INFORMATIK ▪ CONSULTING ▪ SYSTEMS AG

Vorlesung
Automotive Software Engineering
Teil 7 Normen und Standards (1-1)

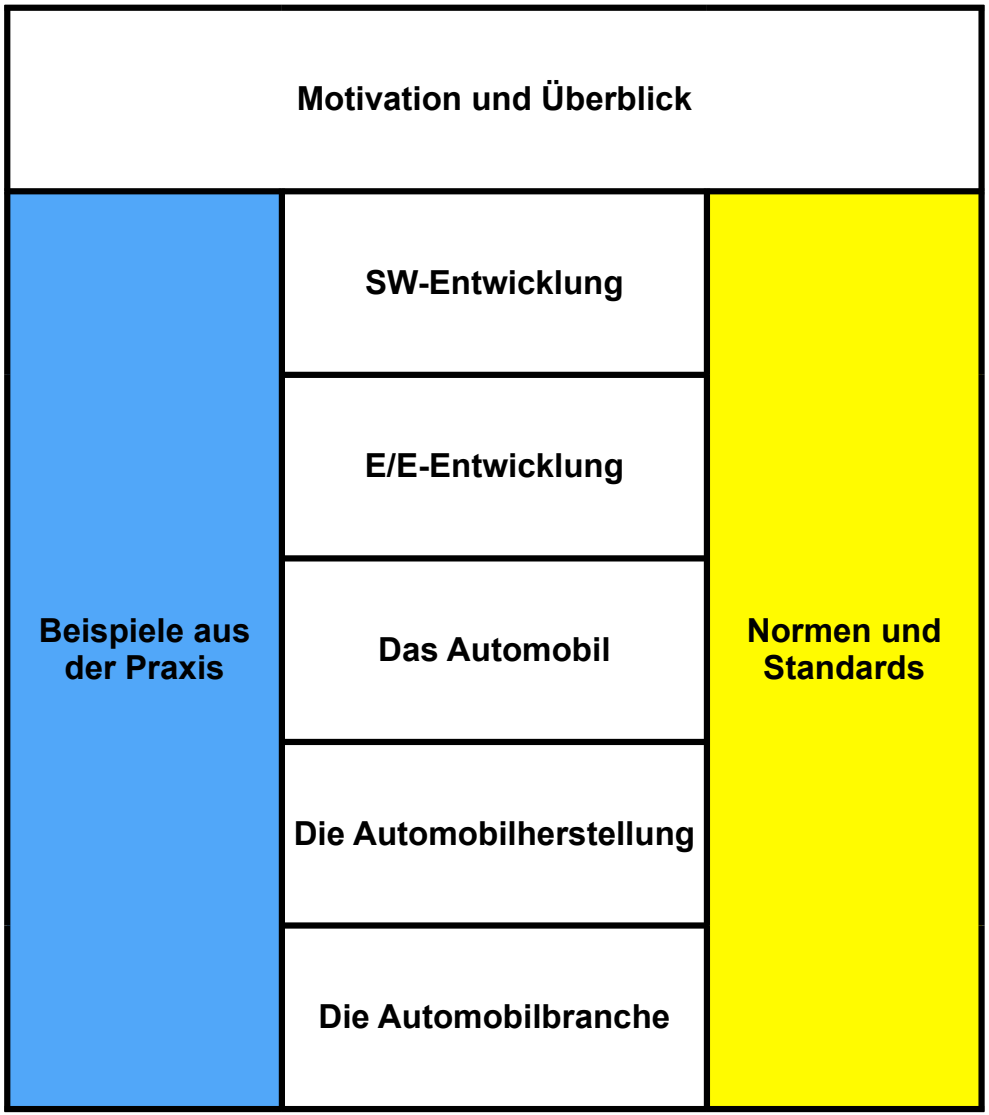
Sommersemester 2014

Prof. Dr. rer. nat. Bernhard Hohlfeld

Bernhard.Hohlfeld@mailbox.tu-dresden.de

Technische Universität Dresden, Fakultät Informatik
Honorarprofessur Automotive Software Engineering

Vorlesung Automotive Software Engineering



AUTOSAR

MOST

CAN

lin

HIS

OSEK/ VDX

ASAM

FlexRay

ISO 26262
Road vehicles - Functional safety

Lernziele
Normen und Standards

- Die Bedeutung von Normen und Standards für industrielle Entwicklung verstehen.
- AUTOSAR Automotive Open System Architecture kennenlernen
  - Motivation
  - Technik
  - Beispiele
- ISO 26262 Road Vehicles Functinal Safety kennenlernen
- Den Begriff COTS einordnen
- Entwurfs- und Codierstandards kennenlernen

7. Normen und Standards

1. AUTOSAR
2. ARTOP
3. ISO 26262 - Road Vehicles - Functional Safety
4. COTS
5. Entwurfs- und Codierstandards

7. Normen und Standards

1. AUTOSAR

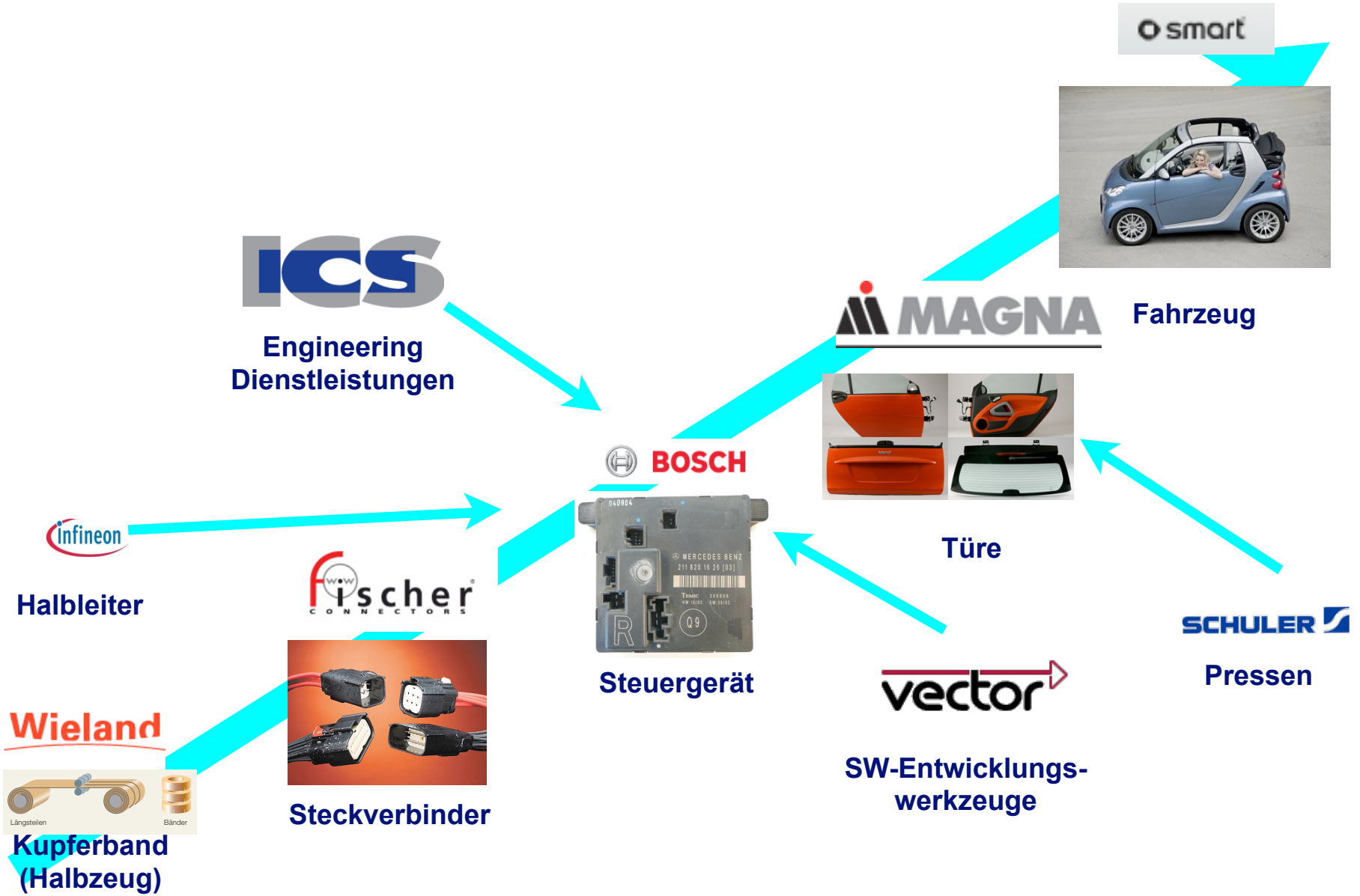
2. ARTOP

3. ISO 26262 - Road Vehicles - Functional Safety

4. COTS

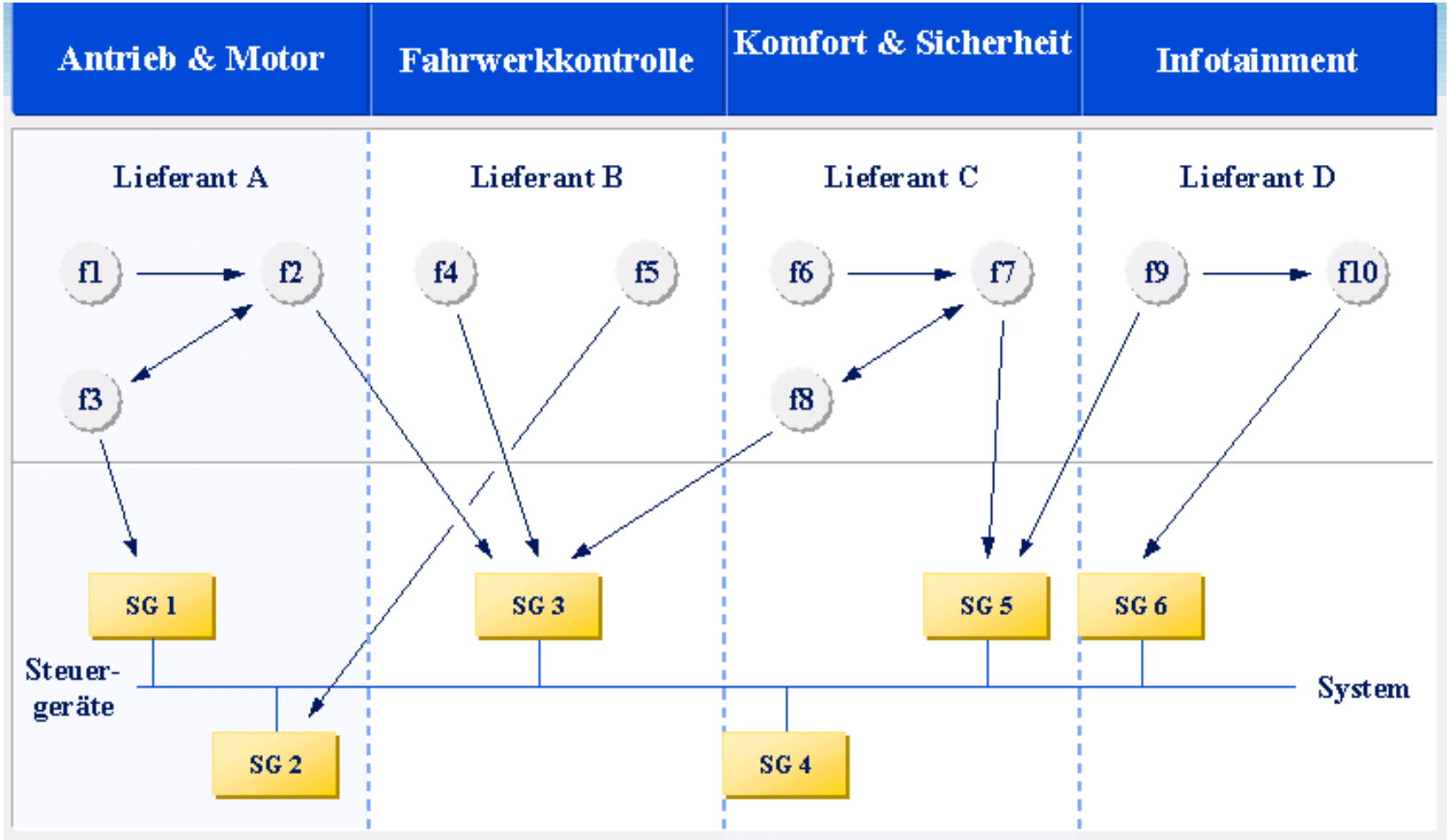
5. Entwurfs- und Codierstandards

Software im Fahrzeug
- siehe Teil 2 Die Automobilbranche



Anforderungen

- Neue Beziehungen, Kompetenzen und Verantwortlichkeiten zwischen OEM und Zulieferer erforderlich



7. Normen und Standards
1. AUTOSAR

1. Organisation
2. Schichtenmodell
3. Systementwicklung
4. Bussysteme im KFZ
5. Software-Architektur
6. Anwendungsbeispiele
7. Geplante AUTOSAR-Anwendungen

統合システムに対応する産業界の枠組みを超えた連携強化

- **統合システムとは情報システムと組込システムで構成される大規模システム**
- **重要な社会インフラの多くは統合システム**
  - **情報システム単体、組込システム単体の信頼性・安全性の確保だけではなく、統合システム全体の信頼性・安全性の確保が重要**
- **統合システムの分類**
  - **単一型統合システム： 単一目的のために情報システムと組込システムから構成されたシステム**
  - **結合型統合システム： 異なる目的で構築された情報システムと組込システムが、それぞれのシステムの利便性を高める目的で結合されたシステム**

単一型統合システムの例

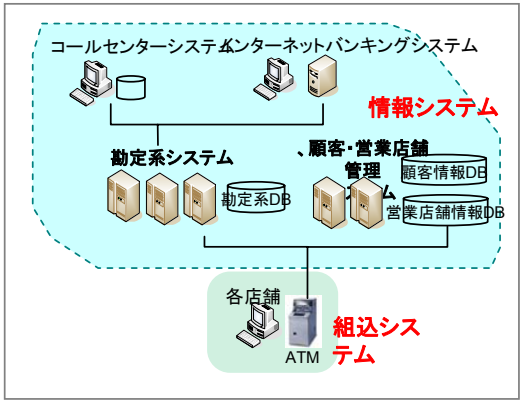
銀行勘定系オンラインシステム

情報システム:

勘定系システム、顧客・営業店舗管理システムなど

組込システム:

ATM、ネットワークルータ、帳票印刷装置など



結合型統合システムの例

交通管制システムとカーナビゲーションシステム

交通管制システム側の目的:

実車両の位置情報や速度情報により、道路交通状況のより正確な把握ができる

カーナビゲーションシステム側の目的:

交通管制システムの渋滞情報から道路の渋滞状況を考慮した経路案内ができる

1. Organisation

2. Schichtenmodell

3. Systementwicklung

4. Bussysteme im KFZ

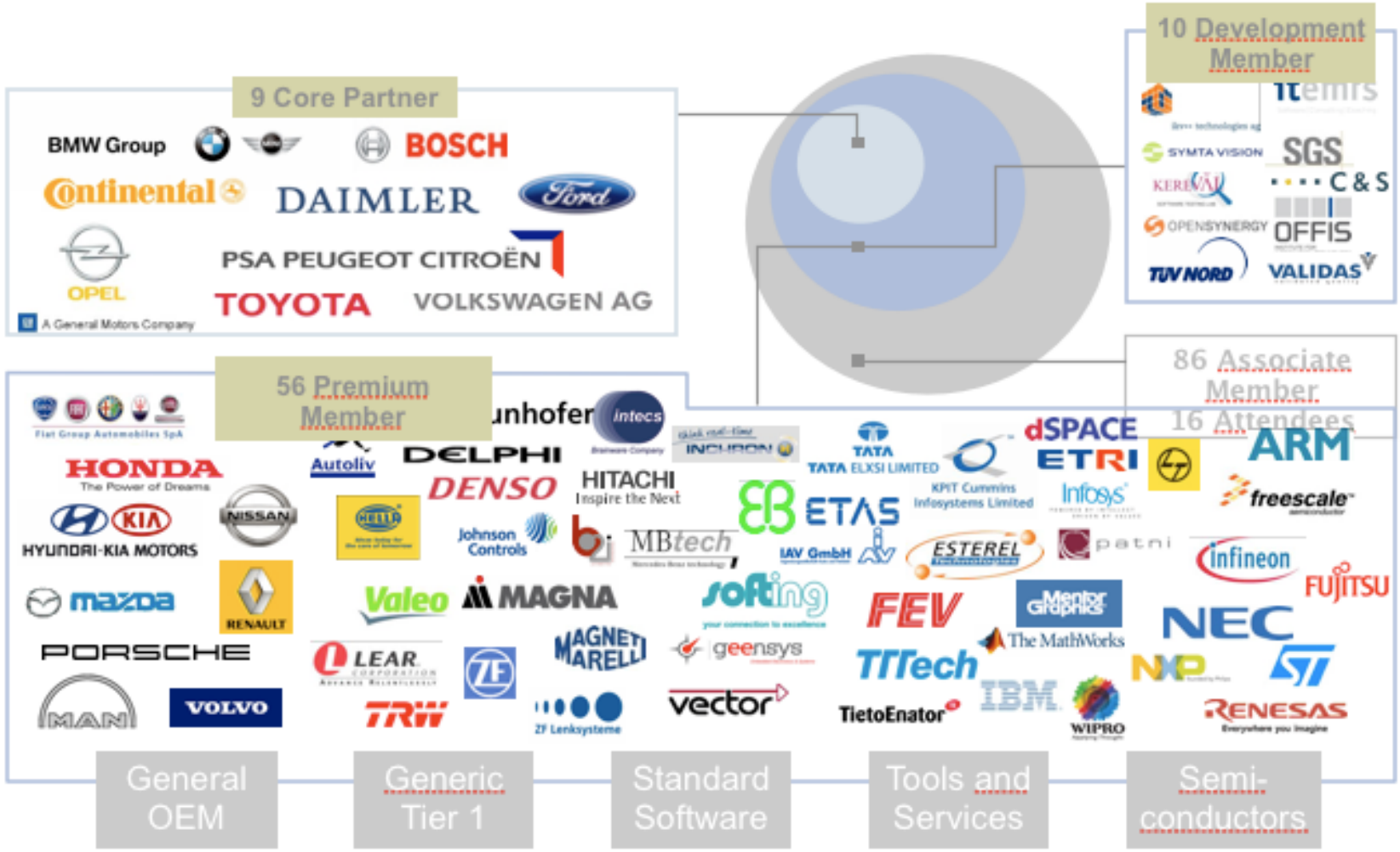
5. Software-Architektur

6. Anwendungsbeispiele

7. Geplante AUTOSAR-Anwendungen

AUTOSAR – Core Partners and Members (Phase II)

- Ca. 170 Firmen (Stand Ende 2009)



Core Partners in Phase III

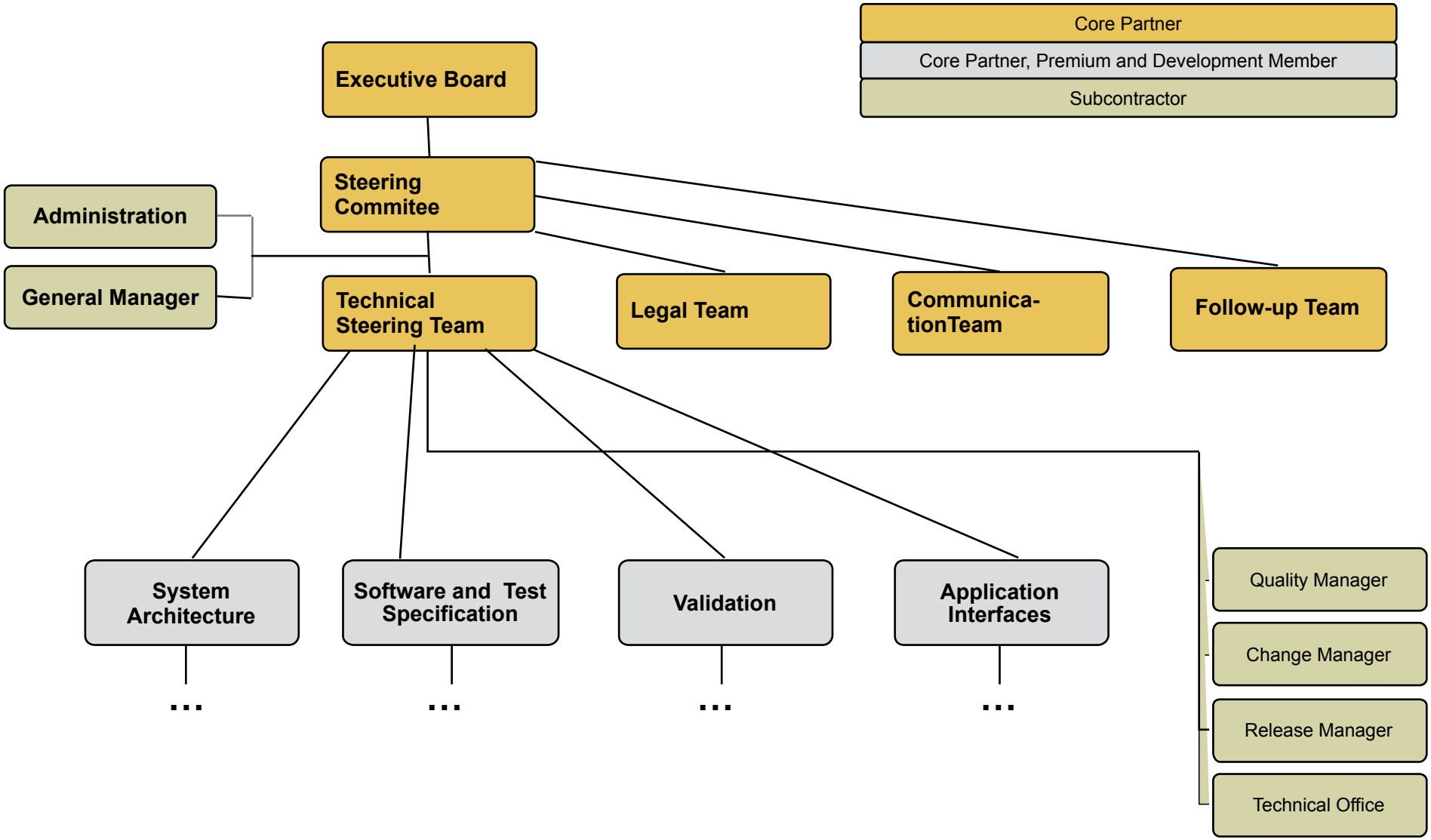
- Initial discussions 2002: BMW, Bosch, Continental, DaimlerChrysler and Volkswagen, partners were joined soon afterwards by Siemens VDO.
- Additional Core Partners 2003: Ford, Peugeot Citroën, Toyota, 2004: GM
- 2008 Siemens VDO became part of Continental.
- Phase III will start with 8 Core Partners

- GM announced to continue in phase III as Premium Member
- The 8 Core Partners agreed on the phase III 2010-2012 development contract
- Phase III planning started and is well under way

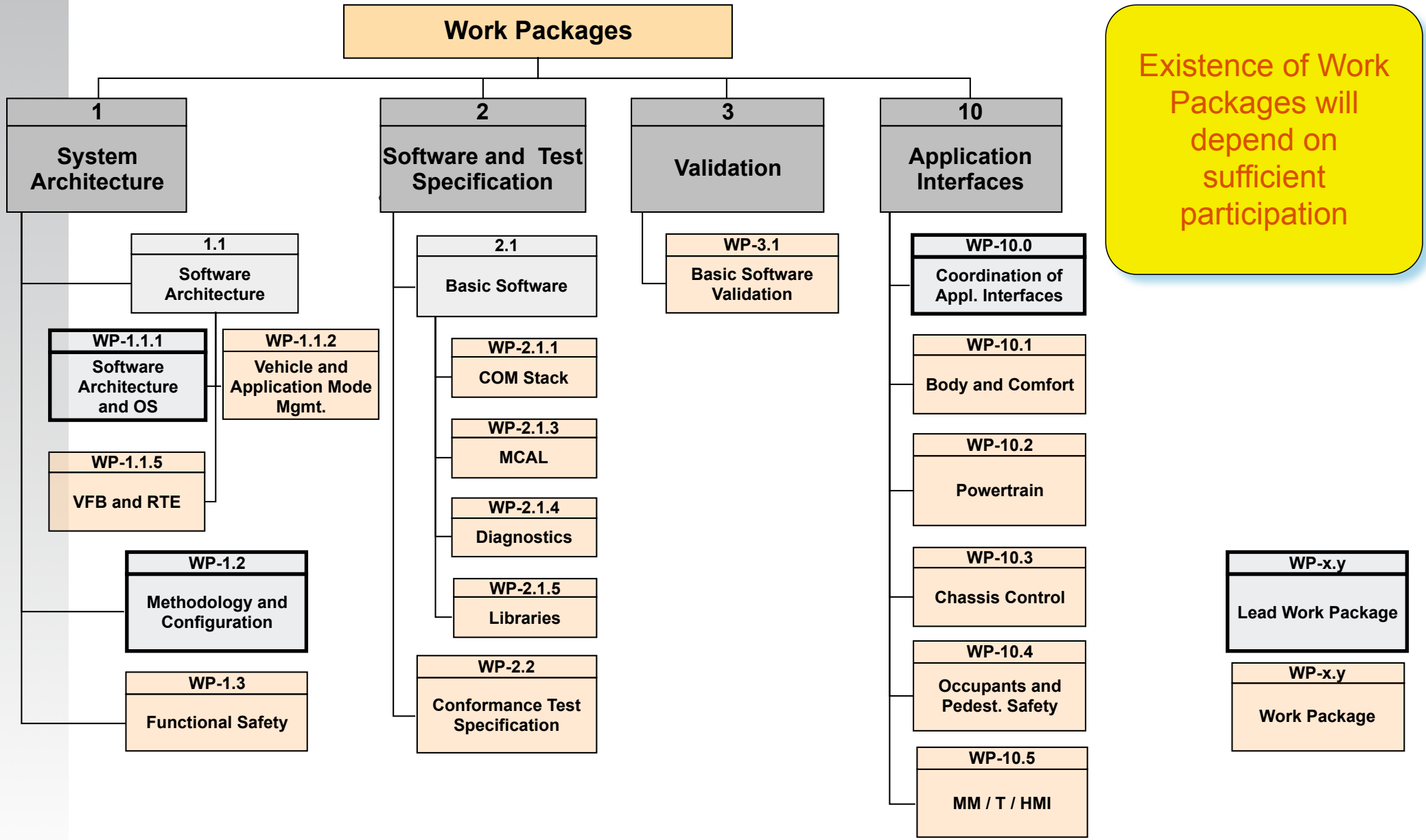
AUTOSAR Membership Levels

- Core Partners
  - Organizational control
  - Technical contributions
  - Administrative control
  - Definition of external Information (web-release, clearance, etc.)
  - Leadership of Working Groups
  - Involvement in Working Groups
  - Utilization of AUTOSAR standard
- Premium Members
  - Leadership of Working Groups
  - Involvement in Working Groups
  - Technical contributions
  - Access to current information
  - Utilization of AUTOSAR standard
- Associate Members
  - Access to finalized documents
  - Utilization of AUTOSAR standard
- Development Members (for small companies with specific expertise)
  - Technical contributions
  - Access to current information
  - Utilization of AUTOSAR standard
- Attendee (e.g. Universities)
  - Support Role

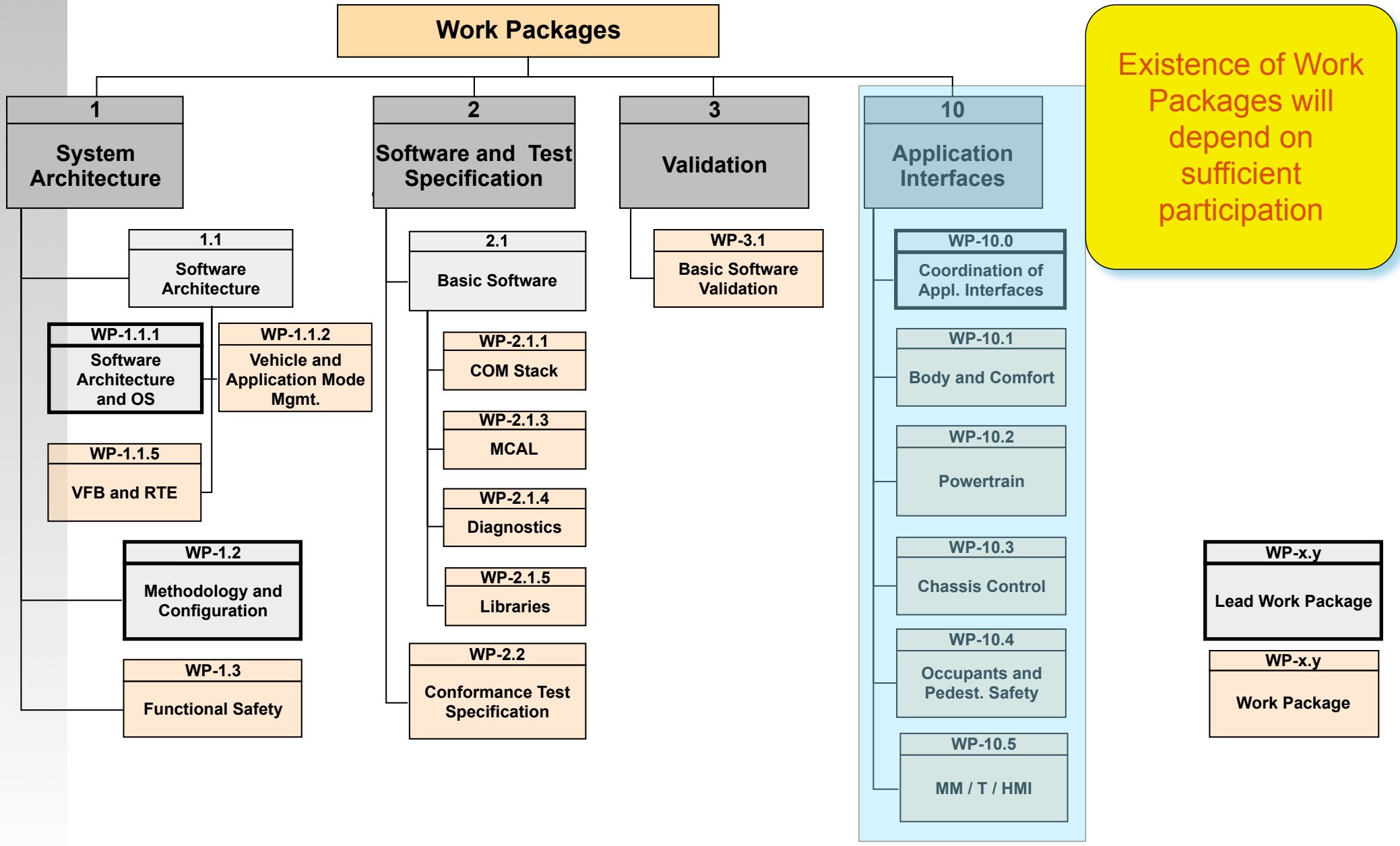
AUTOSAR Phase III Organization



Initial WP Structure Phase III



Initial WP structure Phase III



Elektronische Systeme im Fahrzeug
(4.1 Domänen)

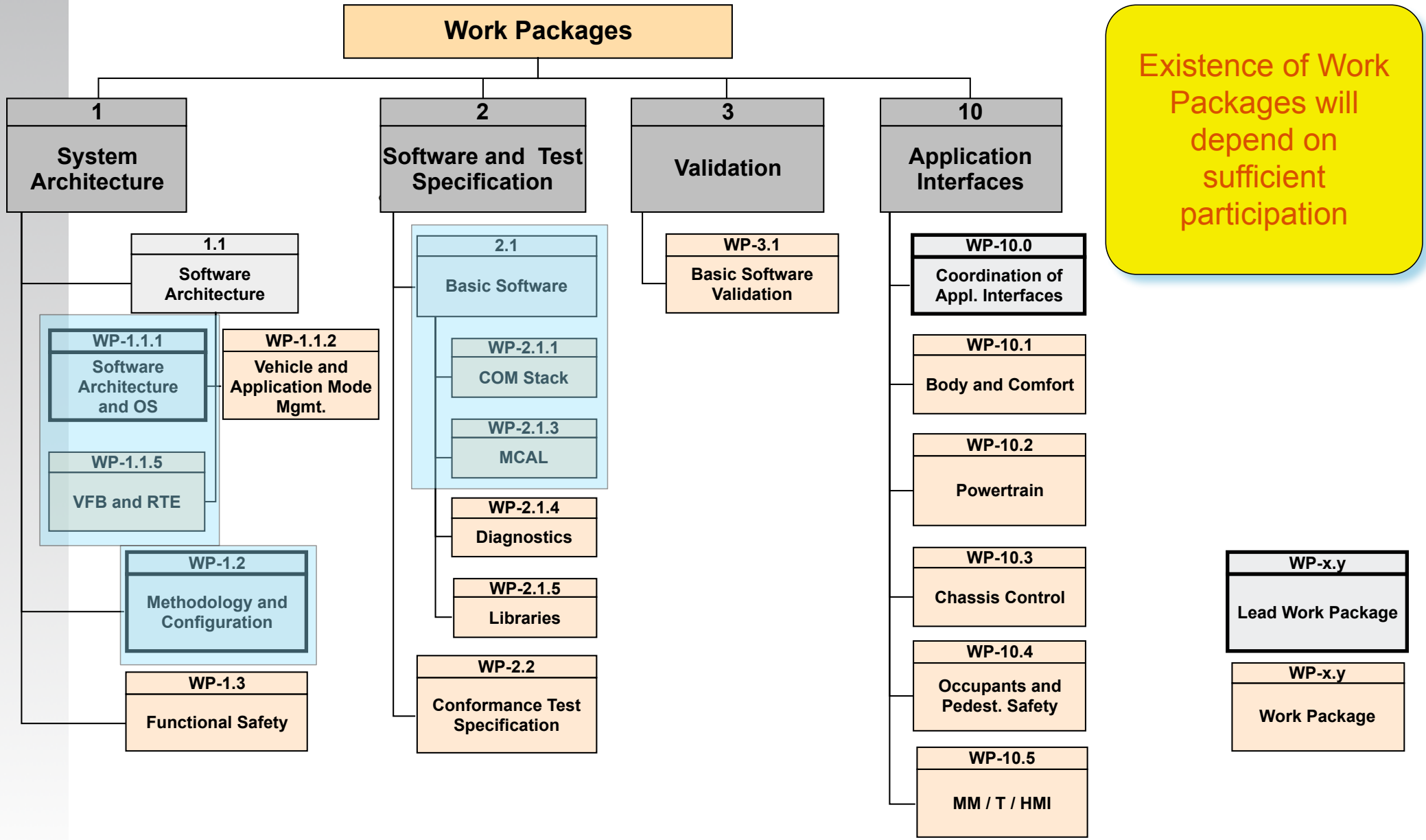
Anwendungsdomänen und elektronische Subsysteme
(in diesem Abschnitt nach Schäuffele / Zurawka: Automotive Software Engineering)

- Antriebsstrang (Powertrain)
- Fahrwerk (Chassis)
- Karosserie (Body)
- Multi-Media (Telematics)

Auch andere Klassifizierungen gebräuchlich
(Beispiel Mercedes-Benz Technik transparent)

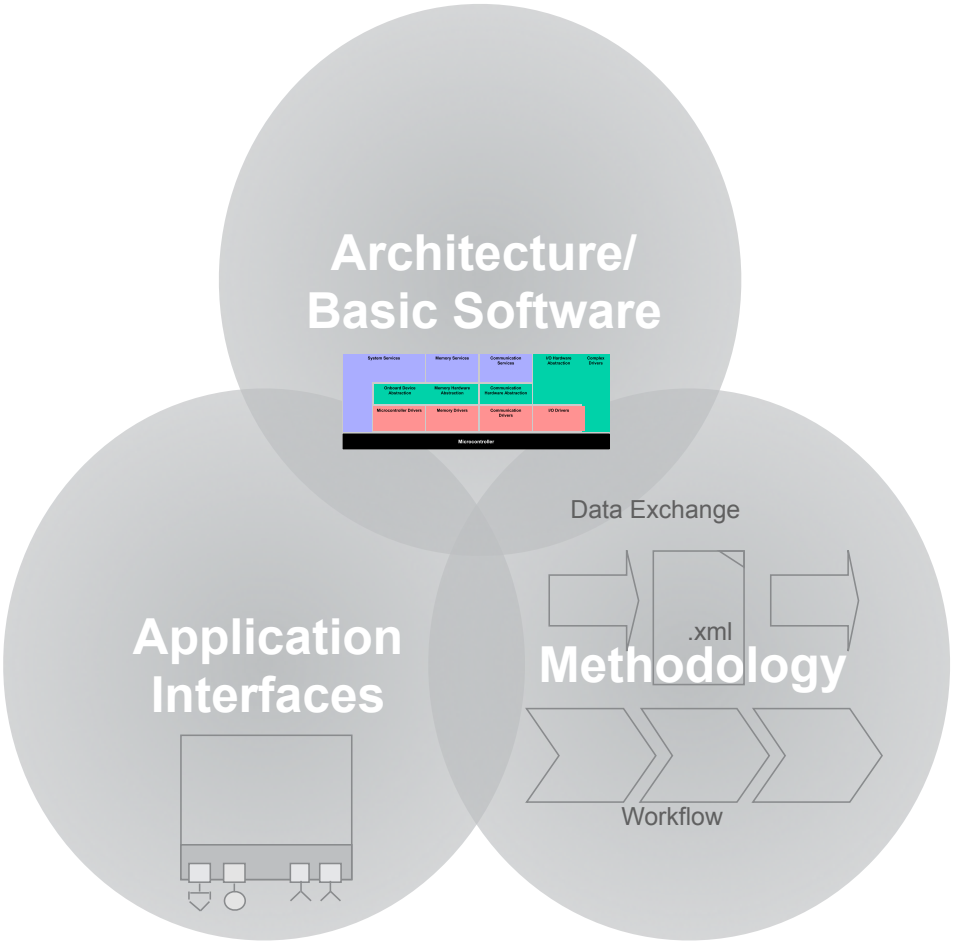
- Aktive Sicherheit
- Passive Sicherheit
- Karosserie
- Fahrwerk
- Innenraumtechnik
- Elektronik
- Motoren/Getriebe

Initial WP structure Phase III



What Daimler expects from AUTOSAR

- Provisioning of an interoperable reliable software kit. Increase of quality and development speed through a comprehensive and standardized design and implementation approach.
- Inter OEM exchange through interoperable software kits
- Reduction of testing effort in the automotive community
- Internal and external libraries for off-the-shelf applications
- Convenient integration into the development chain of Tier1s and OEMs
- Faster SW integration processes
- Standard application interfaces for ‘SW as a product’



Source: AUTOSAR PM Conference 02-2008 / AUTOSAR - a key enabler for comprehensive E/E standardization / S. Wolfsried, Daimler AG

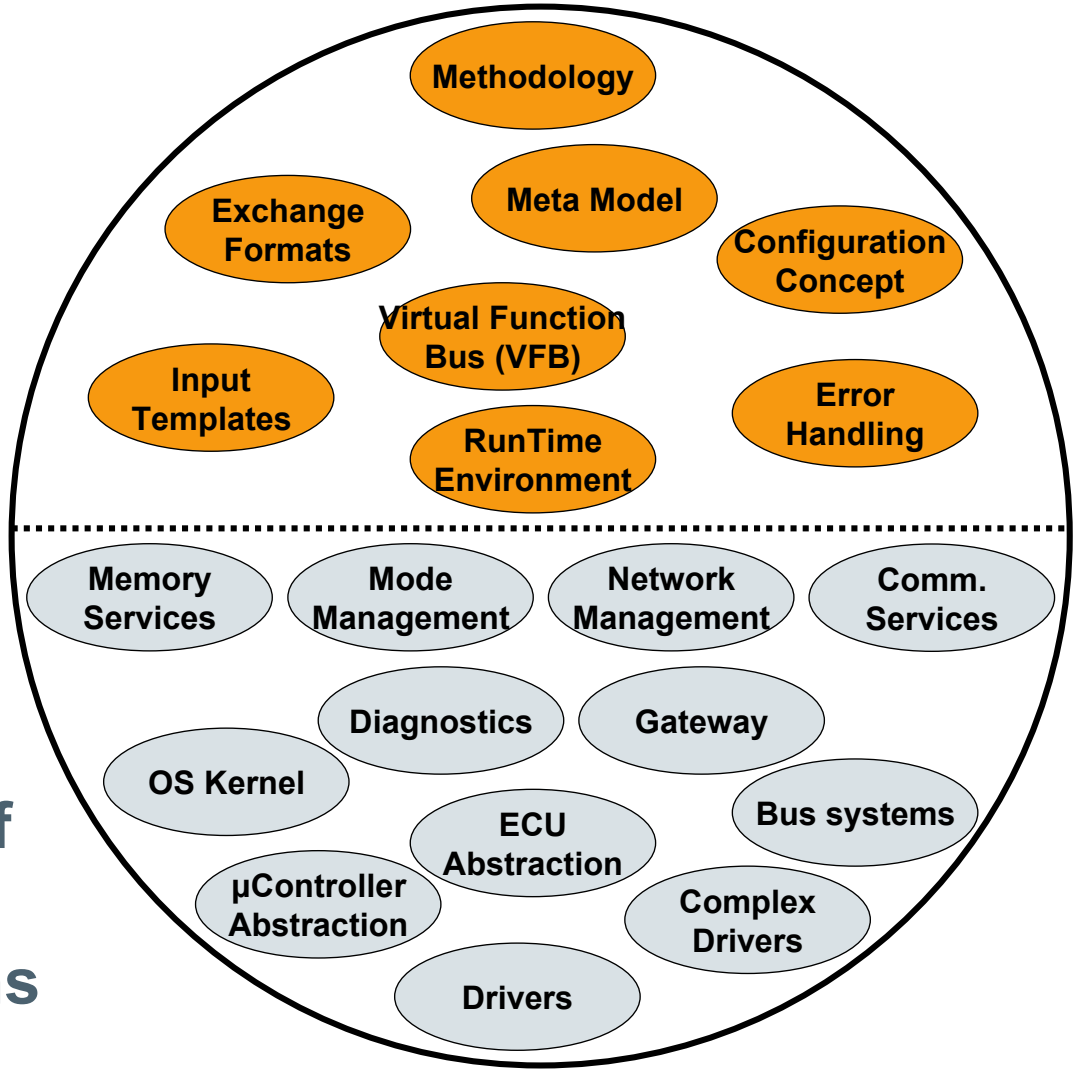
Preconditions for the introduction of AUTOSAR

- Introduction of a standard depends on its maturity and the benefits of its Geplante AUTOSAR-Anwendungen: Daimlers assessment is positive for AUTOSAR 3.0.
- Maturity of a new standard has to be assured
  - The release of AUTOSAR 3.0 has been determined to be the sweet spot for introduction according to our maturity and benefit assessment
- Maintenance shall be managed
  - The AUTOSAR community is seen capable to assure this
- Conformance tests must be available to ensure the standard's continuous integrity
  - Conformance tests are essential for guaranteeing AUTOSAR's integrity as a standard.
  - Conformance tests not yet available
- Today the standard appears overloaded:
  too many requirements with a "one-size-fits-all" approach

Source: AUTOSAR PM Conference 02-2008 / AUTOSAR - a key enabler for comprehensive E/E standardization / S. Wolfsried, Daimler AG

Technical scope of AUTOSAR

New concepts

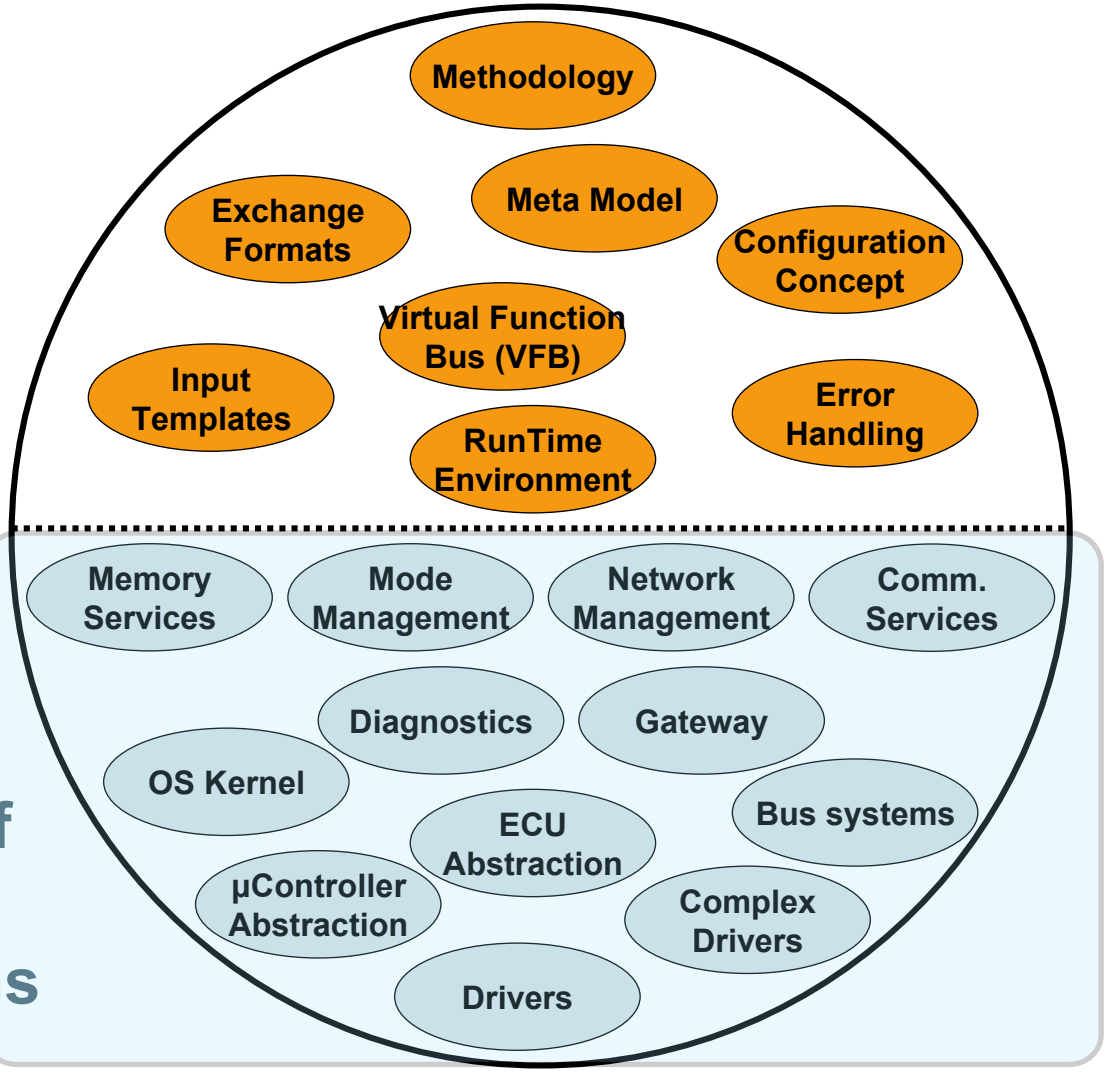


Industry-wide consolidation of ‚existing' basic software designs

Technical scope of AUTOSAR

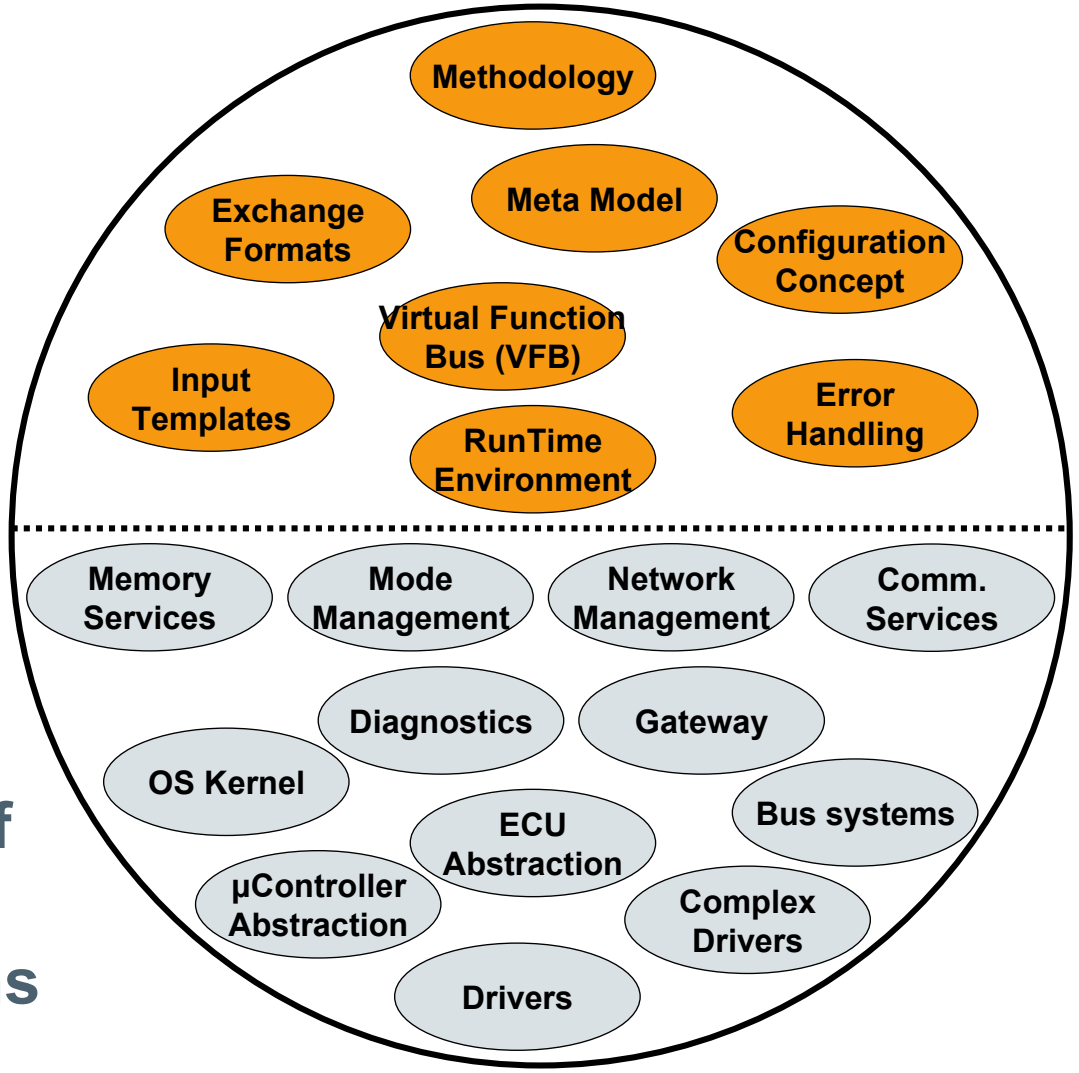
New concepts

Industry-wide consolidation of ‚existing' basic software designs



Technical scope of AUTOSAR

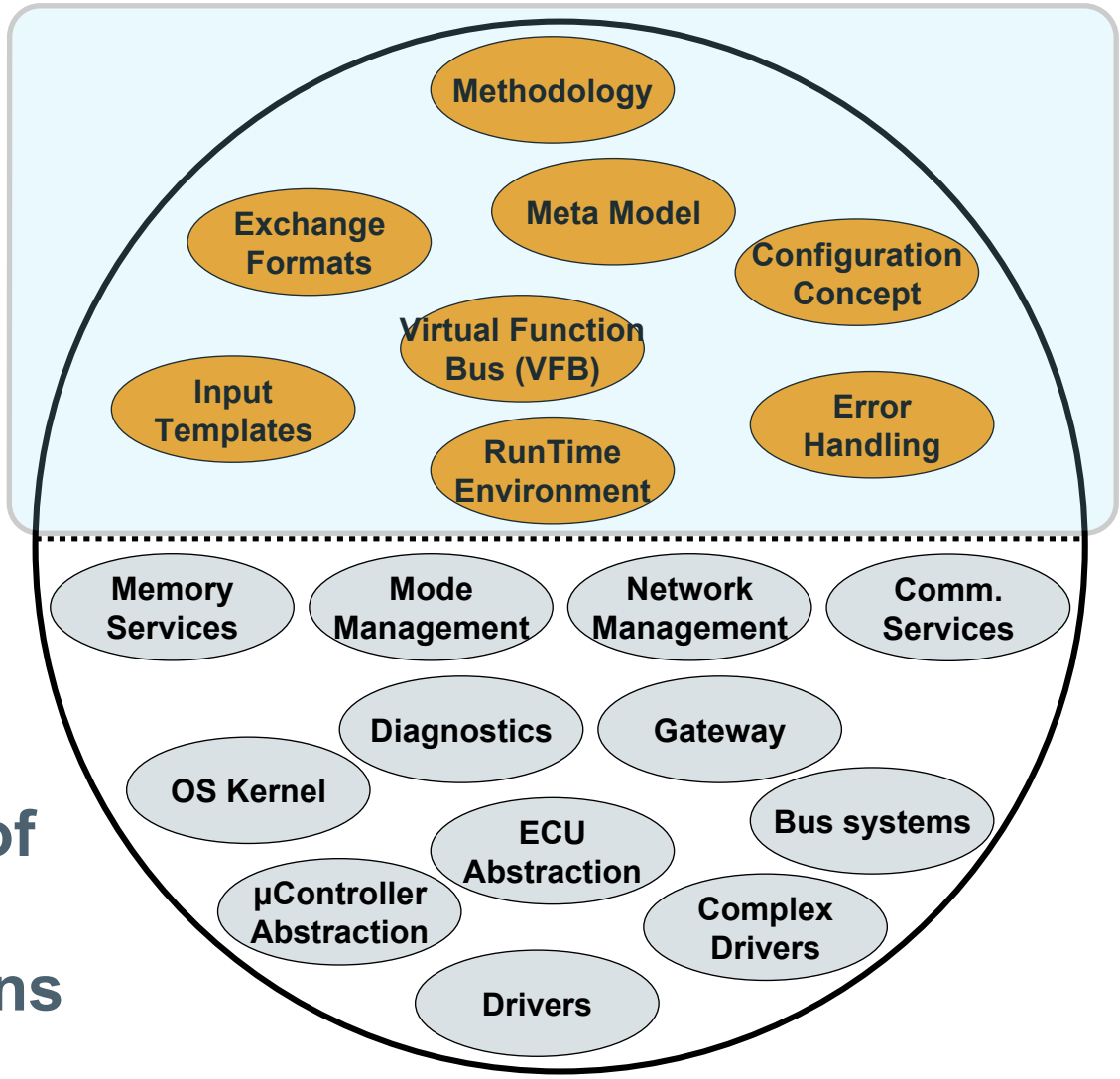
New concepts



Industry-wide consolidation of ‚existing' basic software designs

Technical scope of AUTOSAR

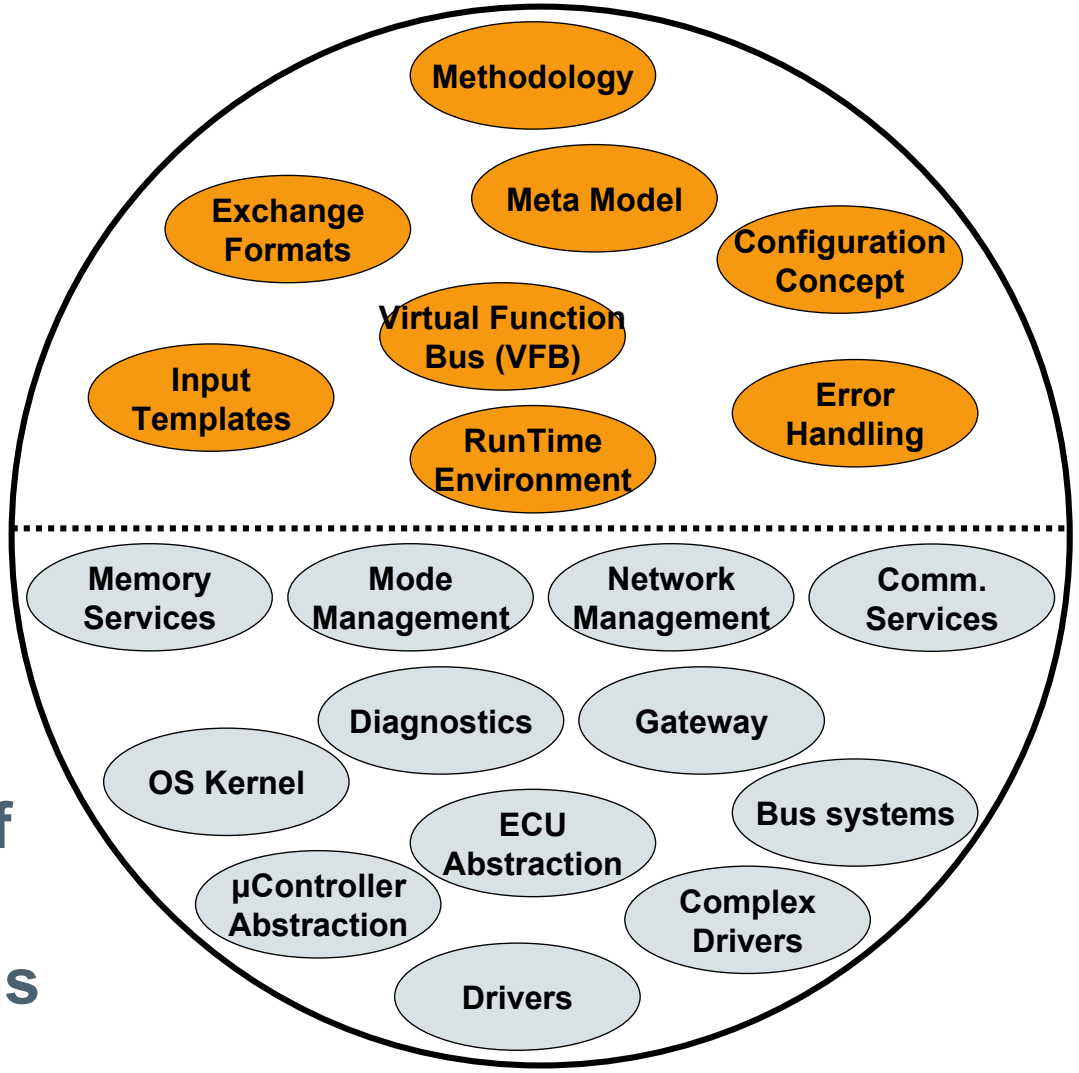
New concepts



Industry-wide consolidation of ‚existing' basic software designs

Technical scope of AUTOSAR

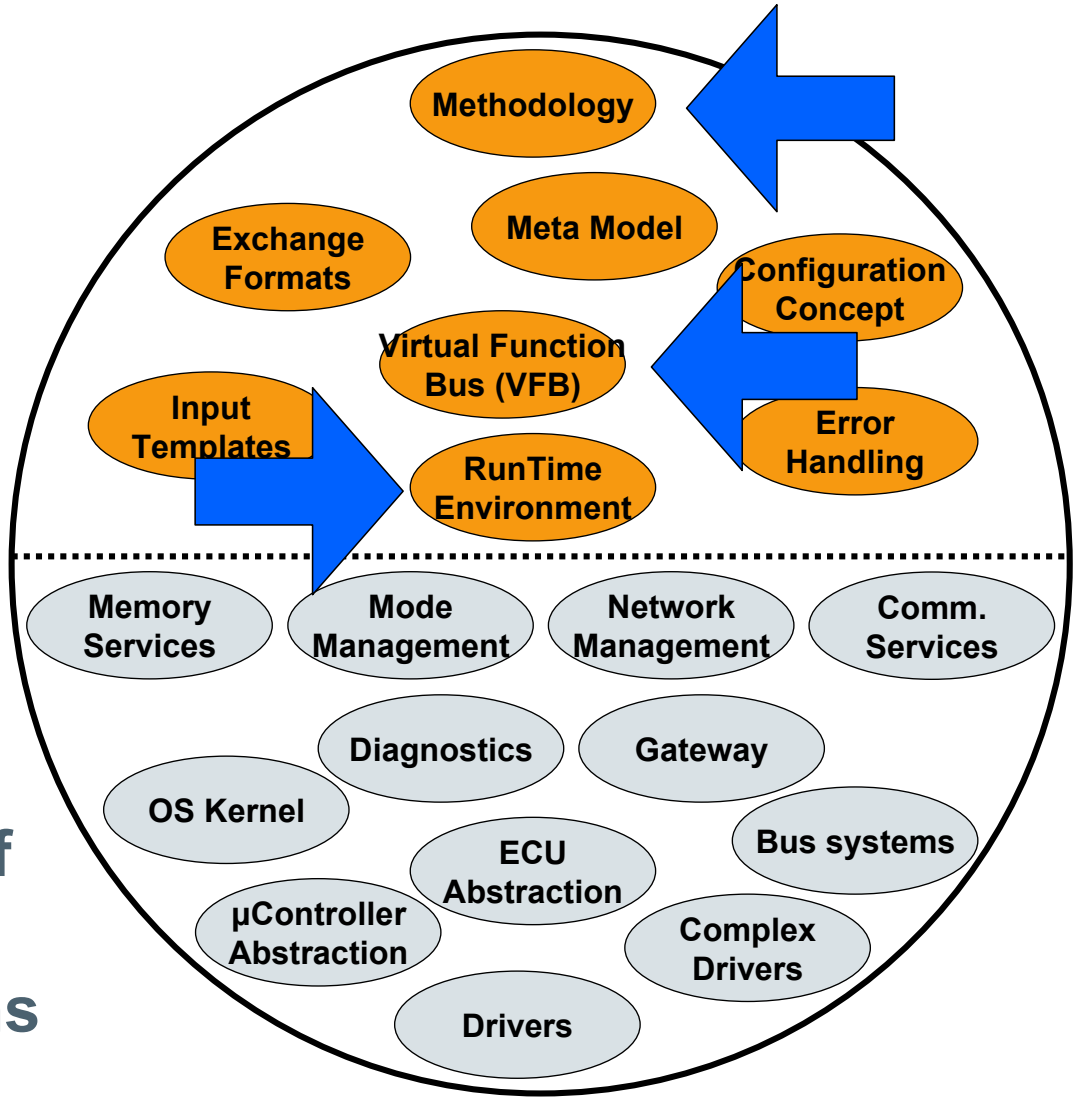
New concepts



Industry-wide consolidation of ‚existing' basic software designs

Technical scope of AUTOSAR

New concepts

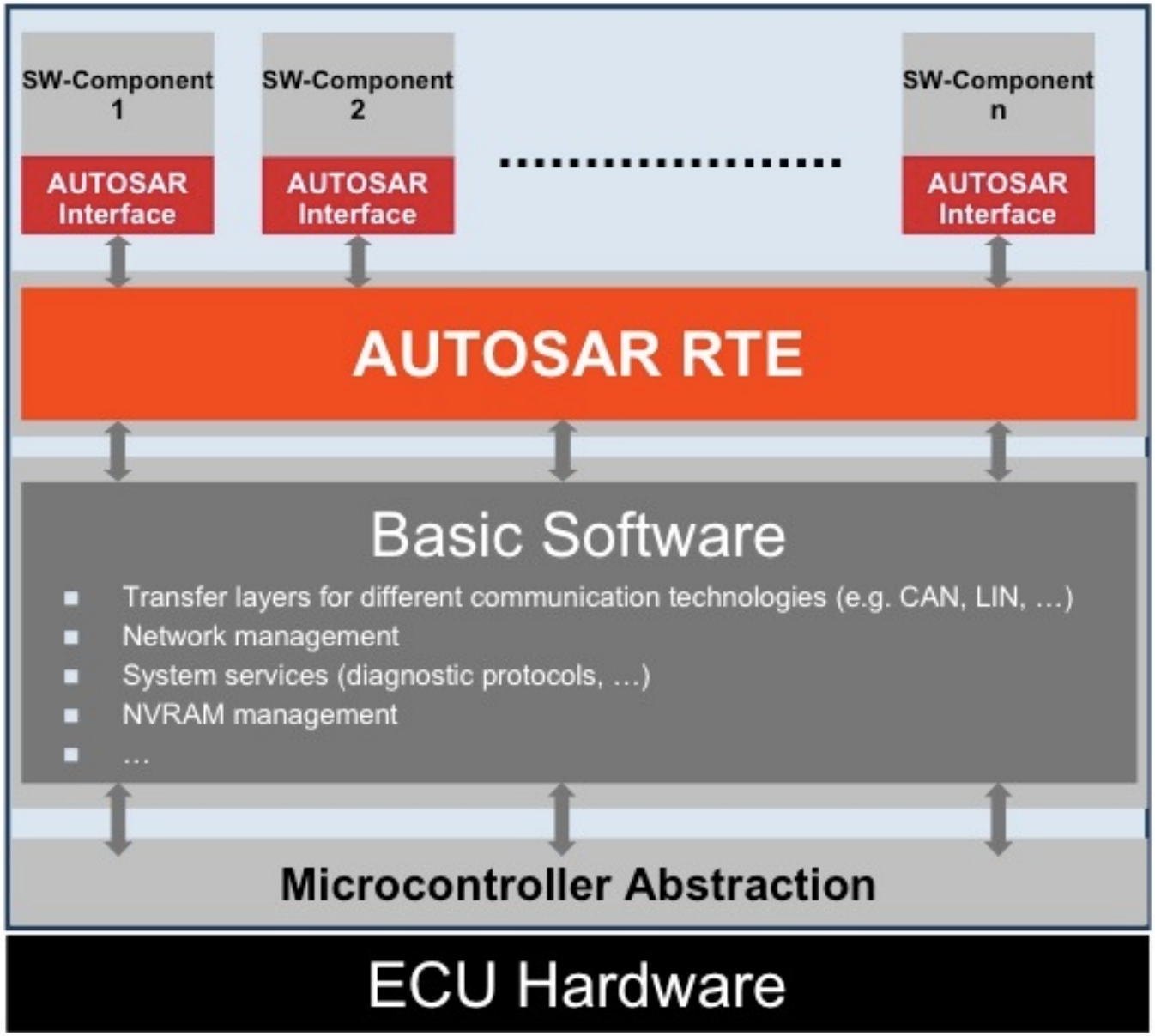


Industry-wide consolidation of ‚existing' basic software designs

7. Normen und Standards
1. AUTOSAR

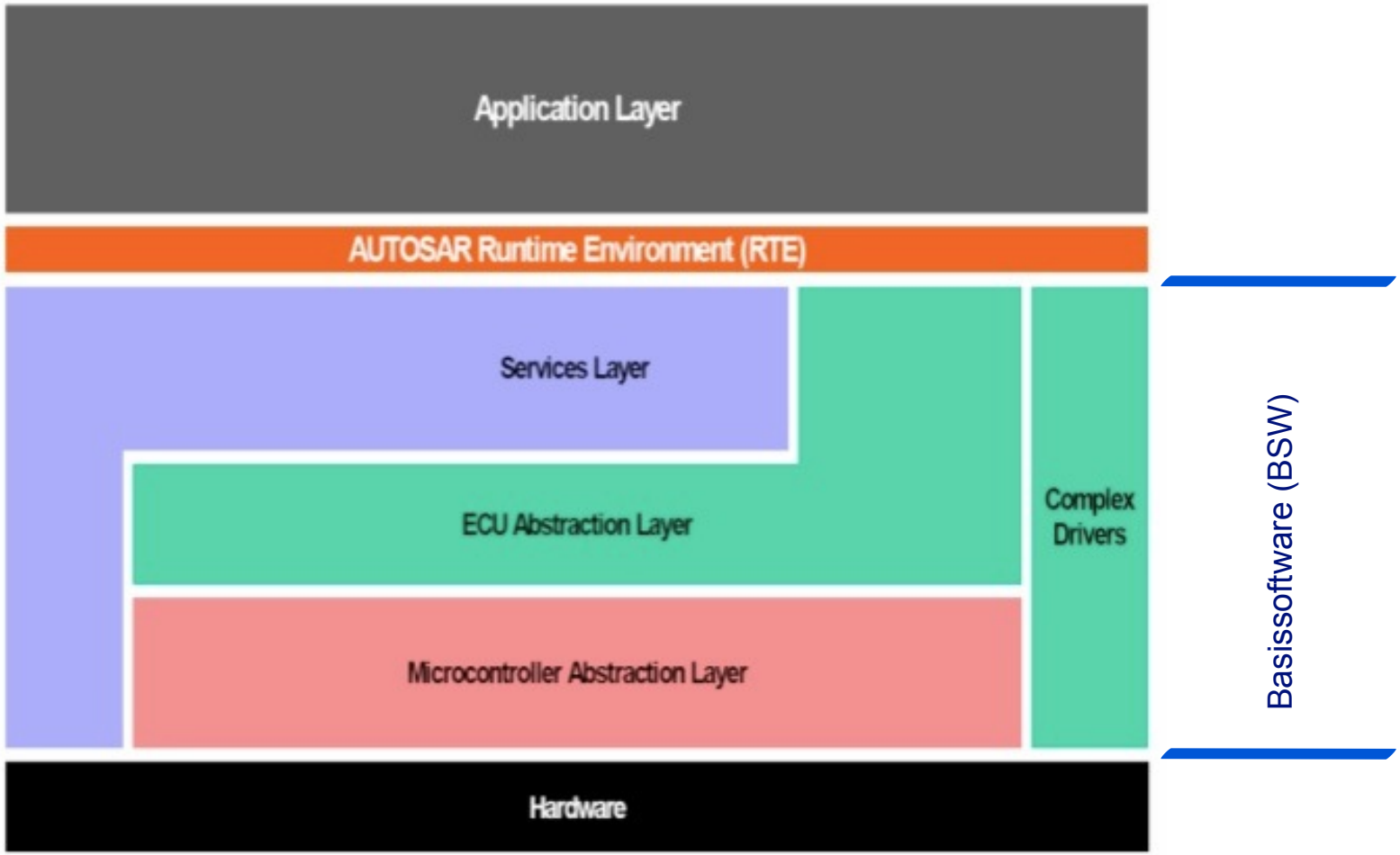
1. Organisation
2. **Schichtenmodell**
3. Systementwicklung
4. Bussysteme im KFZ
5. Software-Architektur
6. Anwendungsbeispiele
7. Geplante AUTOSAR-Anwendungen

AUTOSAR Architekturkonzept



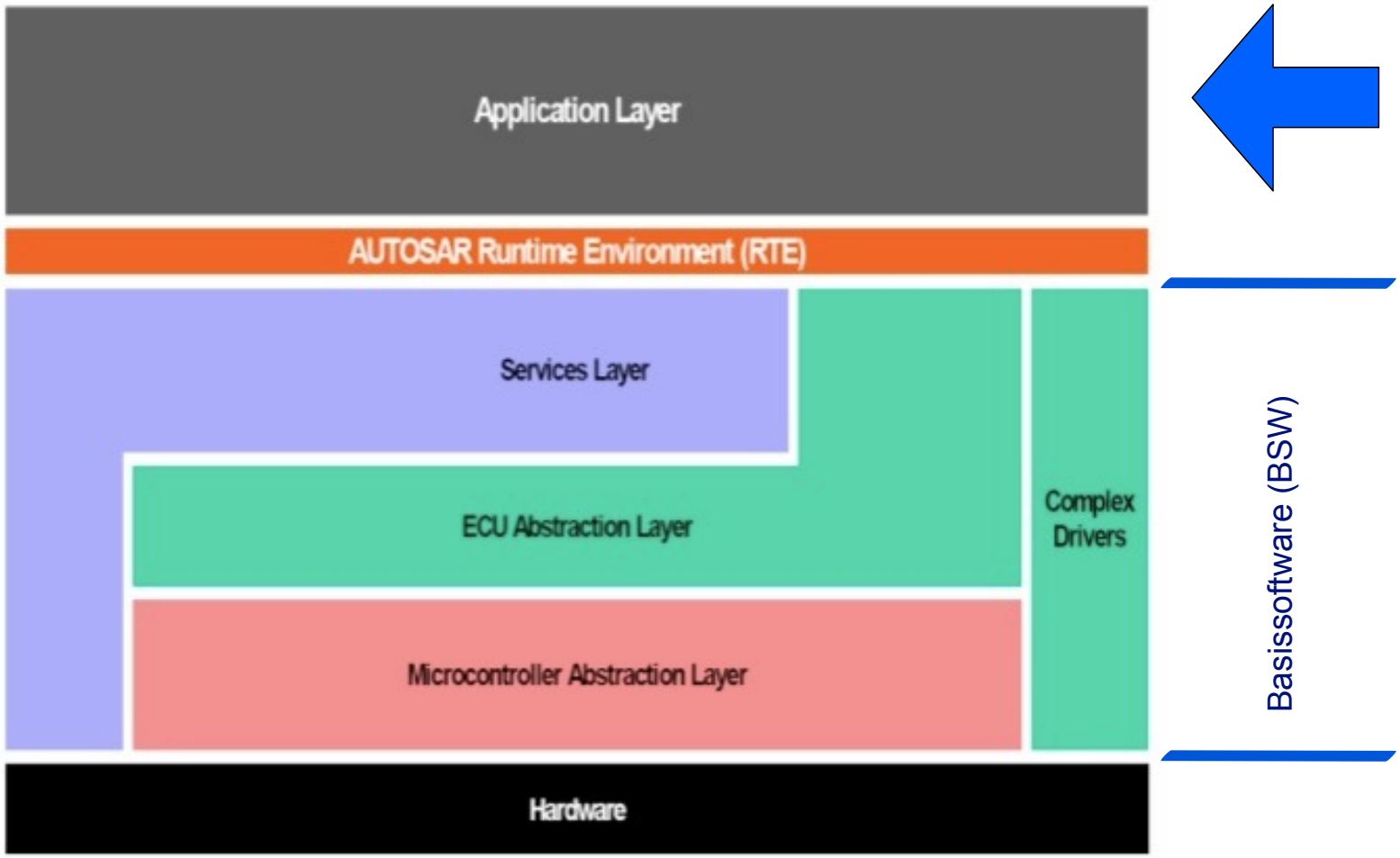
AUTOSAR-Schichtenmodell
Abstraktionsschichten der Steuergerätesoftware

- Anwendungsschicht (Application Layer)
- Laufzeitumgebung (Runtime Environment, RTE)
- Basissoftware (BSW)



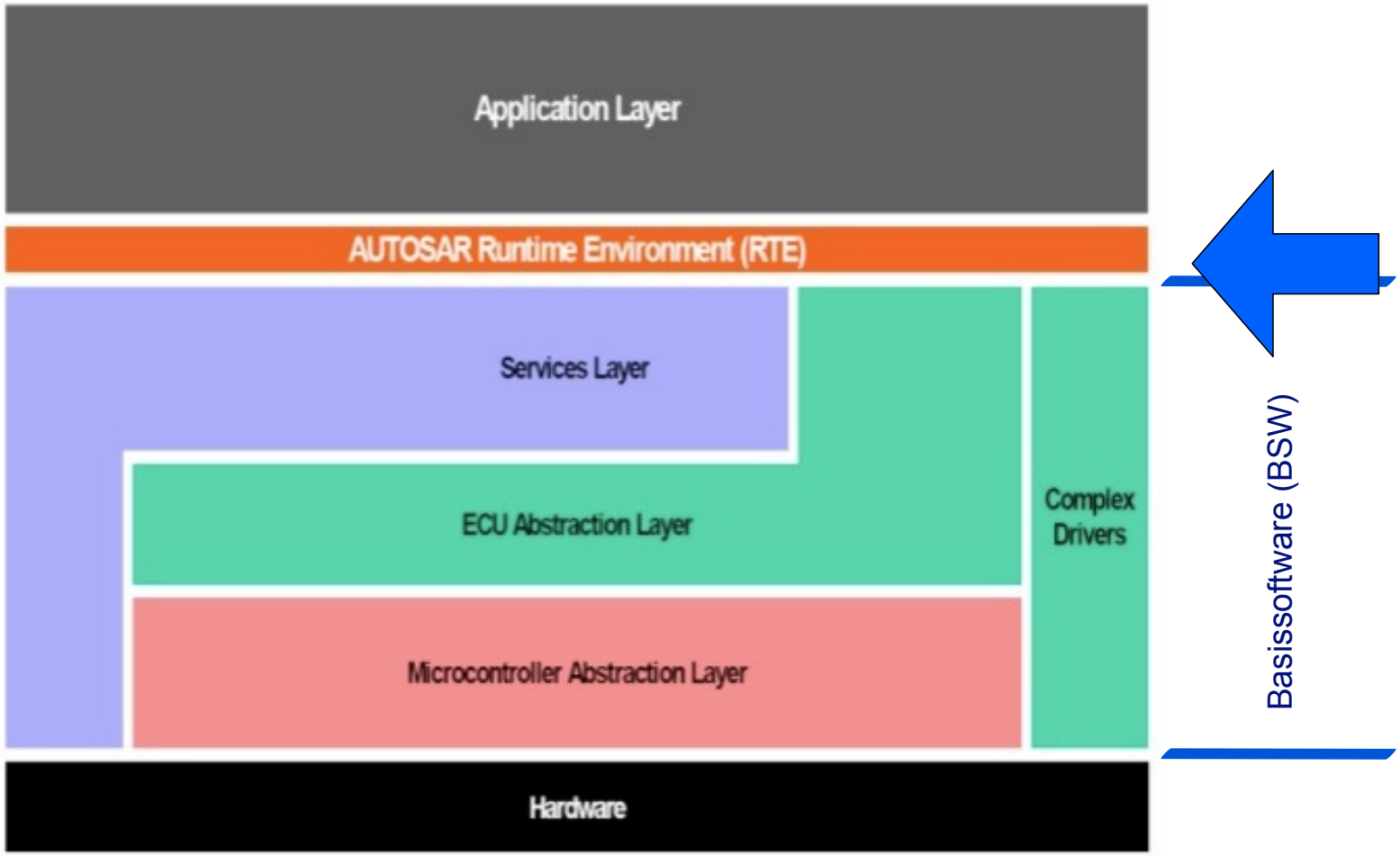
AUTOSAR-Schichtenmodell
Anwendungsschicht (Application Layer)

- Der Application Layer realisiert die Anwendungsfunktionalität des Steuergeräts mittels Anwendungs-Softwarekomponenten (SWC - Software Component).



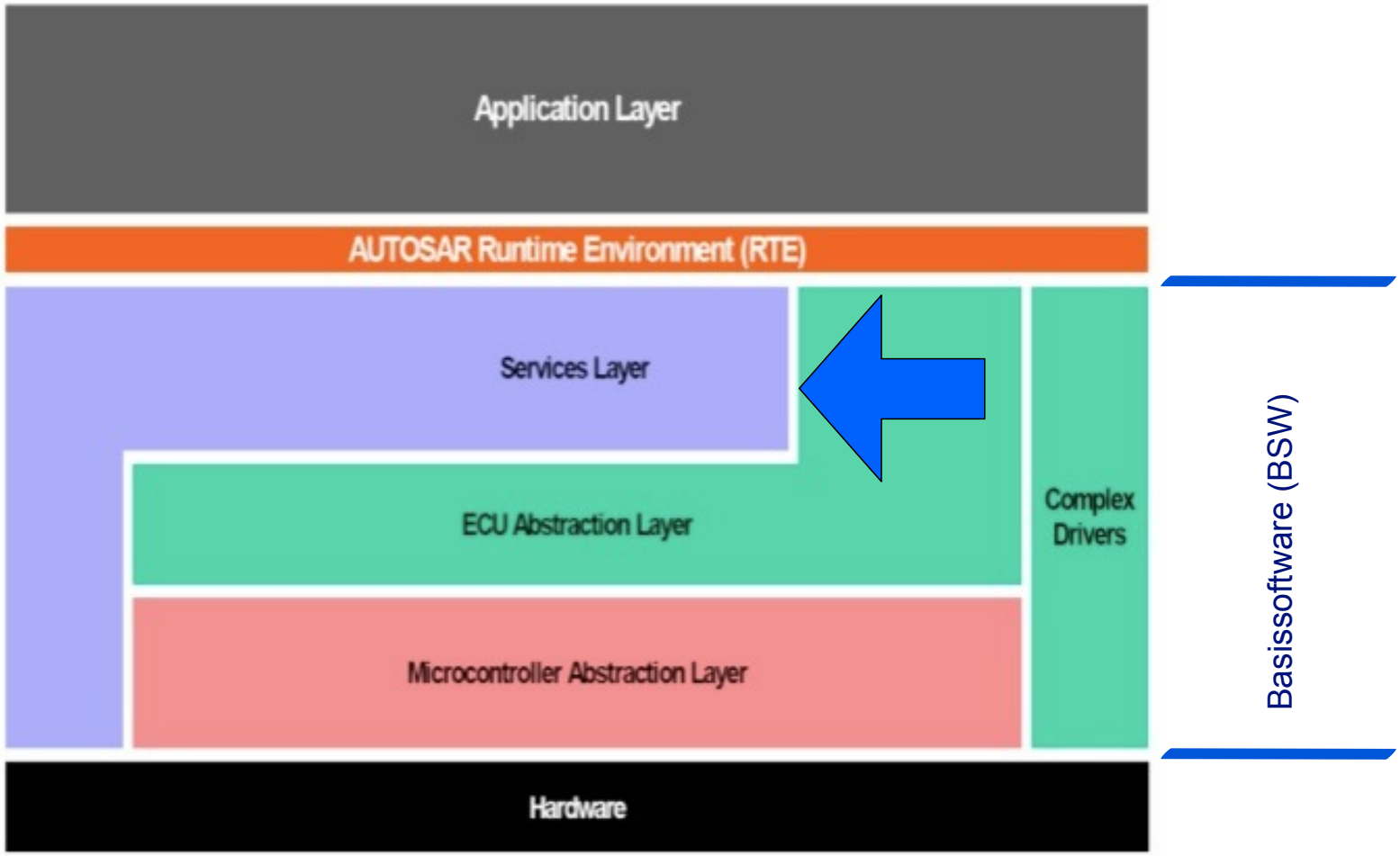
AUTOSAR-Schichtenmodell
Laufzeitumgebung (Runtime Environment, RTE)

- Die Laufzeitumgebung (Runtime Environment, RTE) integriert den Application Layer mit der Basissoftware (BSW). Sie implementiert den Datenaustausch zwischen Anwendungs-Softwarekomponenten (SWC) und steuert die Interaktion zwischen SWCs und der BSW.



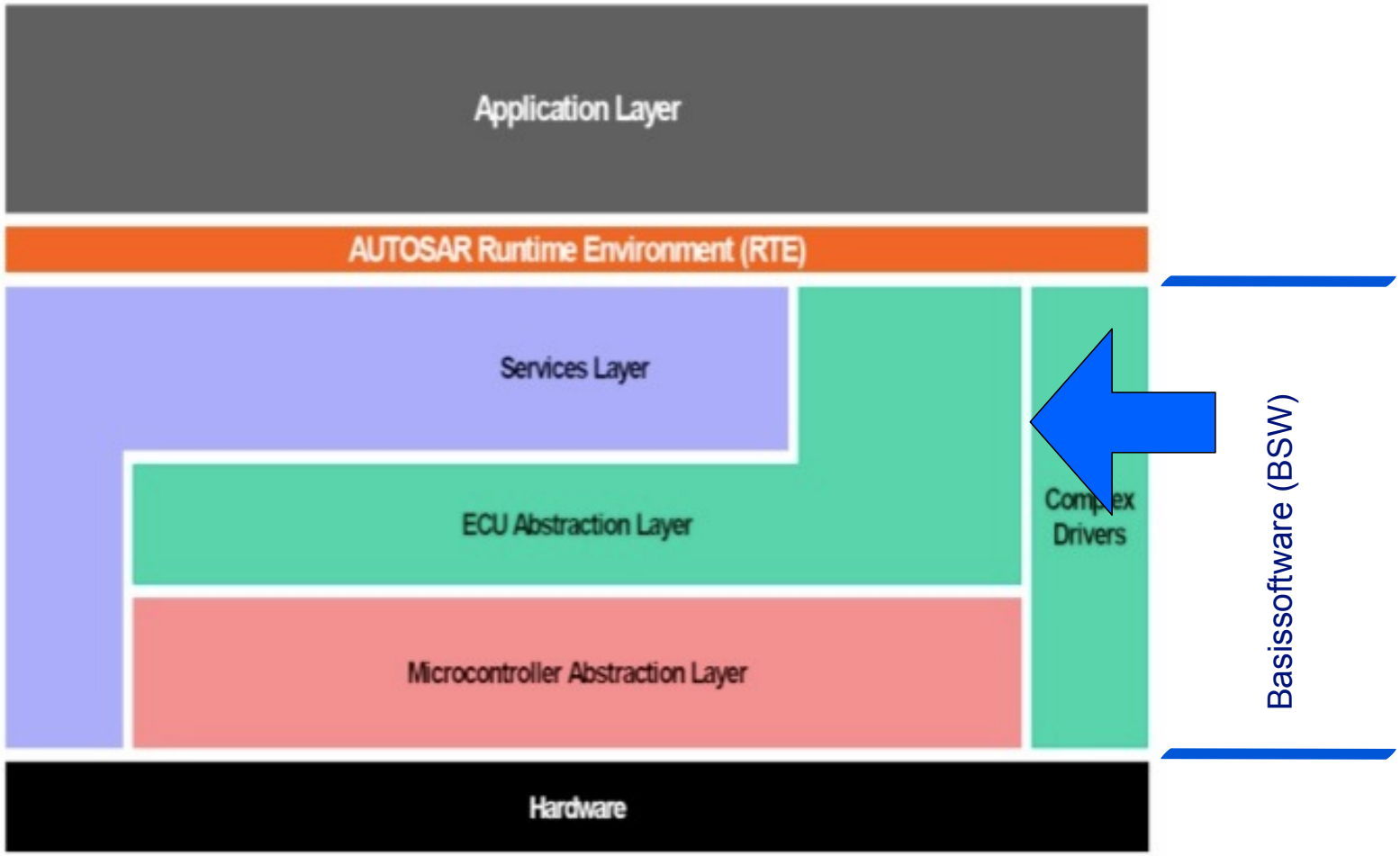
AUTOSAR-Schichtenmodell
BSW - Service Layer

- Der Service Layer stellt verschiedene Arten von Hintergrunddiensten wie Netzwerkdienstze, Speicherverwaltung und Buskommunikationsdienste bereit. Das Betriebssystem ist ebenfalls in dieser Schicht enthalten.



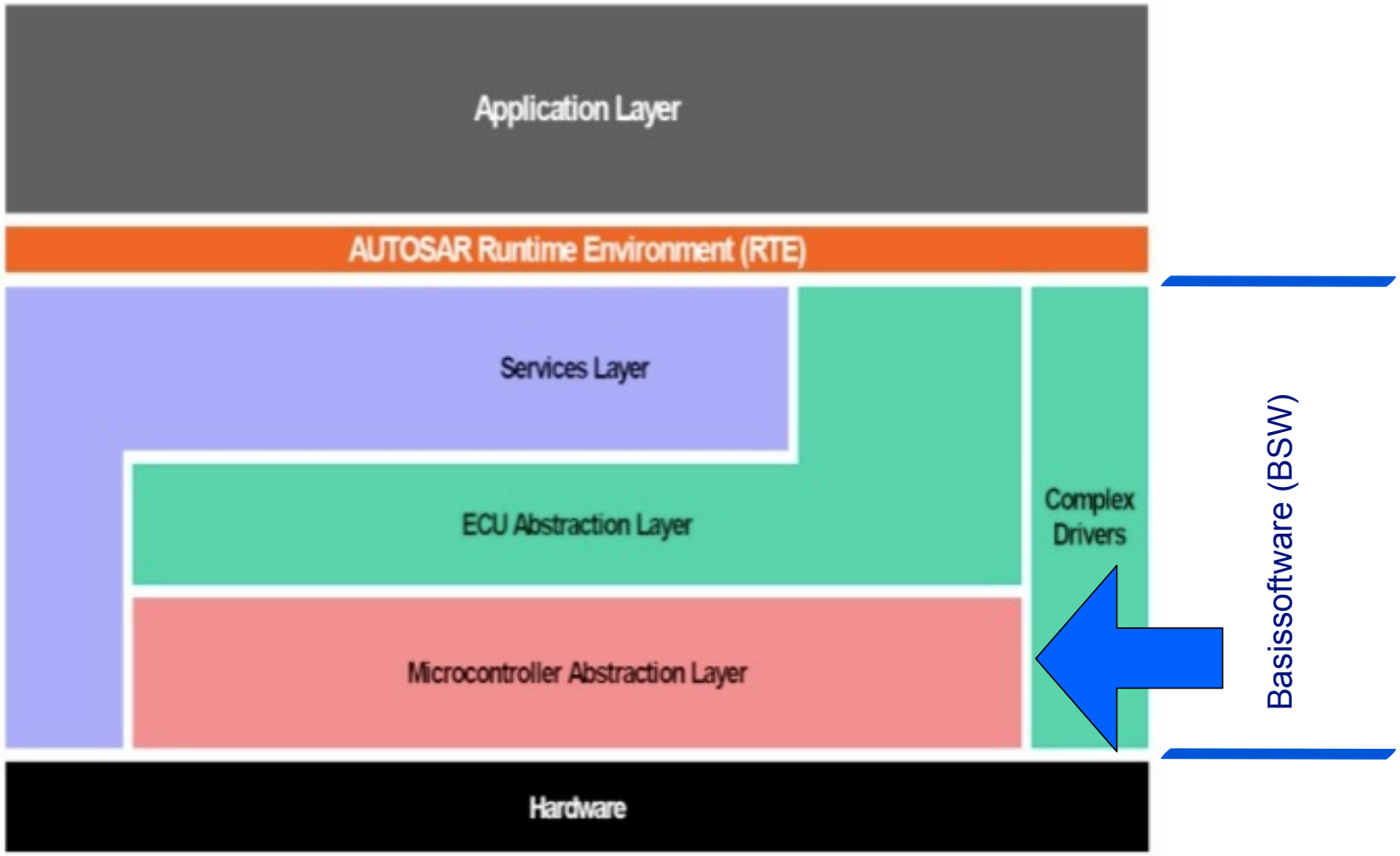
AUTOSAR-Schichtenmodell
BSW - ECU Abstraction Layer

- Der ECU Abstraction Layer bietet einen einheitlichen Zugriff auf alle Funktionalitäten eines Steuergeräts wie Kommunikation, Speicher oder E/A.
- Ziel: Unabhängigkeit der höheren Schichten von der Steuergeräte-Hardware



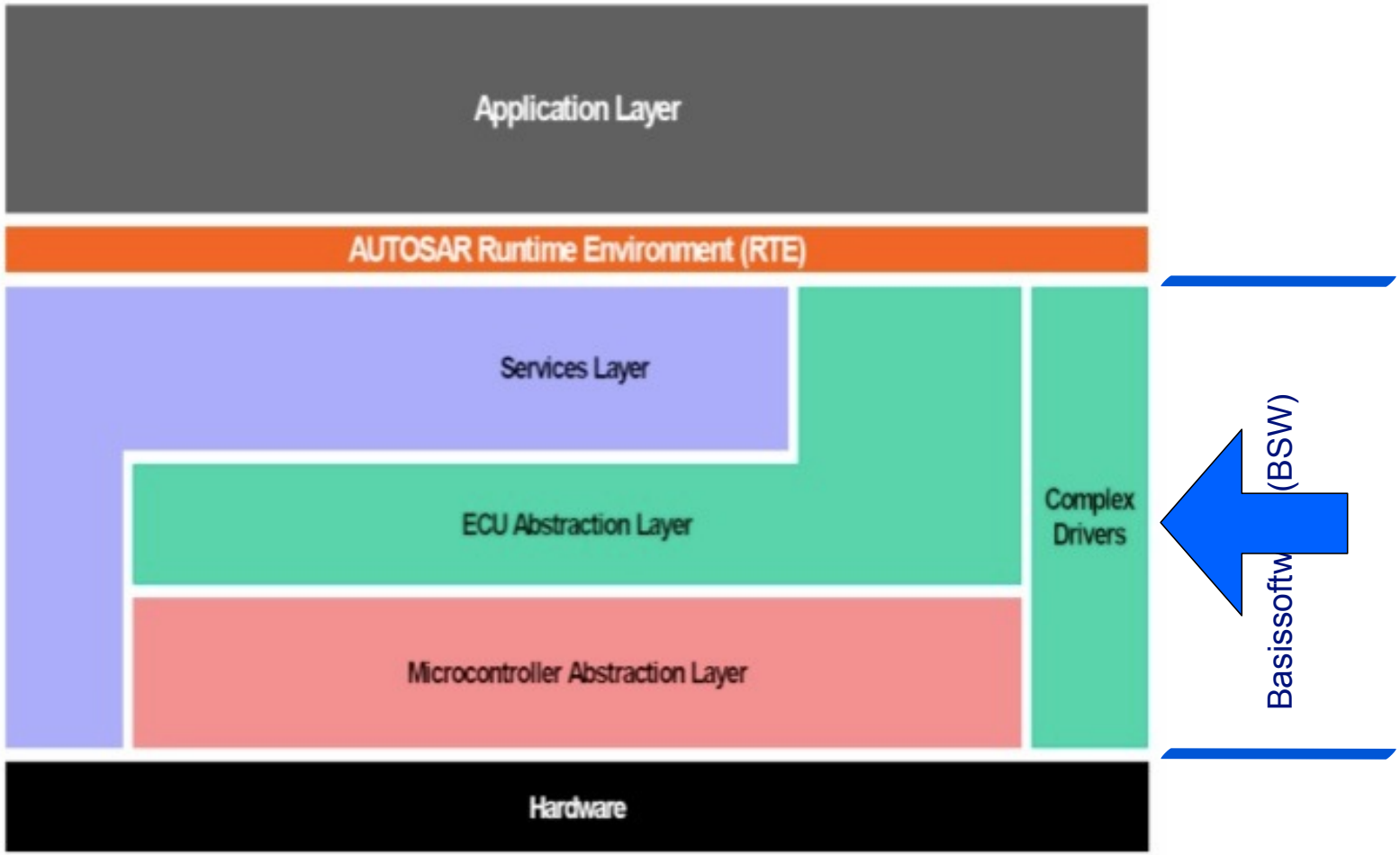
AUTOSAR-Schichtenmodell
BSW - Microcontroller Abstraction Layer

- Der Microcontroller Abstraction Layer (MCAL) bietet beispielsweise Treiber für den Zugriff auf Kommunikation, Speicher und E/A des Mikrocontrollers.
- Ziel: Unabhängigkeit der höheren Schichten von der Mikrocontroller-Hardware



AUTOSAR-Schichtenmodell
BSW - Complex Drivers

- Die Complex Drivers enthalten die in AUTOSAR nicht standardisierten Treiber für die spezifischen Eigenschaften eines Mikrocontrollers oder Steuergeräts.

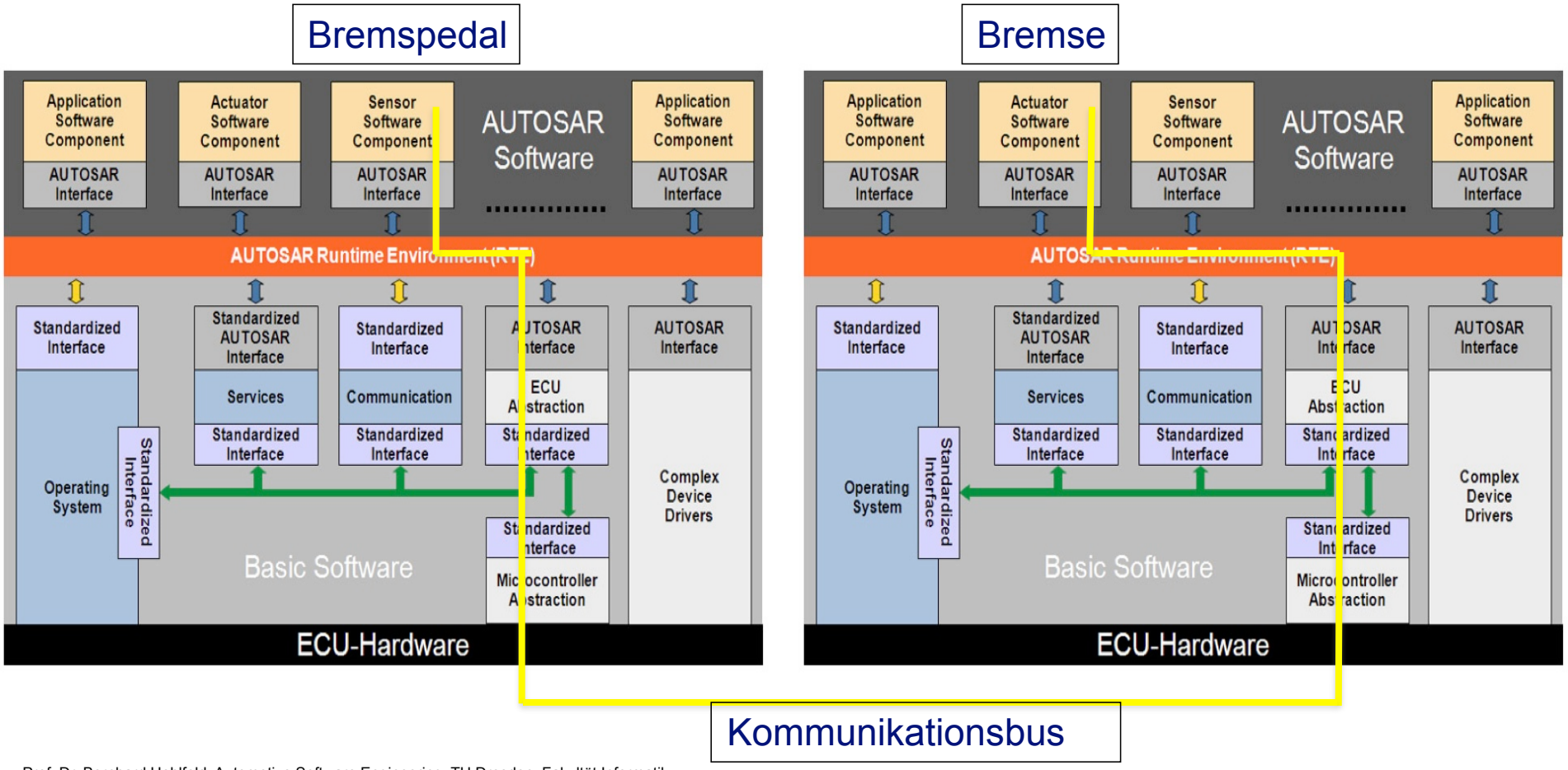


AUTOSAR-Schichtenmodell
BSW - Complex Drivers

- Die Complex Drivers enthalten die in AUTOSAR nicht standardisierten Treiber für die spezifischen Eigenschaften eines Mikrocontrollers oder Steuergeräts.
- Beispiele
  - Sensordatenauswertung
  - Direkter Zugriff auf Mikrocontroller
  - Einfachlösungen für geringe Stückzahlen
  - Zugriffszeiten (siehe unten)
  - Weiterverwendung (siehe unten)

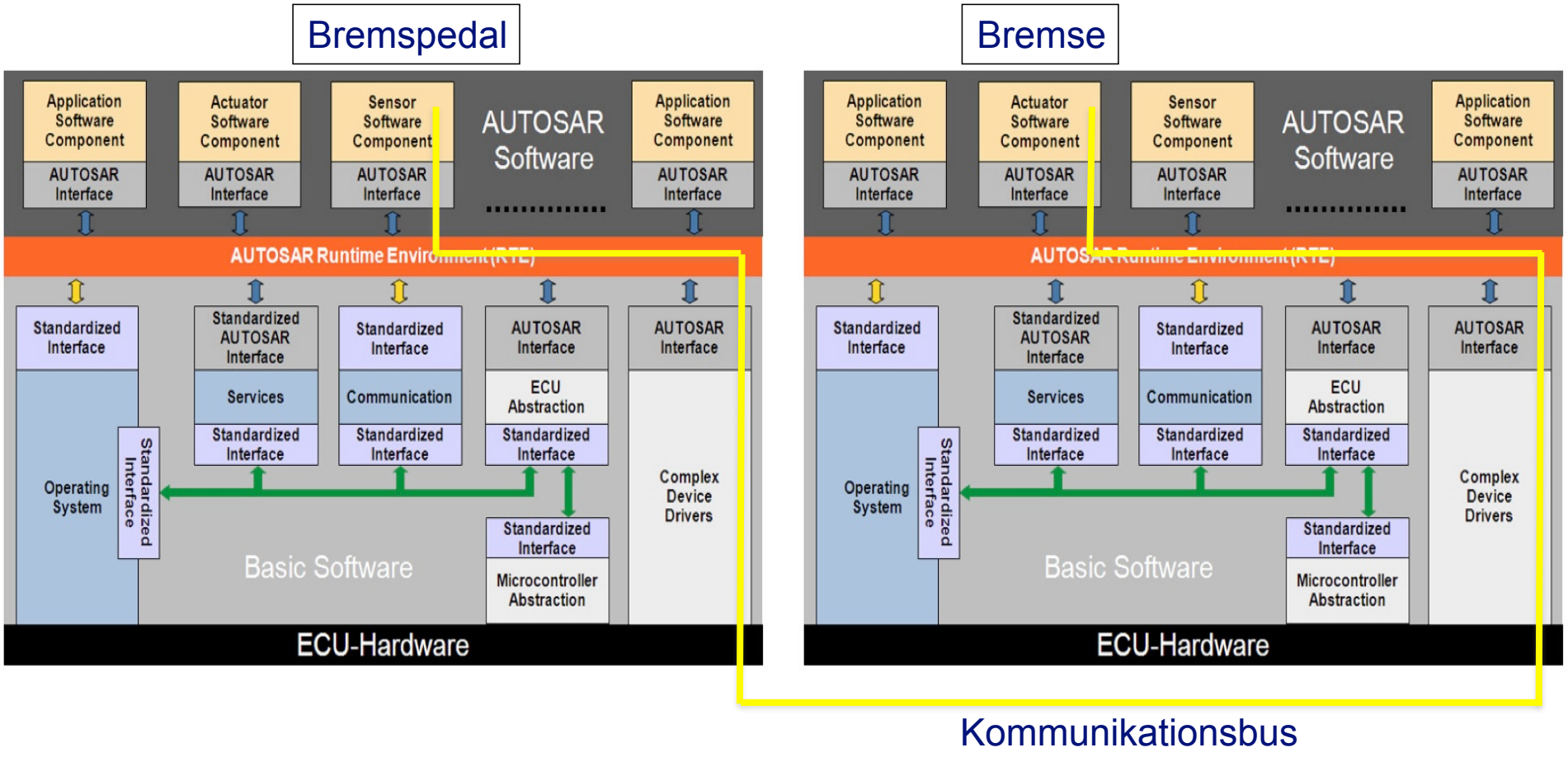
Beispiel Zeitbeschränkungen

- Bremsen mit Verwendung des AUTOSAR-Schichtenmodells:
  Zu langsam durch die vielen Software-Schichten



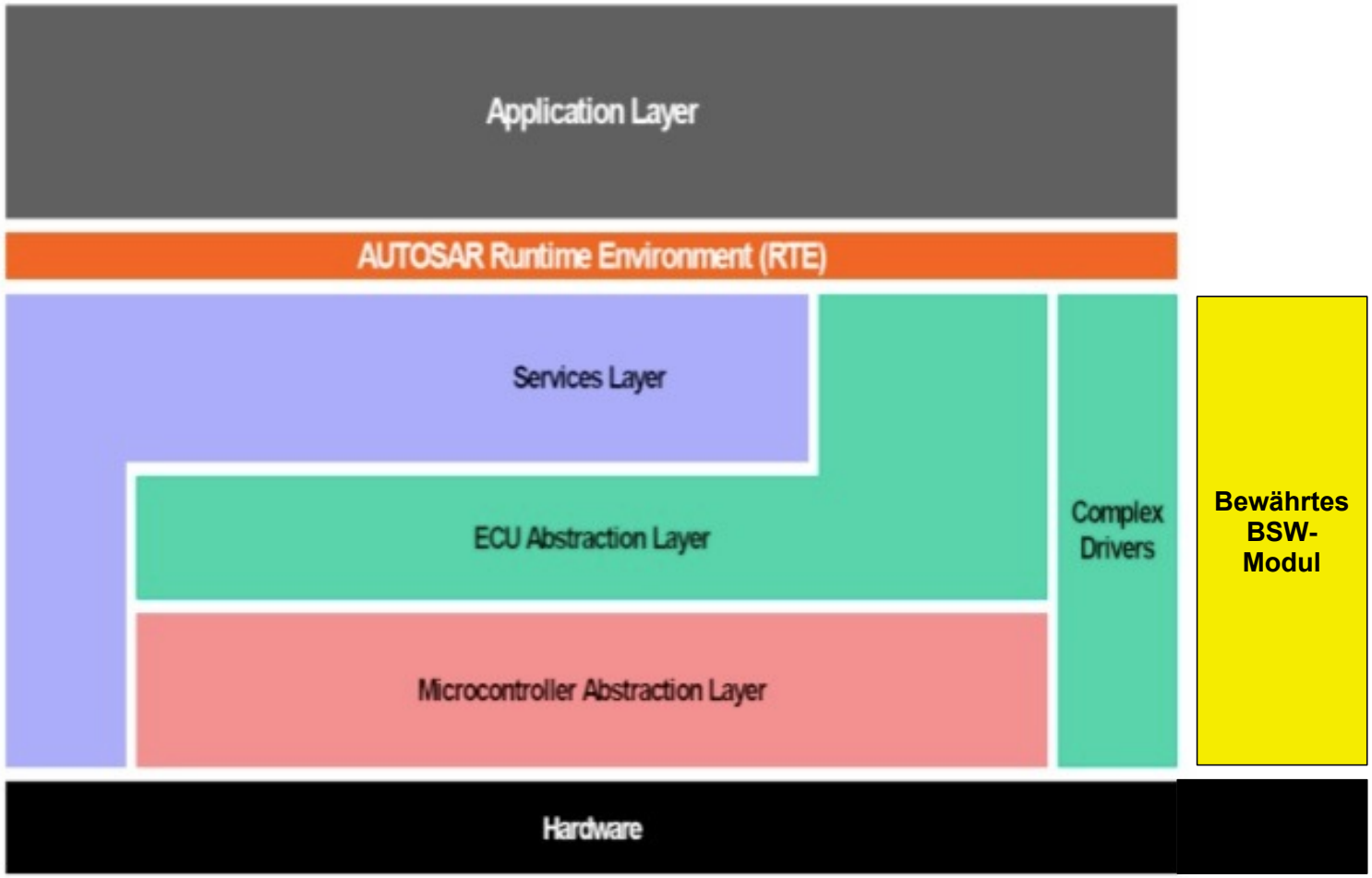
Beispiel Zeitbeschränkungen

- Lösung: Direkter Zugriff auf die ECU-Hardware mit „Complex Drivers“



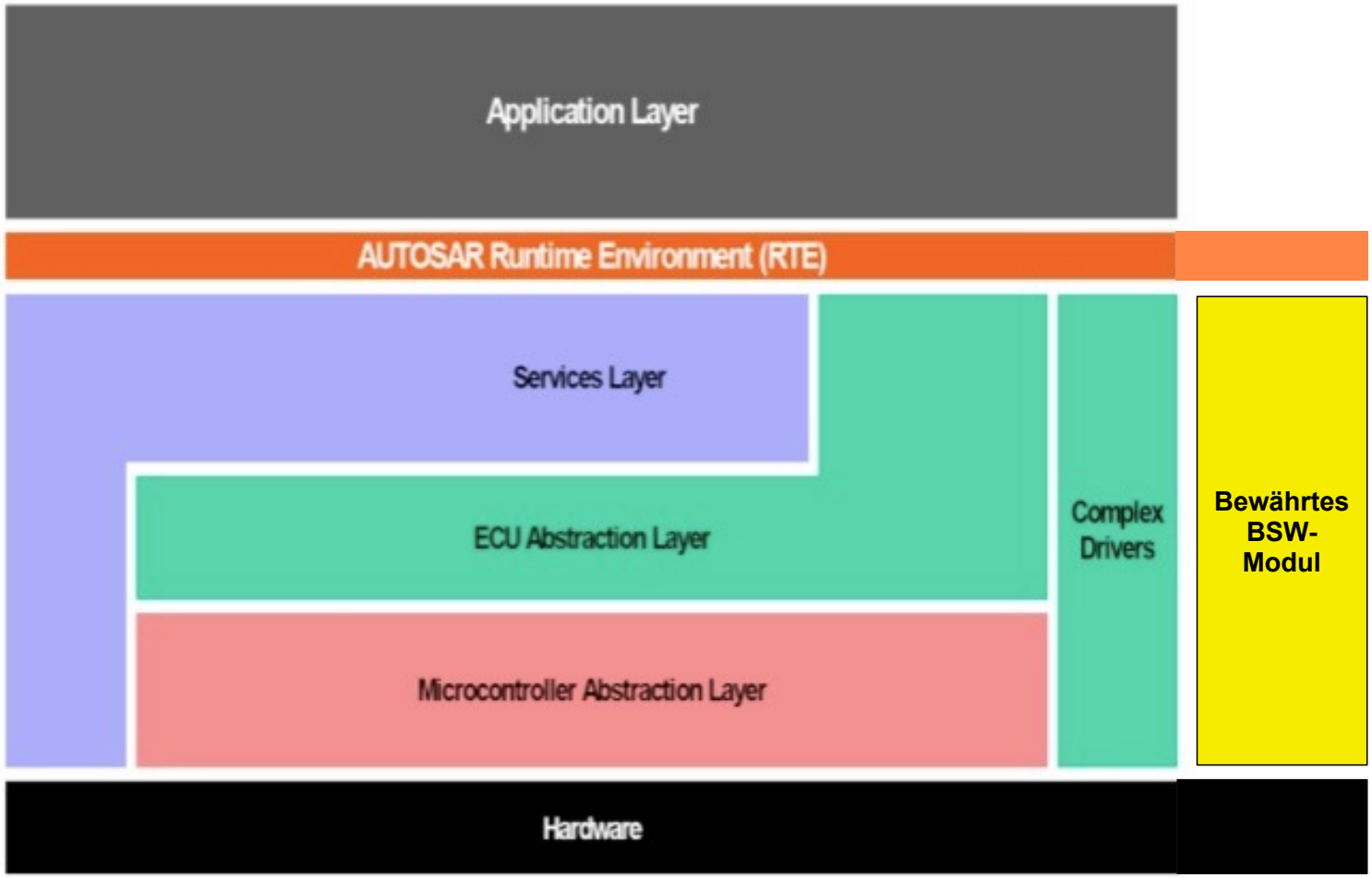
AUTOSAR-Schichtenmodell
BSW - Complex Drivers

- Weiterverwendung



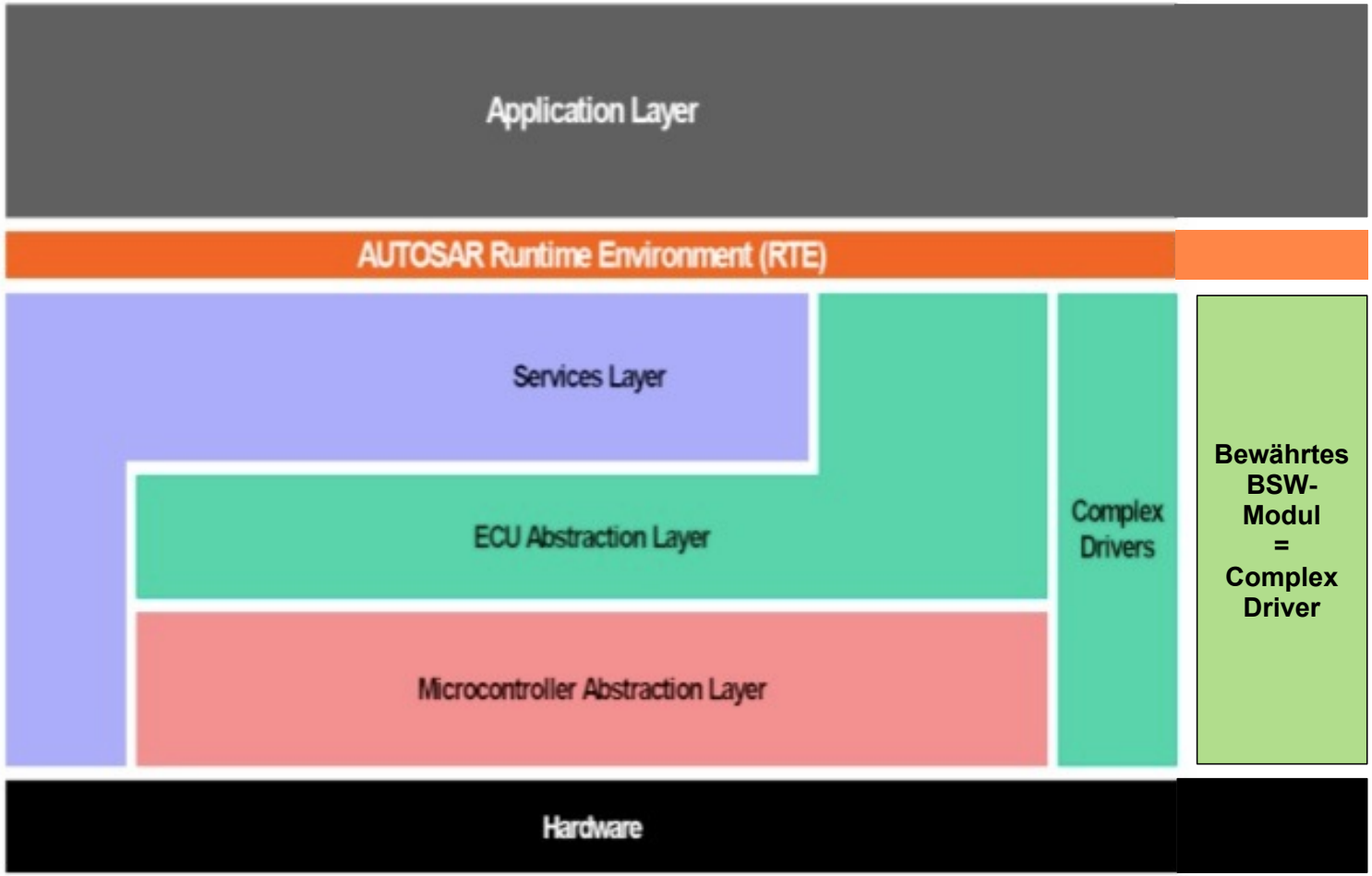
AUTOSAR-Schichtenmodell
BSW - Complex Drivers

- Weiterverwendung
- RTE-Schnittstelle



AUTOSAR-Schichtenmodell
BSW - Complex Drivers

- Weiterverwendung
- RTE-Schnittstelle
- Anwendungssoftware



Case Study
Entwicklung einer Treiberbibliothek für Motorsteuerungen mit AUTOSAR Complex Device Driver (CDD) (1)
Quelle: Vector Informatik GmbH

- Der Kunde

  Die FAW Group Cooperation, die „First Automotive Works“, mit Sitz in Changchun ist der größte chinesische Hersteller von Dieselmotoren, PKWs, sowie mittleren bis schweren Bussen und LKWs. FAW produziert mehr als 2,5 Millionen Fahrzeuge pro Jahr und gehört zu den ersten chinesischen OEMs, die AUTOSAR einsetzen.

- Die Herausforderung

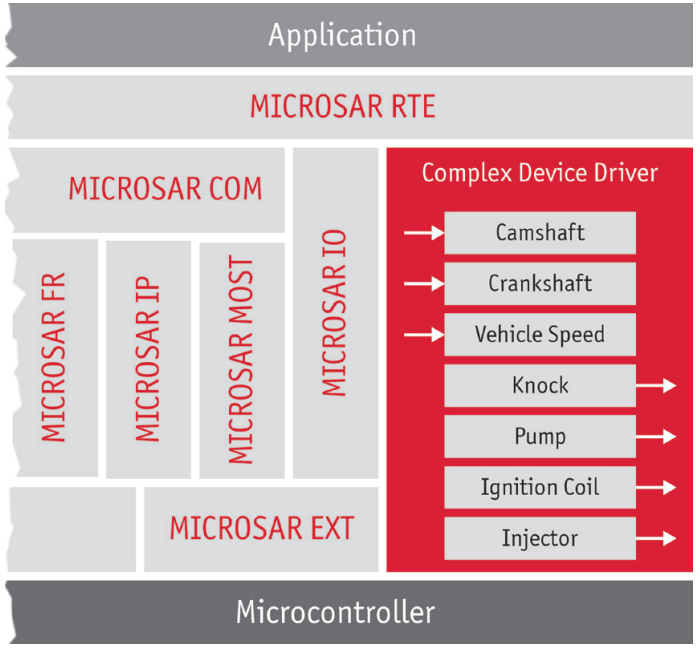
  Entwicklung einer Treiberbibliothek für Motorsteuerungen mit AUTOSAR Complex Device Driver (CDD) FAW entwickelt eine neue Generation ihrer Motorsteuerungen und setzt dabei konsequent AUTOSAR-Basissoftware ein. Die Software wird als eine einzige Plattform realisiert, mit der sich sowohl Benzin- als auch Dieselmotoren steuern lassen. Da in AUTOSAR keine entsprechenden Treiber für Motorsteuerungen definiert sind, möchte FAW die benötigten Sensoren und Aktuatoren aus einer erweiterten AUTOSAR-Treiberbibliotek auswählen und in der gewünschten Anzahl und Ausprägung mit Hilfe einer durchgängigen Werkzeugkette konfigurieren.

FAW

Case Study
Entwicklung einer Treiberbibliothek für Motorsteuerungen mit AUTOSAR Complex Device Driver (CDD) (2)
Quelle: Vector Informatik GmbH

- Die Lösung

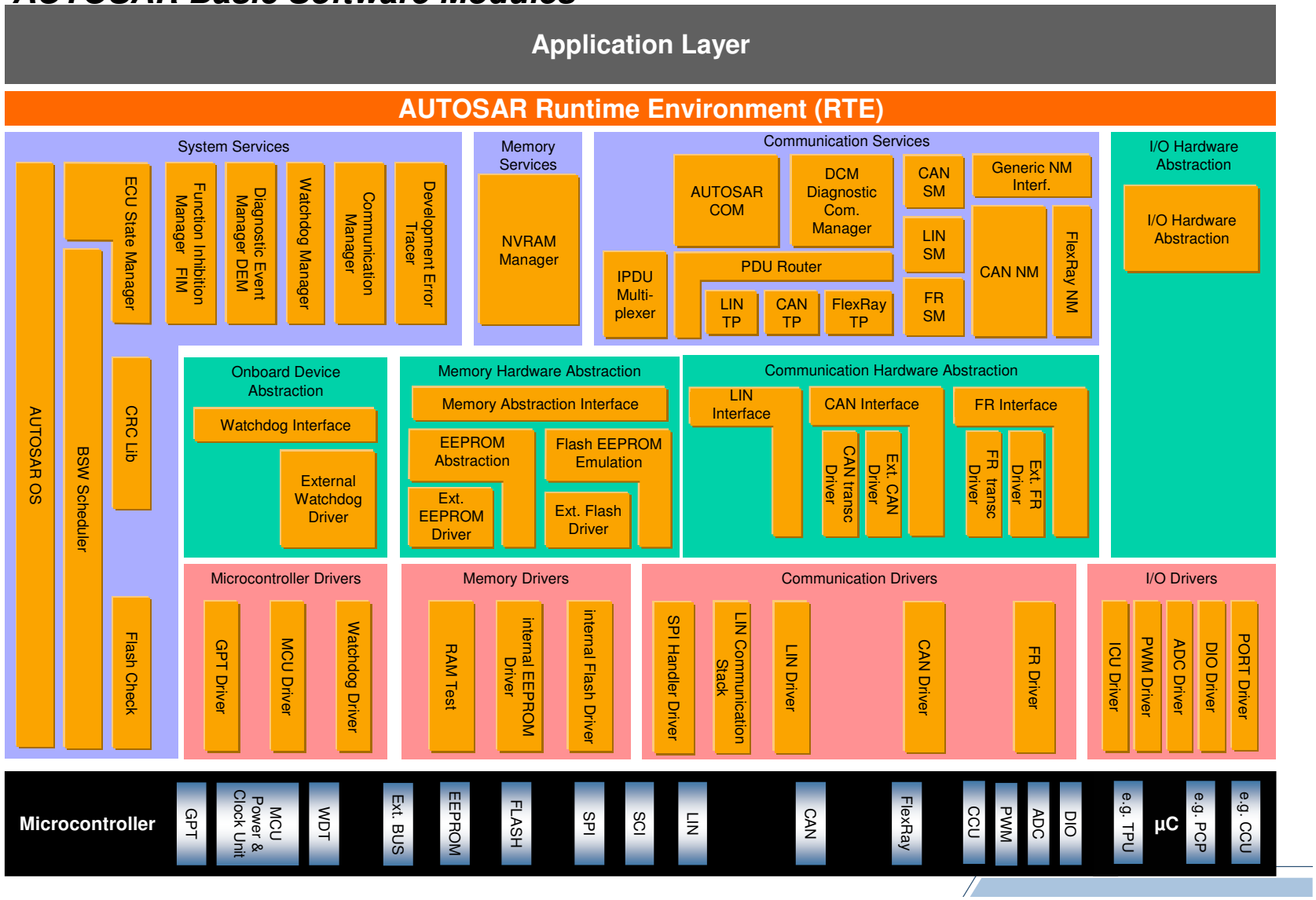
  Konfiguration und Code-Generierung der motorspezifischen Treiber mit der vorhandenen AUTOSAR-Werkzeugkette von Vector Die Treiber zur Ansteuerung der motorspezifischen Sensorik und Aktuatorik wurden von Vector als sogenannte Complex Device Driver (CDD) realisiert. Für die Konfiguration der Trei- ber wurden entsprechende Basissoftware Module Description (BSWMD) Dateien mit Hilfe des DaVinci Configurator Kit erzeugt. Ebenso wurden die Code-Generatoren mit dem DaVinci Configurator Kit erstellt. Der DaVinci Configurator Pro liest die BSWMD-Dateien und bindet die zugehörigen Code Generatoren ein. Damit kann er die Treiber für die Motorsteuerung und die AUTOSAR-Basissoftware konfigurieren.



- Camshaft Nockenwelle
- Crankshaft Kurbelwelle

AUTOSAR Basis Software Module

AUTOSAR Basic Software Modules

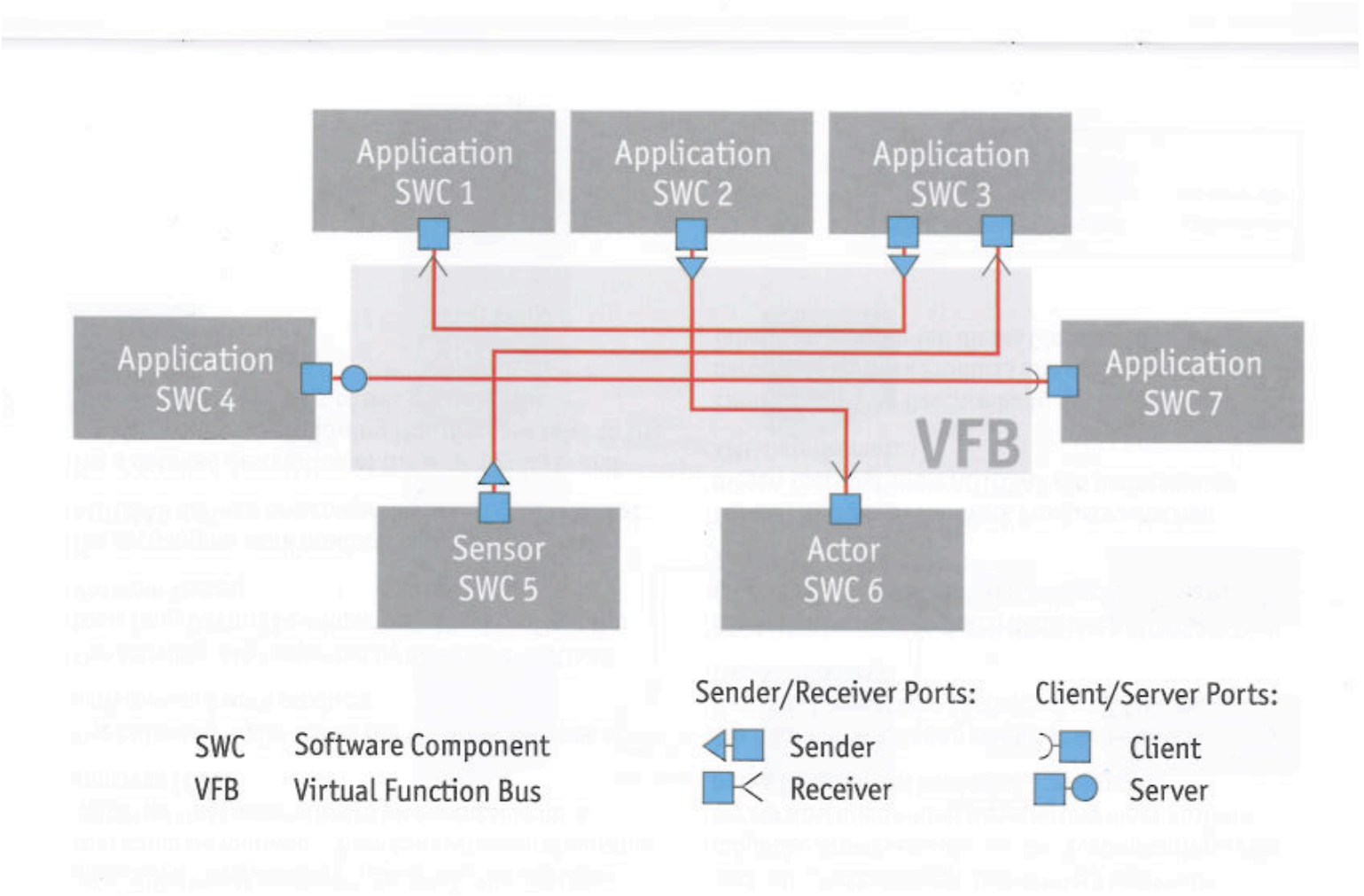


1. Organisation
2. Schichtenmodell
3. **Systementwicklung**
4. Bussysteme im KFZ
5. Software-Architektur
6. Anwendungsbeispiele
7. Geplante AUTOSAR-Anwendungen

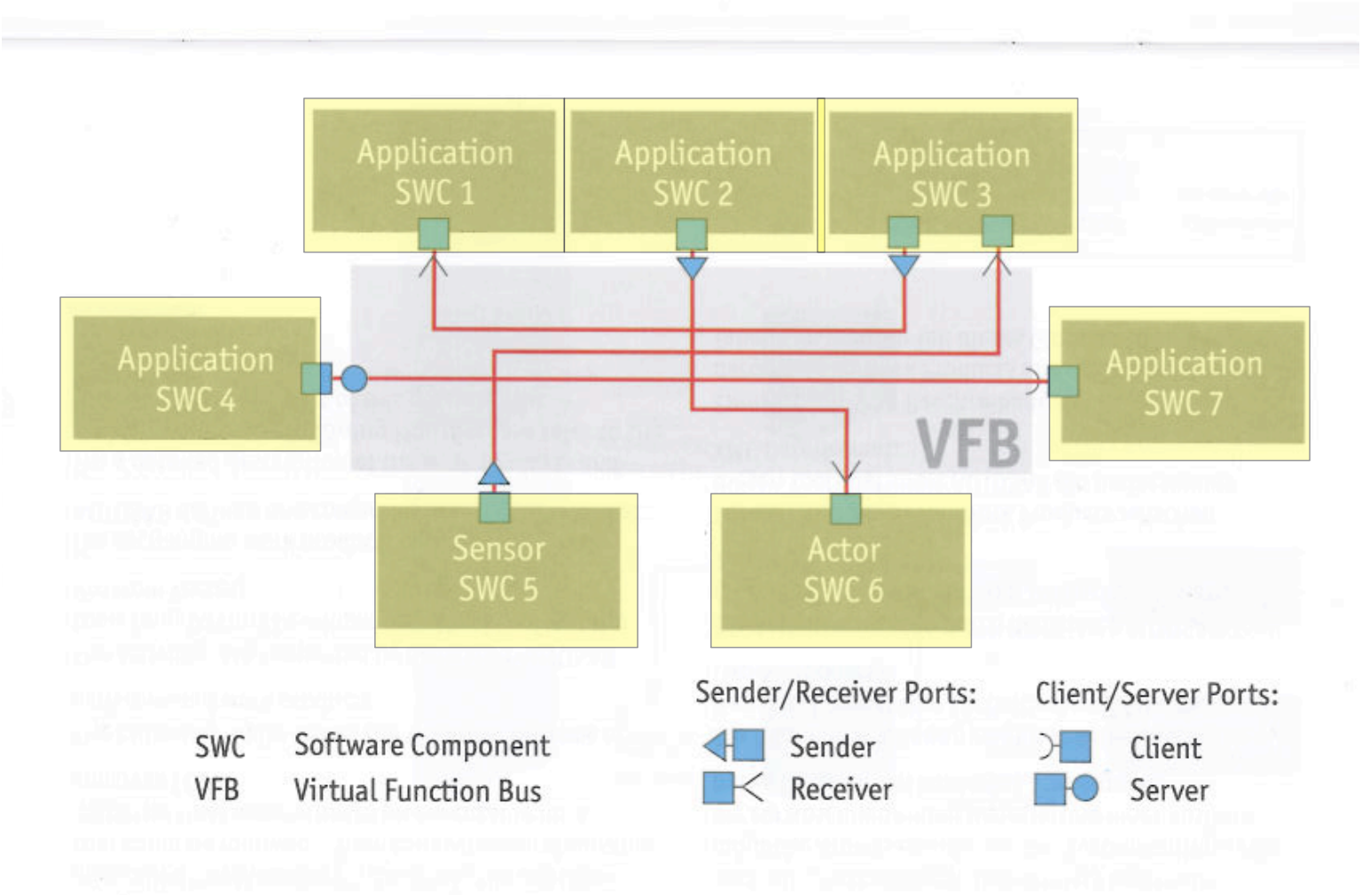
Grundsätzlicher Design-Ansatz:

- Trennung zwischen Steuergerät (Infrastruktur) und Anwendung (Funktionalität)
- Eine Anwendung besteht aus miteinander verbundenen Software Komponenten
- Die Software Komponenten sind atomar, d.h. sie können nicht über mehrere Steuergeräte verteilt werden.
- Die Implementierung der Software Komponenten ist unabhängig vom Steuergerät.
- Methodik
- Beispiele

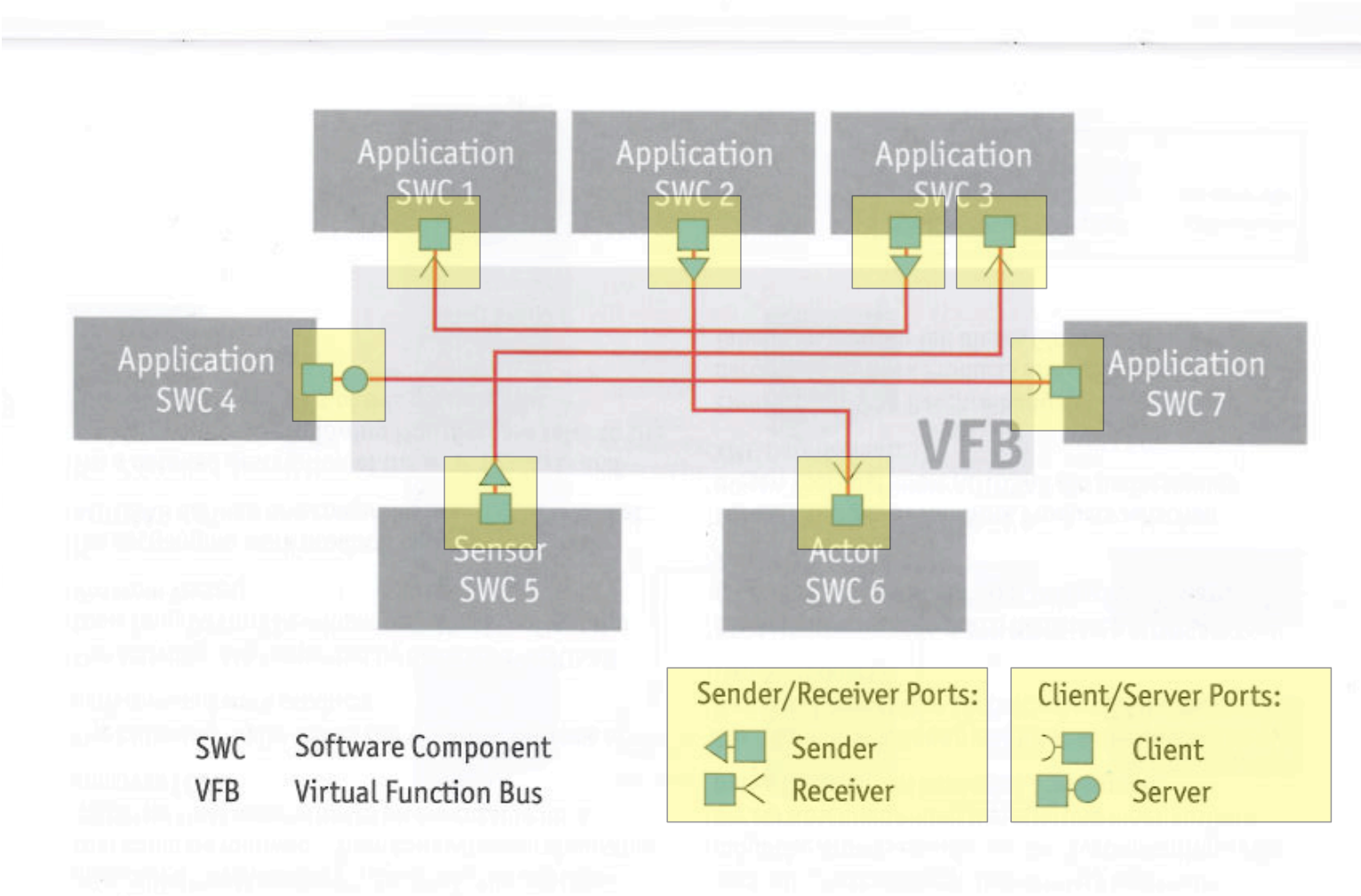
- Die Anwendungssoftware wird unabhängig vom konkreten Steuergerät als ein System von untereinander verbundenen SWCs entworfen



- Die Anwendungssoftware wird unabhängig vom konkreten Steuergerät als ein System von untereinander verbundenen SWCs entworfen

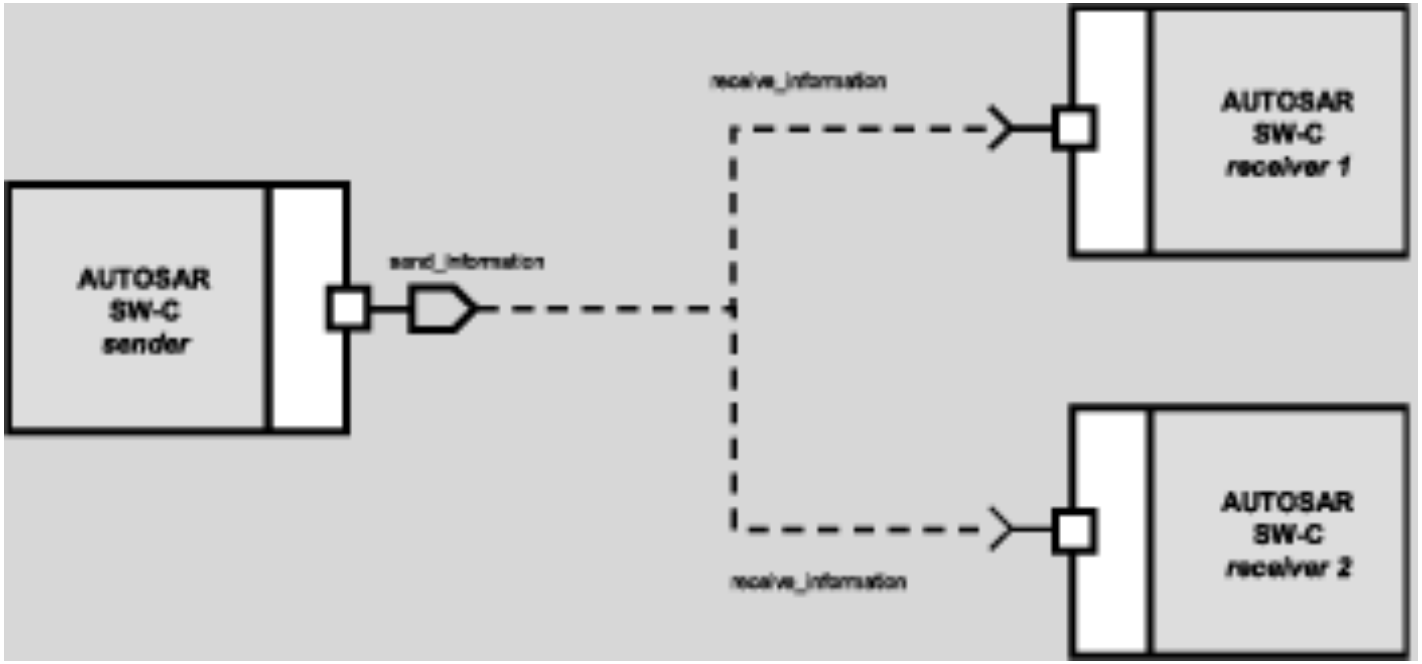


- Sender/Empfänger-Ports
- Client/Server-Ports



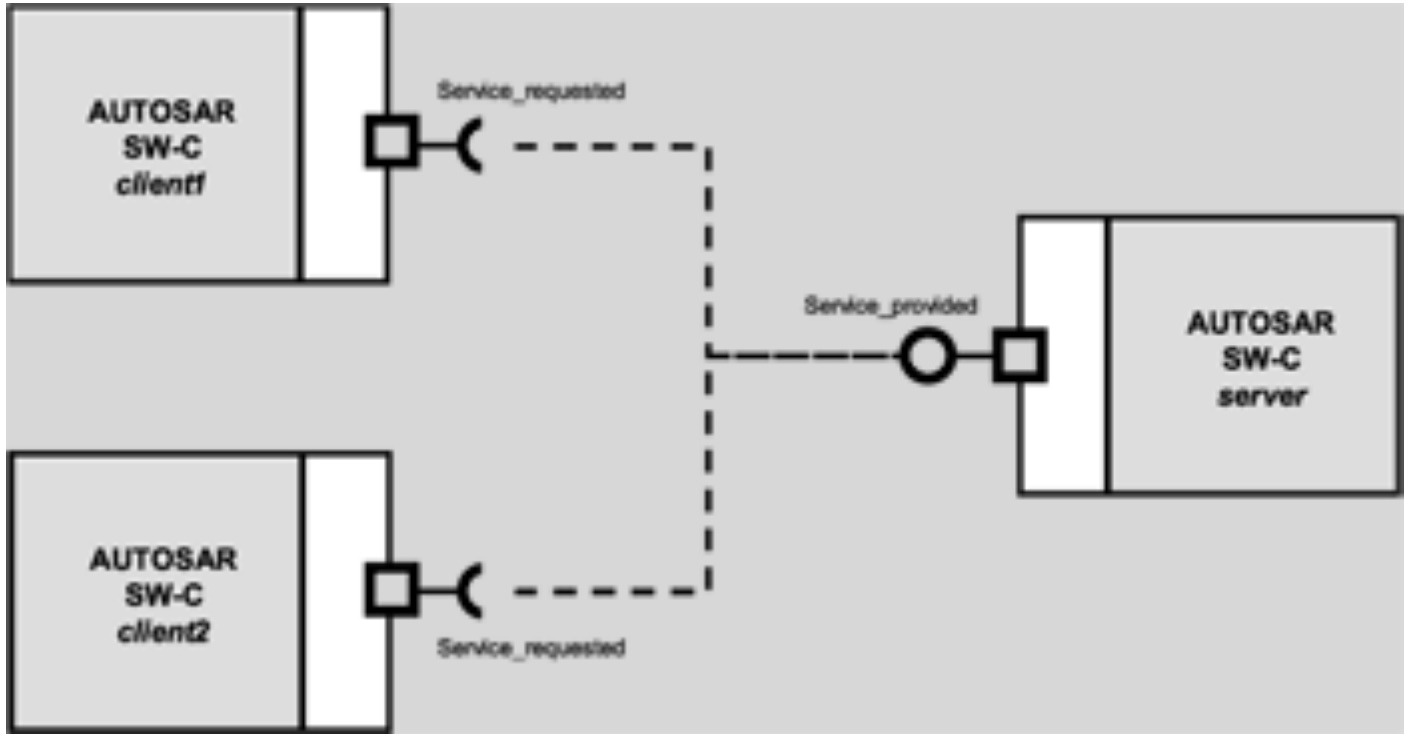
Systementwicklung
Sender/Empfänger-Ports

- The sender-receiver pattern gives solution to the asynchronous distribution of information, where a sender distributes information to one or several receivers. The sender is not blocked (asynchronous communication) and neither expects nor gets a response from the receivers (data or control flow), i.e. the sender just provides the information and the receivers decides autonomously when and how to use this information.
- The sender component does not know the identity or the number of receivers to support transferability and exchange of AUTOSAR Software Components.



Quelle: http://www.autosar.org

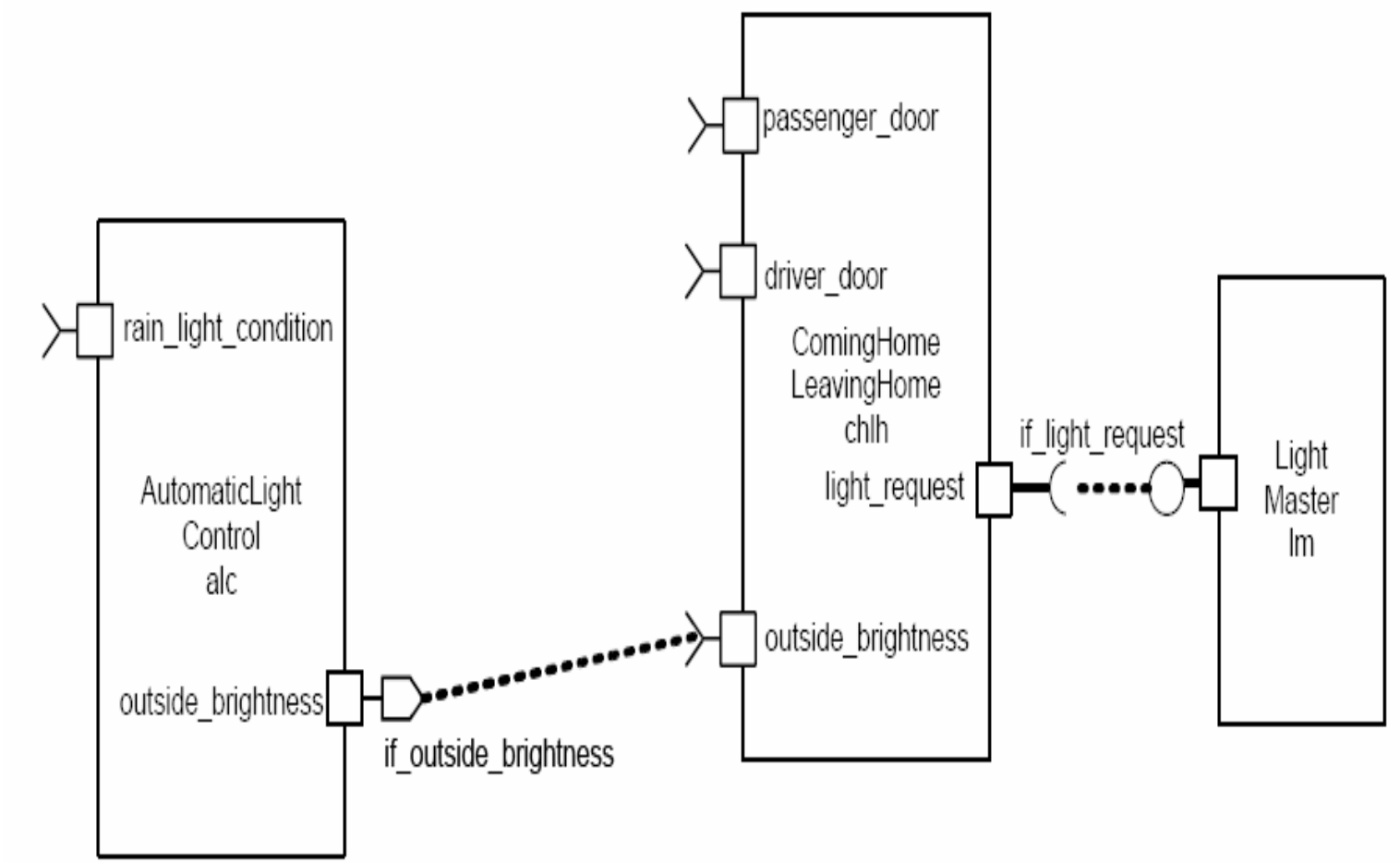
- The client initiates the communication, requesting that the server performs a service, transferring a parameter set if necessary. The server waits for incoming communication requests from a client, performs the requested service, and dispatches a response to the client‘s request.
- The client can be blocked (synchronous communication) or non-blocked (asynchronous communication), respectively, after the service request is initiated until the response of the server is received.



Quelle: http://www.autosar.org

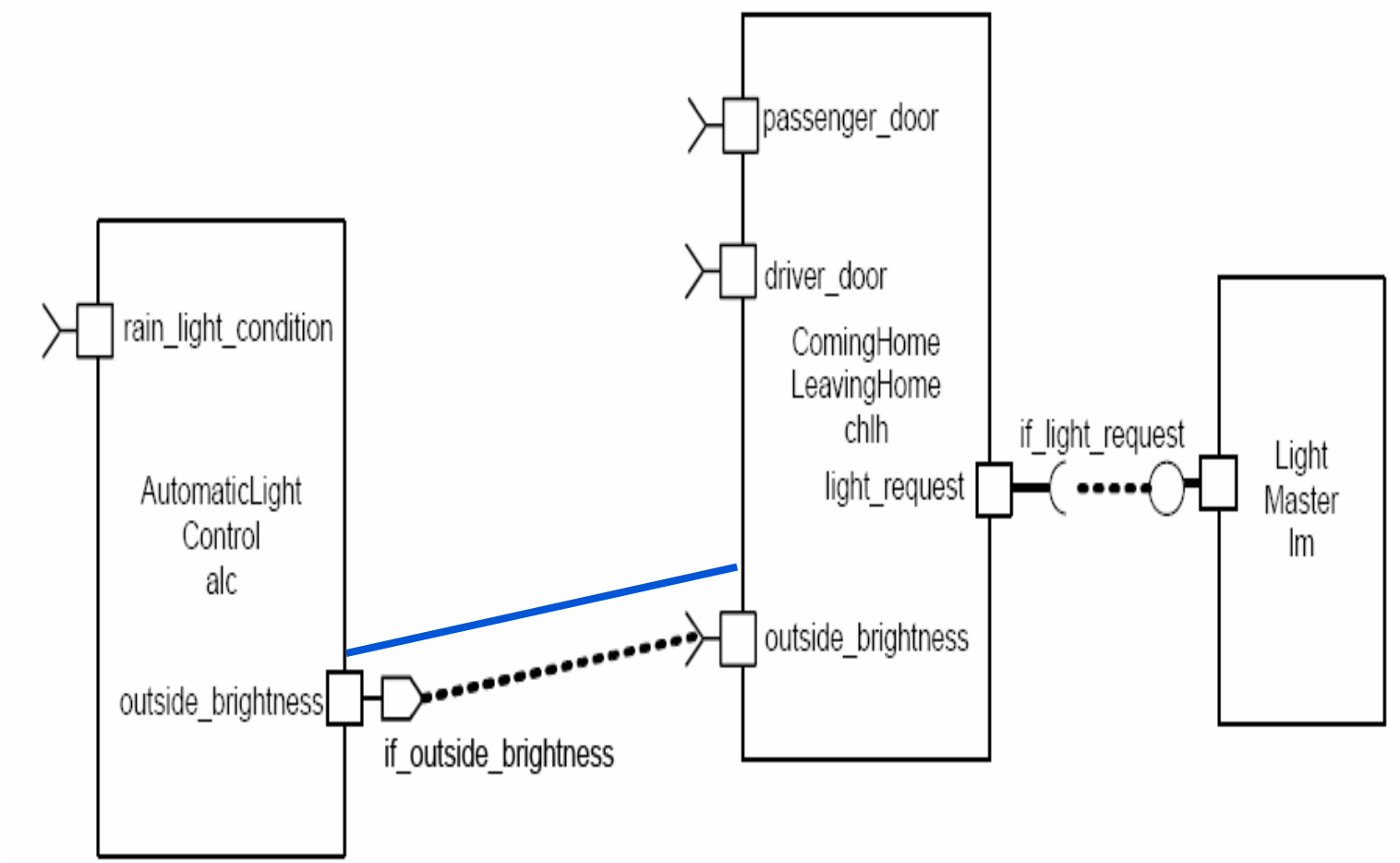
Beispiel

- Regen-Licht-Sensor meldet Helligkeit an Komfort-Lichtsteuerung (Sender/Receiver)
- Komfort-Lichtsteuerung schickt „Licht anschalten“ an Lichtsteuerung (Client/Server)



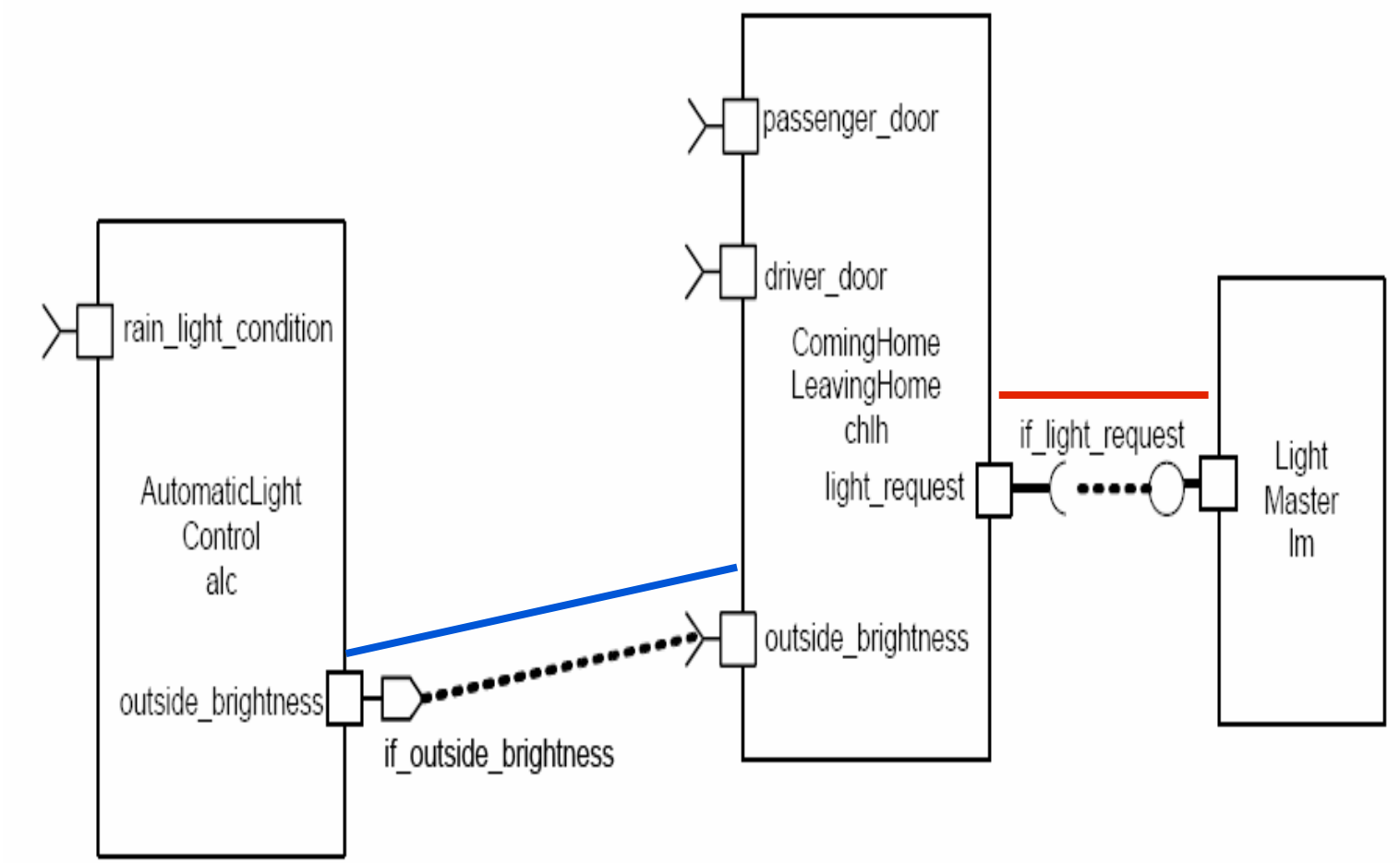
Beispiel

- Regen-Licht-Sensor meldet Helligkeit an Komfort-Lichtsteuerung (Sender/Receiver)
- Komfort-Lichtsteuerung schickt „Licht anschalten“ an Lichtsteuerung (Client/Server)



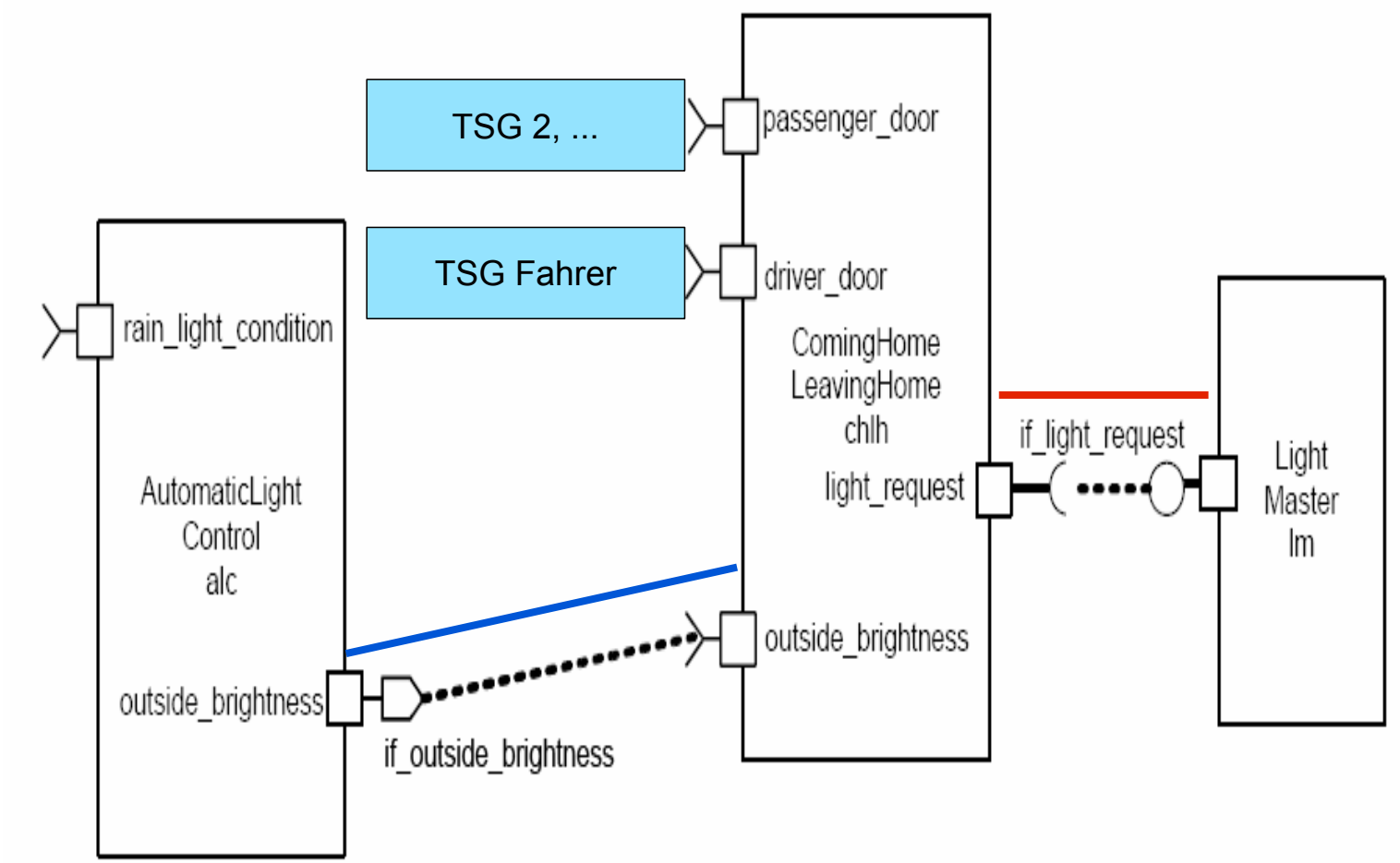
Beispiel

- Regen-Licht-Sensor meldet Helligkeit an Komfort-Lichtsteuerung (Sender/Receiver)
- Komfort-Lichtsteuerung schickt „Licht anschalten“ an Lichtsteuerung (Client/Server)



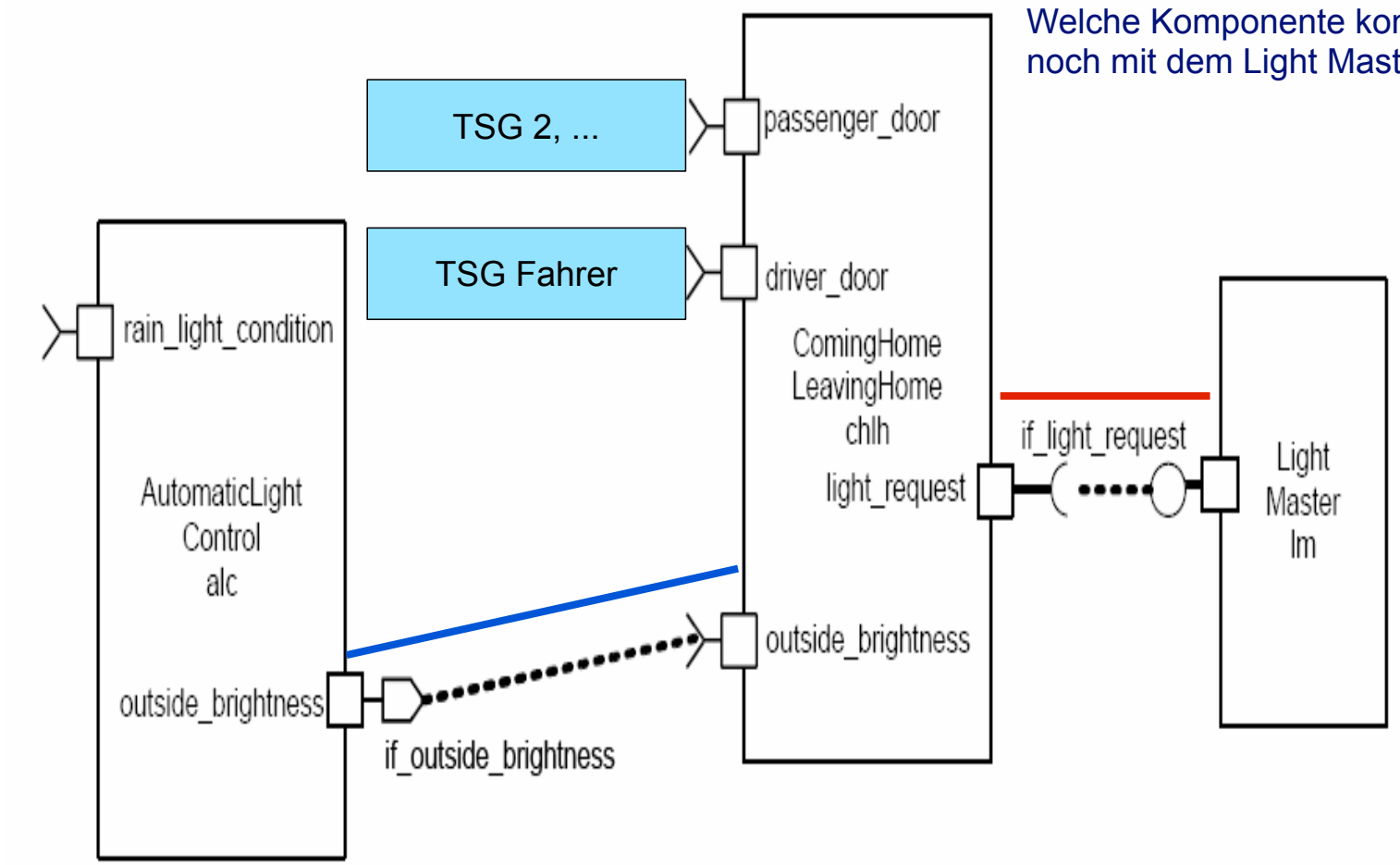
Beispiel

- Regen-Licht-Sensor meldet Helligkeit an Komfort-Lichtsteuerung (Sender/Receiver)
- Komfort-Lichtsteuerung schickt „Licht anschalten“ an Lichtsteuerung (Client/Server)



Beispiel

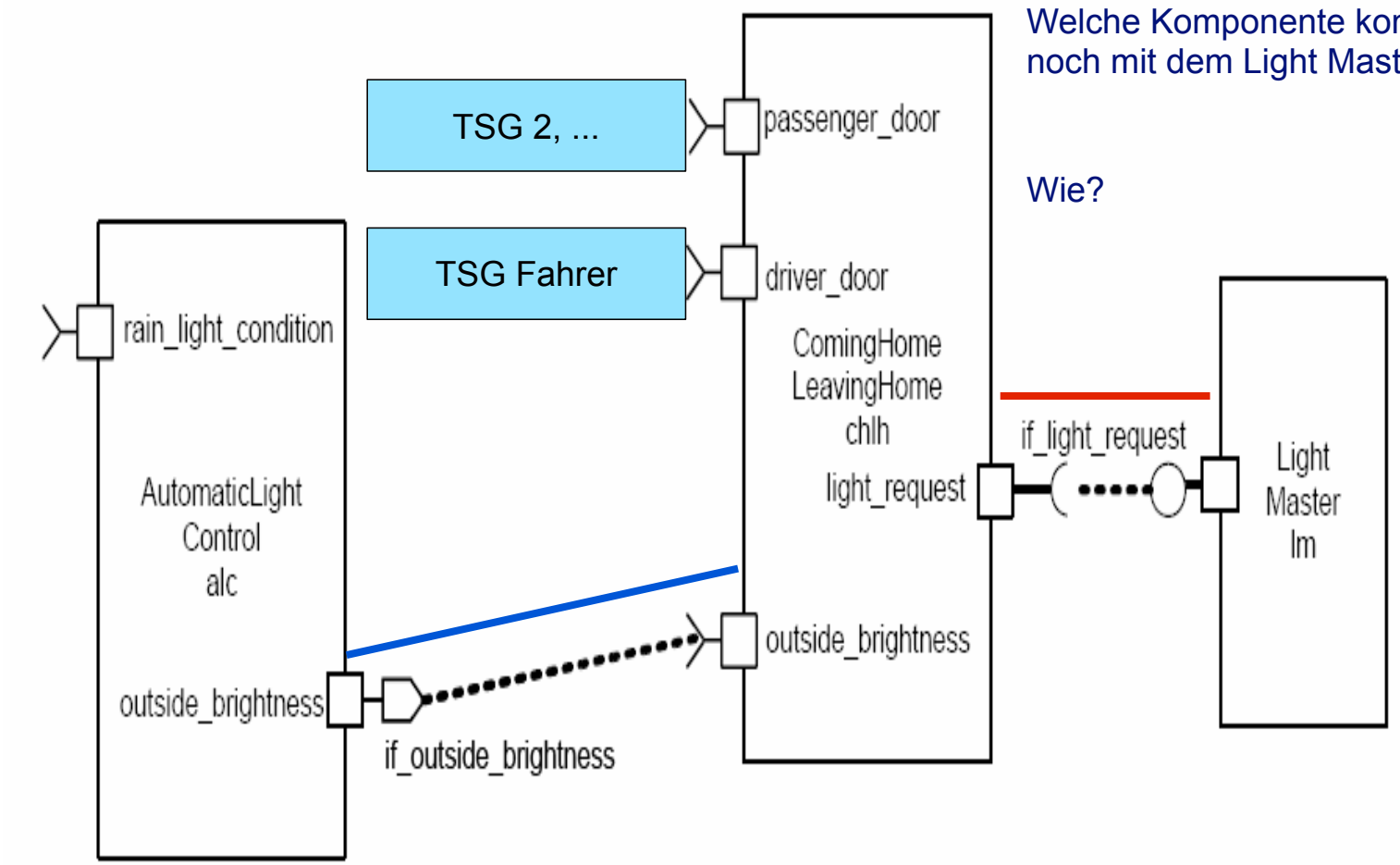
- Regen-Licht-Sensor meldet Helligkeit an Komfort-Lichtsteuerung (Sender/Receiver)
- Komfort-Lichtsteuerung schickt „Licht anschalten“ an Lichtsteuerung (Client/Server)



Welche Komponente kommuniziert noch mit dem Light Master?

Beispiel

- Regen-Licht-Sensor meldet Helligkeit an Komfort-Lichtsteuerung (Sender/Receiver)
- Komfort-Lichtsteuerung schickt „Licht anschalten“ an Lichtsteuerung (Client/Server)



Welche Komponente kommuniziert noch mit dem Light Master?

Wie?

Weitere Informationen zu Ports

- Dr. Stefan Bunzel – AUTOSAR Spokesperson (Continental):
  Hardware-independent Software Development with AUTOSAR
  8. Workshop Automotive Software Engineering
  30 September, 2010, Leipzig
- O. Kindel, M. Friedrich: Softwareentwicklung mit AUTOSAR. Grundlagen, Engineering, Management für die Praxis. dpunkt.verlag, 2009

- Insgesamt 2 x 5 = 10 verschiedene Typen von Ports
- PPort: Provides Interface, Data, Service
- RPort: Requires Interface, Data, Service
- Sender-Receiver Interface
- Client-Server Interface
- Calibration Interface
- Data of AUTOSAR Service
- AUTOSAR Service

Weitere Informationen zu Ports

- Dr. Stefan Bunzel – AUTOSAR Spokesperson (Continental):
  Hardware-independent Software Development with AUTOSAR
  8. Workshop Automotive Software Engineering
  30 September, 2010, Leipzig
- O. Kindel, M. Friedrich: Softwareentwicklung mit AUTOSAR. Grundlagen, Engineering, Management für die Praxis. dpunkt.verlag, 2009

- Insgesamt 2 x 5 = 10 verschiedene Typen von Ports
- PPort: Provides Interface, Data, Service
- RPort: Requires Interface, Data, Service
- Sender-Receiver Interface ✔
- Client-Server Interface
- Calibration Interface
- Data of AUTOSAR Service
- AUTOSAR Service

Weitere Informationen zu Ports

- Dr. Stefan Bunzel – AUTOSAR Spokesperson (Continental):
  Hardware-independent Software Development with AUTOSAR
  8. Workshop Automotive Software Engineering
  30 September, 2010, Leipzig
- O. Kindel, M. Friedrich: Softwareentwicklung mit AUTOSAR. Grundlagen, Engineering, Management für die Praxis. dpunkt.verlag, 2009

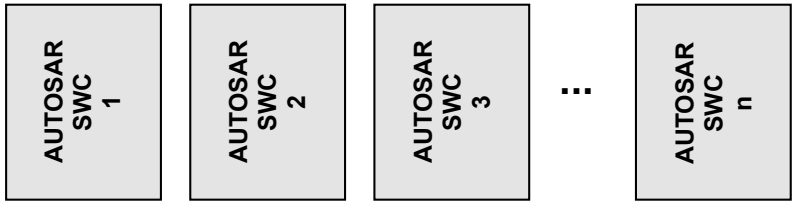
- Insgesamt 2 x 5 = 10 verschiedene Typen von Ports
- PPort: Provides Interface, Data, Service
- RPort: Requires Interface, Data, Service
- Sender-Receiver Interface ✔
- Client-Server Interface ✔
- Calibration Interface
- Data of AUTOSAR Service
- AUTOSAR Service

AUTOSAR Development Methodology
Virtual Functional Bus Concept

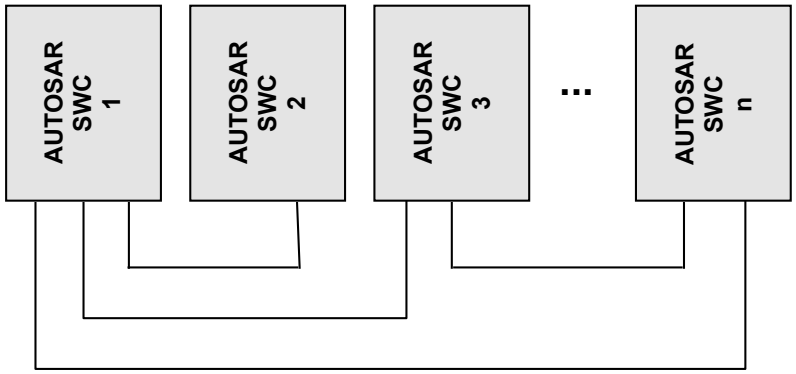
- Application software functionality implemented in „Software Components“ (SWC)

AUTOSAR Development Methodology
Virtual Functional Bus Concept

- Application software functionality implemented in „Software Components“ (SWC)
- Handling of vehicle wide functions, independent from ECUs or network

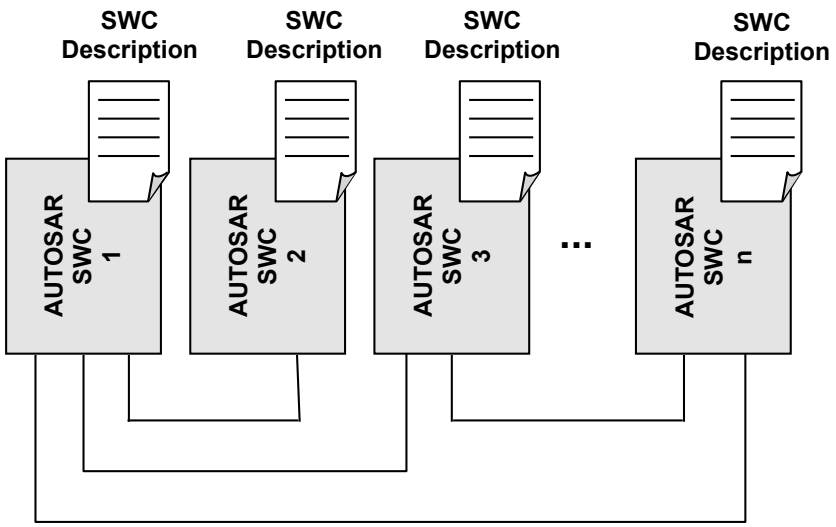


AUTOSAR Development Methodology
Virtual Functional Bus Concept



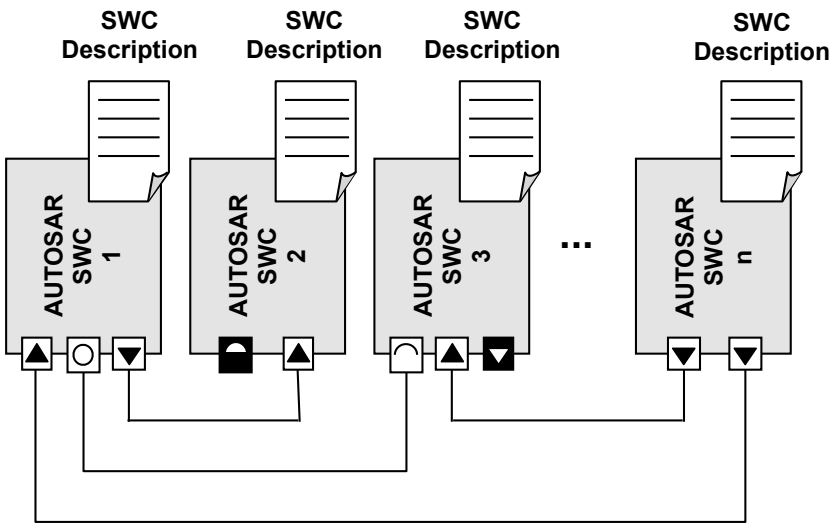
- Application software functionality implemented in „Software Components“ (SWC)
- Handling of vehicle wide functions, independent from ECUs or network
- SWCs can communicate between each other and access functions from the standardized set of infrastructure functions

AUTOSAR Development Methodology
Virtual Functional Bus Concept



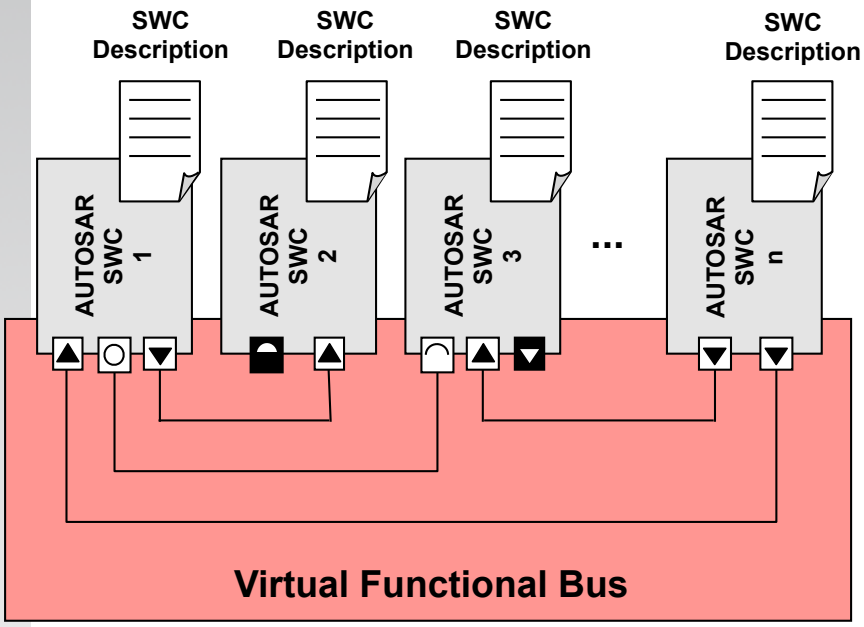
- Application software functionality implemented in „Software Components“ (SWC)
- Handling of vehicle wide functions, independent from ECUs or network
- SWCs can communicate between each other and access functions from the standardized set of infrastructure functions
- Communication needs of a SWC are formally described in a standard template,
  i.e. the SWC Description

AUTOSAR Development Methodology
Virtual Functional Bus Concept



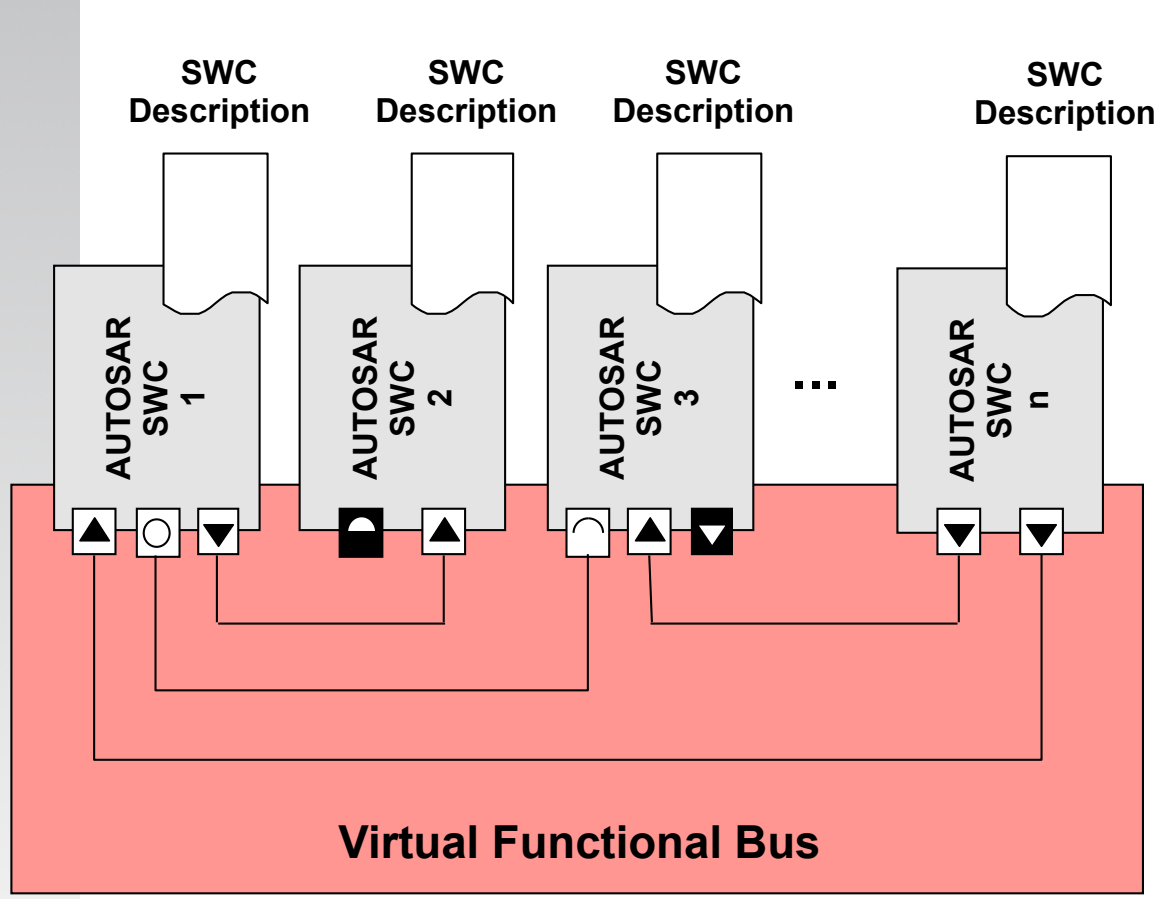
- Application software functionality implemented in „Software Components“ (SWC)
- Handling of vehicle wide functions, independent from ECUs or network
- SWCs can communicate between each other and access functions from the standardized set of infrastructure functions
- Communication needs of a SWC are formally described in a standard template,
  i.e. the SWC Description
- Any SWC interaction runs via „Ports“, which implement different communication paradigms,
  e.g. sender-receiver or client-server

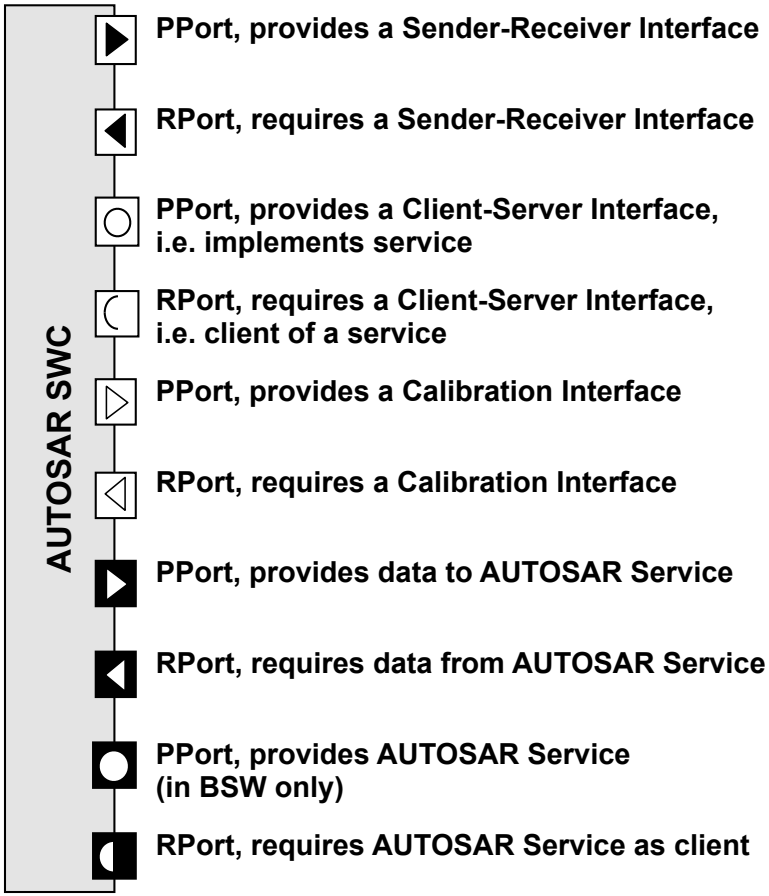
AUTOSAR Development Methodology
Virtual Functional Bus Concept



- Application software functionality implemented in „Software Components“ (SWC)
- Handling of vehicle wide functions, independent from ECUs or network
- SWCs can communicate between each other and access functions from the standardized set of infrastructure functions
- Communication needs of a SWC are formally described in a standard template,
  i.e. the SWC Description
- Any SWC interaction runs via „Ports“, which implement different communication paradigms,
  e.g. sender-receiver or client-server
- The VFB
  - enables a virtual integration of SWCs and
  - allows to formally verify structural and dynamic compatibility of SWCs

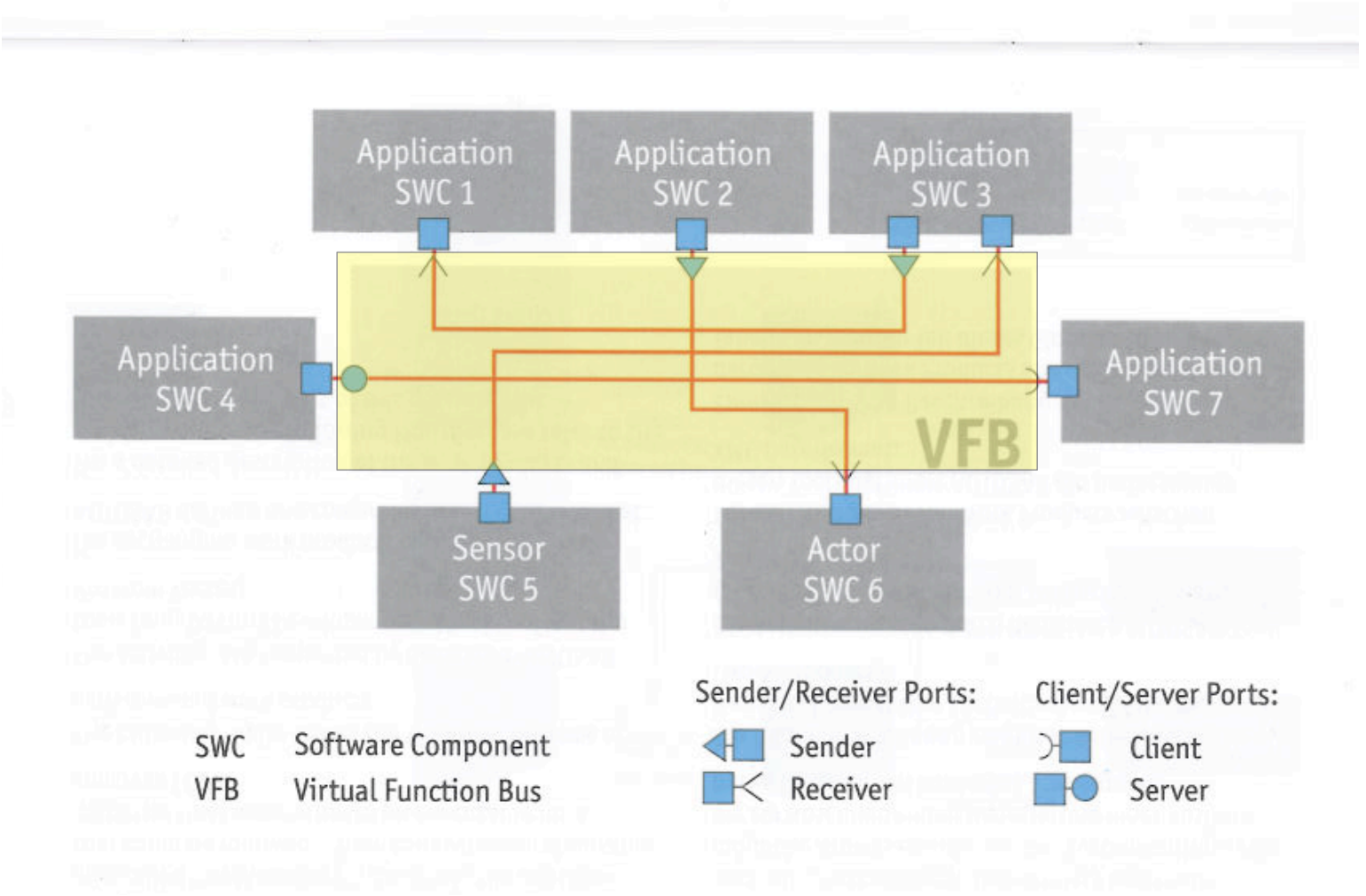
AUTOSAR Development Methodology
Virtual Functional Bus Concept – Ports and Interfaces





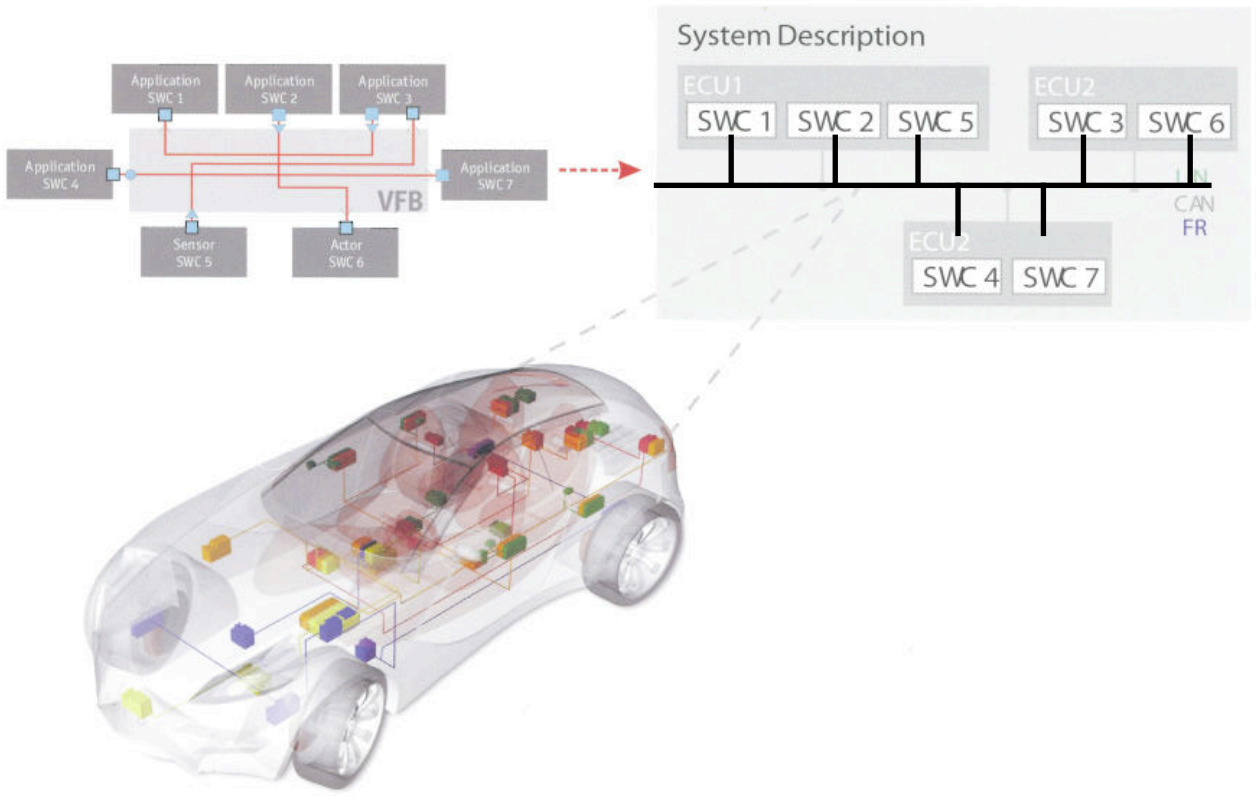
Systementwicklung
Virtual Function Bus (VFB)

- Während der Systementwurfsphase werden die SWCs auf Basis des Virtual Function Bus (VFB) logisch integriert



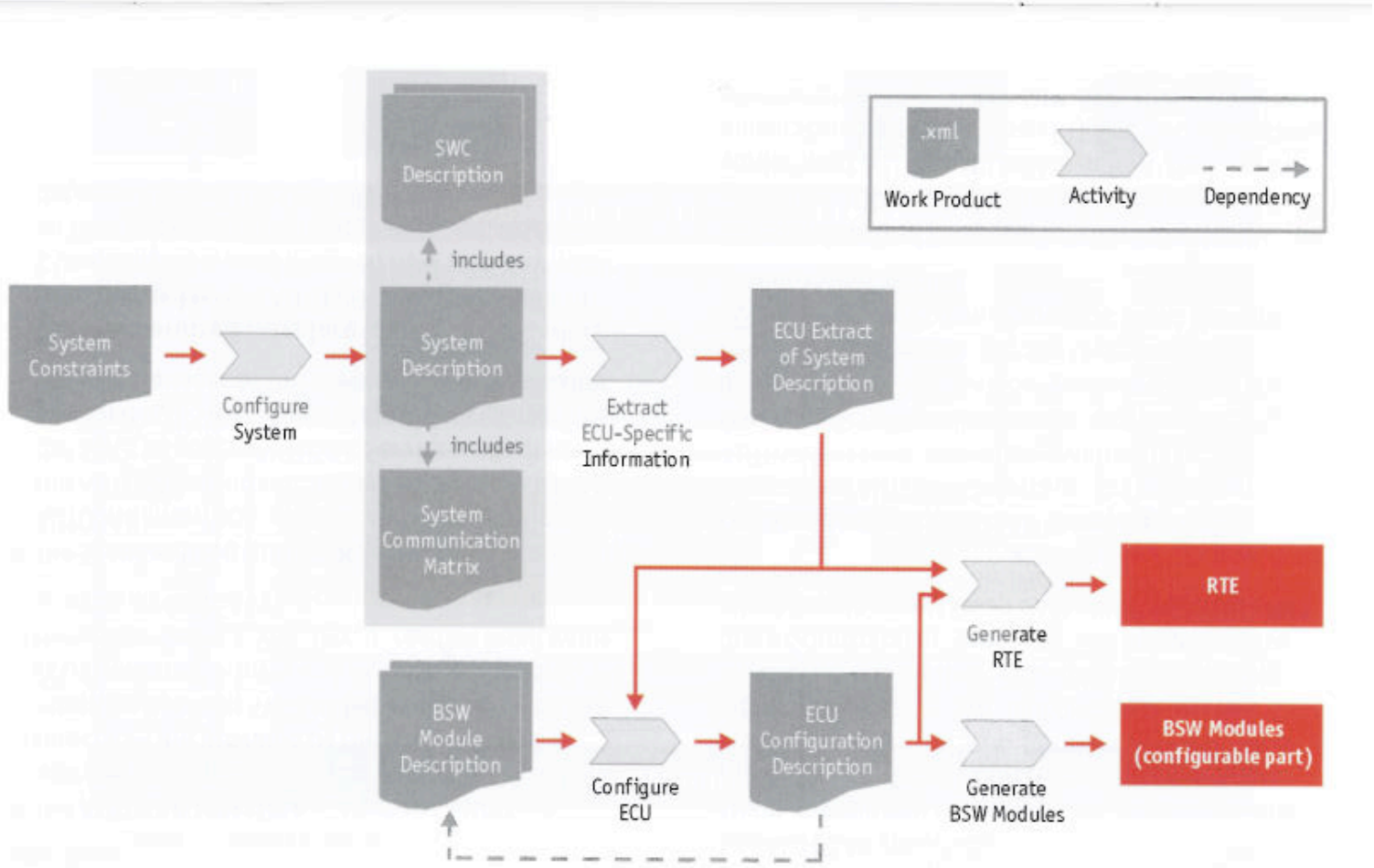
Systementwicklung
System Description

- Zuordnung der Komponenten zu den Steuergeräten
- Beschreibung der Netzwerkkommunikation im Fahrzeug



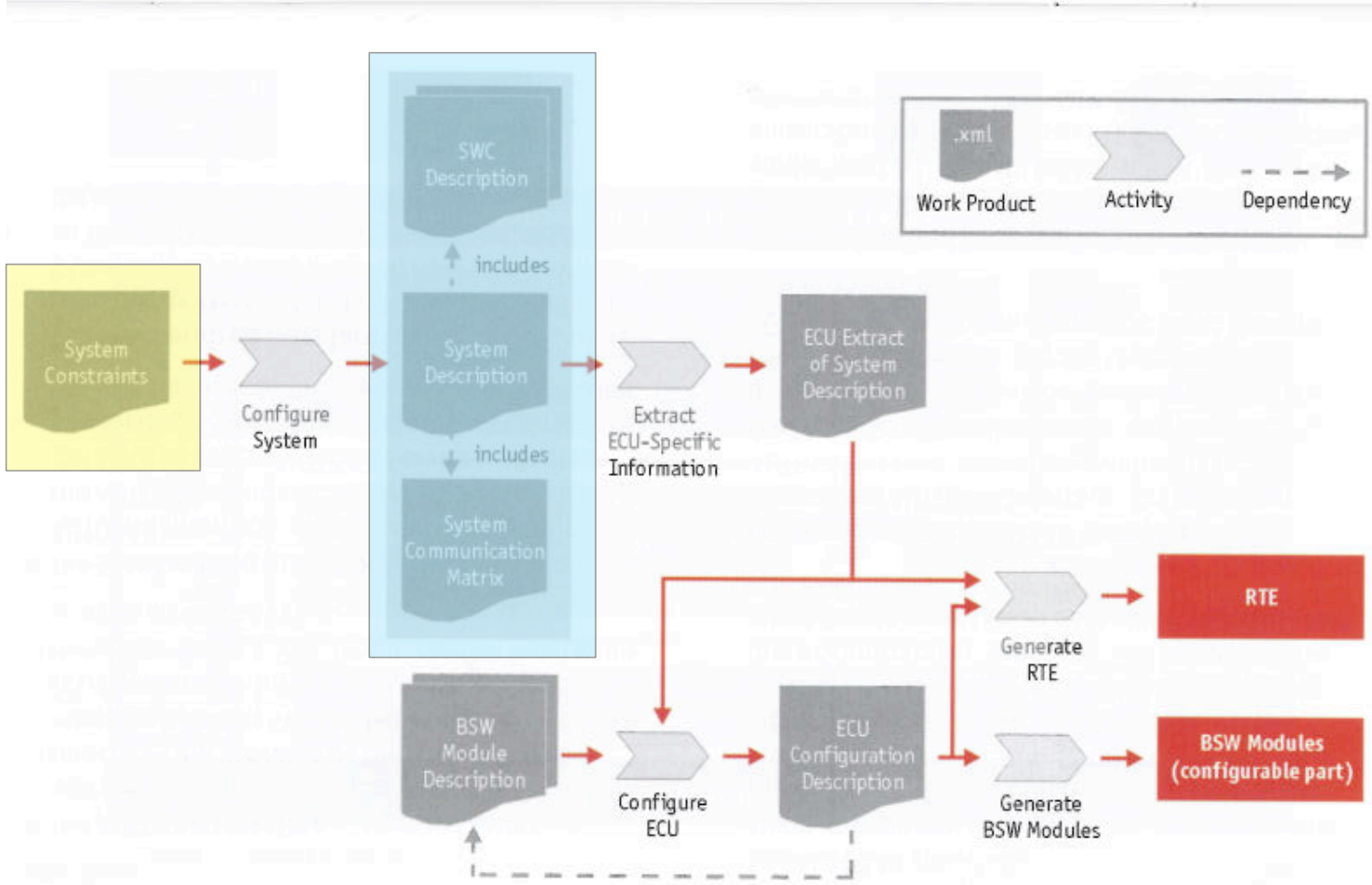
Systementwicklung
AUTOSAR-Methode (1)

- Beschreibt Abläufe von der Systemkonfiguration zur Generierung von Code für Steuergeräte



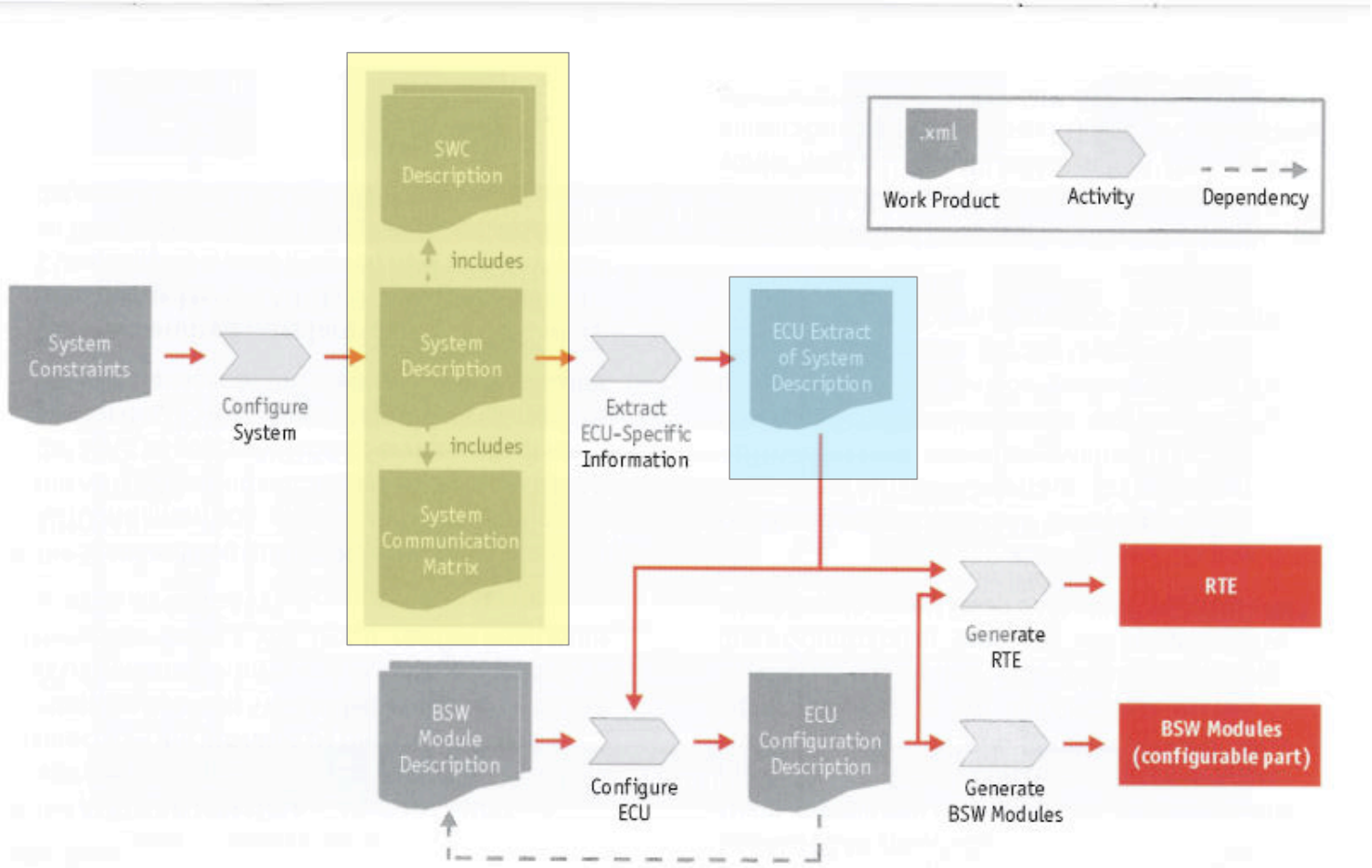
Systementwicklung
AUTOSAR-Methode (2)

- Erstellung von SWC Description, System Description und System Communication-Matrix unter Berücksichtigung der System Constraints
  (Beispiel: Übernahme einer Kommunikationsmatrix aus dem Vorgängerfahrzeug)



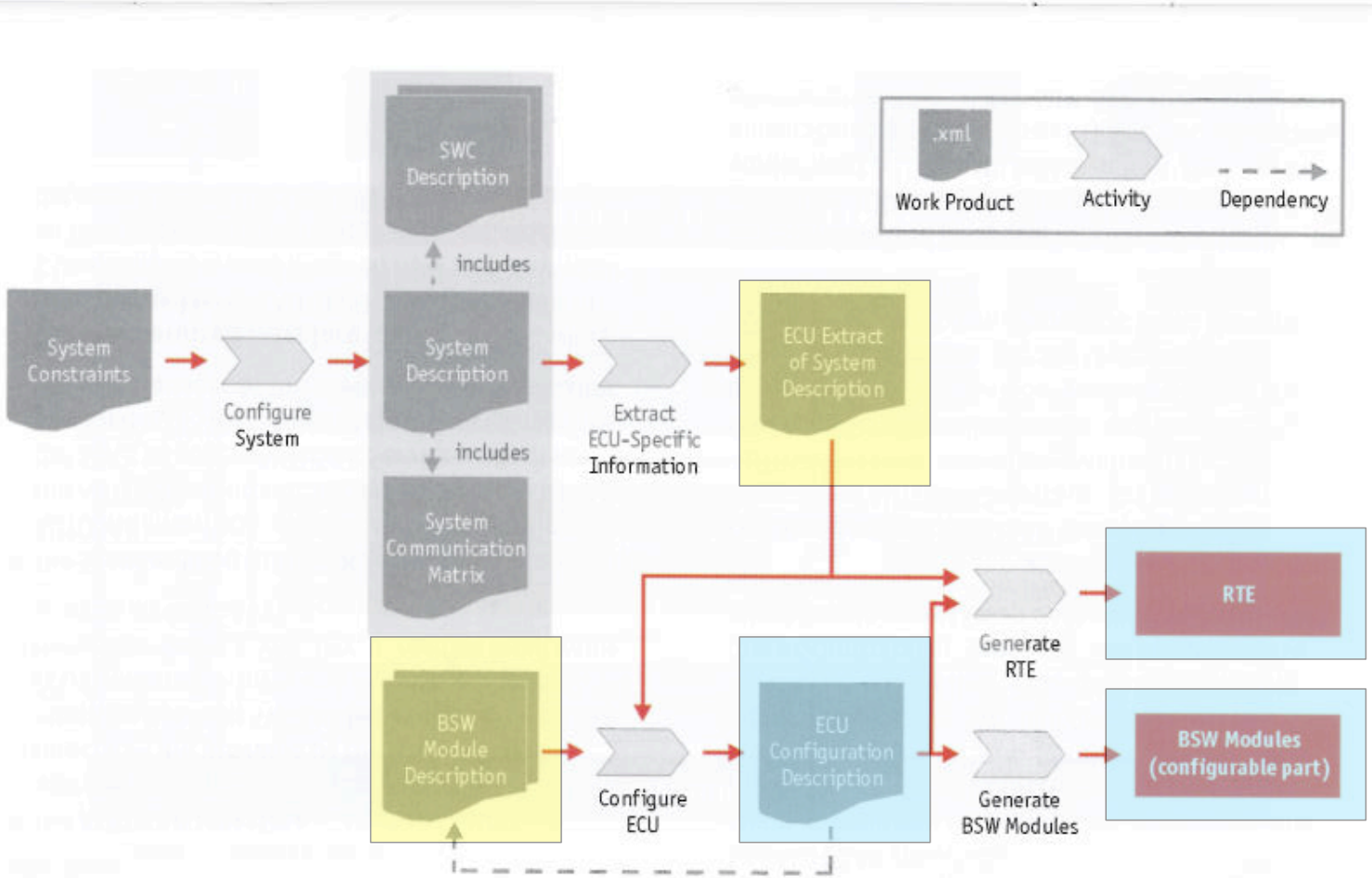
Systementwicklung
AUTOSAR-Methode (3)

- Aus SWC Description, System Description und System Communication-Matrix wird die ECU Extract of System Description generiert



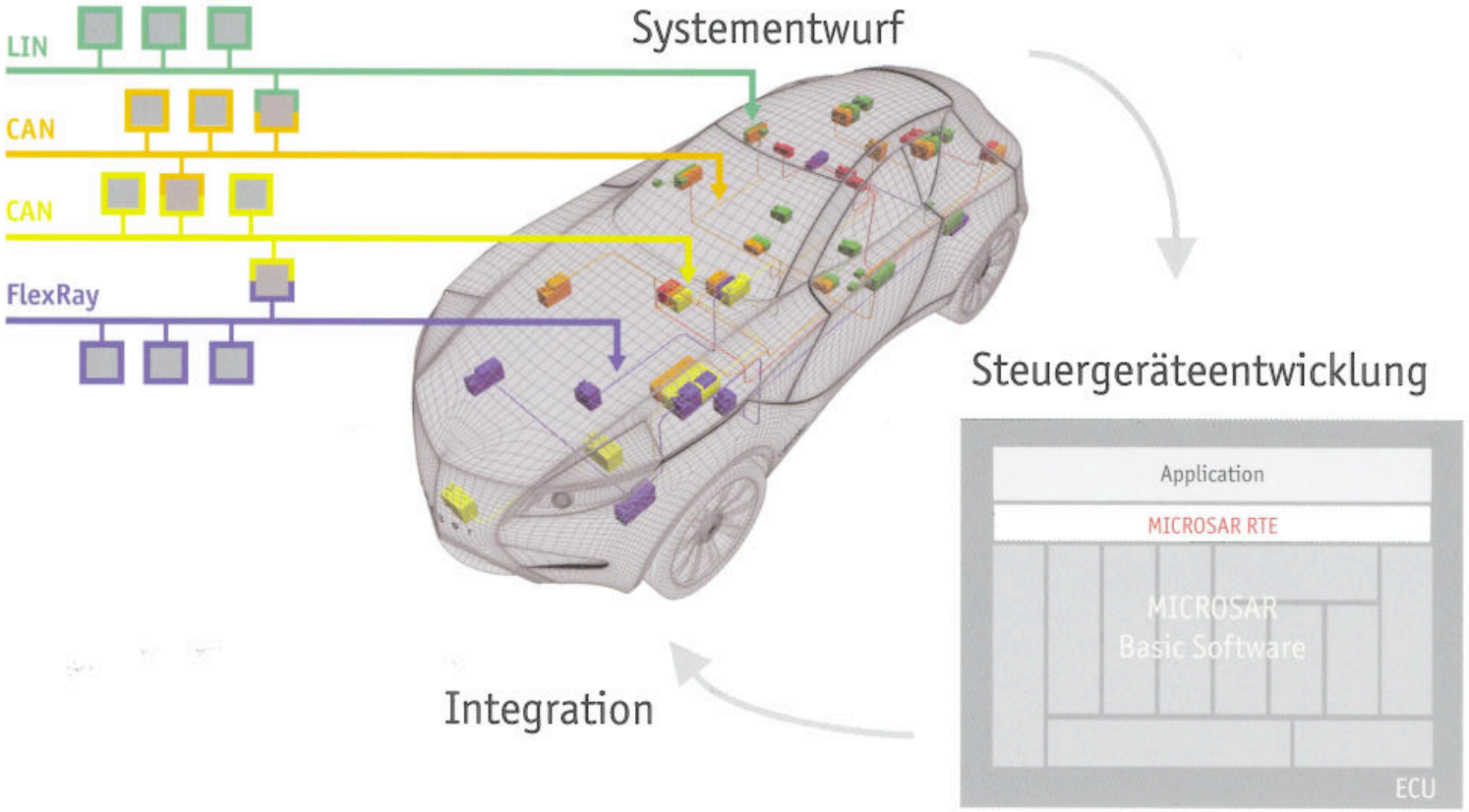
Systementwicklung
AUTOSAR-Methode (4)

- Aus ECU Extract of System Description und BSW Module Description (Herstellerabhängig) werden die ECU Configuration Description, die konkreten BSW-Module und die RTE (Herstellerunabhängig) generiert



Integration ins Fahrzeug

- Steuergeräte
- Bussysteme



1. Organisation
2. Schichtenmodell
3. Systementwicklung
4. **Bussysteme im KFZ**
5. Software-Architektur
6. Anwendungsbeispiele
7. Geplante AUTOSAR-Anwendungen

Klassifikation von Bussystemen - Weiterentwicklung

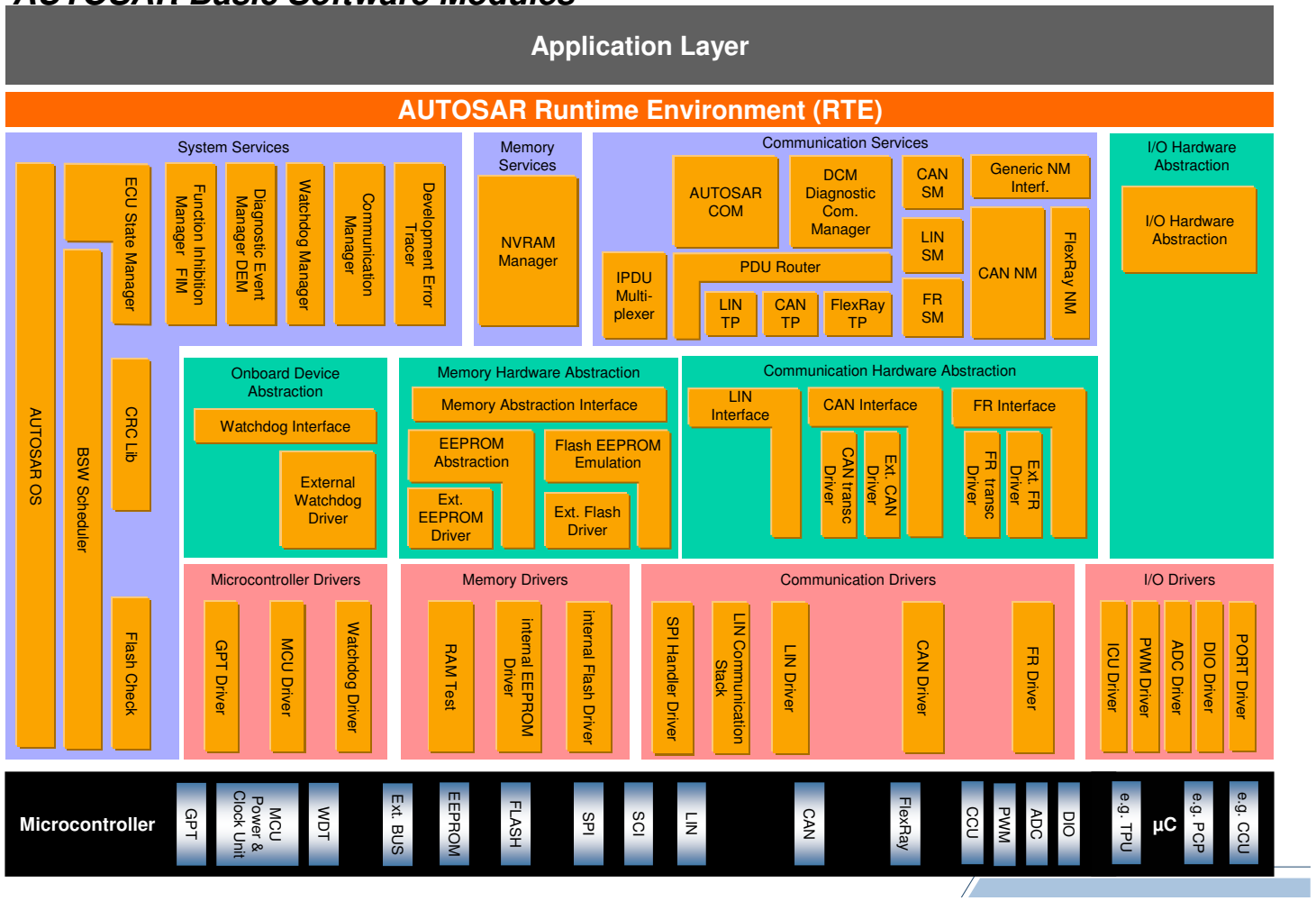
Klasse	Übertragungsraten	Anwendung	Vertreter
Klasse A	Geringe Datenraten (bis 10 kBit/s)	Vernetzung von Aktoren und Sensoren	LIN
Klasse B	Mittlere Datenraten (bis 125 kBit/s)	Komplexe Mechanismen zur Fehlerbehandlung, Vernetzung von Steuergeräten im Komfortbereich	Lowspeed-CAN
Klasse C	Hohe Datenraten (bis 1 MBit/s)	Echtzeitanforderungen, Vernetzung von Steuergeräten im Antriebs- und Fahrwerksbereich	Highspeed-CAN
	bis zu 15 MBit/s	Wie Klasse C, zusätzlich - Mehr ECUs / CAN - Schnellere Kommunikation über lange CAN	CAN FD (Flexible Data Rate)
Klasse C+	Sehr hohe Datenraten (bis 10 MBit/s)	Echtzeitanforderungen, Sicherheitsanforderungen, Vernetzung von Steuergeräten im Antriebs- und Fahrwerksbereich	FlexRay
Klasse D	Sehr hohe Datenraten (> 10 MBit/s)	Vernetzung von Steuergeräten im Telematik- und Multimediabereich Verliert an Bedeutung	MOST
		in Einführung	Ethernet

Quelle: BOSCH: Kraftfahrtechnisches Taschenbuch, Vieweg+Teubner, 27. Auflage, 2011.

Gelbe Felder ergänzt

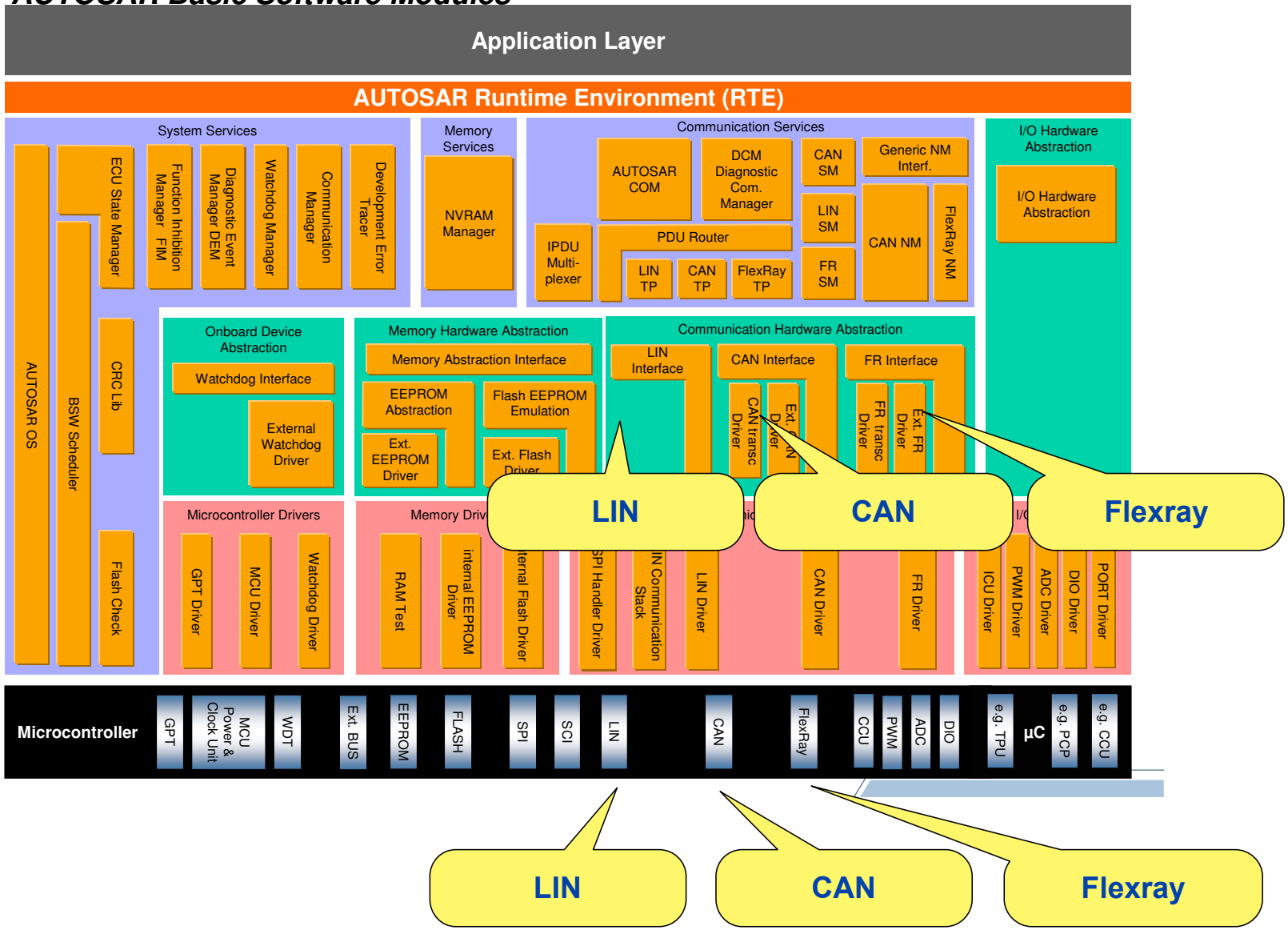
AUTOSAR Basis Software Module

AUTOSAR Basic Software Modules



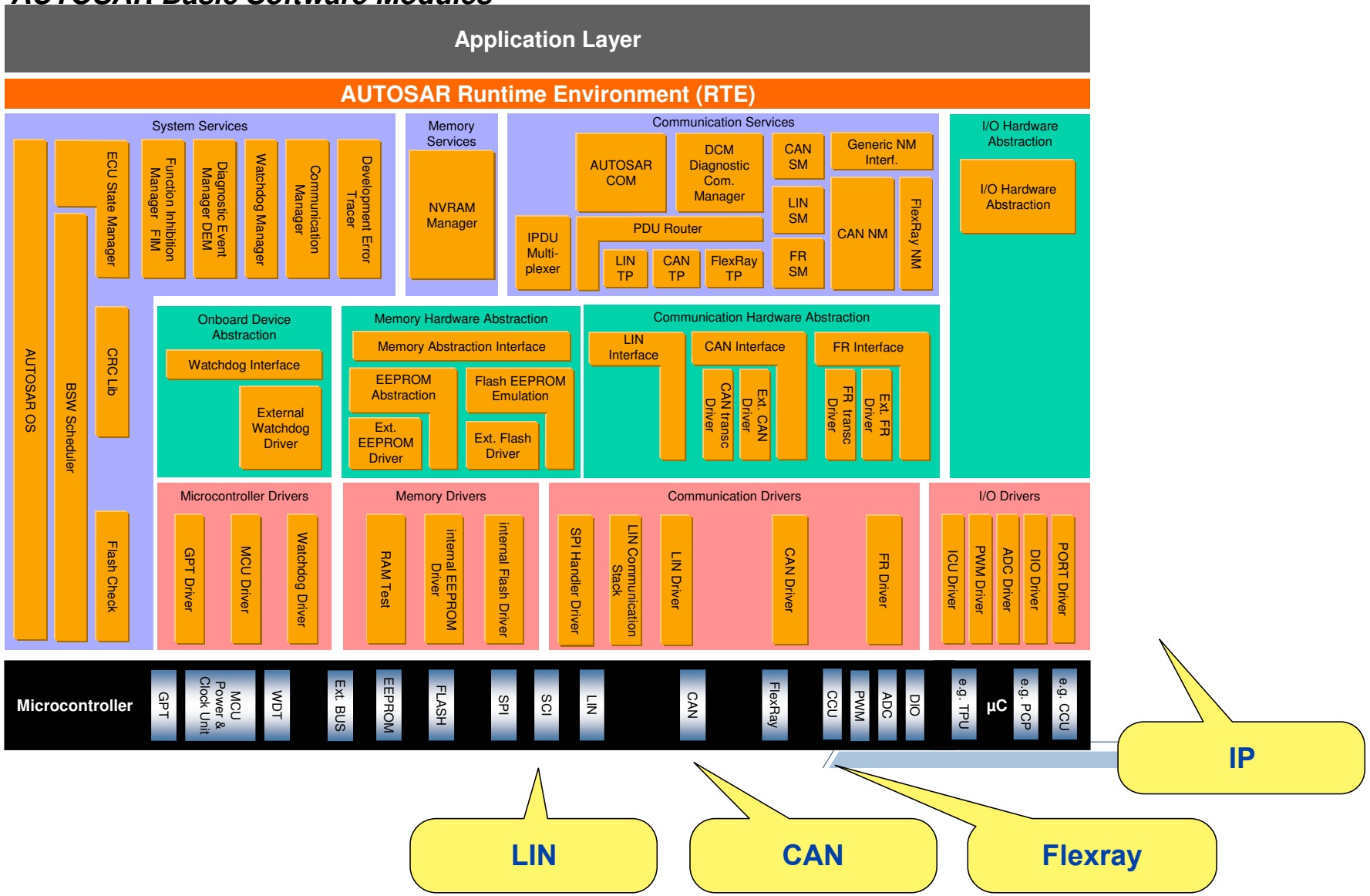
AUTOSAR Basis Software Module

AUTOSAR Basic Software Modules



AUTOSAR Basis Software Module

AUTOSAR Basic Software Modules



TCIP/IP-Erweiterung in Release 4.0

AUTOSAR Release 4.0
TCP/IP Extension of the CommStack – Overview