AUTomotive Open System ARchitecture

Seminar Software Engineering for Automotive Systems WS 07/08

Florian Leitner
florian.leitner@uni-konstanz.de

Outline

- Introduction to AUTOSAR
- AUTOSAR Software Architecture
- Modeling AUTOSAR Systems
  - Unified Modeling Language
  - MATLAB / SIMULINK

Introduction to AUTOSAR

- What is AUTOSAR?
- Who is AUTOSAR?
- Why do we need AUTOSAR?
- Main Motivations and Goals

What is AUTOSAR?

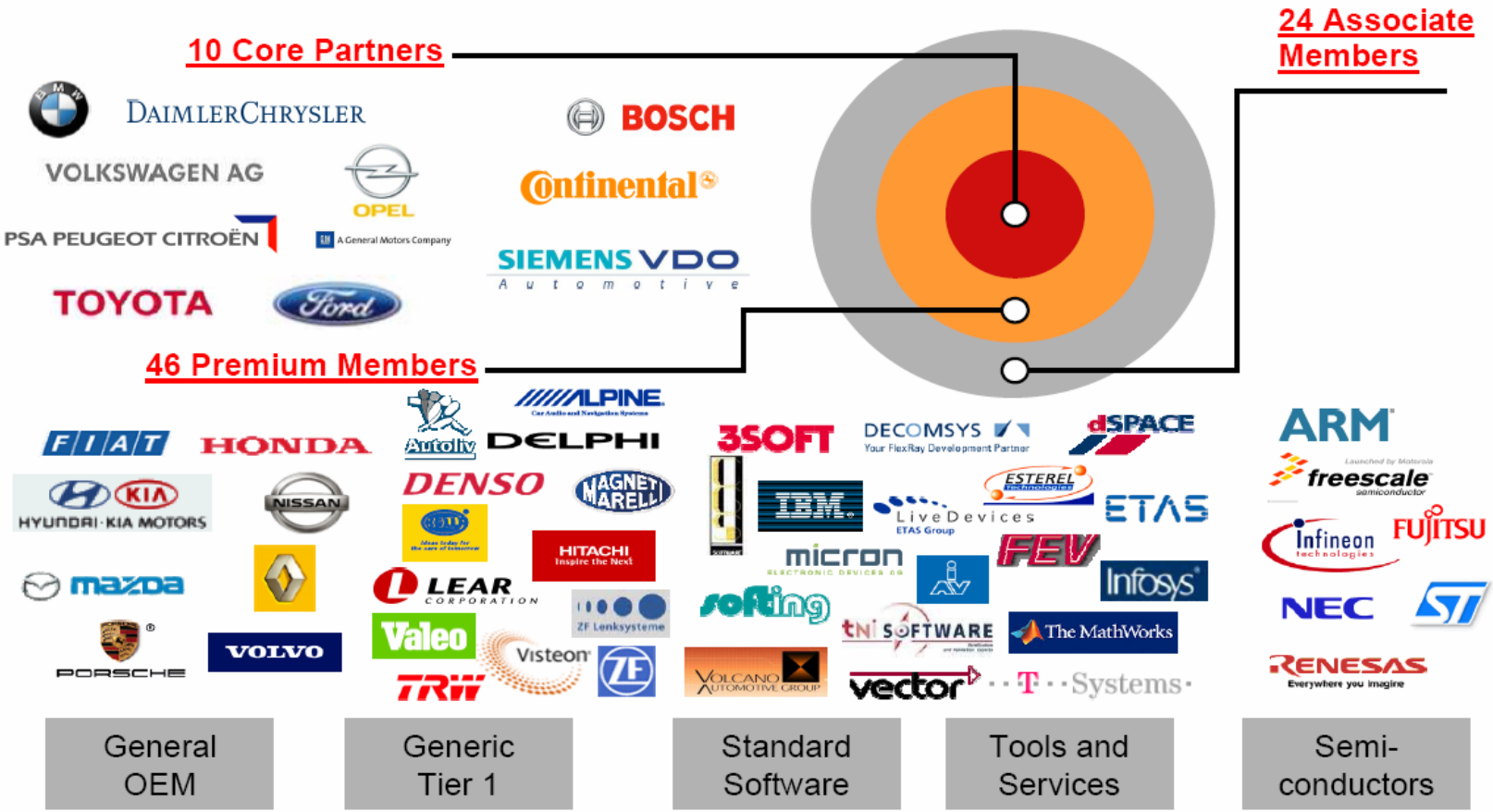
- www.autosar.org:
  „AUTOSAR (AUTomotive Open System ARchitecture) is an open and standardized automotive software architecture, jointly developed by automobile manufacturers, suppliers and tool developers.“

- The AUTOSAR consortium was founded in 2003 by BMW, DaimlerChrysler, Bosch, Continental, Volkswagen and Siemens VDO. Later Ford, General Motors, Toyota and PSA (Peugeot Citroen) joined the consortium as core members.

- The AUTOSAR standard will serve as a platform for future vehicle applications

Who is AUTOSAR?

Worldwide, OEMs and suppliers participate in AUTOSAR.



Why do we need AUTOSAR?

- Increasing complexity of software in automotive systems
- There is no standardized software architecture

In a single car*	1994	2008
Control Units	40	60
MIPS	45	1150
MHz	85	2000
MCU-Storage (Program + Data)	1,1 MB+160 kB	19 MB+1,25 MB
Transistors	21 m.	340 m.

(* without Infotainment)

Source: ElektronikNet.de http://www.elektroniknet.de/home/automotive/autosar/multiple-applikationen-in-steuergeraeten/

Main Motivations

- Management of E/E complexity associated with growth in functional scope
- Flexibility for product modification, upgrade and update
- Scalability of solutions within and across product lines
- Improved quality and reliability of E/E systems

Main Goals

- Fulfillment of future vehicle requirements, such as, availability and safety, SW upgrades/ updates and maintainability
- Increased scalability and flexibility to integrate and transfer functions
- Higher penetration of "Commercial off the Shelf" SW and HW components across product lines
- Improved containment of product and process complexity and risk
- Cost optimization of scalable systems

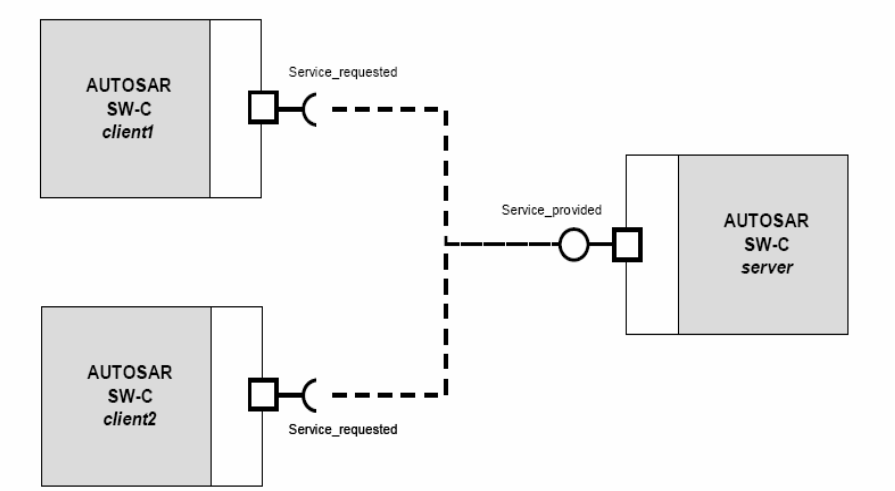
AUTOSAR Software Architecture

- AUTOSAR Software Component
- Communication Patterns
- Layered Software Architecture
- AUTOSAR Basic Software

AUTOSAR Software Components

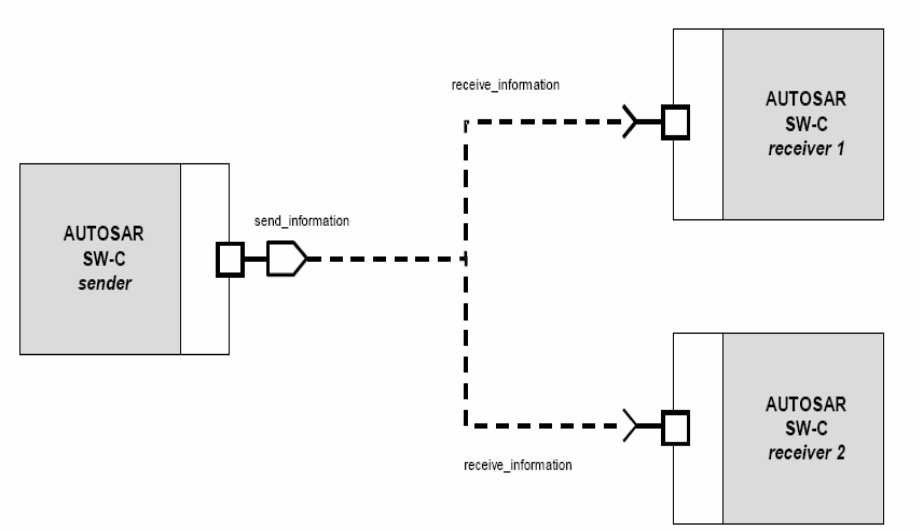
- Fundamental design concept is **separation between infrastructure and application**
- Application consists of interconnected software components
- atomic software component: a software component can not be distributed over several ECUs
- AUTOSAR software component implementation is independent from the infrastructure

Communication Patterns



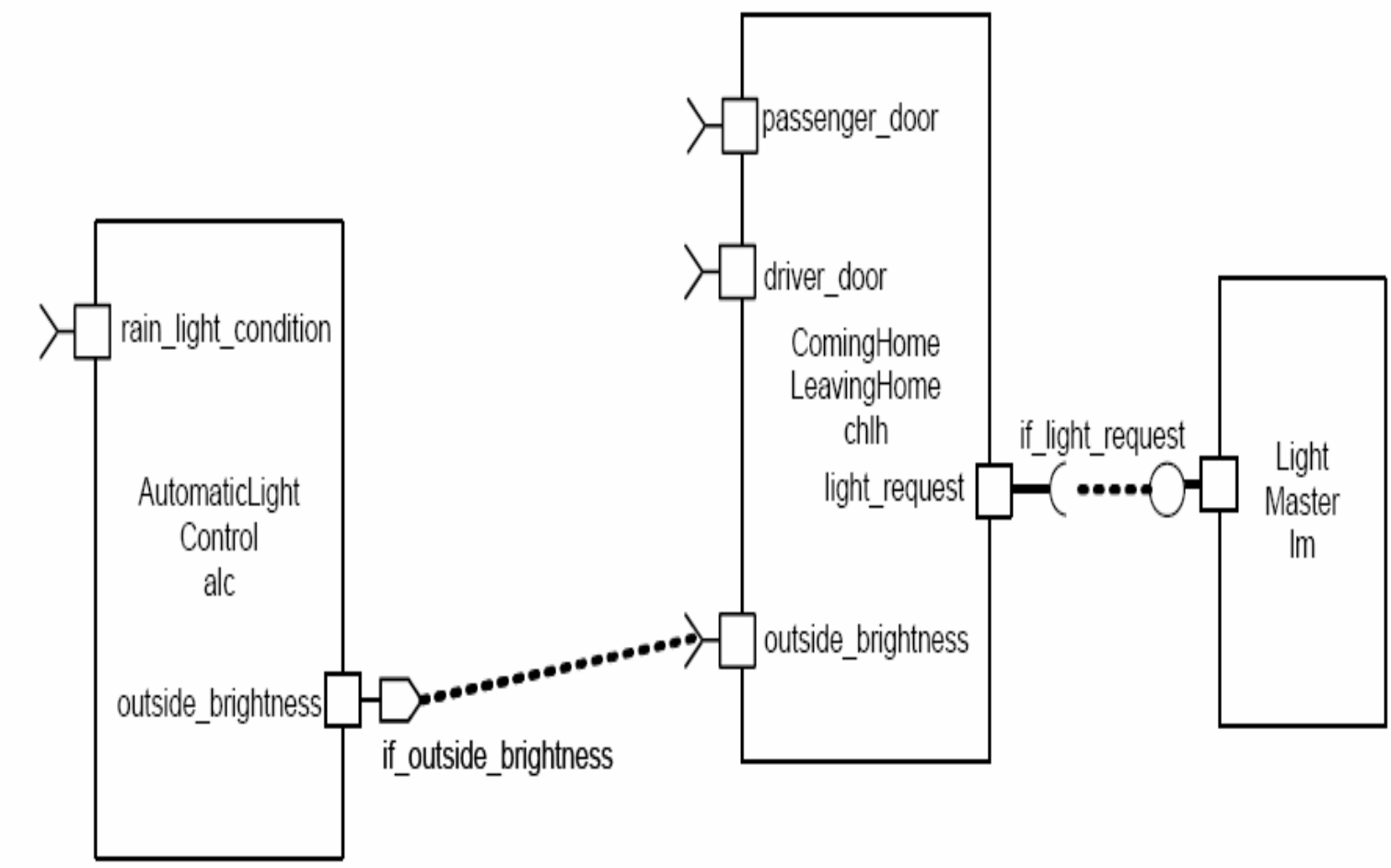
- Client – Server
  - Server: provider of Service
  - Client: user of service
  - Client initiates communication
  - Single component can be server and client
  - Synchronous and asynchronous communication is possible

Communication Patterns

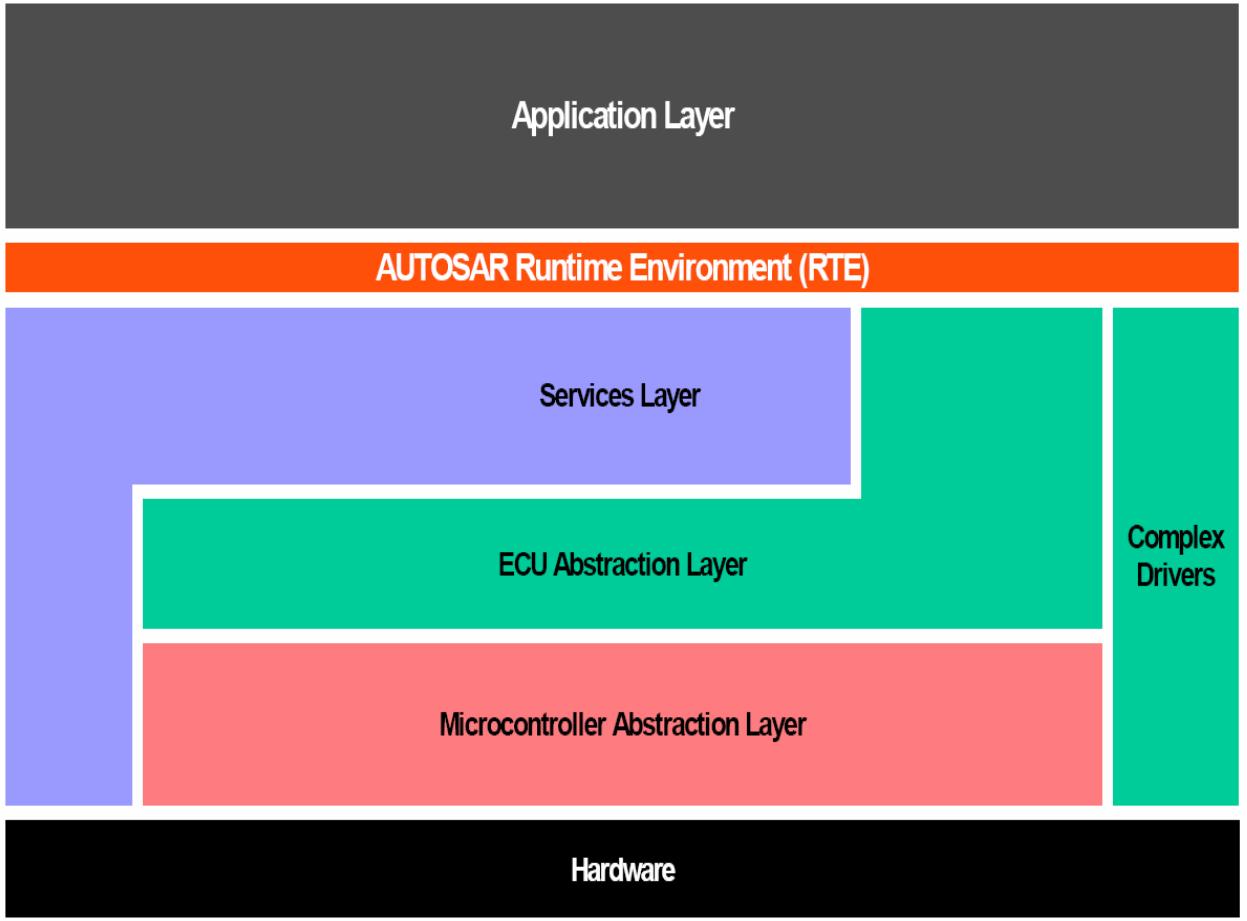


- Sender – Receiver
  - Asynchronous distribution of information
  - No response from receivers
  - Sender does not know the number or identity of the receivers

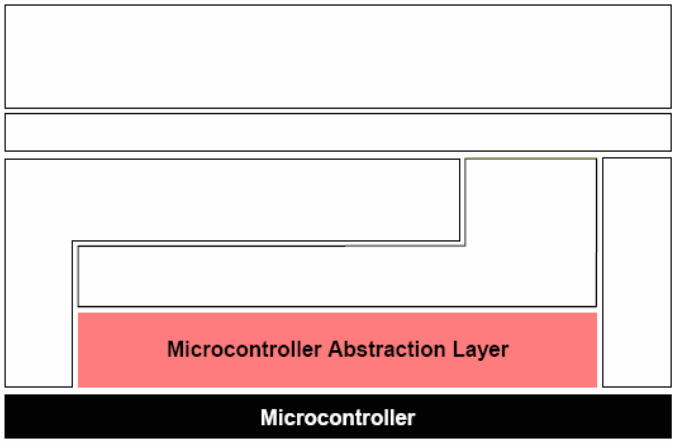
Example


passenger_door
driver_door
ComingHome
LeavingHome
chlh
light_request
outside_brightness
rain_light_condition
AutomaticLight
Control
alc
outside_brightness
if_outside_brightness
if_light_request
Light
Master
lm

Layered Software Architecture

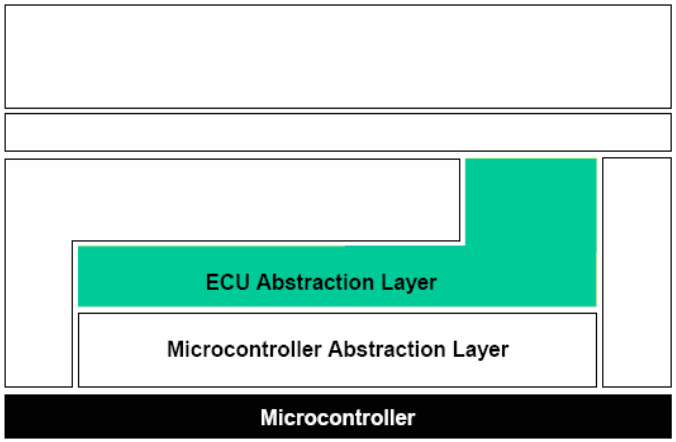


Layered Software Architecture



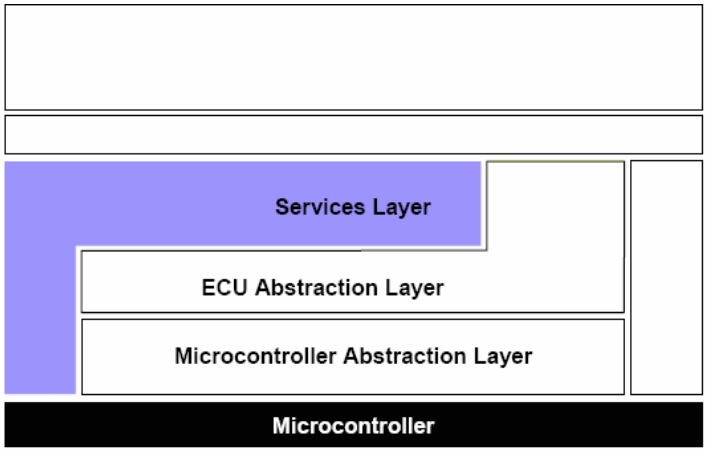
- Microcontroller Abstraction Layer:
  - lowest software layer of the Basic Software
  - Makes higher software independent of Microcontroller

Layered Software Architecture



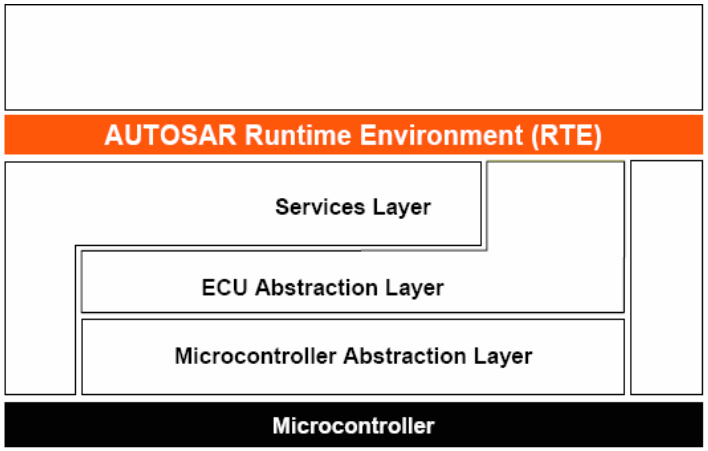
- ECU Abstraction Layer:
  - Interfaces the drivers of Microcontroller Abstraction Layer
  - Makes higher software layers independent of ECU hardware layout
  - Offers access to I/O Signals

Layered Software Architecture



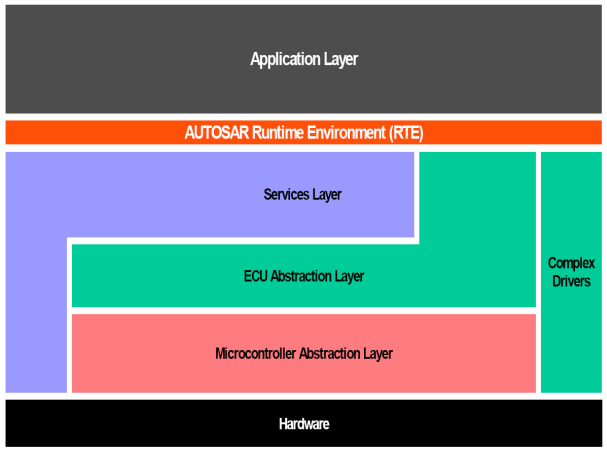
- Service Layer:
  - Highest layer of the Basic Software
  - Offers Memory Services, Diagnostic Services, ECU state management
  - Provides basic services for application and basic software modules

Layered Software Architecture



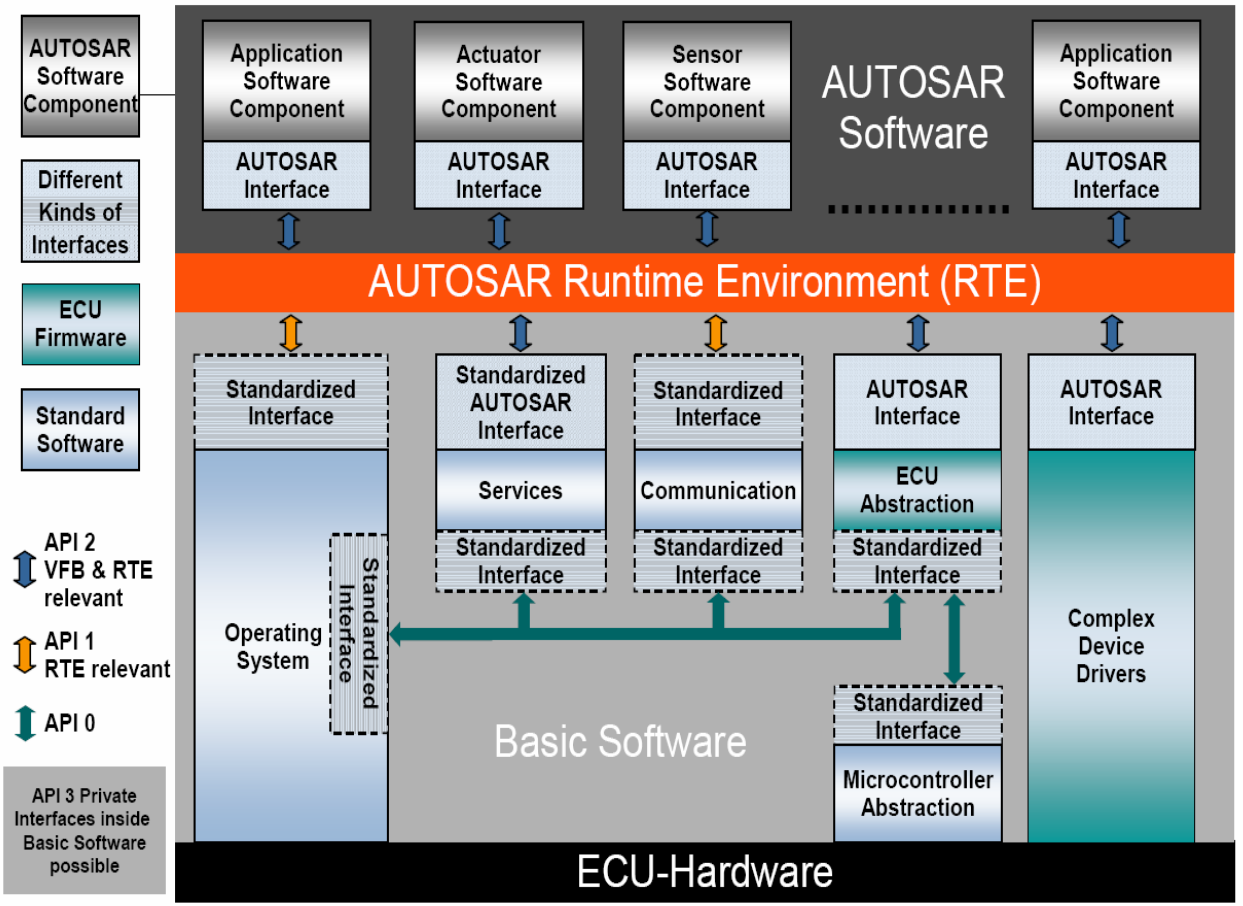
- Runtime Environment:
  - Middleware Layer
  - Provides communication services for the application software
  - Makes AUTOSAR Software Components independent from the mapping to a specific ECU

Layered Software Architecture

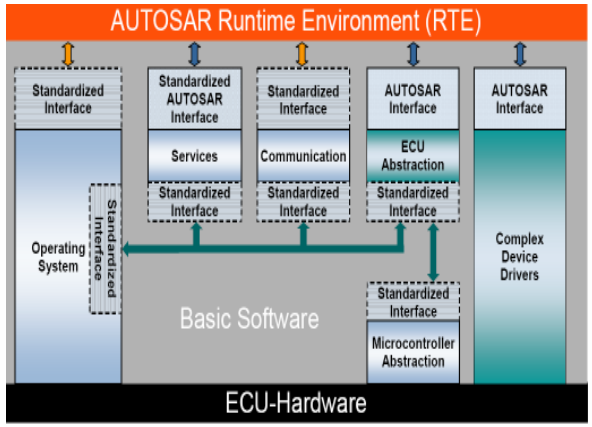


- Application Layer:
  - "component style"
  - Software components communicate with other components and/or services via the RTE

AUTOSAR Software Architecture



AUTOSAR Basic Software



- **Services:** All system services (e.g. NVRAM, flash and memory management, diagnostic protocols)
- **Communication:** communication framework (e.g. CAN, LIN, FlexRay...), the I/O management and the network management.
- **Operating System:** standard OSEK OS (ISO 17356-3) is used as basis for the AUTOSAR OS. The can differ from ECU to ECU but if a proprietary OS is used the interfaces to the AUTOSAR Components have to be AUTOSAR compliant.
- **Microcontroller Abstraction:** All access to the Hardware is routed through the Microcontroller Abstraction layer. This is needed to avoid direct access to microcontroller registers from higher-level software.
- **ECU Abstraction:** provides a software interface to the electrical values of any specific ECU to ensure the hardware independence of higher-level software
- **Complex Device Driver (CDD):** provides direct access to hardware for resource critical applications (e.g. injection control, electric valve control)

Modeling AUTOSAR Systems

- Unified Modeling Language
  - UML and embedded systems
  - UML and AUTOSAR
- VW AUTOSAR Pasat Sedan
  - Proceeding

UML and embedded System

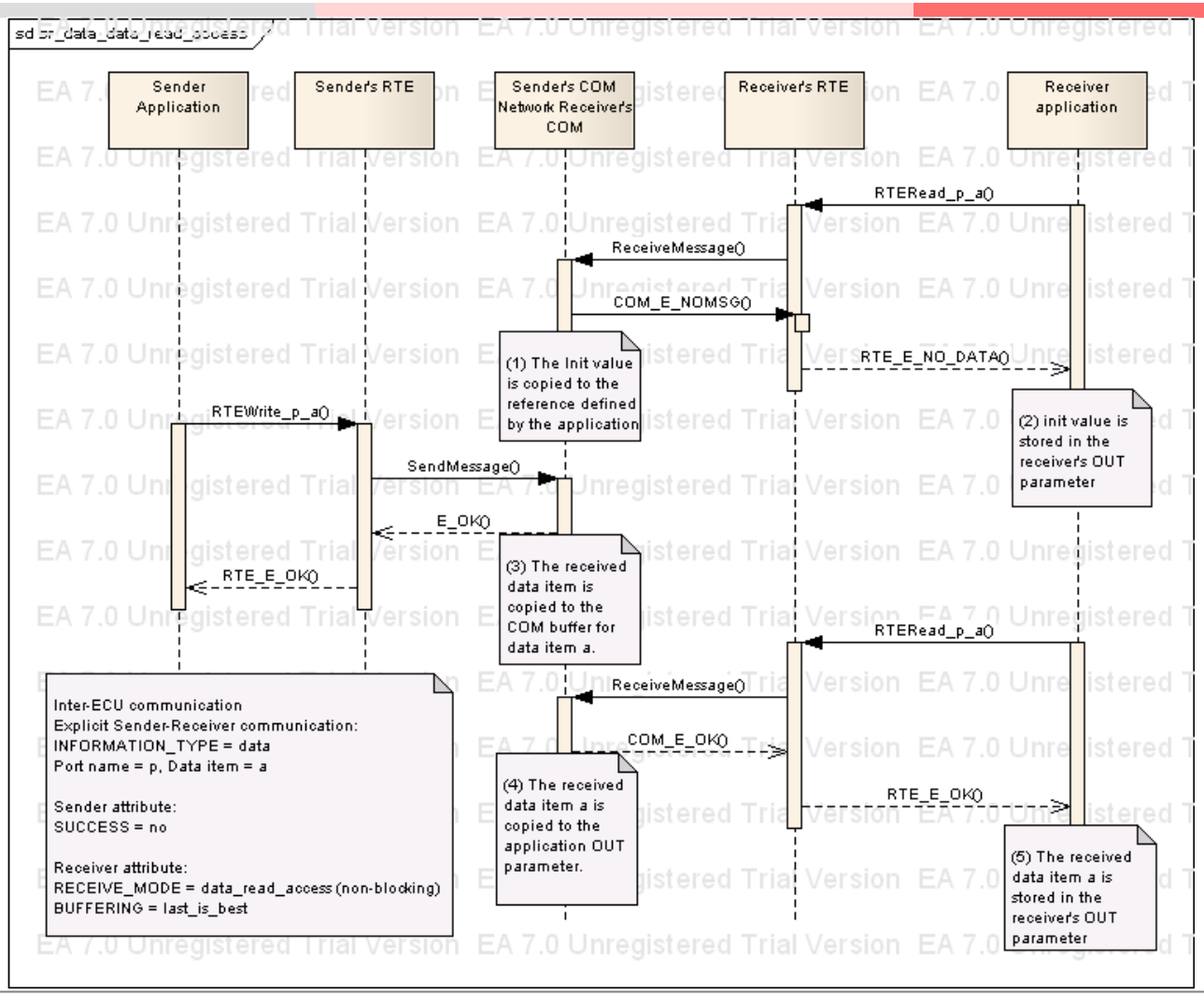
- Object Oriented approach is new in the embedded software field
  - Embedded software is often highly optimized, resource and time critical

- BUT: to deal with the increasing complexity a model driven development approach is necessary

UML and AUTOSAR

- There is a special UML Profile for Enterprise Architect (Sparx Systems)
- Theoretically any UML 2.0 tool can be used. Enterprise Architect is the tool of choice because of its low price

UML Example


sd sr_data_data_read_access
Sender Application
Sender's RTE
Sender's COM Network Receiver's COM
Receiver's RTE
Receiver application
RTERead_p_a()
ReceiveMessage()
COM_E_NOMSG()
(1) The Init value is copied to the reference defined by the application
RTE_E_NO_DATA()
(2) init value is stored in the receiver's OUT parameter
RTEWrite_p_a()
SendMessage()
E_OK()
RTE_E_OK()
(3) The received data item is copied to the COM buffer for data item a.
RTERead_p_a()
ReceiveMessage()
COM_E_OK()
RTE_E_OK()
(4) The received data item a is copied to the application OUT parameter.
(5) The received data item a is stored in the receiver's OUT parameter
Inter-ECU communication
Explicit Sender-Receiver communication:
INFORMATION_TYPE = data
Port name = p, Data item = a
Sender attribute:
SUCCESS = no
Receiver attribute:
RECEIVE_MODE = data_read_access (non-blocking)
BUFFERING = last_is_best

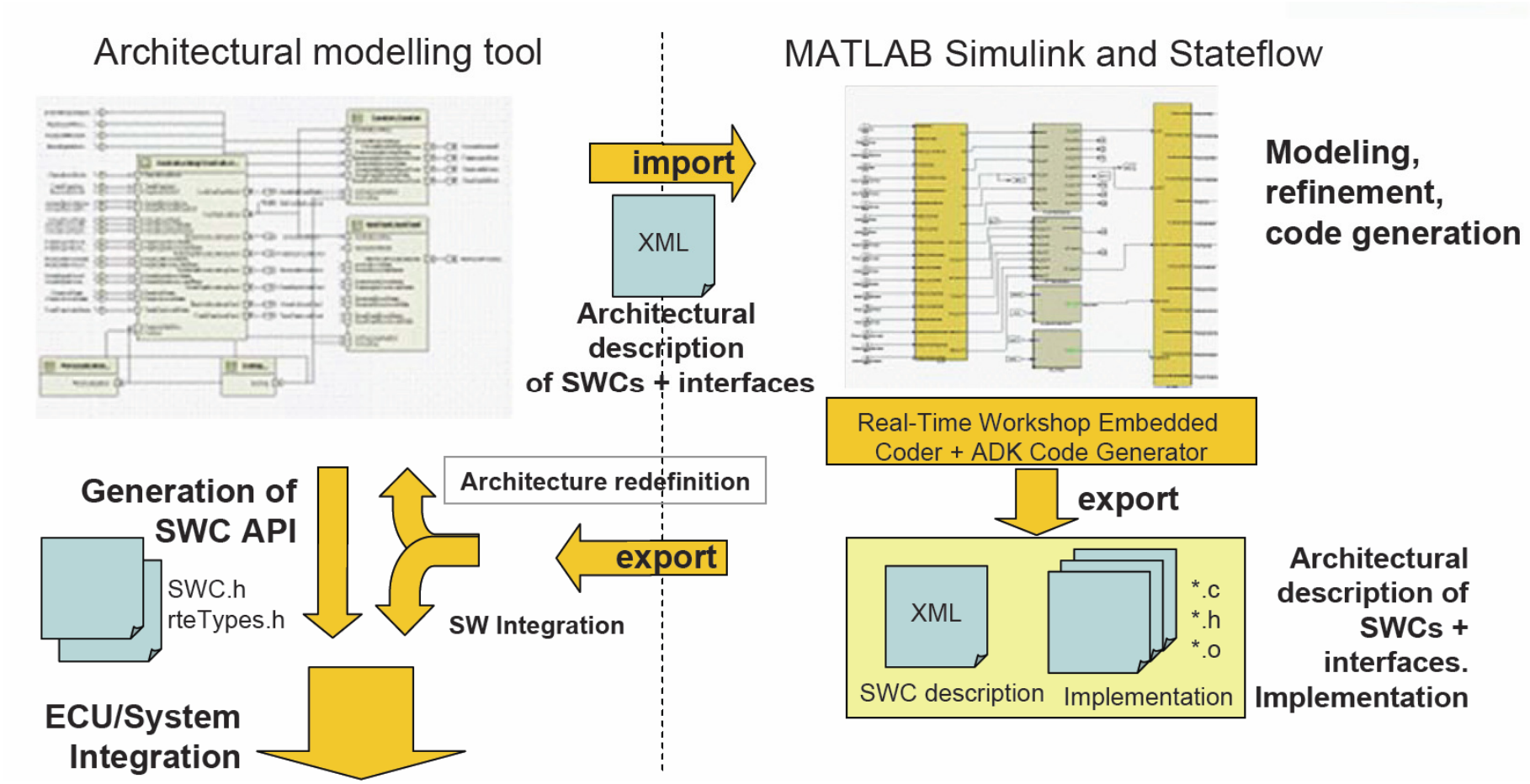
VW AUTOSAR Pasat Sedan

- Volkswagen AG and HELLA have jointly developed a fully functional Body/Comfort ECU for a Volkswagen series-production vehicle which is furnished with AUTOSAR compatible software.

Proceeding

- Create architectural model with interfaces
- Create SWC Description and import architecture definition to Simulink
- Implementation of the SWC behavior with MATLAB and Simulink
- Simulation and Test with Simulink
- Generating AUTOSAR code + SWC Description with Real-Time Workshop Embedded Coder + ADK
- Software and ECU Integration
- ECU tests with CANoe (Emulation Tool) and final in-vehicle tests

Proceeding


Architectural modelling tool
MATLAB Simulink and Stateflow
import
XML
Architectural description of SWCs + interfaces
Modeling, refinement, code generation
Real-Time Workshop Embedded Coder + ADK Code Generator
export
Architecture redefinition
Generation of SWC API
SWC.h
rteTypes.h
export
SW Integration
XML
*.c
*.h
*.o
SWC description
Implementation
Architectural description of SWCs + interfaces. Implementation
ECU/System Integration

References

- [1] AUTOSAR GbR: Technical Overview
  - http://www.autosar.org/download/AUTOSAR_TechnicalOverview.pdf
- [2] AUTOSAR GbR: About AUTOSAR
  - http://www.autosar.org/
- *[3]* AUTOSAR GbR: Layered Software Architecture
  - http://www.autosar.org/download/AUTOSAR_LayeredSoftwareArchitecture.pdf
- *[4] VOLKSWAGEN AG, Carmeq GmbH: AUTOSAR-Compliant Functional Modeling with MATLAB, Simulink, Stateflow and Real-Time Workshop Embedded Coder of a Serial Comfort Body Controller, at Mathworks Automotive Conference 2007*
  - http://www.mathworks.com/industries/auto/mac2007/proceedings/day2/presentations/2_autosar.pdf

Questions and Answers...

Thank You!