



Processes and tools for the development of modular embedded automotive software

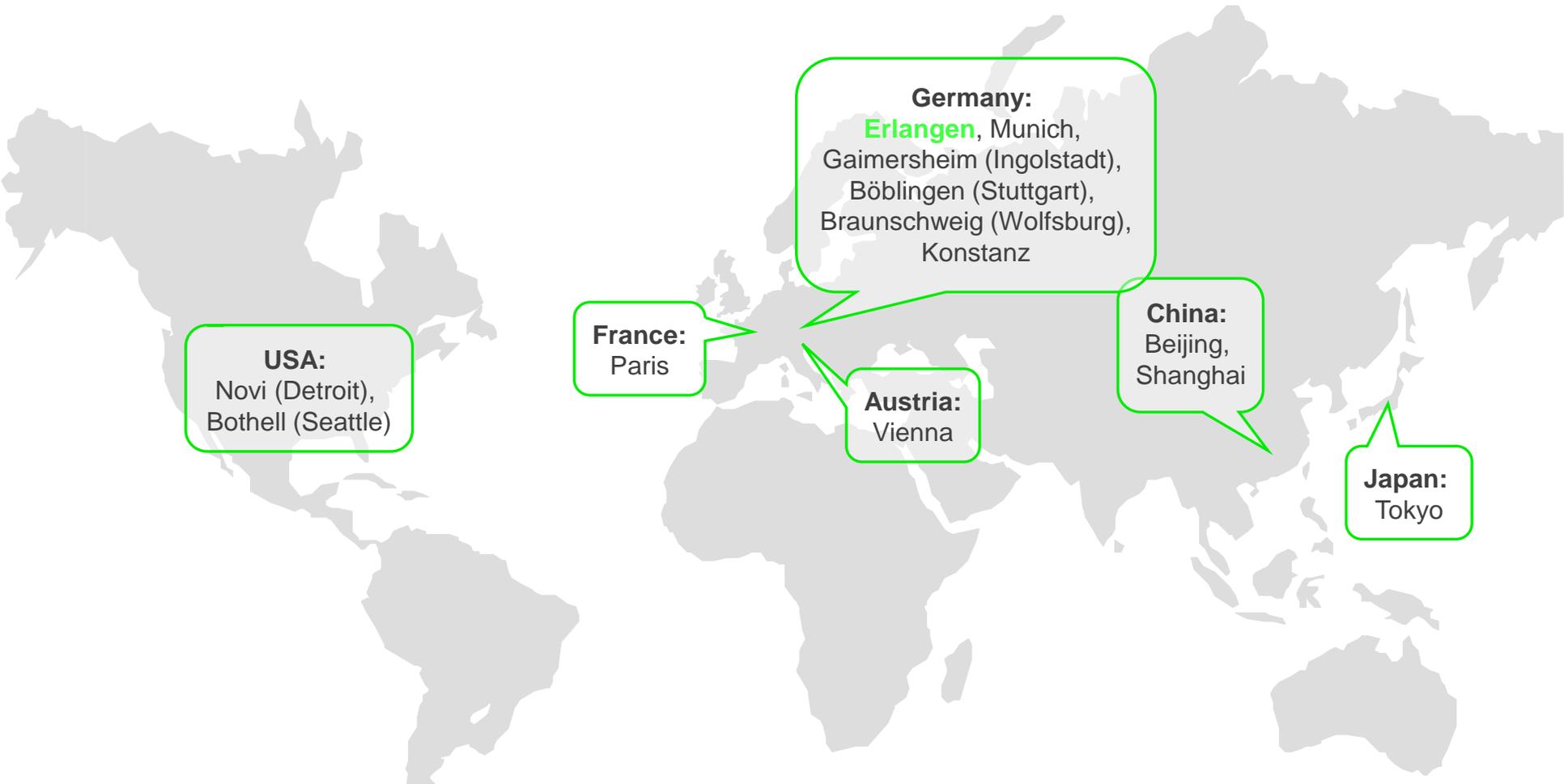


Agenda

- Company overview
- Automotive HMI development
- “Under the hood” software development
- Conclusions/Q&A

EB Automotive Software

Engineering competence close to our customers



Success Story

OSEK and AUTOSAR compliant standard core for BMW



The Company

Headquartered in Munich, Germany, the BMW AG is one of the major manufacturers of performance and luxury cars. It owns and produces the MINI brand, and is the parent company of Rolls-Royce Motor Cars. BMW produces more than one million cars per year.

The Challenge

BMW started using standard software in series production as early as 2001. The Standard Core includes up to 56 basic software modules. In addition, BMW substituted selected standard modules with ones developed on their own. BMW searched for a company to take over the integration work.

The Solution

EB has become the main supplier of the BMW Standard Core in 1997. EB has developed most of basic software and has integrated the complete Standard Core versions SC3, SC4, SC5, SC6 and BAC2.1 on a great variety of hardware platforms.

The Benefit

EB's expertise in OSEK and AUTOSAR software development has helped make efficient project specific integrations and portings of the BMW Standard Core.

Success Story

Modular infotainment system at AUDI



The Company

Headquartered in Ingolstadt, Germany, AUDI AG is one of the major manufacturers of luxury cars and is a 99%-owned subsidiary (as of 2004) of Volkswagen, the largest car manufacturer in Europe. Audi produces more than 960,000 cars per year with several renowned models in its line-up of A-series cars, which include the A4, A6, and A8.

The Solution

In successful co-operation, Audi Electronics Venture GmbH (AEV) and EB have developed jointly the truly modular infotainment system for the Audi A5, A6 and A8 as well as the Audi Q5, which was introduced in fall 2008 to the public.

The Challenge

Beginning of 2005 Audi started off their latest generation of the Multi-Media technology (MMI 3G) and sought a development partner who not only had a deep understanding of Automotive user interfaces but who would also manage the powerful project with many different country, feature and display variants and over 1400 different user dialogs.

The Benefit

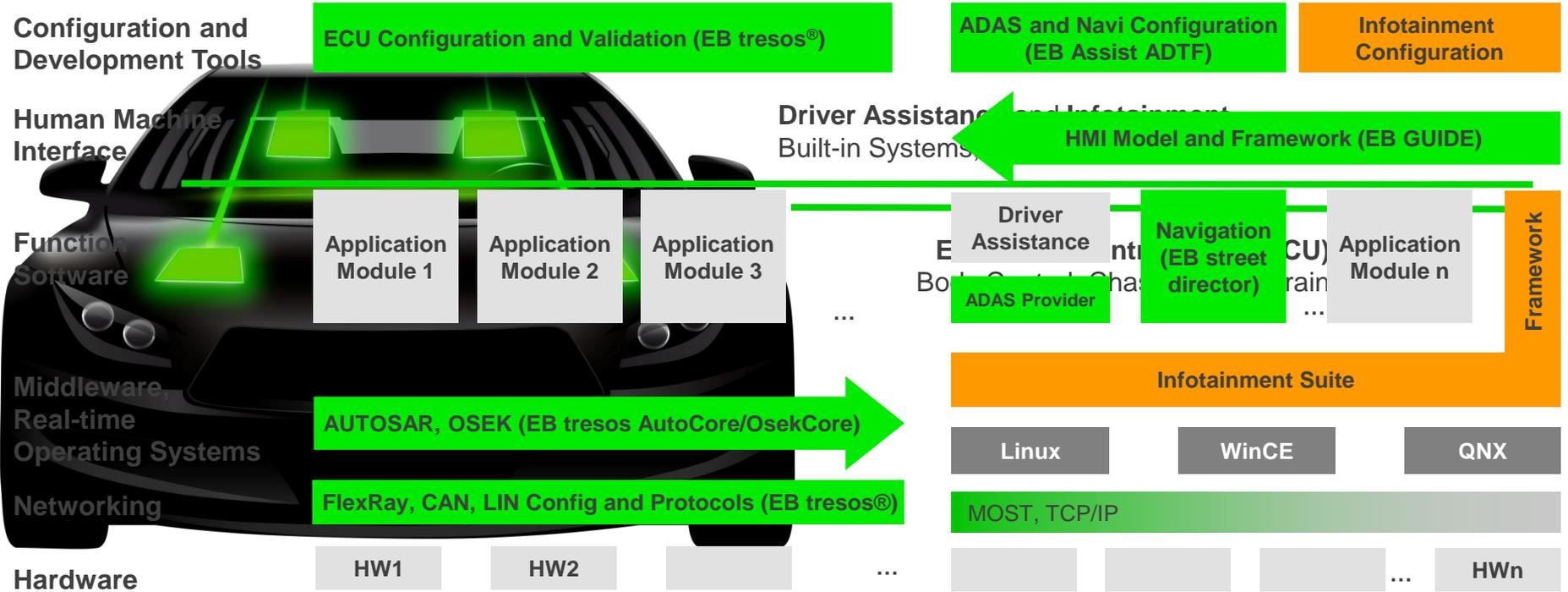
Audi benefited from the customized system which at the same time features an extremely high reusability through modularity and usage of open standards, e.g. OSGi. Additionally, EB's integrated HMI development tool EB GUIDE with automatic code generation helped to manage complexity.

Trends in Automotive Embedded Software Development

- Dealing with increased complexity
- Dealing with increased legislation
- How do we get better products, faster, at a lower cost?

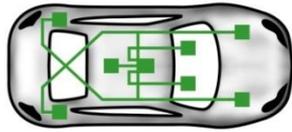


EB Automotive Software



Solutions for the automotive world

ECU Software



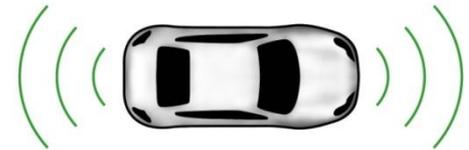
- EB provides software modules for BMW standard core
- Leading AUTOSAR technical partner for JasPar in Japan
- Supplier of FlexRay solutions for BMW X5

Infotainment Software

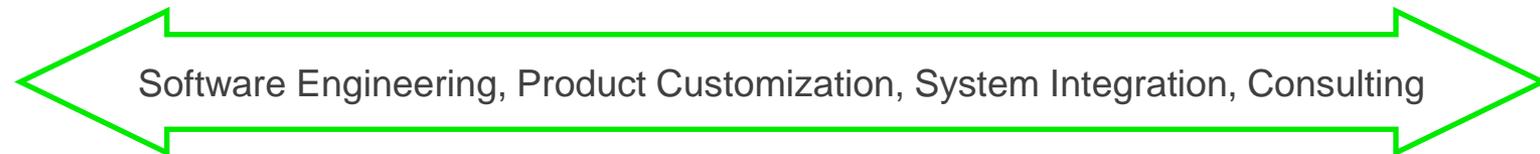


- Audi A6 and Q7 Generic User Interface developed by EB
- Navigation and speech dialog software for the new Audi A1

Driver Assistance



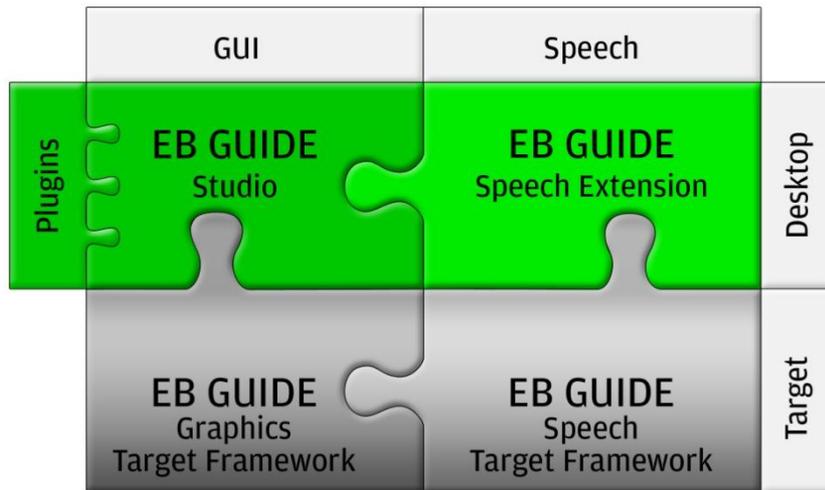
- ADTF distribution partner for AEV (Audi Electronics Venture)
- Advance Driver Assistance development competenc





Infotainment Software

EB GUIDE – One product line for HMI development



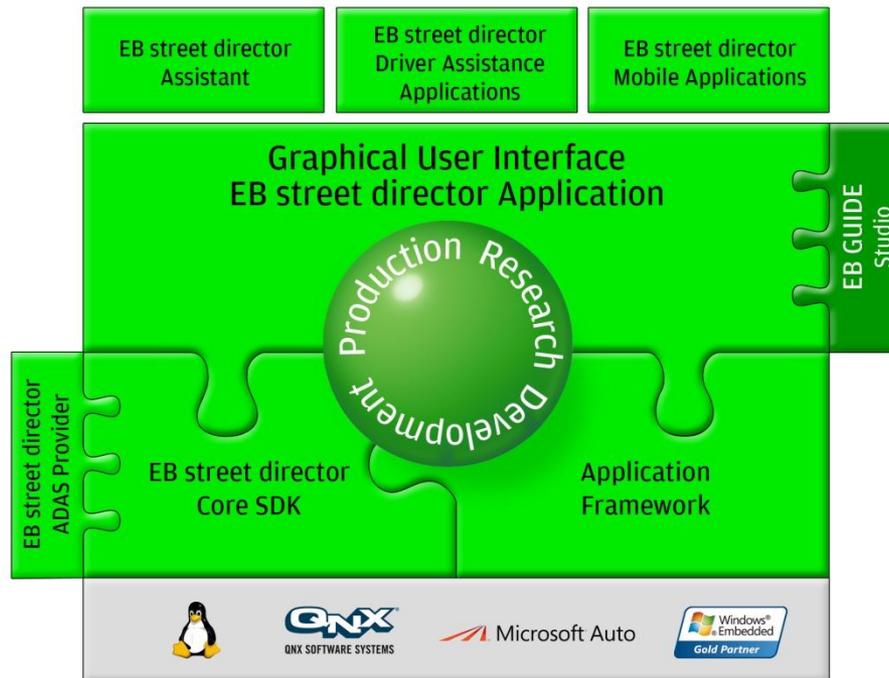
- **EB GUIDE Studio**
Tool for graphical HMI development
- **EB GUIDE Speech Extension**
Tool add-on for speech dialog design
- **EB GUIDE Speech Target Framework (STF)**
Runtime for speech dialog, connecting to Automatic Speech recognition and Text-to-Speech engine
- **EB GUIDE Graphics Target Framework (GTF)**
Runtime for graphical user interface
- **EB GUIDE Studio Plugins**
Functional extensions of EB GUIDE standard tooling





Infotainment Software

EB street director – Versatile navigation software



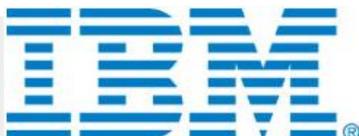
- EB street director components
- EB GUIDE
- 3rd Party

- **EB street director Application**
Complete navigation solution
- **EB street director Core SDK**
Navigation development environment
- **EB street director Assistant**
Connected content manager
- **EB street director ADAS Provider**
Map based electronic horizon
- **Driver Assistance Applications**
Extensions for EB street director
e.g. curve warning
- **Mobile Applications for EB street director**
e.g. picture viewer, MP3 and video player



Agenda

- **Process overview**
- Define AUTOSAR software application architecture, design and implementation using Rational Rhapsody
- Configure and generate AUTOSAR ECU BSW with EB tresos Studio/AutoCore



Industry solution: Automotive Open System Architecture (AUTOSAR)

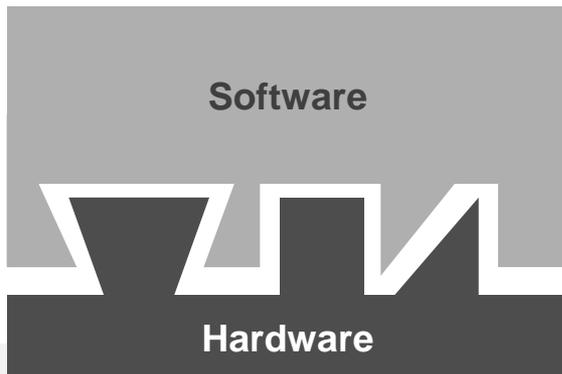
An open and standardized automotive software architecture, jointly developed by automobile manufacturers, suppliers, and tool developers

Goals

- **Standardization of basic system functions** as an OEM wide “standard core” solution
- **Scalability** to different vehicle and platform variants
- **Transferability** of functions throughout the network
- **Integration** of functional modules from **multiple suppliers**
- Higher penetration of **commercial-off-the-shelf software and hardware** components across product lines

Challenge: Conventional proprietary interface

Expensive and repetitive integration effort for common non-differentiating functionalities

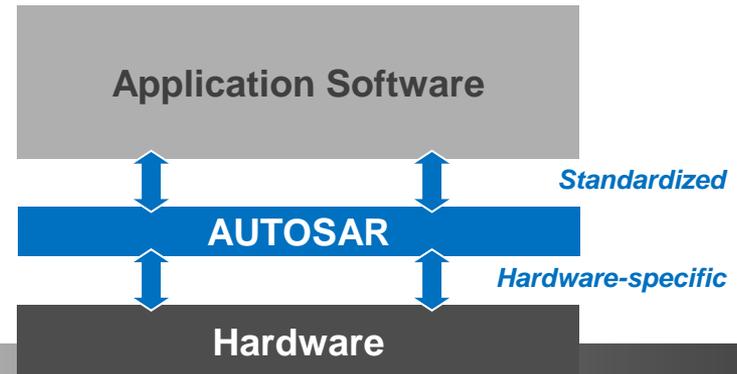


Industry impact

- Software will be largely independent of the microcontroller chosen
- Development processes will be simplified
 - Reduced development time and costs
- Reuse of software increases at OEM as well as at suppliers
 - Enhanced quality and efficiency

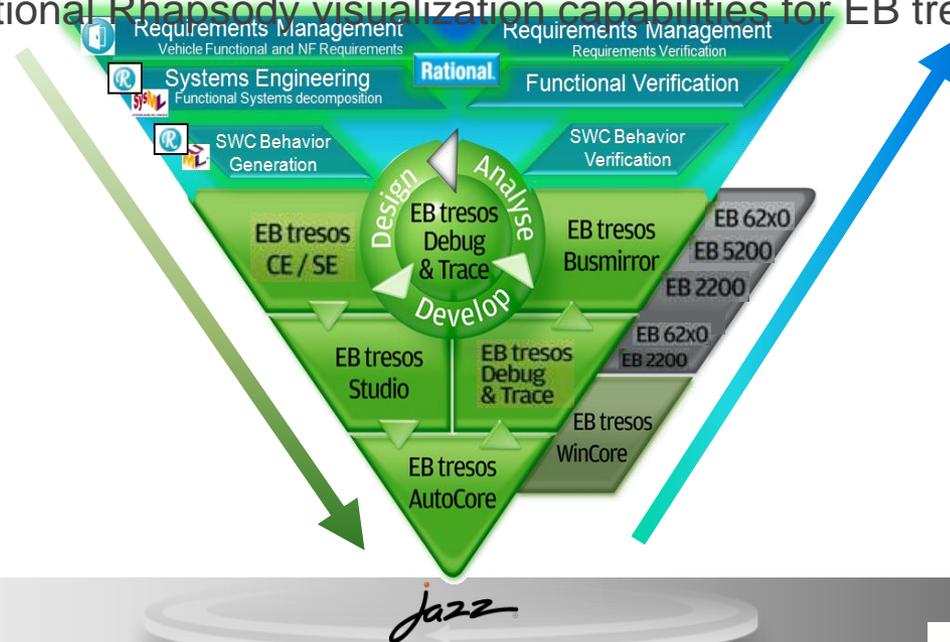
Solution: AUTOSAR standardized interface

Modular, scalable, transferable, and re-usable throughout the industry



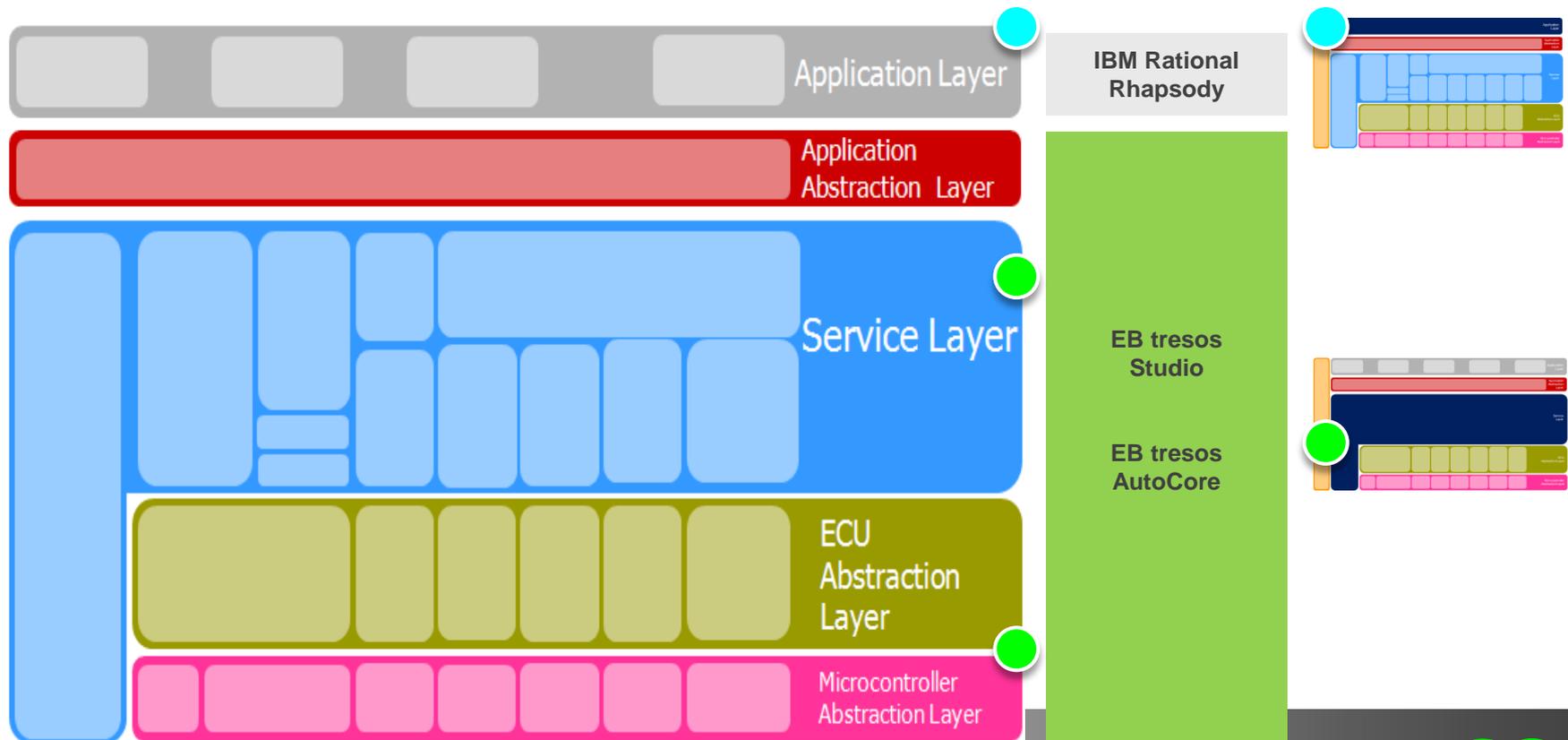
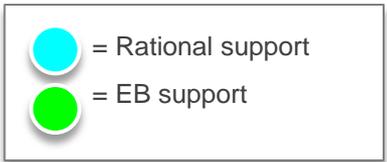
Providing tools based on open standards

- Connect ECU RTE and BSW configurations to system requirements and application models
 - Avoiding costly manual linkages for demonstrating traceability
- Leverage integrations to change, software configuration , asset , and quality management
 - Establishing repeatable, automated and documented workflows
 - Improving collaboration amongst teams and increasing efficiency
- Adding IBM Rational Rhapsody visualization capabilities for EB tresos simulation



AUTOSAR's layered architecture requires dedicated tools

IBM Rational software and Elektrobit provide best of breed tools



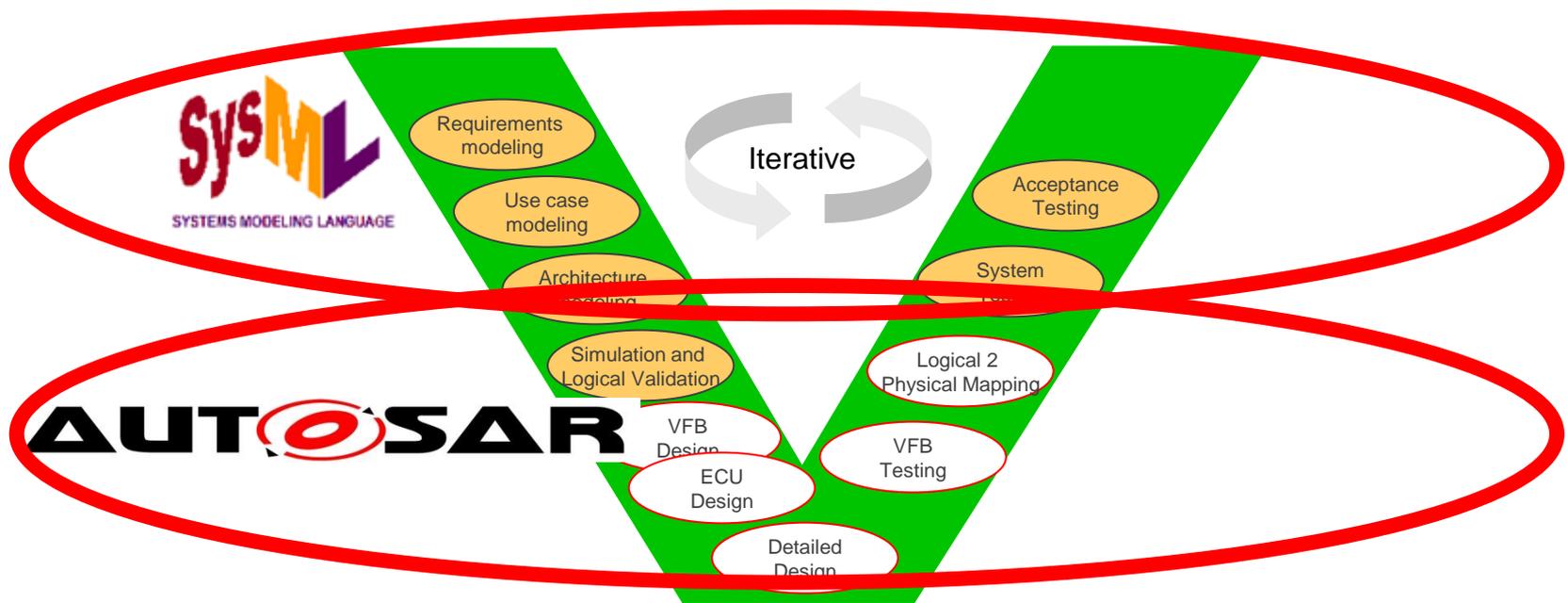
Agenda

- Process overview
- **Define AUTOSAR software application architecture, design and implementation using Rational Rhapsody**
- Configure and generate AUTOSAR ECU BSW with EB tresos Studio/AutoCore

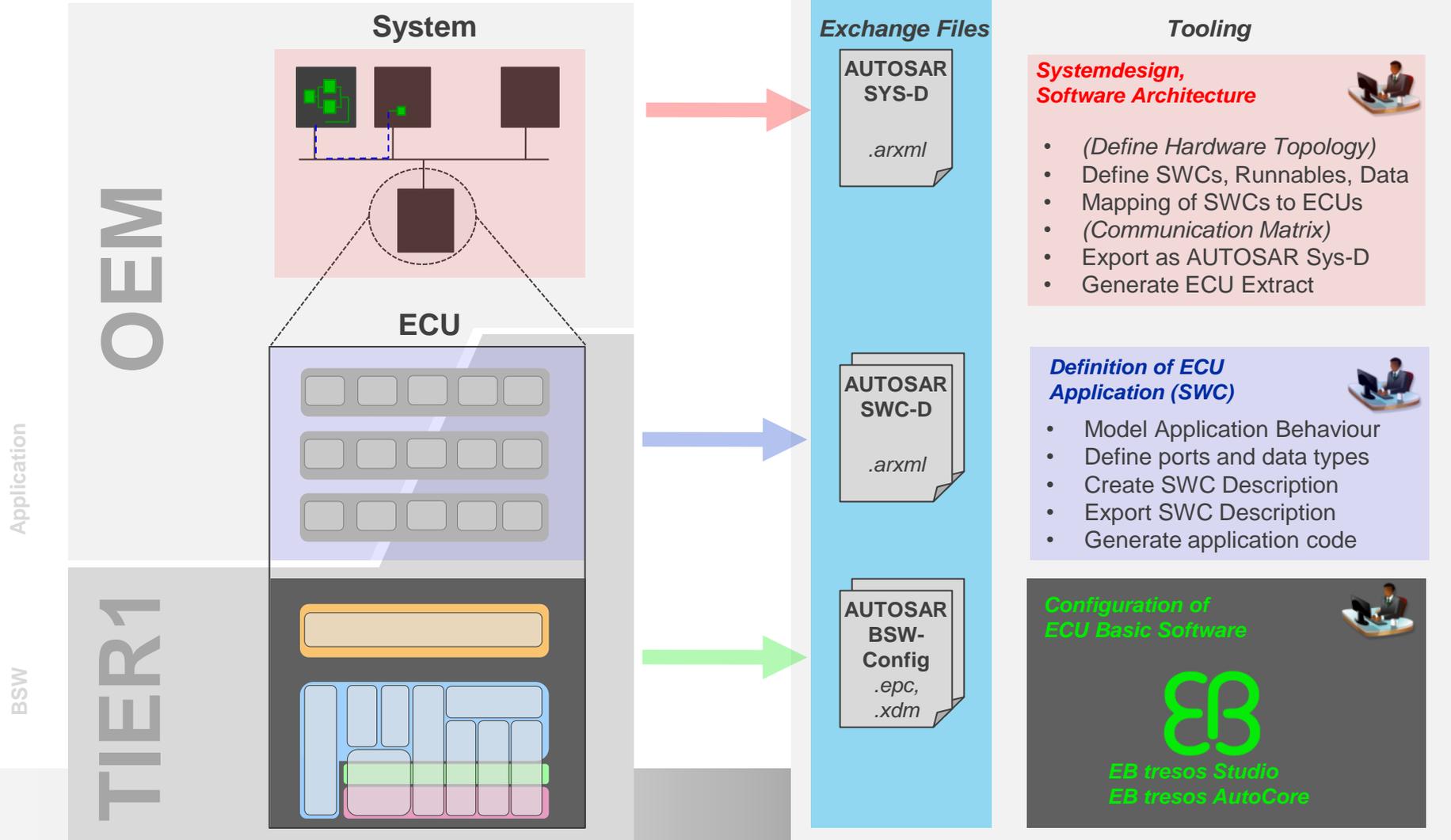


Systems Development – Where AUTOSAR & SysML Fits

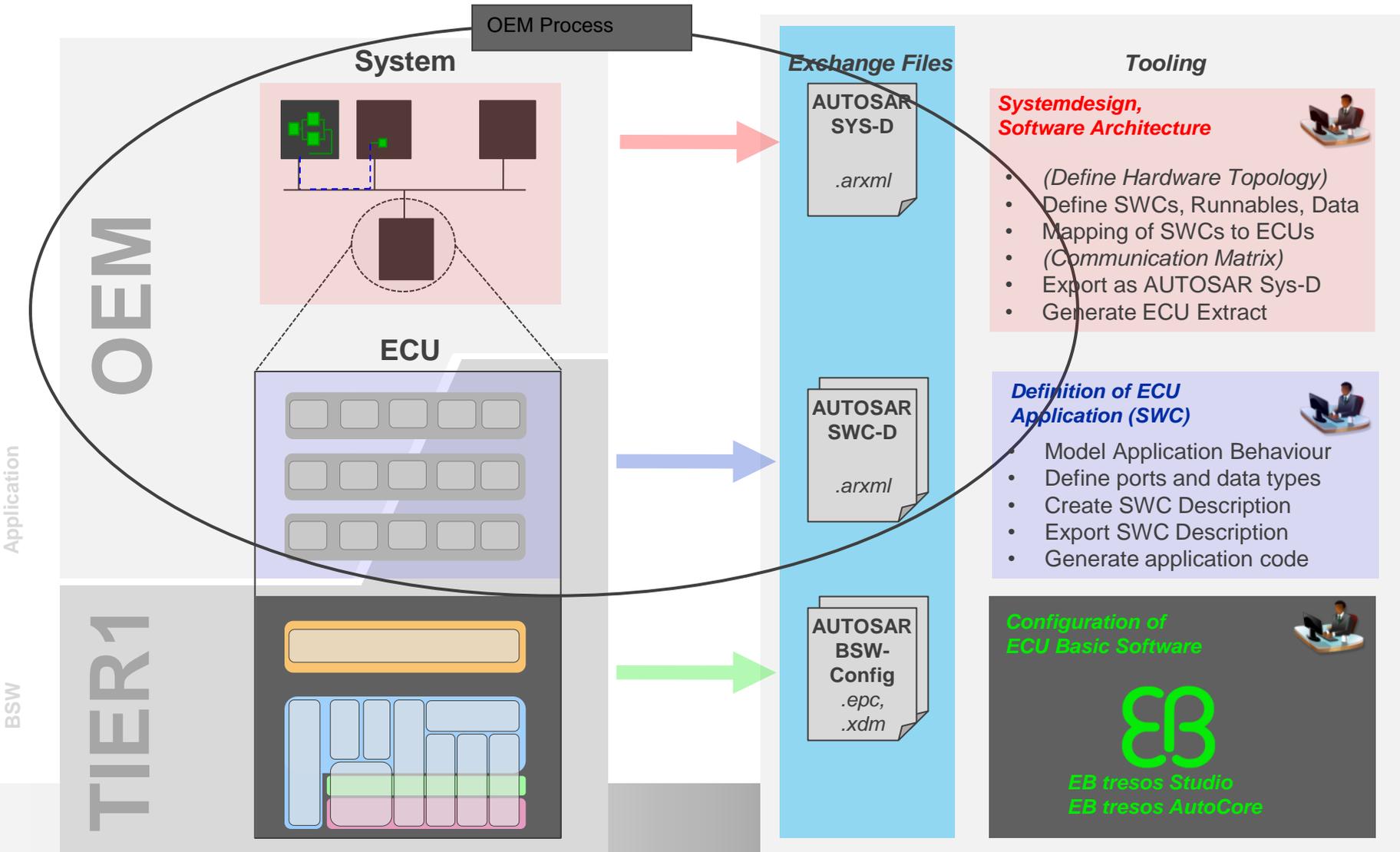
- SysML Provide Analysis Focus
- AUTOSAR Provide Design & Implementation Focus



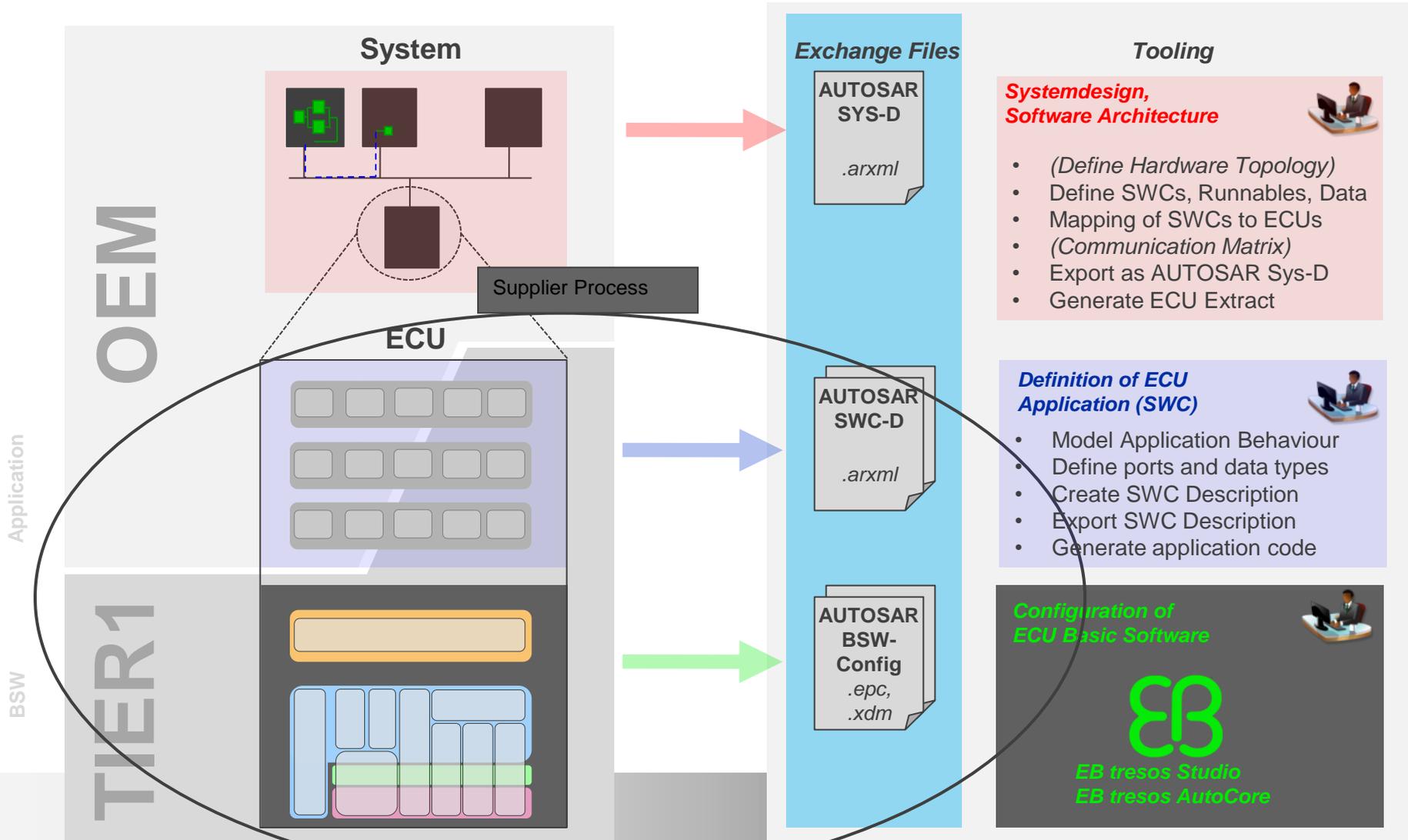
AUTOSAR Methodology



AUTOSAR Methodology



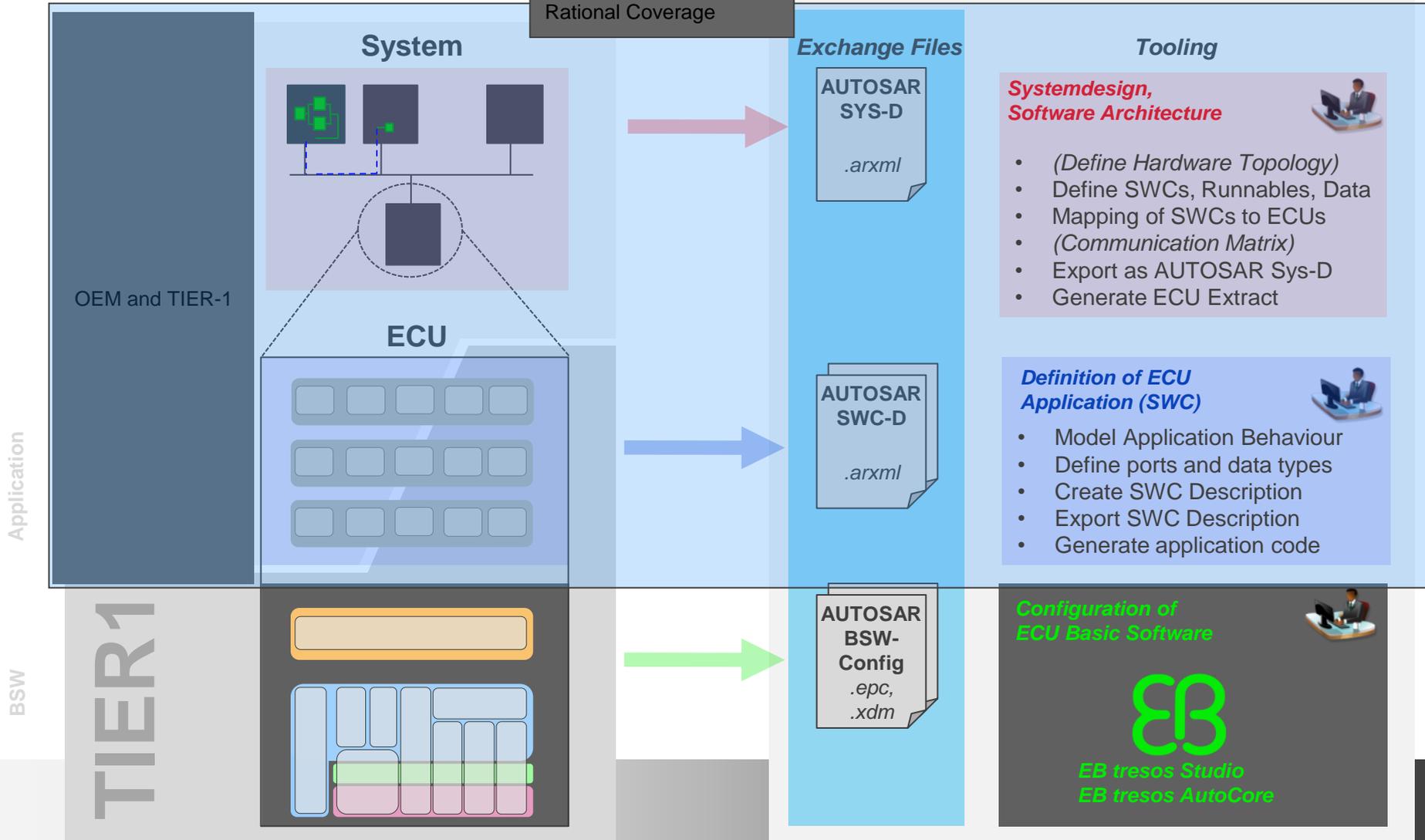
AUTOSAR Methodology



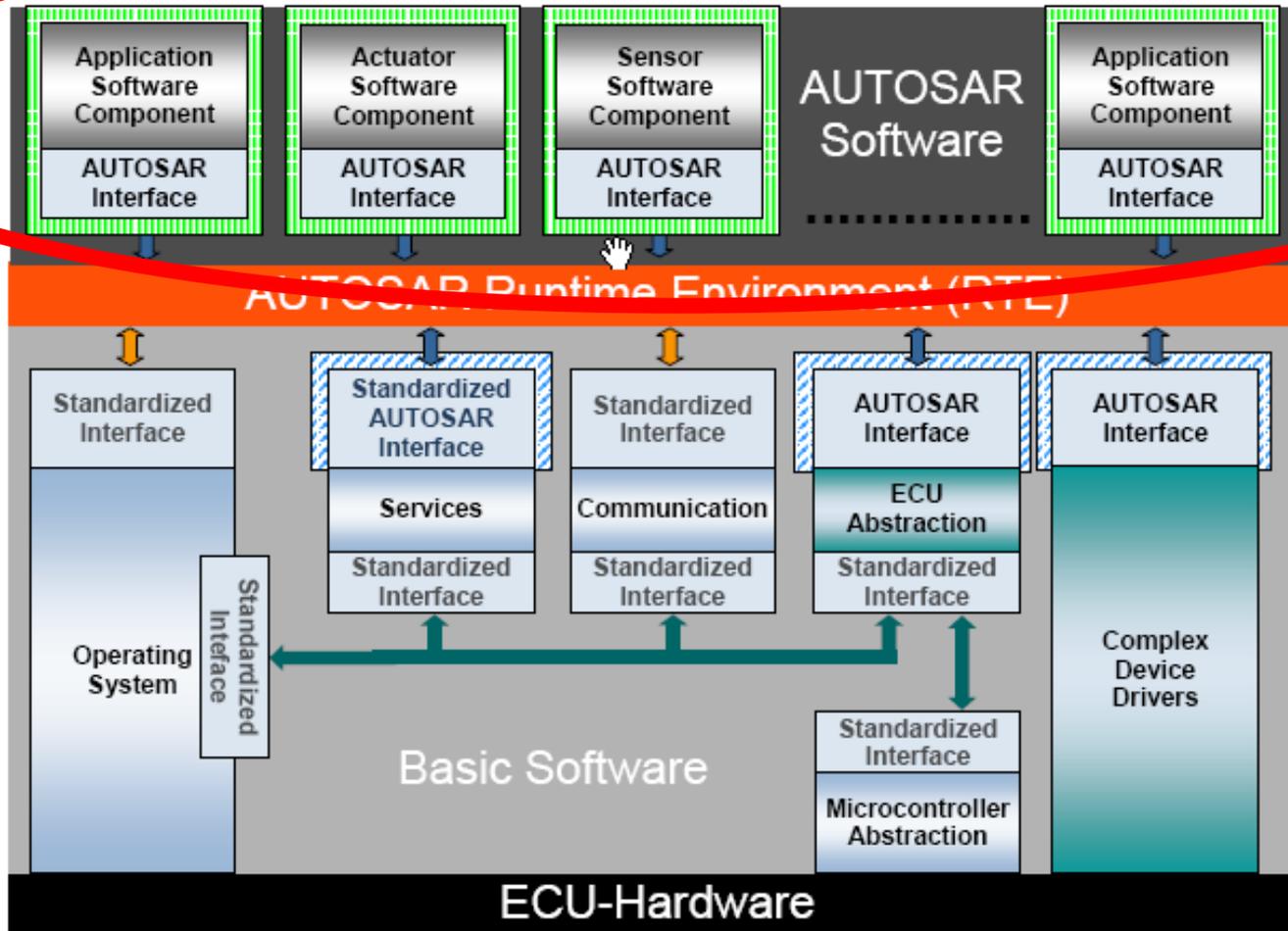
AUTOSAR Methodology



Rational Coverage



AUTOSAR ECU Architecture



The software component template describes these components completely



Of these software components only the AUTOSAR Interface side can be fully described in the software component template

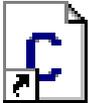


AUTOSAR Modeling Tools

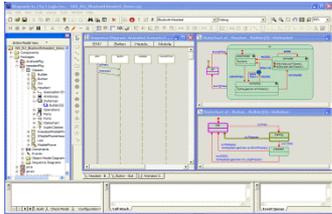
- AUTOSAR System Authoring
 - Defines the architecture and integrates the algorithms
- AUTOSAR Behavioral Modeling Tools (BMT Implementation)
 - Implement the algorithms that plug into the architectures



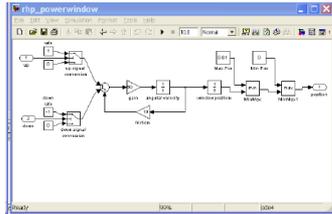
C Code BMT



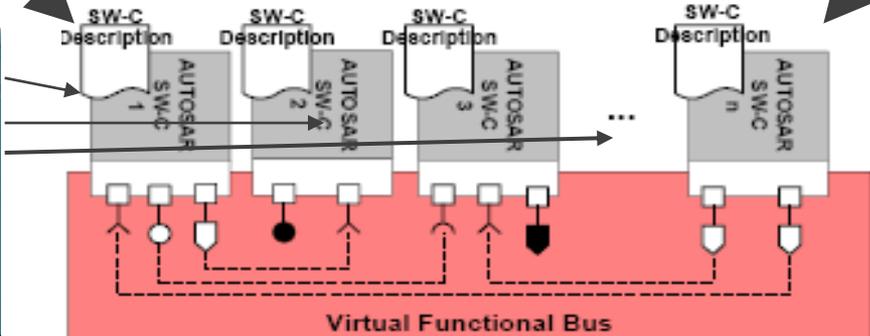
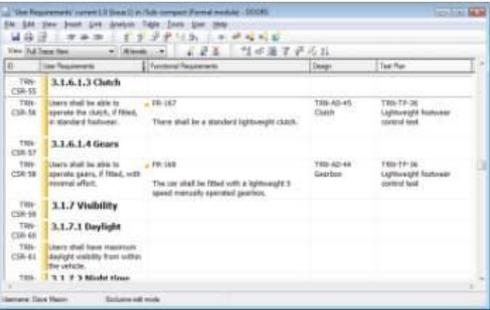
My User Code



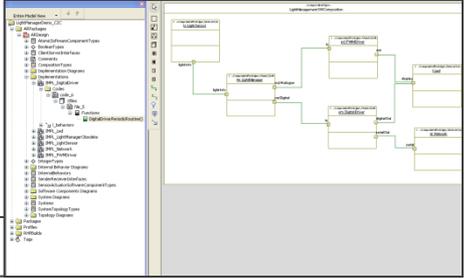
Simulink BMT



DOORS



Rhapsody Systems Modeling

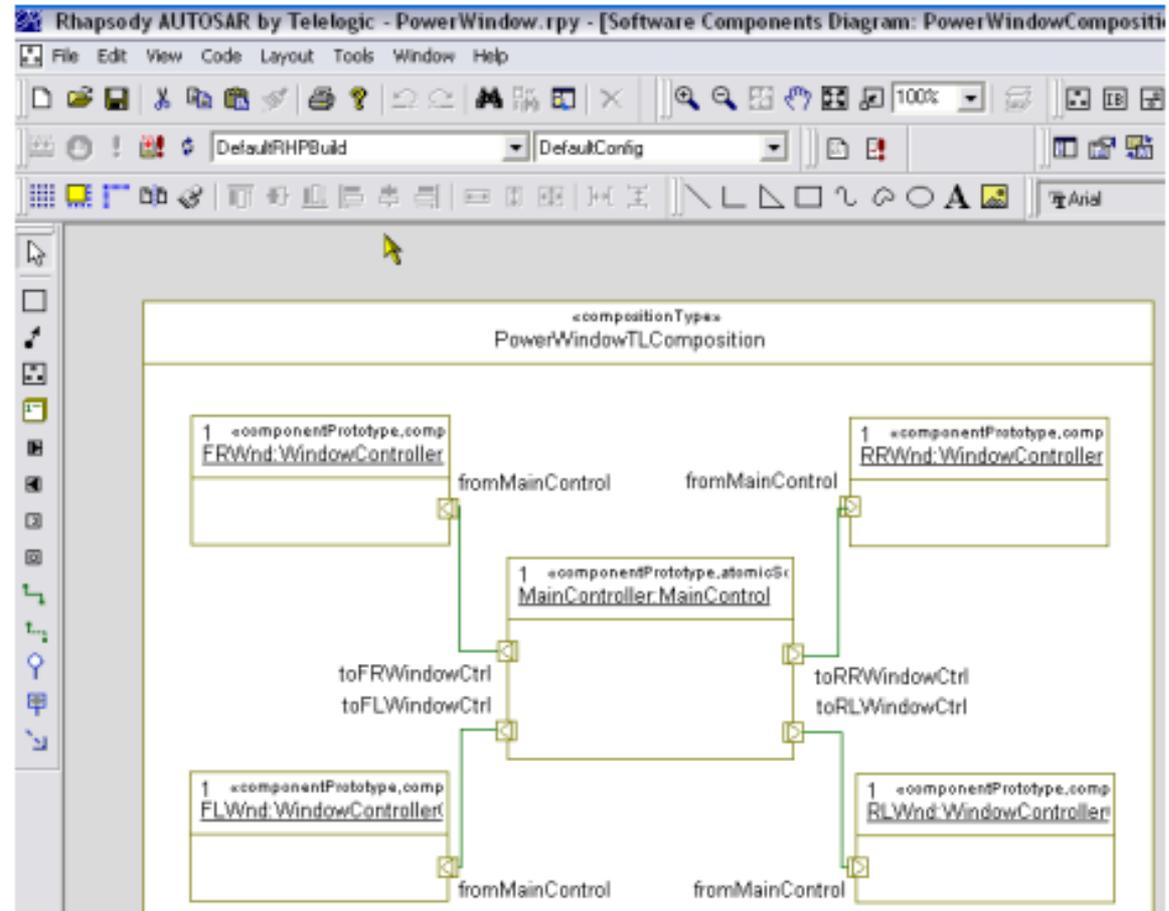


Rhapsody Systems Modeling : Award-Winning Solution!



Software Component Diagram

- Defines the software architecture of the functional system



Example 1: AUTOSAR SW-C Port Binding Table

From Component Protot...	From Port	From Port Interface	To Component ...	To Port	To Port Interface	Connector Type
ECU1Application	SensLPFiltered	SensLPFiltered_3	ECU1SW	SensLPFiltered	SensLPFiltered_2	DelegationConnectorPrototype
ECU1ApplicationSens	SensDigital	SensDigital				
ECU1ApplicationSens	SensRawValue	SensRawValue				
ECU1SW	SensLPFiltered	SensLPFiltered_2	ECU2SW	SensLPFiltered	SensLPFiltered	AssemblyConnectorPrototype
ECU1SW	SensLPFiltered	SensLPFiltered_2				
ECU2Application	SensLPFiltered	SensLPFiltered_1	ECU2SW	SensLPFiltered	SensLPFiltered	DelegationConnectorPrototype
ECU2Application	Actu2OnOff	Actu2OnOff_1	ECU2SW	Actu2OnOff	Actu2OnOff_2	DelegationConnectorPrototype
ECU2Application2Actu	Actu2OnOff	Actu2OnOff				
ECU2Application2Actu	ActuAnalog	ActuAnalog				
ECU2SW	SensLPFiltered	SensLPFiltered				
ECU2SW	Actu2OnOff	Actu2OnOff_2				
ECU2SW	Actu2OnOff	Actu2OnOff_2	ECU3SW	Actu2OnOff	Actu2OnOff_4	AssemblyConnectorPrototype
ECU2SW	SensLPFiltered	SensLPFiltered				
ECU3Application	Actu2OnOff	Actu2OnOff_3	ECU3SW	Actu2OnOff	Actu2OnOff_4	DelegationConnectorPrototype
ECU3Application	ActuADValue	ActuADValue_1	ECU3SW	ActuADValue	ActuADValue	DelegationConnectorPrototype
ECU3SW	Actu2OnOff	Actu2OnOff_4				
ECU3SW	ActuADValue	ActuADValue				
ECU3SW	ActuADValue	ActuADValue				
ECU3SW	Actu2OnOff	Actu2OnOff_4				

Table/Matrix

-  Refresh
-  Toggle empty row
-  Add model element
-  Switch Rows and Columns



Example 2: AUTOSAR System Signal to DataElement Mapping Table

signal	dataElement	N
Signal.RainSensorActivationHmiRqst	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiperStimuli.WashWipeHmiCtrl.RainSensorActivationHmiRqst.RainSen...	
Signal.RainSensorActivationHmiRqst	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.WasherWiperCtrl.RainSensorActivationHmiRqst.RainSensorActi...	
Signal.RainSensorDeviceEvent_LIN	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiperStimuli.RainSensor_LINMaster_ctrl.RainSensorDeviceEvent_LIN....	
Signal.RainSensorDeviceEvent_LIN	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiperStimuli.RainSensor_LIN_hdrl.RainSensorDeviceEvent_LIN.Devic...	
Signal.RainSensorSensitivityHmiRqst	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiperStimuli.WashWipeHmiCtrl.RainSensorSensitivityHmiRqst.RainSen...	
Signal.RainSensorSensitivityHmiRqst	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.WasherWiperCtrl.RainSensorSensitivityHmiRqst.RainSensorSen...	
Signal.RainSensorDeviceIndication_LIN	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiperStimuli.RainSensor_LIN_hdrl.RainSensorDeviceIndication_LIN....	
Signal.RainSensorDeviceIndication_LIN	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiperStimuli.RainSensor_LINMaster_ctrl.RainSensorDeviceIndication...	
Signal.ResponseError_Stalks_LIN		
Signal.RsActivationRqst	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.RainSensor.RsActivationRqst.RsActivationRqst	
Signal.RsActivationRqst	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.WasherWiperCtrl.RsActivationRqst.RsActivationRqst	
Signal.RsActivationStatus	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiperStimuli.WashWipeHmiCtrl.RsActivationStatus.RsActivationStatus	
Signal.RsActivationStatus	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.WasherWiperCtrl.RsActivationStatus.RsActivationStatus	
Signal.RsAmbientTemp	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.RainSensor.RsAmbientTemp.RsAmbientTemp	
Signal.RsAmbientTemp	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.WasherWiperCtrl.RsAmbientTemp.RsAmbientTemp	
Signal.RsFailure	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.RainSensor.RsFailure.RsFailure	
Signal.RsFailure	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.WasherWiperCtrl.RsFailure.RsFailure	
Signal.RsFailure_UB		
Signal.RsRainIntensityStatus	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.RainSensor.RsRainIntensityStatus.RsRainIntensityStatus	
Signal.RsRainIntensityStatus	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.WasherWiperCtrl.RsRainIntensityStatus.RsRainIntensityStatus	
Signal.RsRainIntensityStatus_UB		
Signal.RsSensitivityRqst	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.RainSensor.RsSensitivityRqst.RsSensitivityRqst	
Signal.RsSensitivityRqst	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.WasherWiperCtrl.RsSensitivityRqst.RsSensitivityRqst	
Signal.RsVehicleSpeed	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.RainSensor.RsVehicleSpeed.RsVehicleSpeed	
Signal.RsVehicleSpeed	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.WasherWiperCtrl.RsVehicleSpeed.RsVehicleSpeed	
Signal.RsVehicleType	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.RainSensor.RsVehicleType.RsVehicleType	
Signal.RsVehicleType	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.WasherWiperCtrl.RsVehicleType.RsVehicleType	
Signal.RsWasherStatus	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.WasherWiperCtrl.RsWasherStatus.RsWasherStatus	
Signal.RsWasherStatus	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.RainSensor.RsWasherStatus.RsWasherStatus	
Signal.RsWindScreenType	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.WasherWiperCtrl.RsWindScreenType.RsWindScreenType	
Signal.RsWindScreenType	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.RainSensor.RsWindScreenType.RsWindScreenType	
Signal.RsWiperParkStatus	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.WasherWiperCtrl.RsWiperParkStatus.RsWiperParkStatus	
Signal.RsWiperParkStatus	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.RainSensor.RsWiperParkStatus.RsWiperParkStatus	
Signal.RsWipingRqst	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.RainSensor.RsWipingRqst.RsWipingRqst	
Signal.RsWipingRqst	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.WasherWiperCtrl.RsWipingRqst.RsWipingRqst	
Signal.RsWipingRqst_UB		



Example 2: AUTOSAR System Signal to DataElement Mapping Table

signal	dataElement
Signal.RainSensorActivationHmiRqst	VehicleProject.system
Signal.RainSensorActivationHmiRqst	VehicleProject.system
Signal.RainSensorDeviceEvent_LIN	VehicleProject.system
Signal.RainSensorDeviceEvent_LIN	VehicleProject.system
Signal.RainSensorSensitivityHmiRqst	VehicleProject.system
Signal.RainSensorSensitivityHmiRqst	VehicleProject.system
Signal.RainSensorDeviceIndication_LIN	VehicleProject.system
Signal.RainSensorDeviceIndication_LIN	VehicleProject.system
Signal.ResponseError_Stalks_LIN	
Signal.RsActivationRqst	VehicleProject.system
Signal.RsActivationRqst	VehicleProject.system
Signal.RsActivationStatus	VehicleProject.system
Signal.RsActivationStatus	VehicleProject.system
Signal.RsAmbientTemp	VehicleProject.system
Signal.RsAmbientTemp	VehicleProject.system
Signal.RsFailure	VehicleProject.system
Signal.RsFailure	VehicleProject.system
Signal.RsFailure_UB	
Signal.RsRainIntensityStatus	VehicleProject.system
Signal.RsRainIntensityStatus	VehicleProject.system
Signal.RsRainIntensityStatus_UB	
Signal.RsSensitivityRqst	VehicleProject.system
Signal.RsSensitivityRqst	VehicleProject.system
Signal.RsVehicleSpeed	VehicleProject.system
Signal.RsVehicleSpeed	VehicleProject.system
Signal.RsVehicleType	VehicleProject.system
Signal.RsVehicleType	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.WasherWiperCtrl.RsVehicleType.RsVehicleType
Signal.RsWasherStatus	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.WasherWiperCtrl.RsWasherStatus.RsWasherStatus
Signal.RsWasherStatus	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.RainSensor.RsWasherStatus.RsWasherStatus
Signal.RsWindScreenType	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.WasherWiperCtrl.RsWindScreenType.RsWindScreenType
Signal.RsWindScreenType	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.RainSensor.RsWindScreenType.RsWindScreenType
Signal.RsWiperParkStatus	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.WasherWiperCtrl.RsWiperParkStatus.RsWiperParkStatus
Signal.RsWiperParkStatus	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.RainSensor.RsWiperParkStatus.RsWiperParkStatus
Signal.RsWipingRqst	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.RainSensor.RsWipingRqst.RsWipingRqst
Signal.RsWipingRqst	VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.WasherWiperCtrl.RsWipingRqst.RsWipingRqst

Execution Manager Instance Reference

Context:
 VehicleProject.system_2.WasherWiperSubSystemTest_Type.WasherWiper.RainSensor.RsActivationRqst.RsActivationRqst

- [-] RainSensor
 - [+] [-] RsFailure
 - [+] [-] RsRainIntensityStatus
 - [+] [-] RsWipingRqst
 - [+] [-] RsActivationRqst
 - [x] [-] RsActivationRqst
 - [+] [-] RsAmbientTemp
 - [+] [-] RsSensitivityRqst
 - [+] [-] RsWasherStatus
 - [+] [-] RsVehicleSpeed
 - [+] [-] RsVehicleType
 - [+] [-] RsWindScreenType
 - [+] [-] RsWiperParkStatus
 - [+] [-] RsFailure
 - [+] [-] RsRainIntensityStatus
 - [+] [-] RsWipingRqst
 - [+] [-] RsActivationRqst
 - [+] [-] RsAmbientTemp
 - [+] [-] RsSensitivityRqst
 - [+] [-] RsWasherStatus
 - [+] [-] RsVehicleSpeed

OK Cancel

Explicit AUTOSAR Implementation

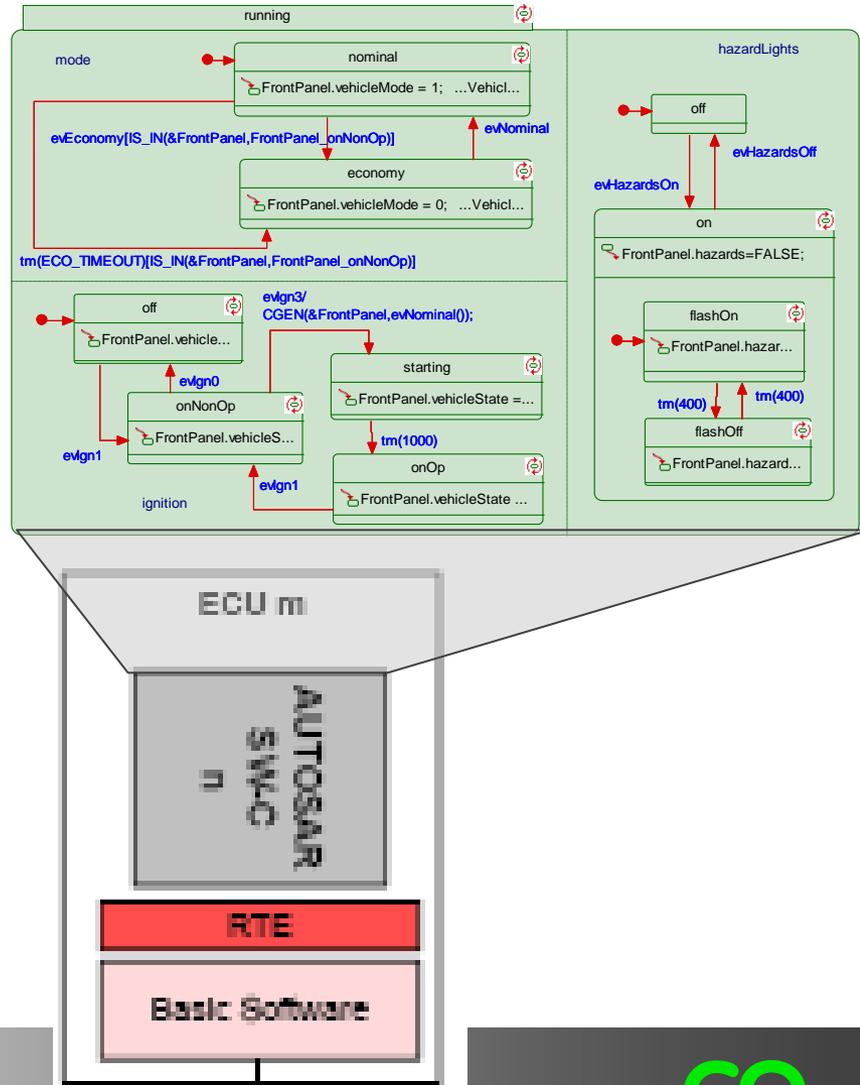
The screenshot displays an AUTOSAR development environment. On the left, the 'Entire Model View' tree shows a project named 'LightManagerDemo_C2C' with a sub-package 'ARDesign' containing 'Implementations' and 'Impl_DigitalDriver'. The 'Impl_DigitalDriver' package is circled in red. The main workspace shows a 'LightManagementSWComposition' diagram with several components: 'Is:LightSensor', 'Im:LightManager', 'pd:PwmDriver', 'om:DigitalDriver', 'L:Led', and 'nt:Network'. Connections are shown between these components, such as 'lightInfo' from the sensor to the manager, and various outputs from the manager to the driver and LED. A 'Function: DigitalDriverPeriodicRoutine in file_5' window is open in the foreground, showing the implementation of the periodic routine function.

```
void DigitalDriverPeriodicRoutine()  
  
    unsigned int lightLevel = 0;  
    unsigned char enabled = 0;  
  
    /* Read the control */  
    enabled = Rte_IRead_DigitalDriverPeriodicRoutine_in_control();  
  
    if(enabled) {  
        lightLevel = Rte_Read_DigitalDriverPeriodicRoutine_in_lightLevel();  
  
        Rte_Write_digitalOut_setLight(lightLevel);  
    }  
  
    Rte_Call_serialOut_setLight(lightLevel);
```



Rhapsody AUTOSAR BMT Implementation

- Modeling the behavior of the AUTOSAR Software Component using UML/SysML Graphical Models
 - Statecharts
 - Activity Diagrams (Flowcharts)
 - C code
- Generate code for the behavior of an AUTOSAR Software Component
 - Regarding the related AR definitions, targeting the RTE



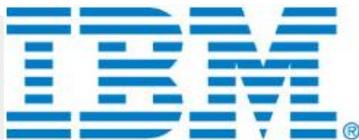
DEMO Rational Rhapsody:

- **Define AUTOSAR software application architecture, design and implementation using Rational Rhapsody**
 - Define an AUTOSAR Software Architecture
 - Creating Application SWC
 - Defining SW Composition
 - Define an AUTOSAR Software Design & Implementation
 - Defining SWC Internal-Behavior
 - Defining SWC Implementation
 - Define some aspects of an AUTOSAR System
 - Mapping of SWC to ECU Instance
 - Mapping of DataElementPrototype / ClientServerOperation to SystemSignal
 - Defining the System TopLevelSWComposition



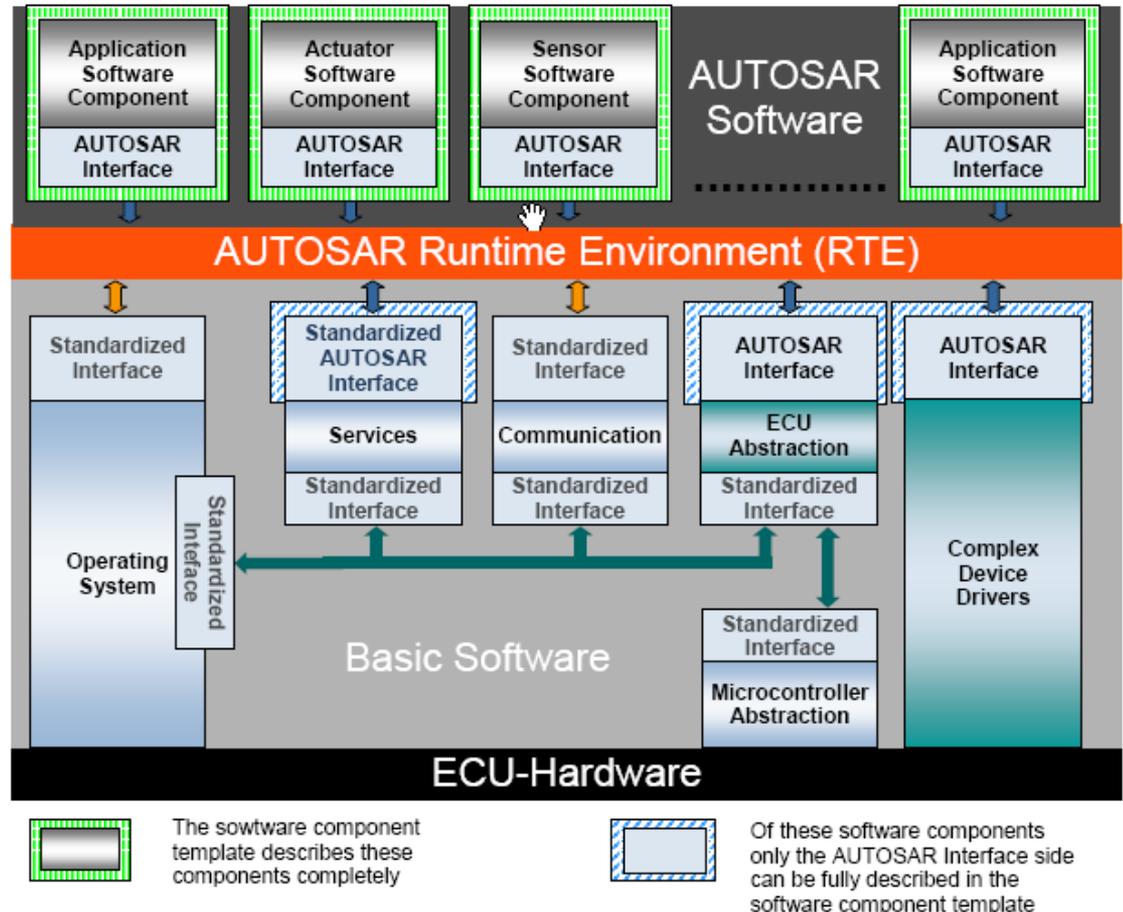
Agenda

- Process overview
- Define AUTOSAR software application architecture, design and implementation using Rational Rhapsody
- **Configure and generate AUTOSAR ECU BSW with EB tresos Studio/AutoCore**



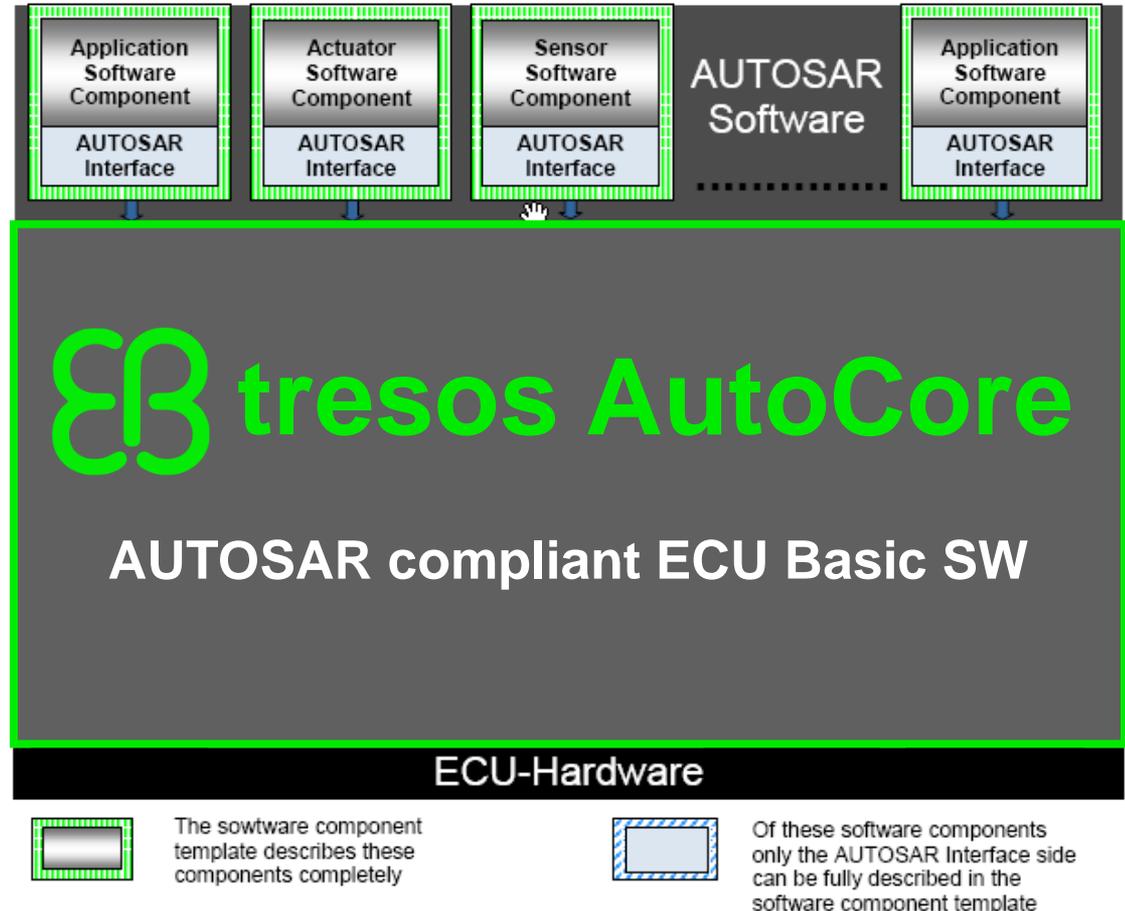
Configure & generate AUTOSAR ECU BSW

tresos Studio/AutoCore



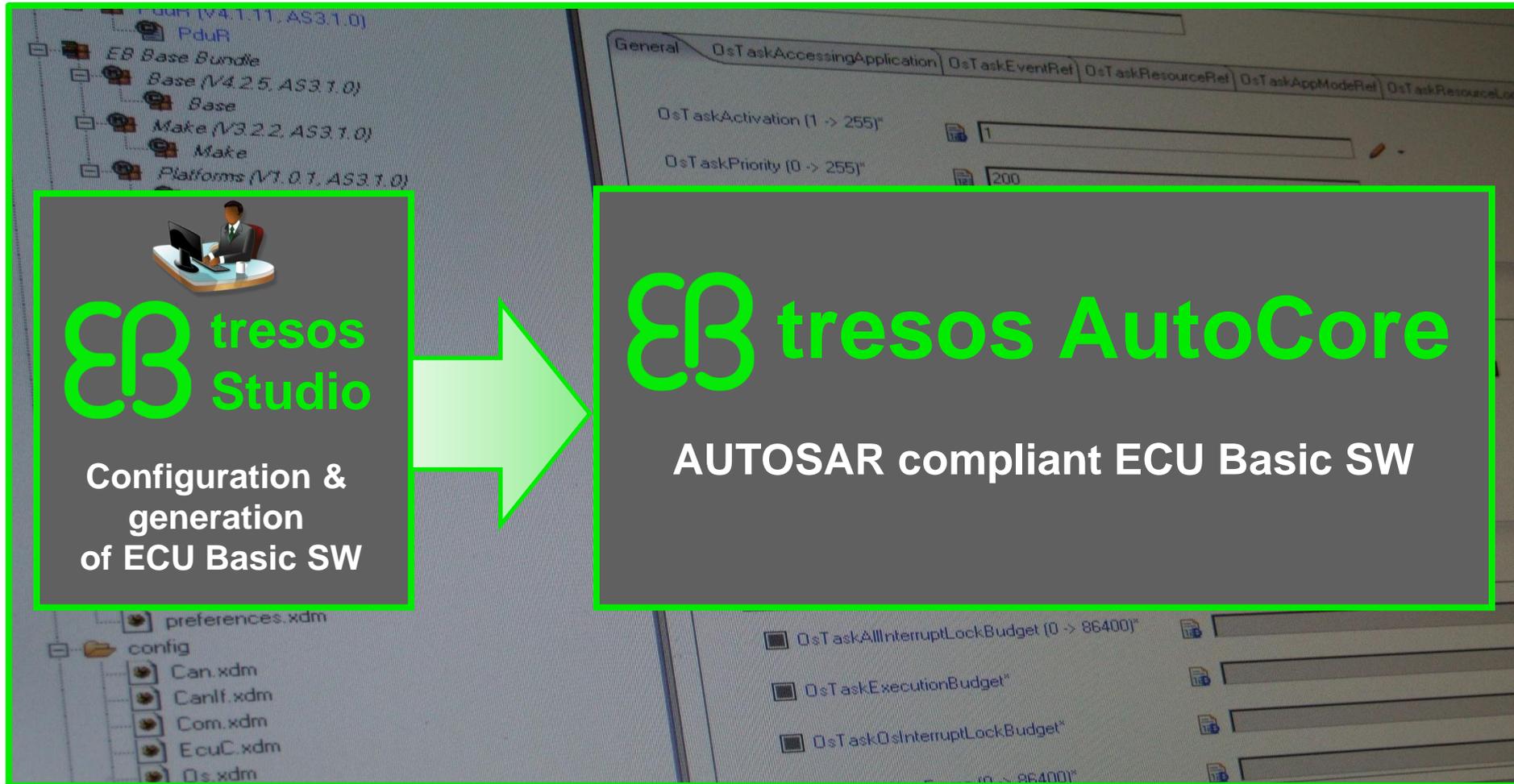
Configure & generate AUTOSAR ECU BSW

EB tresos Studio/AutoCore



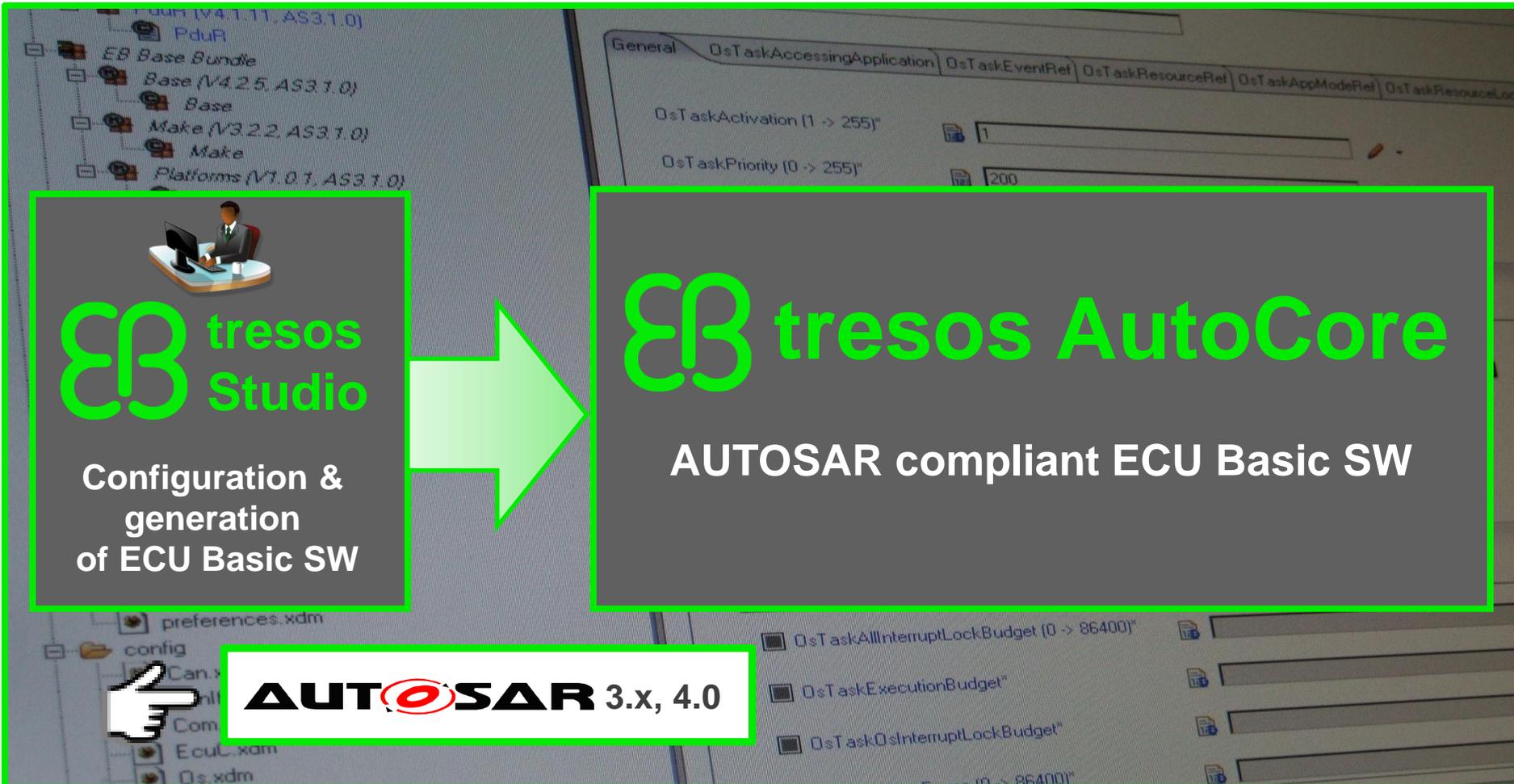
Configure & generate AUTOSAR ECU BSW

EB tresos Studio/AutoCore

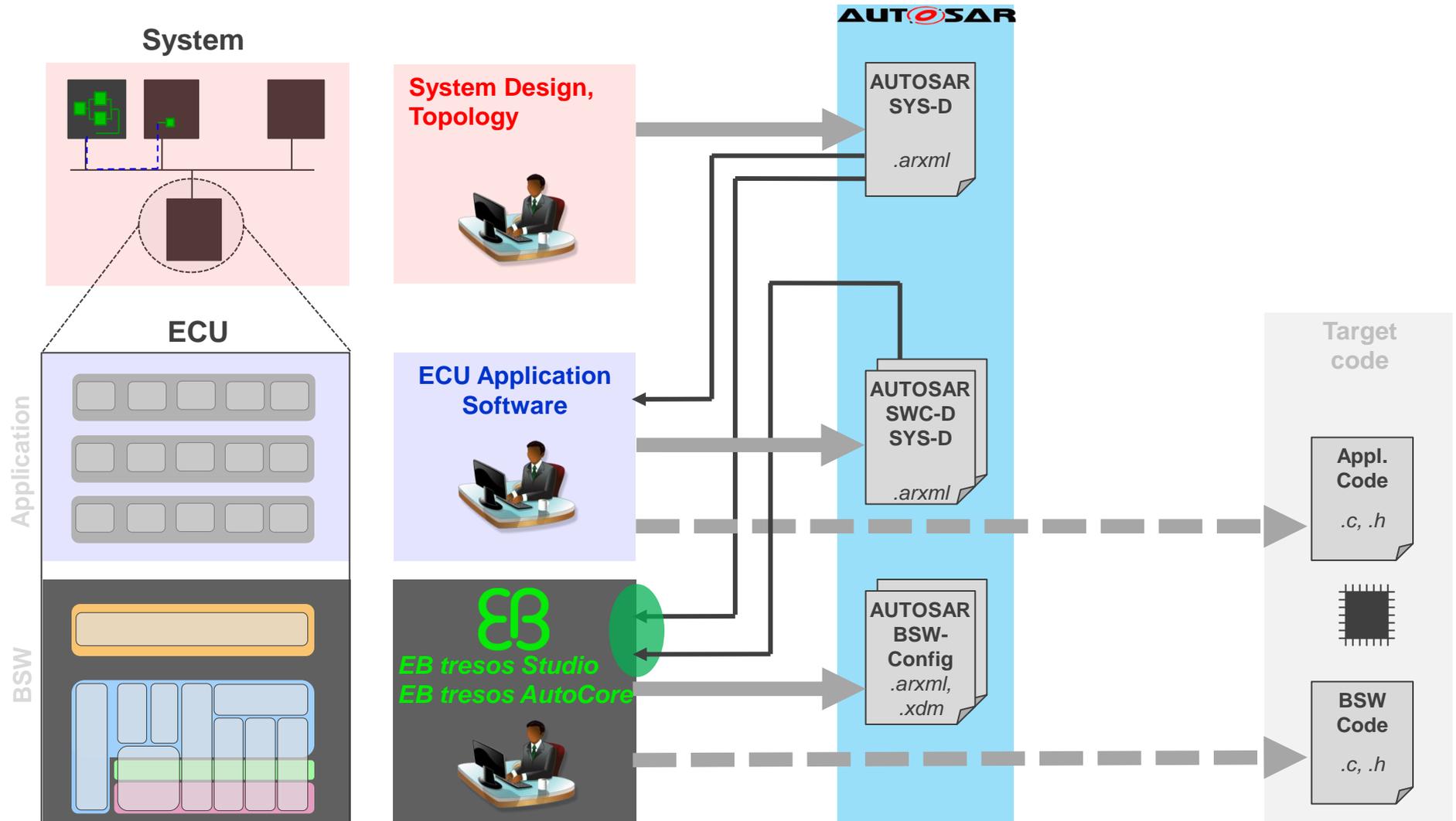


Configure & generate AUTOSAR ECU BSW

EB tresos Studio/AutoCore



AUTOSAR Workflow: Exchange Files





Thank you!