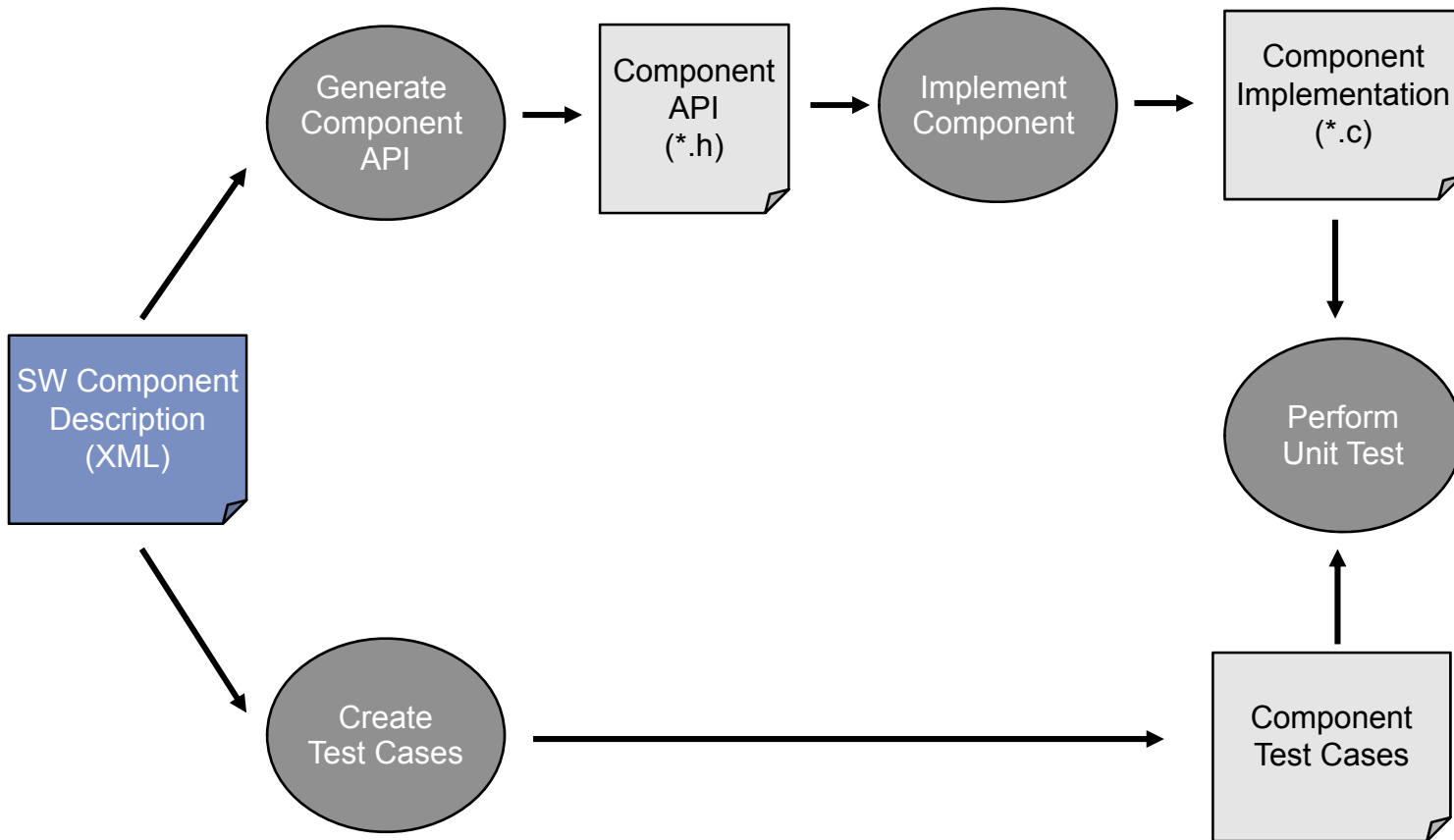
Workflows TIER1

ECU Configuration



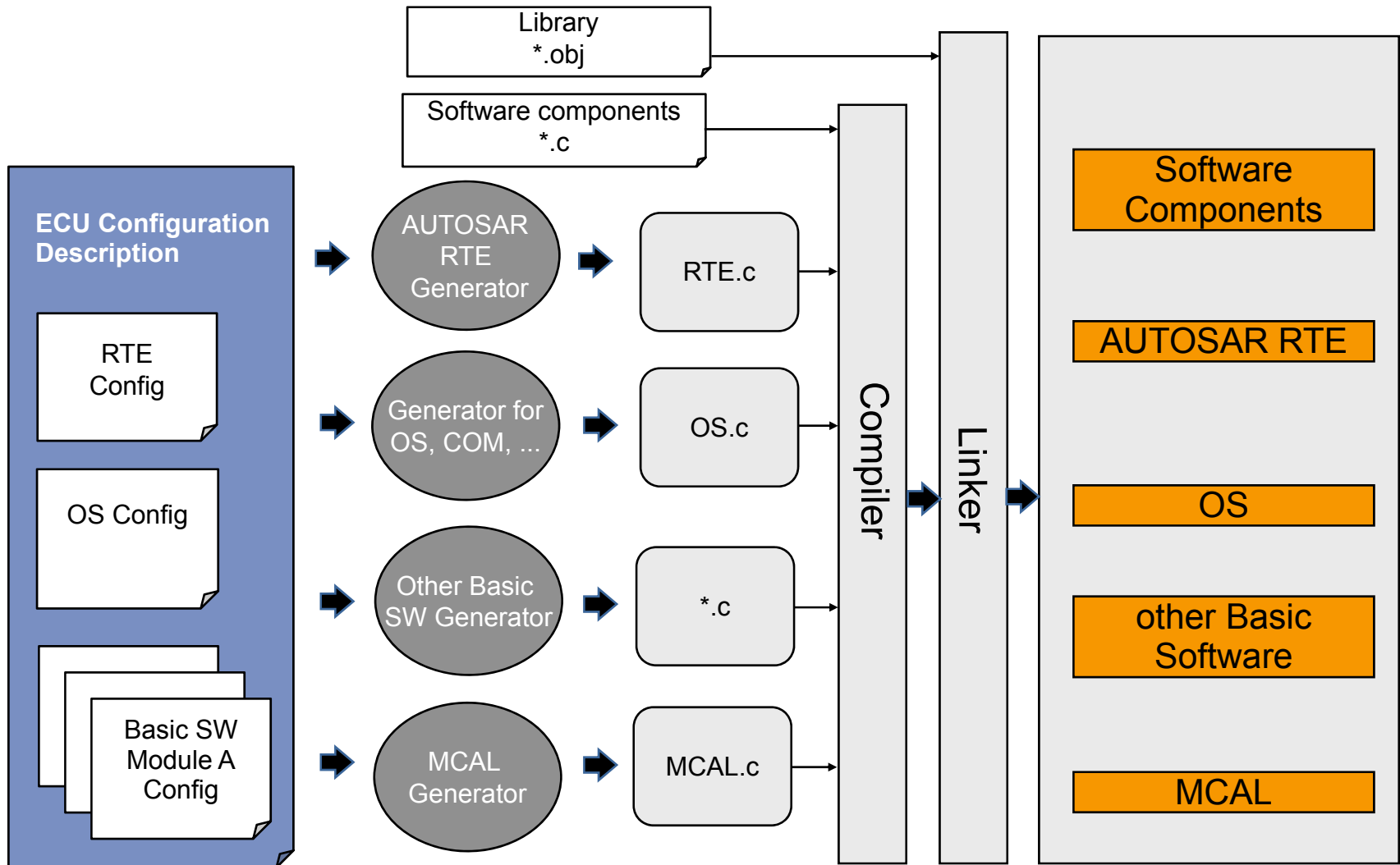
Workflows TIER1

Component development process



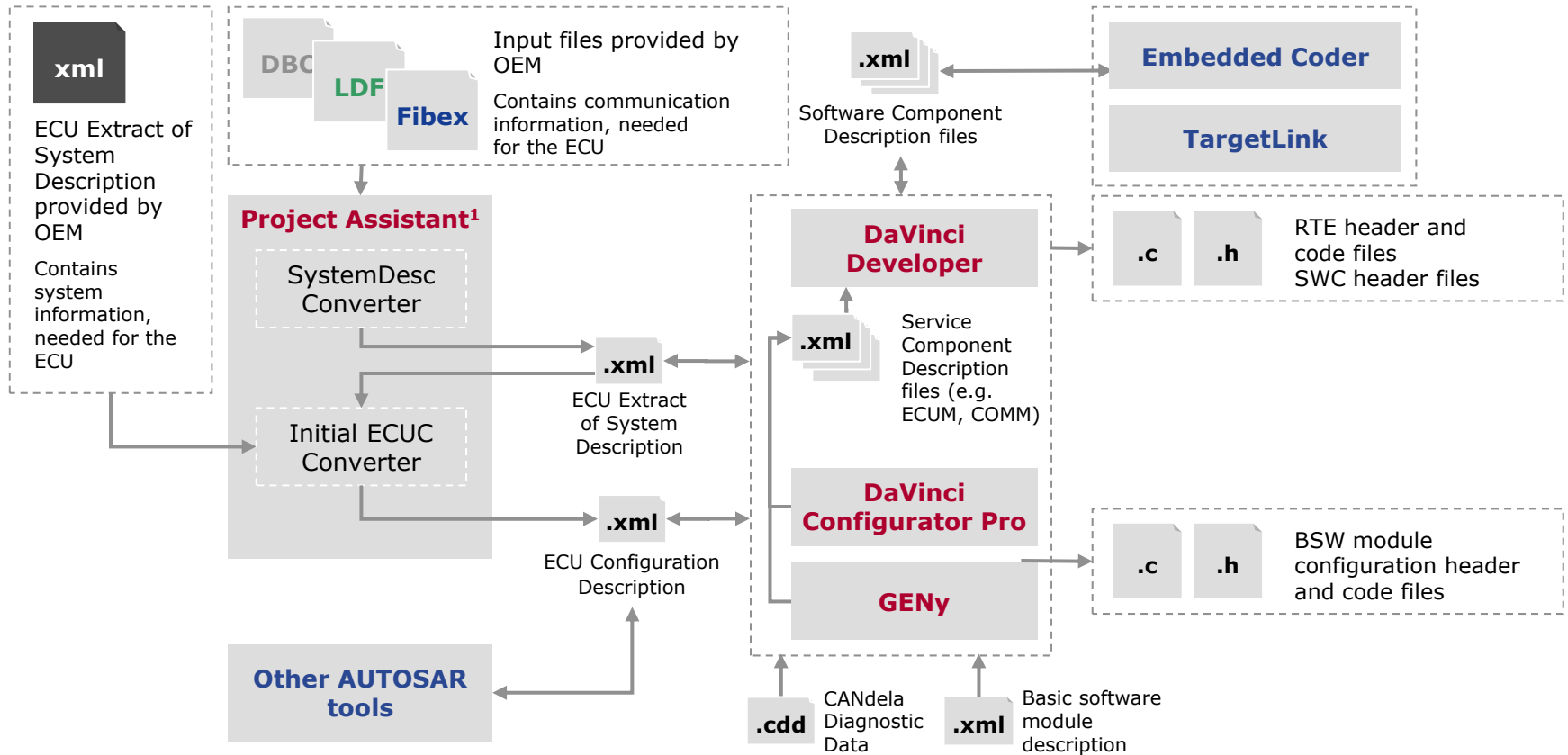
Workflows TIER1

Executable ECU



Workflows TIER1

Tool Support



¹ The Project Assistant is part of the Vector tool chain

- ▶ The webinar series about AUTOSAR
 - ▶ 2011-10-06 ECU development with AUTOSAR - An introduction for AUTOSAR beginners
 - ▶ 2011-10-19 Introduction to the AUTOSAR Method for ECU development
 - ▶ 2011-11-02 AUTOSAR Configuration Process - How to handle 1000s of parameters
 - ▶ 2011-11-15 Reliable Testing of AUTOSAR ECUs (in preparation)
 - ▶ 2011-12-06 MICROSAR Safe: AUTOSAR basic software for safety-relevant ECUs (ISO 26262)
 - ▶ Various Webinars on MICROSAR OS – the AUTOSAR operating system from Vector

- ▶ Registration to the upcoming Webinars and the list of recorded Webinars:
 - ▶ http://www.vector.com/vi_webinars_en.html
 - ▶ http://www.vector.com/vi_autosar_webinars_en.html (only AUTOSAR Webinars)

- ▶ The overview of Vector's training services:
http://www.vector.com/vi_training_en.html

- ▶ We stay online for some more minutes to answer your questions. Please ask your questions in the Q&A window.

- ▶ Contact data for additional questions, product information or presentation :
 - ▶ matthias.wernicke@vector.com
 - ▶ +49 (0) 711 80670 466

Thank you for your attention.

For detailed information about Vector
and our products please have a look at:

www.vector.com

Author:

Vector Informatik GmbH

Ingersheimer Str. 24

70499 Stuttgart