

# AUTOSAR Configuration Process - How to handle 1000s of parameters

Webinar 2011-11-02

## > Introduction

AUTOSAR ECU Configuration Description

ECU Configuration Workflow

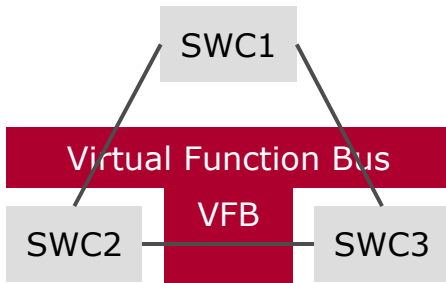
Multi-User Workflow

Webinar Series

# Introduction

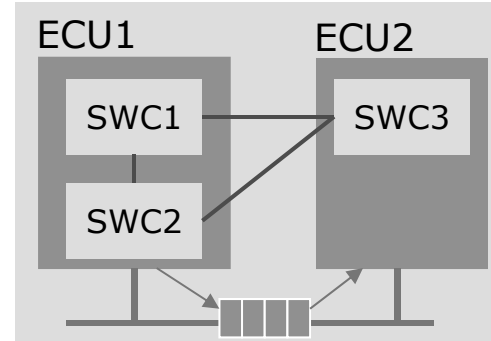
## AUTOSAR Method

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



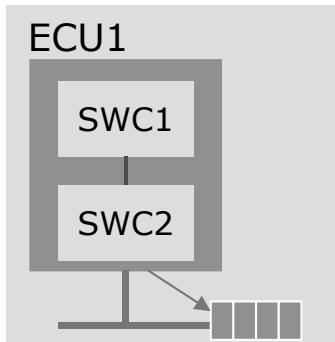
Software Component Description\*

..and distributed to ECUs

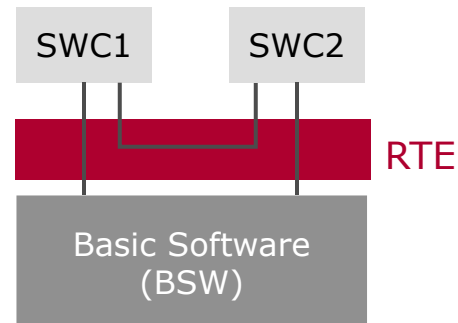


System Description\*

An extract is created for each ECU...



The ECU is configured based on the ECU Extract.



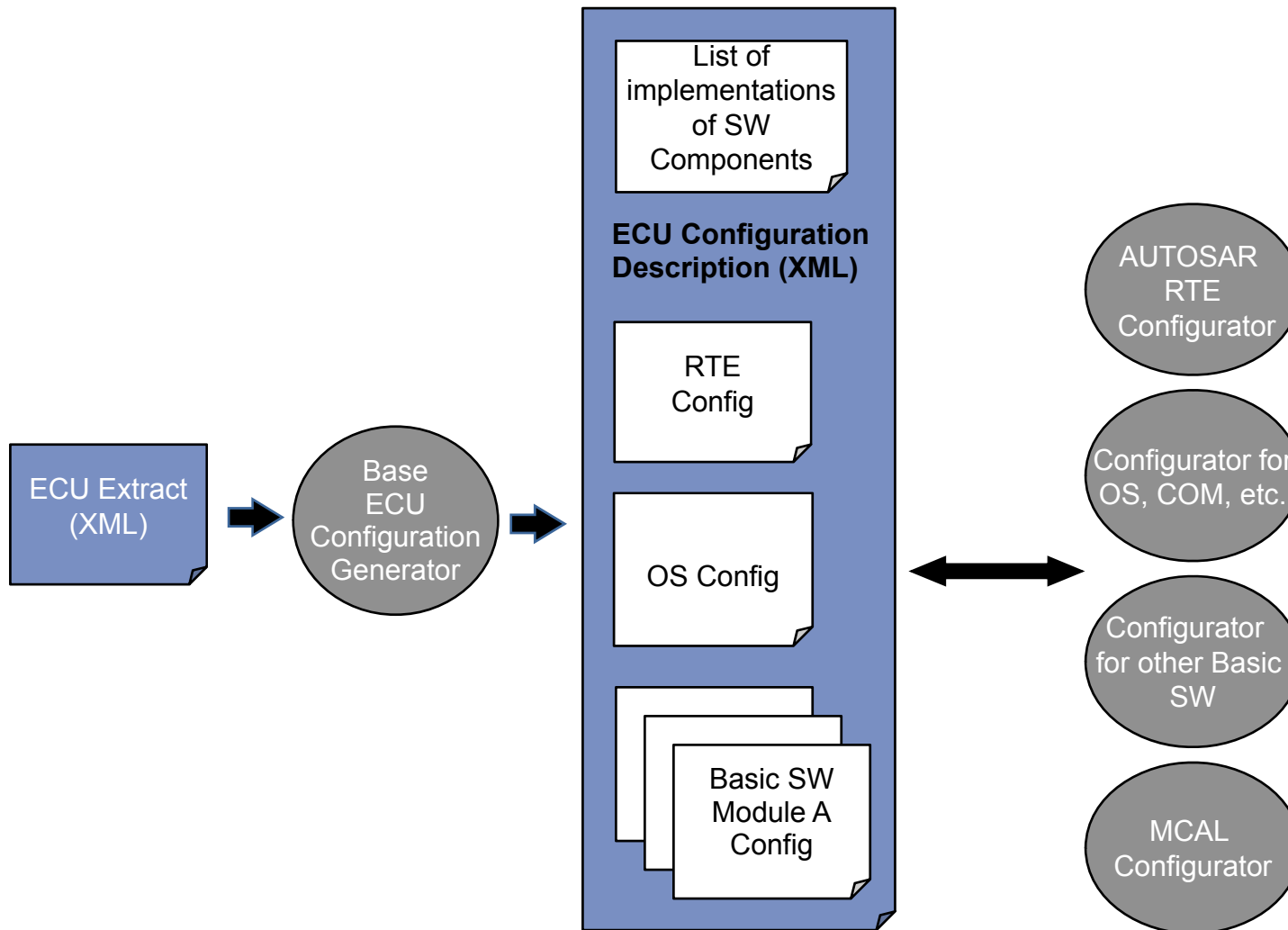
ECU Configuration Description (ECUC)\*

\* AUTOSAR

Extract of System Description\*

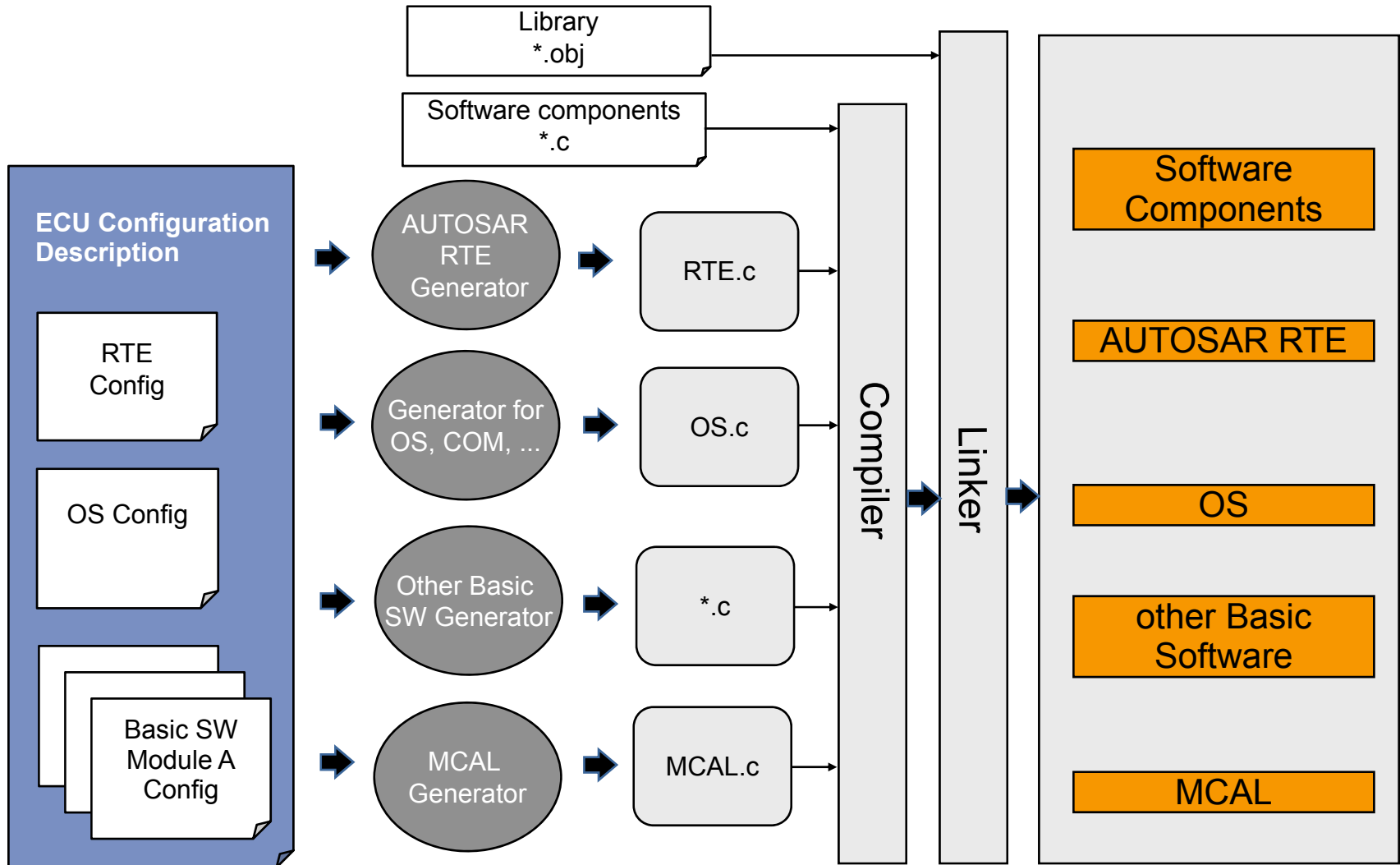
# Introduction

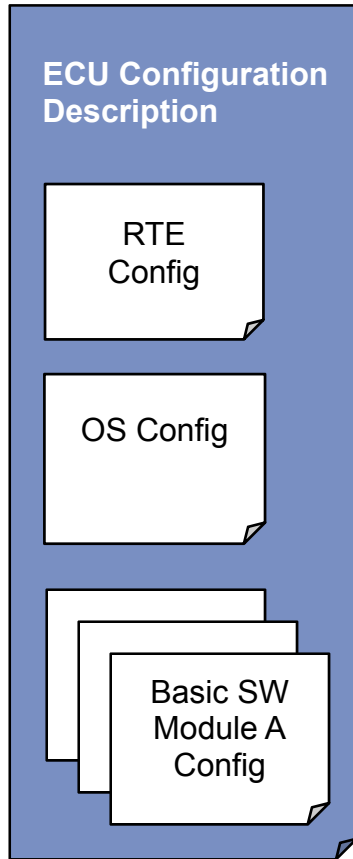
## AUTOSAR Method



# Introduction

## AUTOSAR Method





- ▶ 1000s of parameters
- ▶ Large XML file
- ▶ Realistic file size
  - ▶ Small ECU: e.g. 10 MB
  - ▶ Large ECU: e.g. 50 MB

```
<<ECU-CONFIGURATION>>
<MODULE-CONFIGURATION UID="67aee67e-12e0-4bc8-b209-628ct3b915c8">
  <SHORT-NAME=Board</SHORT-NAME>
  <DESC>
    <L-2 L="FOR-ALL">Contains all controller-specific parameters for the generator</L-2>
  </DESC>
  <DEFINITION-REF DEST="MODULE-DEF">MICROSAR/Board</DEFINITION-REF>
  <IMPLEMENTATION-CONFIG-VARIANT-VARIANT-PRE-COMPILE</IMPLEMENTATION-CONFIG-VARIANT>
  <MODULE-DESCRIPTION-REF DEST="BSV-IMPLEMENTATION">MICROSAR/Board_Impl</MODULE-DESCRIPTION-REF>
  <CONTAINERS>
    <CONTAINER UID="ddd10167-ee3c-41d1-8cc8-fbe85248e1c1">
      <SHORT-NAME=BoardGeneral</SHORT-NAME>
      <DEFINITION-REF DEST="PARAM-CONF-CONTAINER-DEF">MICROSAR/Board/BoardGeneral</DEFINITION-REF>
      <PARAMETER-VALUES>
        <FLOAT-VVALUE>
          <DEFINITION-REF DEST="FLOAT-PARAM-DEF">MICROSAR/Board/BoardGeneral/BoardOscillator</DEFINITION-REF>
          <VALUE=4</VALUE>
          <FLOAT-VVALUE>
            <BOOLEAN-VVALUE>
              <DEFINITION-REF DEST="BOOLEAN-PARAM-DEF">MICROSAR/Board/BoardGeneral/BoardAtomicBitAccessnBitfield</DEFINITION-REF>
              <VALUE=false</VALUE>
              <BOOLEAN-VVALUE>
                <ENUMERATION-VVALUE>
                  <DEFINITION-REF DEST="ENUMERATION-PARAM-DEF">MICROSAR/Board/BoardGeneral/BoardAtomicVariableAccess</DEFINITION-REF>
                  <VALUE=Atomic16BitAccess</VALUE>
                  <ENUMERATION-VVALUE>
                    <INTEGER-VVALUE>
                      <DEFINITION-REF DEST="INTEGER-PARAM-DEF">MICROSAR/Board/BoardGeneral/BoardWidthOfPointer</DEFINITION-REF>
                      <VALUE=4</VALUE>
                      <INTEGER-VVALUE>
                        <ENUMERATION-VVALUE>
                          <DEFINITION-REF DEST="ENUMERATION-PARAM-DEF">MICROSAR/Board/BoardGeneral/BoardStructAlignment</DEFINITION-REF>
                          <VALUE=Auto</VALUE>
                          <ENUMERATION-VVALUE>
                            <DEFINITION-REF DEST="ENUMERATION-PARAM-DEF">MICROSAR/Board/BoardGeneral/BoardArrayAlignment</DEFINITION-REF>
                            <VALUE=Byte</VALUE>
                            <ENUMERATION-VVALUE>
                              <ENUMERATION-VVALUE>
                                <DEFINITION-REF DEST="ENUMERATION-PARAM-DEF">MICROSAR/Board/BoardGeneral/BoardStructInArrayAlignment</DEFINITION-REF>
                                <VALUE=Byte</VALUE>
                                <ENUMERATION-VVALUE>
                                  <INTEGER-VVALUE>
                                    <DEFINITION-REF DEST="INTEGER-PARAM-DEF">MICROSAR/Board/BoardGeneral/BoardAlignmentFillPattern</DEFINITION-REF>
                                    <VALUE=0</VALUE>
                                    <INTEGER-VVALUE>
                                      <ENUMERATION-VVALUE>
                                        <DEFINITION-REF DEST="ENUMERATION-PARAM-DEF">MICROSAR/Board/BoardGeneral/BoardSizeOfPduIdType</DEFINITION-REF>
                                        <VALUE=Two_Byte</VALUE>
                                        <ENUMERATION-VVALUE>
                                          <DEFINITION-REF DEST="ENUMERATION-PARAM-DEF">MICROSAR/Board/BoardGeneral/BoardSizeOfPduLengthType</DEFINITION-REF>
                                          <VALUE=Two_Byte</VALUE>
                                          <ENUMERATION-VVALUE>
                                            <ENUMERATION-VVALUE>
                                              <DEFINITION-REF DEST="ENUMERATION-PARAM-DEF">MICROSAR/Board/BoardGeneral/BoardSizeOfEnum</DEFINITION-REF>
                                              <VALUE=One_Byte</VALUE>
                                              <ENUMERATION-VVALUE>
```

Introduction

**> AUTOSAR ECU Configuration Description**

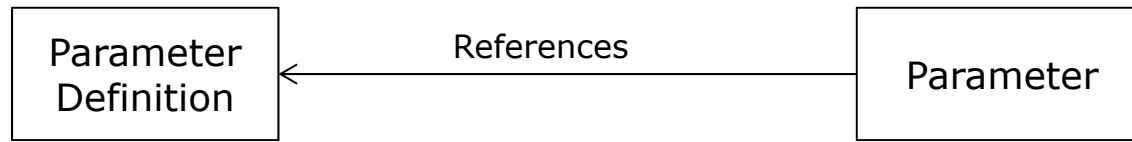
ECU Configuration Workflow

Multi-User Workflow

Webinar Series

# AUTOSAR ECU Configuration Description

## What is a parameter?



- ▶ Name
- ▶ Multiplicity (e.g. 0..1, 1..n)
- ▶ Type (integer, enum, ...)
- ▶ Min/Max values, literals
- ▶ Description
- ▶ ...

- ▶ Value of the parameter within the configuration of a concrete ECU

Example:

Name: *NvmBlockUsedCrcType*

Multiplicity: *1:1*

Type: *enum (NVM\_NOCRC, NVM\_CRC16, NVM\_CRC32)*

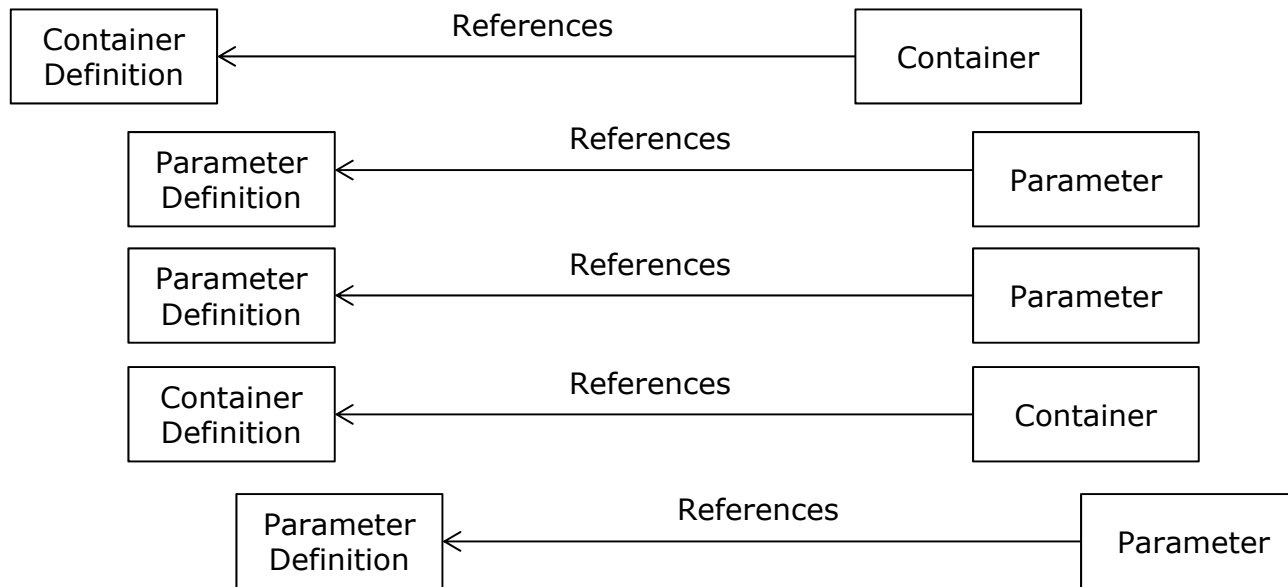
Example:

*NVM\_CRC16*

# AUTOSAR ECU Configuration Description

## Containers and Parameters

- ▶ Parameters are hierarchically organized in **Containers**
- ▶ A container consists of
  - ▶ Parameters
  - ▶ sub-containers



Example:

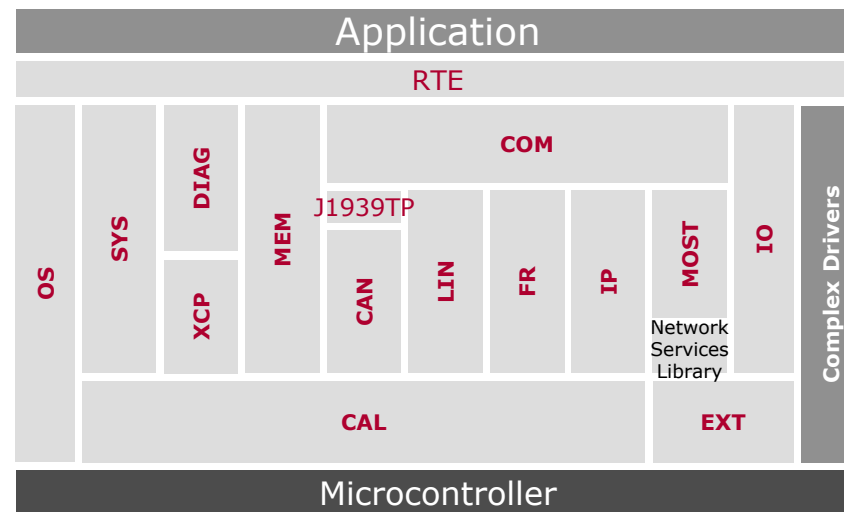
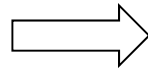
Name: *NvmBlockDescriptor*

Multiplicity: 1:65536

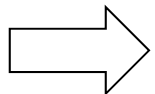
# AUTOSAR ECU Configuration Description

## Basic Software Module Description

- ▶ Basic Software Module Description (BSWMD) contains
  - ▶ Parameter definitions
  - ▶ Preconfigured values, recommended values
- ▶ XML format
  - ▶ Schema defined by AUTOSAR
  - ▶ Content defined by BSW vendor
  - ▶ May contain AUTOSAR standard parameters, and vendor specific parameters
- ▶ Typically one file per BSW module



- ▶ Degree of freedom is reduced by
  - ▶ Parameters (indirectly) given by the OEM
    - > Network communication description
    - > Parts of the diagnostic description
  - ▶ Parameters given by preconfiguration (preconfigured by the basic SW supplier)
- ▶ But
  - ▶ Still there are lots of parameters to be set
  - ▶ Additional challenges due to project situation
    - > Several developers work in parallel at the same ECU
    - > BSW is supplied by several vendors (e.g. Vector, micro-controller manufacturer)



**Key for success: configuration tools**

- ▶ Comfort Views
  - ▶ Structured display of the ECU configuration
- ▶ Assistance for creating the configuration
  - ▶ Automatic baud rate calculation, hardware masks
  - ▶ Automatic fragmentation of EEPROM blocks, ...
  - ▶ Consistency checks (e.g. is configuration of FullCAN objects consistent)
- ▶ Support for integration MCALs
  - ▶ Integration of MCAL configuration GUI and code generators
  - ▶ Migration (platform switch)
- ▶ Support for integration of add-on modules
  - ▶ Include the configuration of the add-on module to the overall ECU configuration (based on a BSWMD)
  - ▶ Ensure consistency between add-on modules and AUTOSAR standard modules

Introduction

AUTOSAR ECU Configuration Description

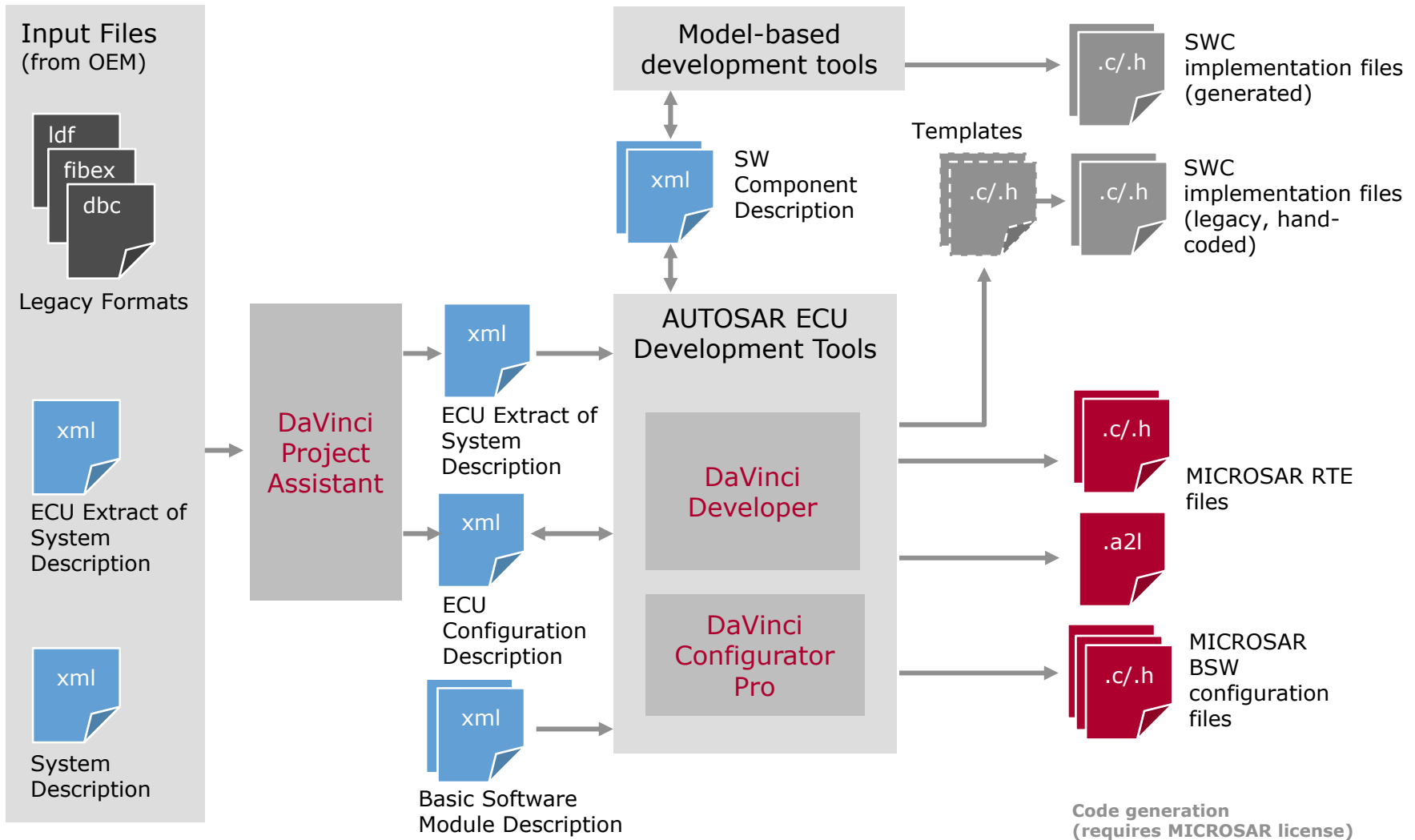
**> ECU Configuration Workflow**

Multi-User Workflow

Webinar Series

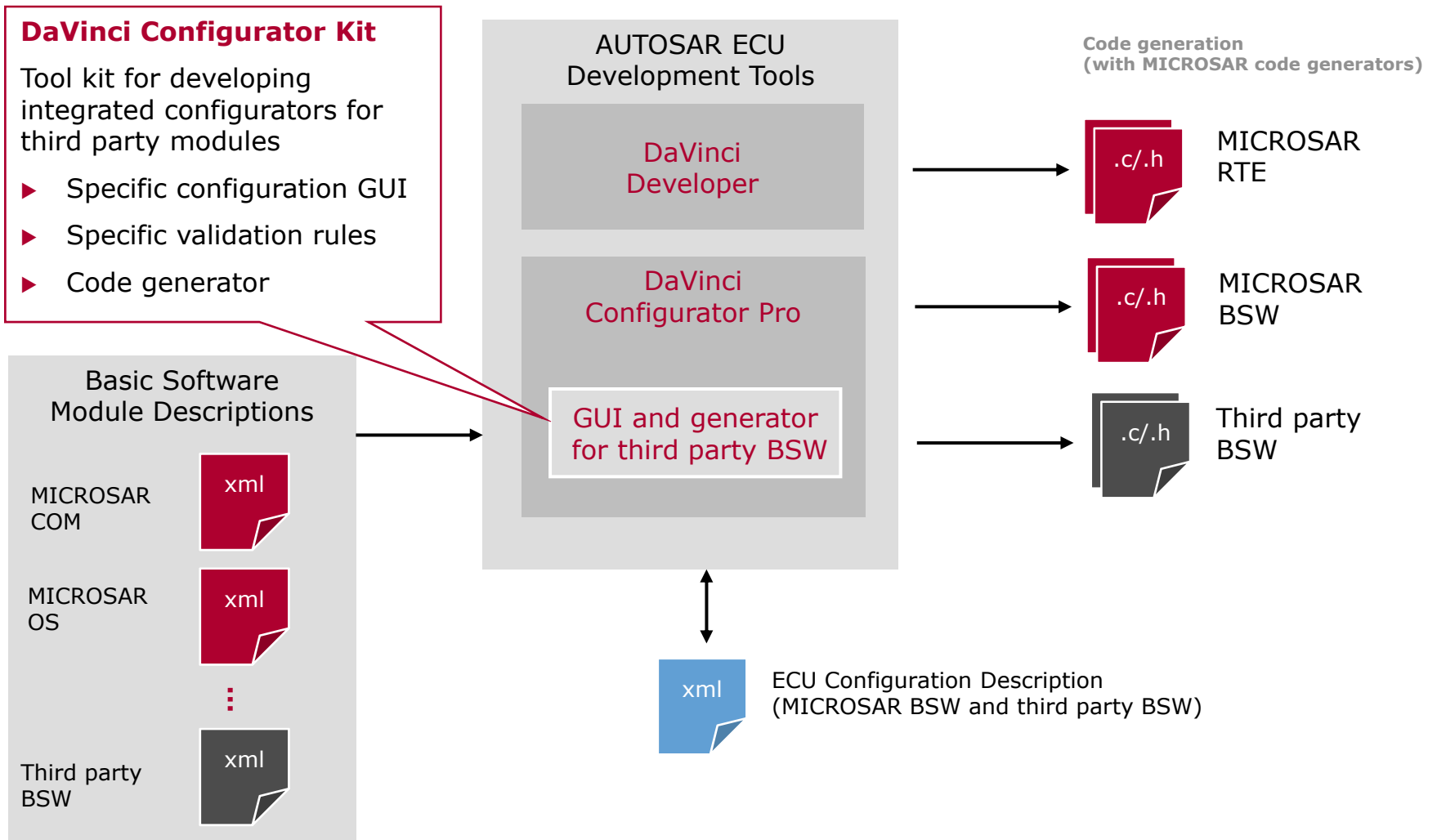
# ECU Configuration Workflow

## Overview



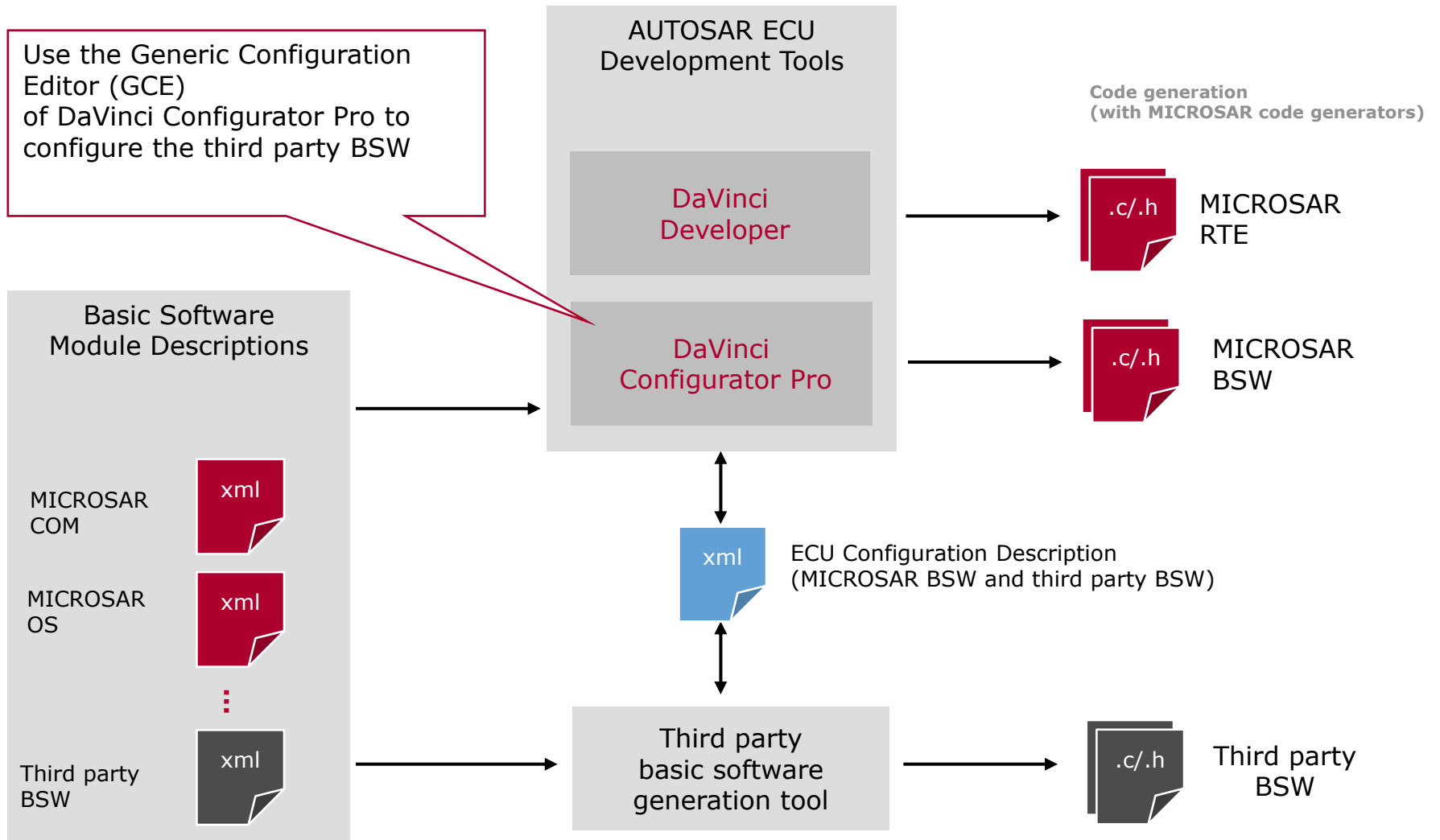
# ECU Configuration Workflow

## Integration of third party BSW (Option 1)



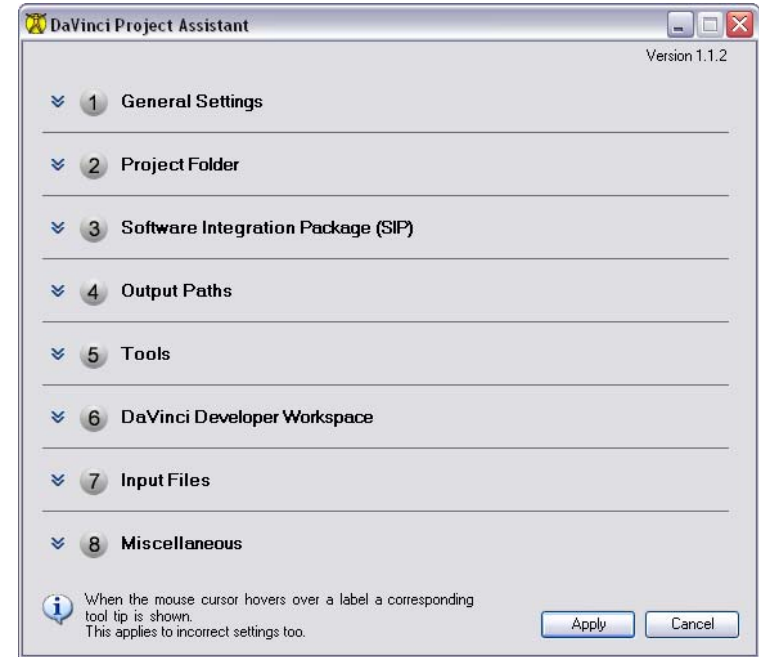
# ECU Configuration Workflow

## Integration of third party BSW (Option 2)



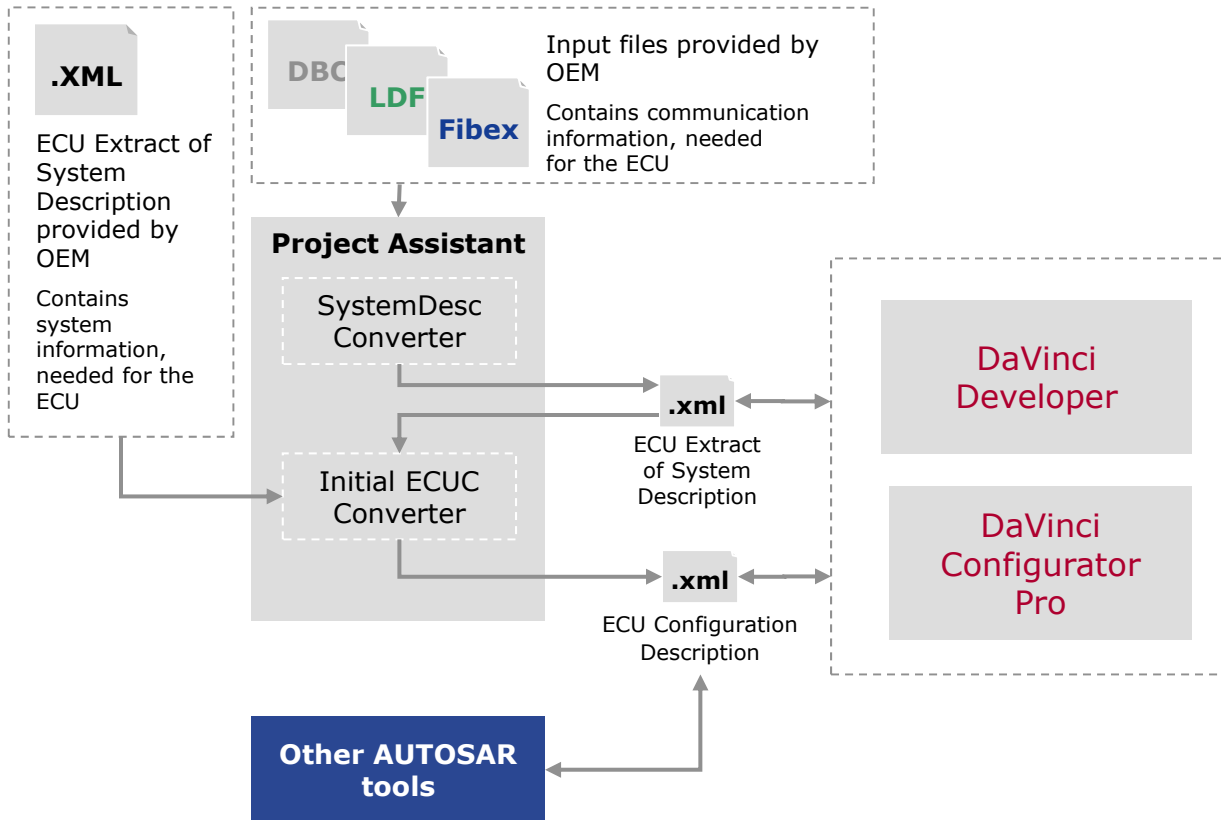
### Project Assistant

- ▶ Collects all necessary administrative information
  - ▶ Paths
  - ▶ BSW Licenses
  - ▶ Input Files
- ▶ Supports typical workflows
  - ▶ Setup of new projects
  - ▶ Automatic update of projects (e.g. new communication matrix)
- ▶ Converts dbc, fibex, ldf to AUTOSAR ECU Extract of System Description
- ▶ Creates links in Windows start menu to conveniently open the project with the DaVinci tools
- ▶ Assistant is integrated in the DaVinci tools



# ECU Configuration Workflow

## DaVinci Project Assistant



Introduction

AUTOSAR ECU Configuration Description

ECU Configuration Workflow

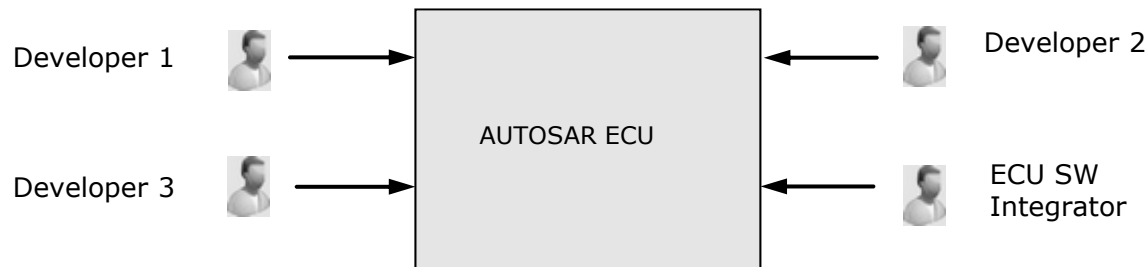
**> Multi-User Workflow**

Webinar Series

# Multi-User Workflow

## General

- ▶ Several developers are working in parallel at the same AUTOSAR ECU
- ▶ Each developer must be able to make changes independently from other developers (no blocking time accepted)
- ▶ Integration effort must be small
- ▶ Configuration artifacts (AUTOSAR XML files) must be managed consistently with other artifacts of ECU development (code files, specification documents, test environments, ...)



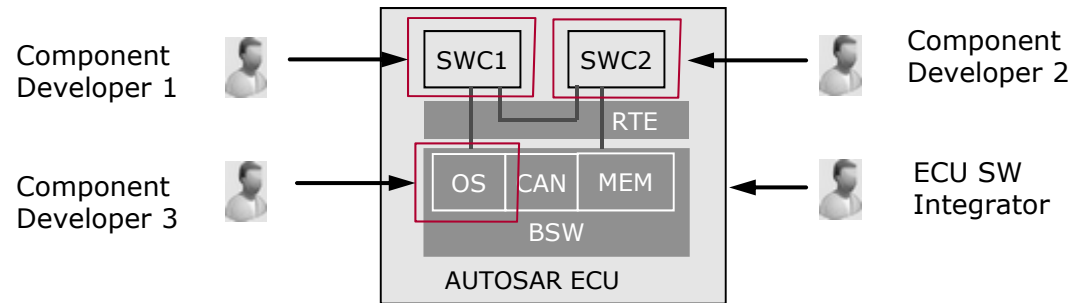
# Multi-User Workflow

## Responsibilities

Several possible ways to organize responsibilities, e.g.

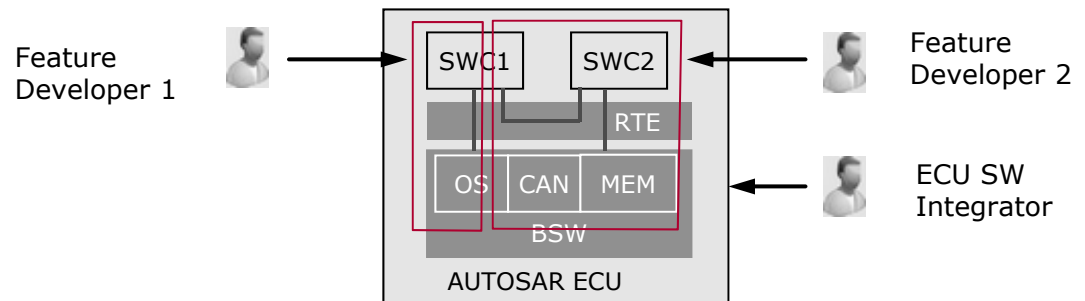
### Component responsables

- ▶ Component responsables are assigned to individual components
- ▶ Software Components (SWC), Basic Software (BSW) modules or clusters



### Feature responsables

- ▶ Feature developer makes changes in potentially all SWCs or BSW modules
- ▶ Overlapping modifications – several users change e.g. same SWC
- ▶ No strict assignment of user to SWC/BSW-Module



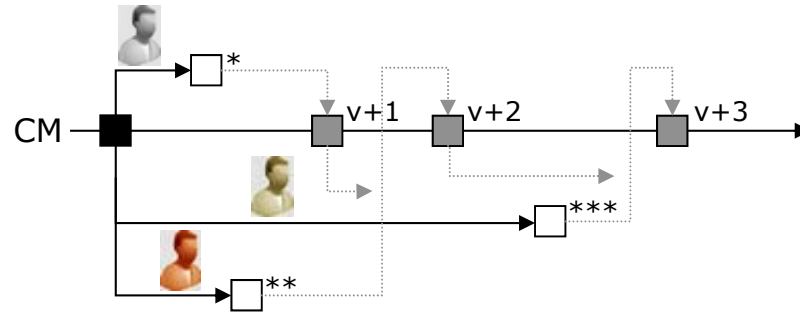
# Multi-User Workflow

## Access Strategy

Two general approaches regarding development process/access strategy

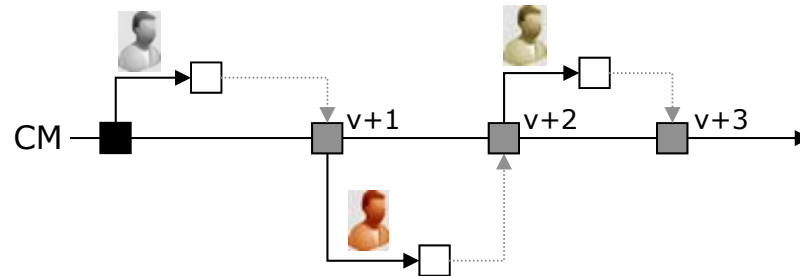
### Concurrent editing

- ▶ Several persons change in parallel the same objects (e.g. SWCs or BSW module configurations)
- ▶ Typically realized via branches in a configuration management (CM) system
- ▶ Requires diff/merge



### Exclusive editing

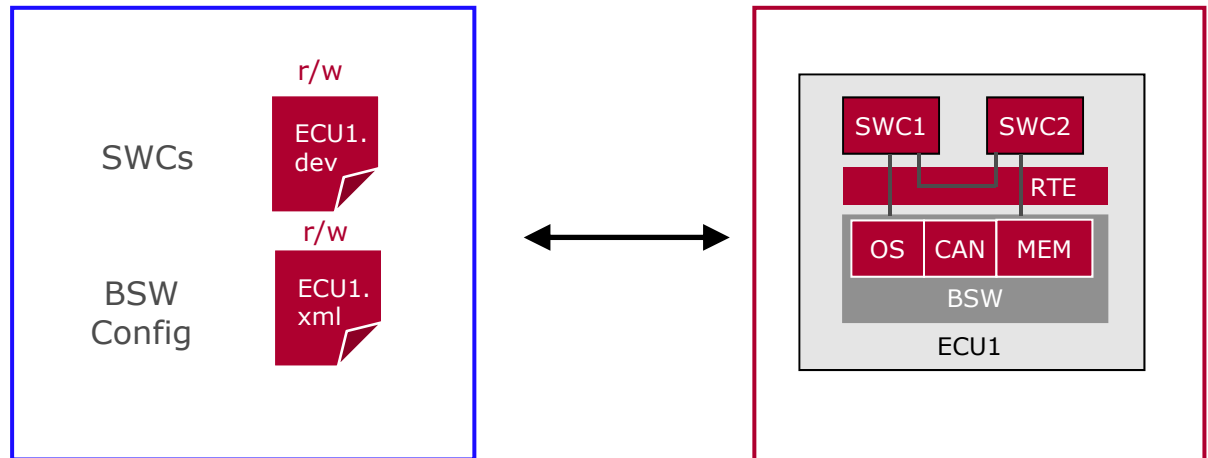
- ▶ Only one person allowed to change an object
- ▶ Typically realized via check-out control strategy of CM system
- ▶ Requires sequencing of work packages – difficult to organize!
- ▶ Avoids merge



# Multi-User Workflow

## File Structure

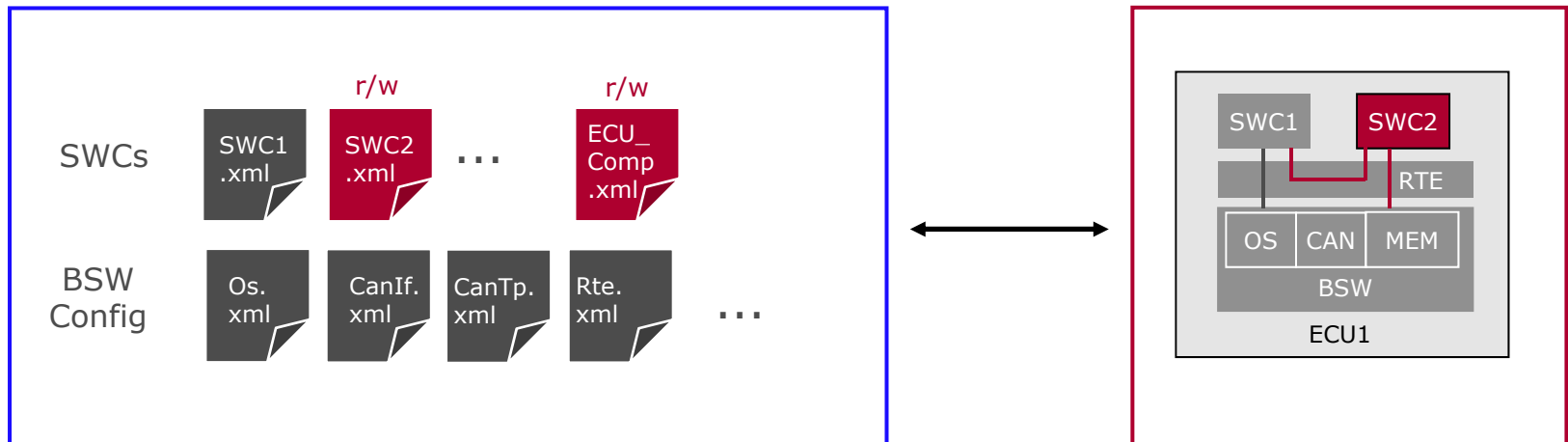
- ▶ Granularity Option 1: Monolithic
  - ▶ All SWCs of the ECU within one file (DEV workspace)
  - ▶ All BSW module configurations within one ECUC file



# Multi-User Workflow

## File Structure

- ▶ Granularity Option 2: One file per module
  - ▶ Separate file for each SWC of the ECU (DCF Workspace)
  - ▶ Separate ECUC split file for each BSW module configuration of the ECU



- ▶ 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 AUTOSAR Testing
  - ▶ 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_webinars_en.html)
  
- ▶ The overview of Vector's training services:  
[http://www.vector.com/vi\\_training\\_en.html](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 :
  - ▶ [embedded@de.vector.com](mailto:embedded@de.vector.com)

Thank you for your attention.

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

[www.vector.com](http://www.vector.com)

Author:

Vector Informatik GmbH

Ingersheimer Str. 24

70499 Stuttgart