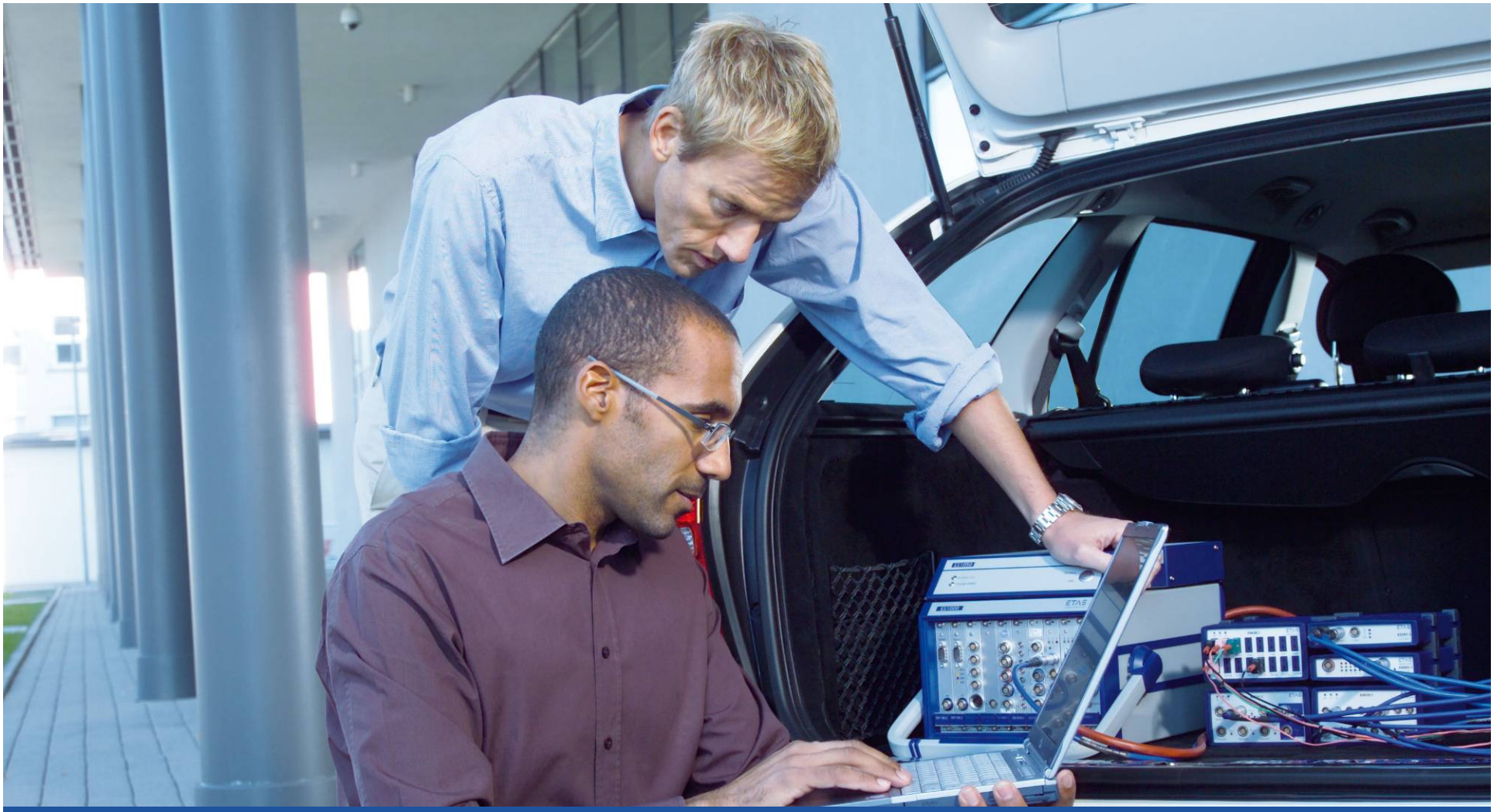
AUTOSAR-Software mit ASCET (Dr. Kai Pinnow – ETAS GmbH)
TU Darmstadt, Industrie Kolloquium, 26. Mai 2009

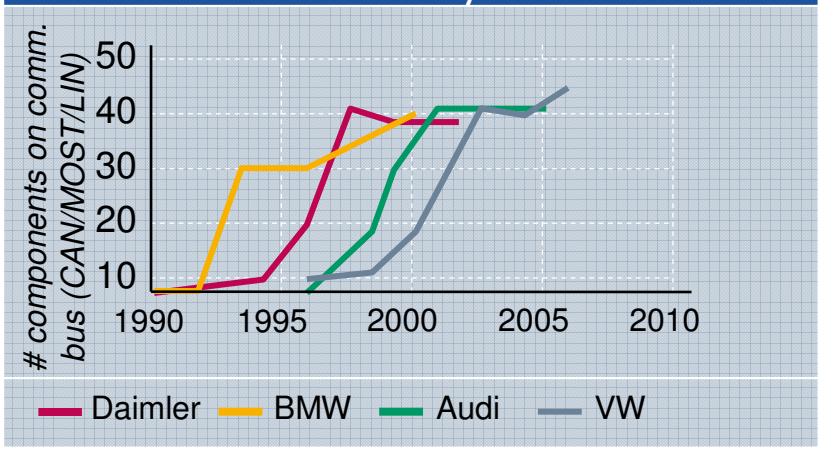


AUTOSAR-Software mit ASCET
Übersicht

- AUTOSAR (AUTomotive Open System ARchitecture)
  - Komplexitätskrise: "cooperate on standards, compete on innovation"
  - Partnerschaft von Automobilherstellern und –zulieferern
  - Standardisierung von Basis-Softwarearchitekturen und -schnittstellen
  - Hardware-Abstraktion und gemeinsames Runtime-Environment
  - Mehr Flexibilität, Skalierbarkeit, Qualität und Zuverlässigkeit durch eine vereinheitlichte AUTOSAR-Methodologie
- ETAS
  - Firmenprofil und Produkte
- ASCET – Modellbasierte Softwareentwicklung
  - Regelungstechnische Blockdiagramme und Zustandsautomaten
  - AUTOSAR kompatible Modellierung
  - C-Code Generatoren für Echtzeitsysteme

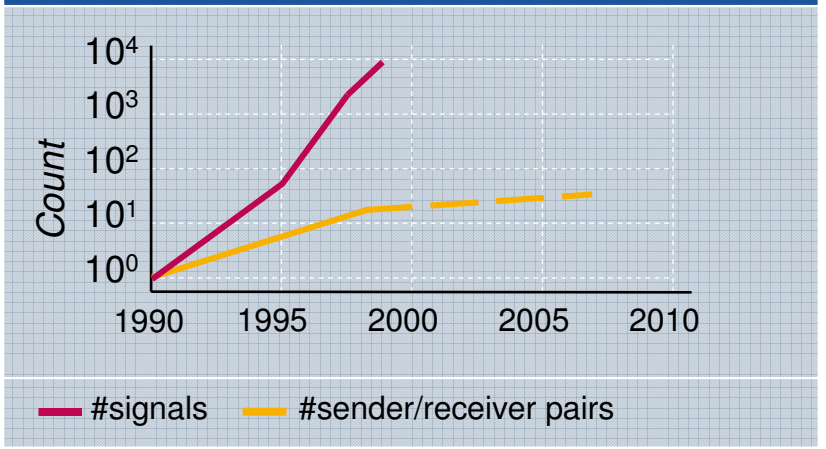
Komplexitätskrise
Die Komplexität von E/E-Systemen im Automobil wächst zunehmend

Anzahl der Systeme



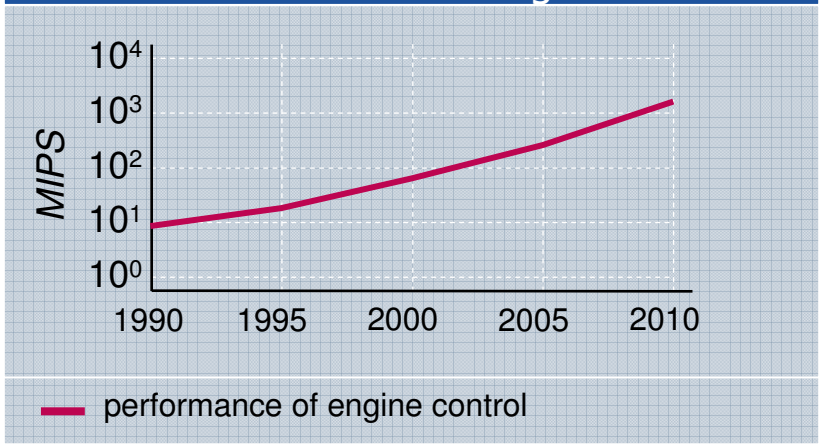
Quelle: VW 2005, Fachkongreß Automobil-Elektronik

Kommunikation



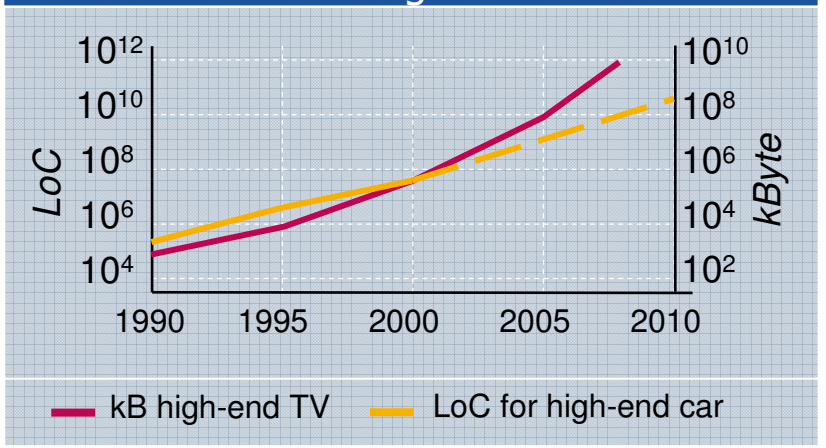
Quelle: BMW, Frischkorn, BoCSE 2002

Rechenleistung



Quelle: NEC, 2006 (TOP57)

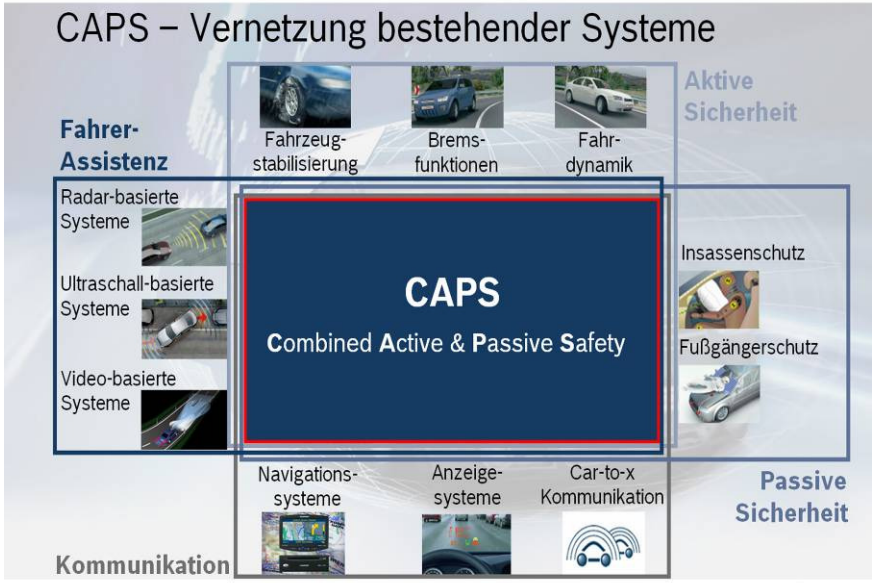
Codegröße



Quellen: Daimler-Chrysler 2004; Philips

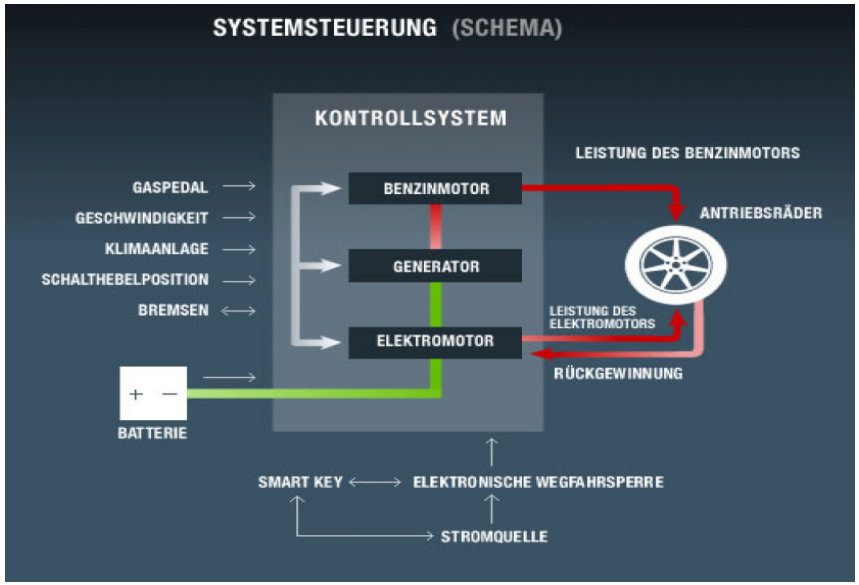
Komplexitätskrise
Beispiele für innovative komplexe Systeme und deren Komplexitätstreiber

Vernetzung von Fahrerassistenz- und Sicherheitssystemen (CAPS)[1]



- Darstellung neuer Funktionen durch Verknüpfung bestehender Systeme (ACC+ESP -> Predictive Brake Assistant)
- Mehrfache Nutzung von Sensorsignalen
- Einfache Integration neuer Systeme und Funktionen durch modularen Aufbau

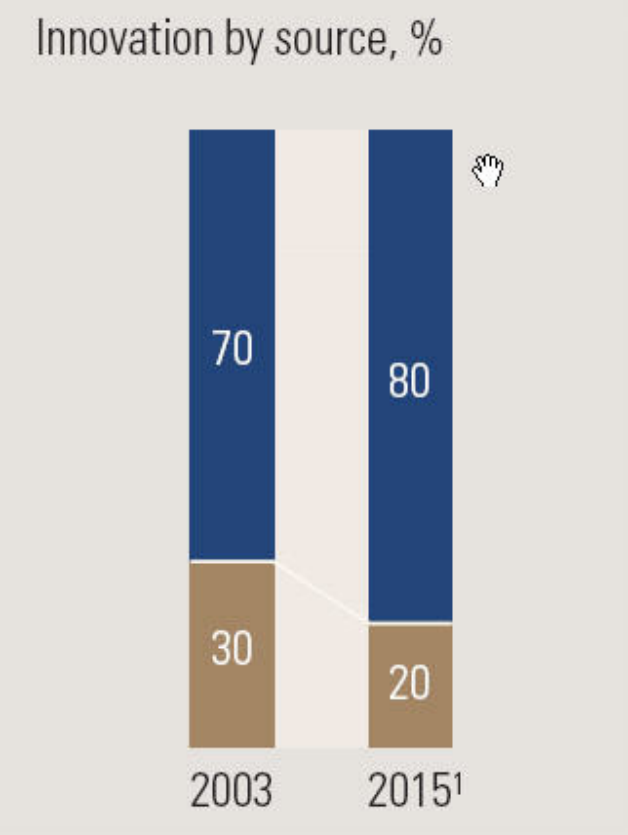
Zentrale Hybrid-Antriebssteuerung[2]

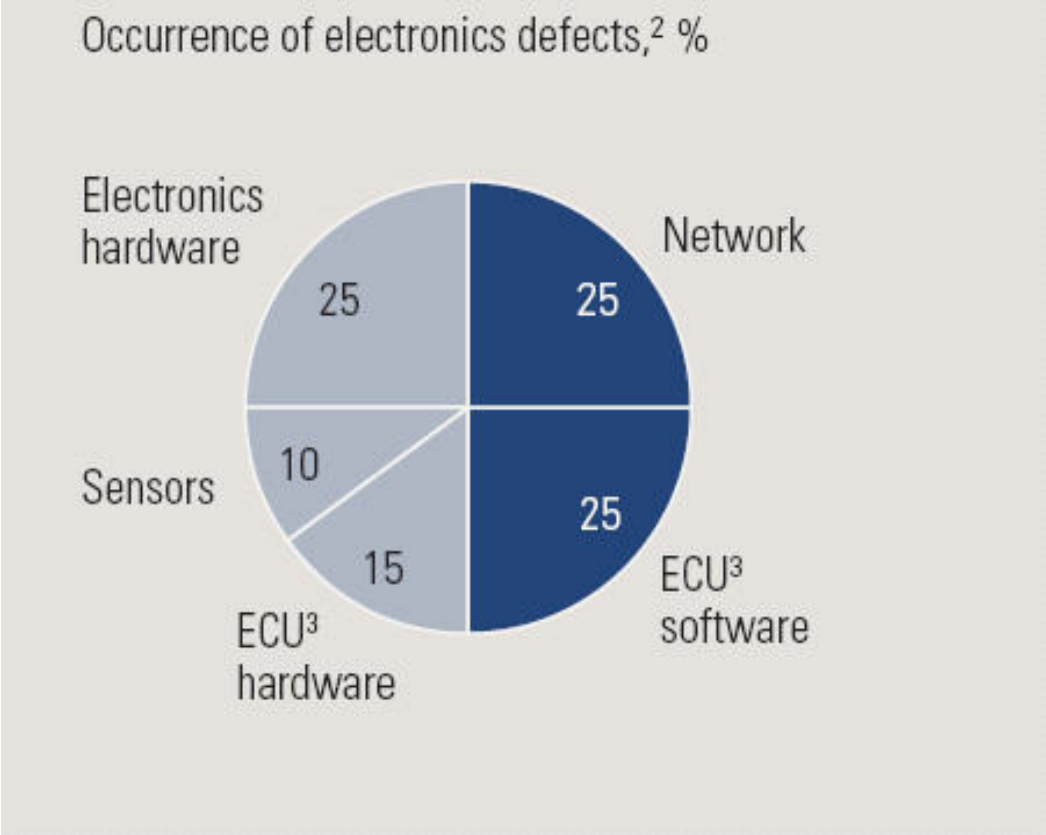


- Zentrale Steuerung der Hybrid-Komponenten (Verbrennungs- und Elektromotor, Generator & Batterie, Getriebe – bislang noch eher getrennte Domänen)
- Zentrales Energiemanagement (Klimaanlage, Licht, Infotainment,..)

Quellen: [1] Bosch, [2] Toyota

Komplexitätskrise
Software als Innovationstreiber und Qualitätsthema

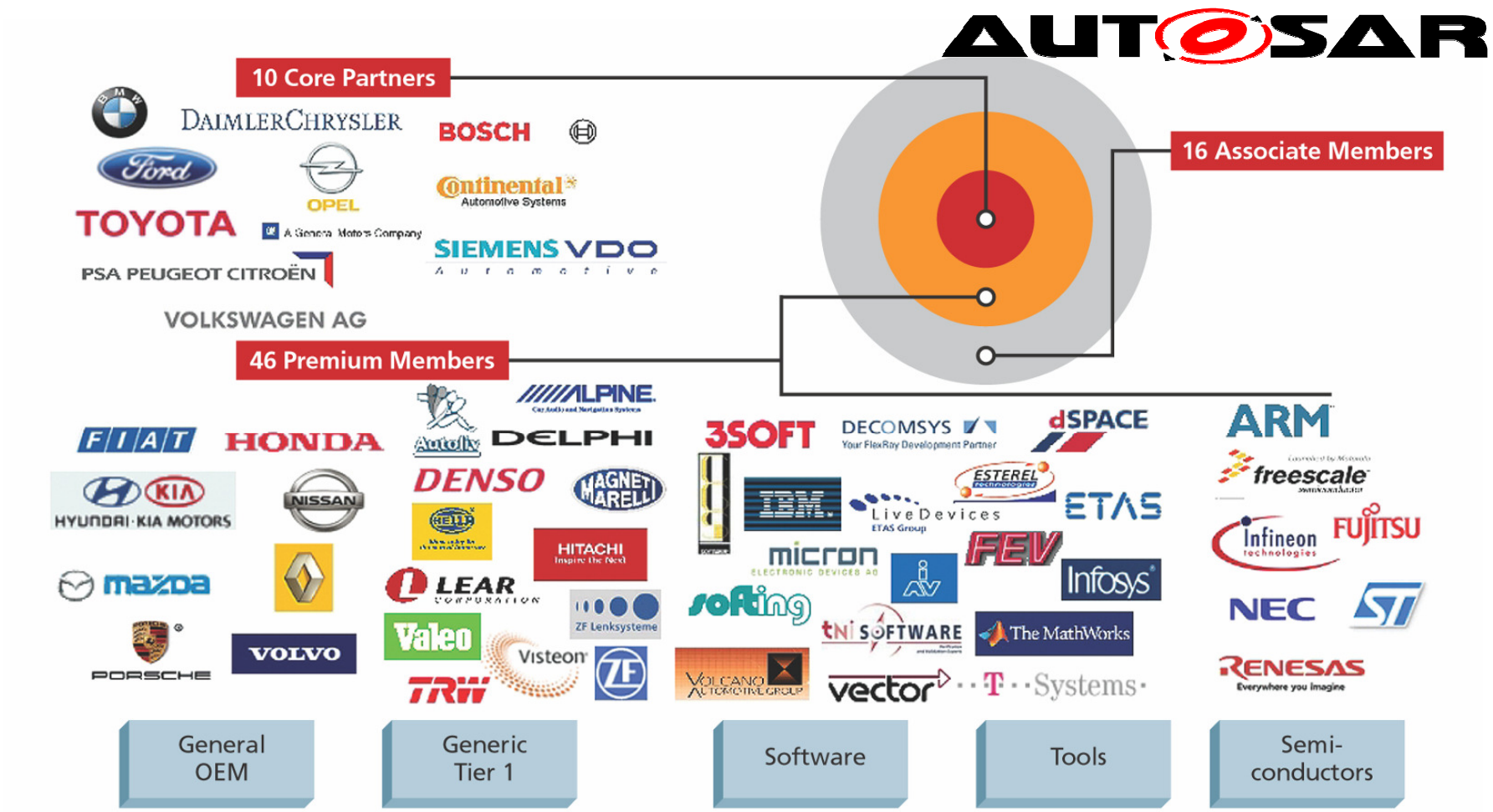




[1]Projected.
[2]German cars, 2003; data representative for all luxury cars with >50 electronic control units.
[3]ECU = electronic control units (the embedded controllers that combine chips and code to control function of vehicle).

Source: 2003 McKinsey–PTW HAWK survey (Institute for Production Management, at the Technical University of Darmstadt); McKinsey analysis

Partnerschaft von Automobilherstellern und -zulieferern
AUTOSAR-Initiative hat breite Zustimmung der Industrie gefunden



Quelle: ETAS RealTimes magazine 2/2005

Partnerschaft von Automobilherstellern und -zulieferern
AUTOSAR-Initiative findet breite Zustimmung

AUTOSAR

Pressemitteilung 3. Dezember 2008

Erste AUTOSAR-Konferenz in den USA mit positivem Echo

Mehr als 100 nordamerikanische Fachleute, Akademiker und Journalisten kamen zur ersten „Open Conference" der Entwicklungspartnerschaft AUTOSAR (Automotive Open Systems Architecture), die Ende Oktober 2008 in Detroit stattfand.

„Der Erfolg dieser Konferenz zeigt die weltweit zunehmende Bedeutung von AUTOSAR", so Gerulf Kinkelin, Sprecher von AUTOSAR und Innovation Area Manager für Elektrik, Elektronik und Telematik bei PSA Peugeot Citroën. „Wir werden unsere Bemühungen fortführen, den AUTOSAR-Standard auch außerhalb von Europa zu etablieren." Die Konferenz fällt in eine Zeit, in der zum Einen erstmals Fahrzeuge mit AUTOSAR in Serie gehen, und andererseits die internationale Ausrichtung von AUTOSAR immer wichtiger wird.

Großes Interesse fand der einführende Vortrag von Jim Buczkowski, Direktor für elektronisches und elektrisches Systemdesign bei Ford. Er erläuterte die Vorteile offener Standards und die wichtige Rolle von AUTOSAR bei künftigen Neuvorstellungen. "Das belegen die Roadmaps der Automobilhersteller und Zulieferer", so Buczkowski. Die Frage-Antwort-Runde mit den Core-Partnern der Entwicklungspartnerschaft wurde ebenfalls rege genutzt.

In weiteren Vorträgen stellten Unternehmen wie Delphi-
ETAS, Hyundai, KPIT, Magneti Marelli, Tata Ele
Erfahrungen bei der Umsetzung von AUTC
integralen Bestandteil der Standard kü
mobilentwicklung spielen wird

Quellen:
www.autosar.org

AUTOSAR

Pressemitteilung August 2008

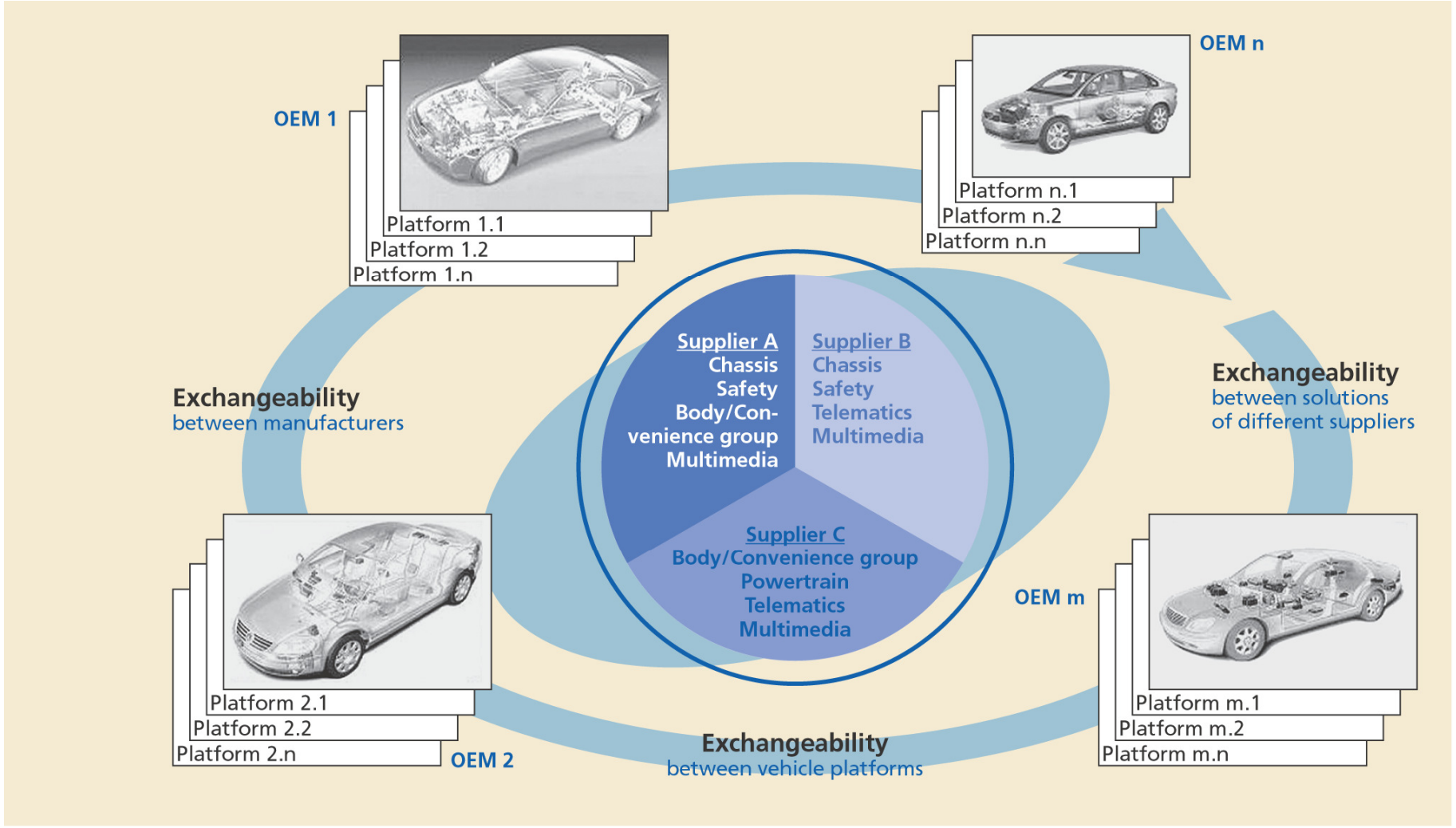
AUTOSAR veröffentlicht neue Spezifikationen: Release 3.1 mit On-Board-Diagnose

Die Entwicklungspartnerschaft AUTOSAR veröffentlichte am 15. August 2008 neue Spezifikationen, mit denen erstmals Regeln zur Einbindung der On-Board-Diagnose (OBD) gemäß dem OBDII-Standard festgelegt werden. Für das neue Release 3.1 haben die Entwickler elf Dokumente des vorangegangenen Release 3.0 angepasst. „Diese Erweiterung ist wichtig für Europa und Japan, aber insbesondere für den Markt in den USA. Die dortigen Vorschriften der OBD sind im internationalen Vergleich am umfangreichsten. Von nun an kann AUTOSAR weltweit eingesetzt werden", erläutert Dr. Jürgen Mössinger, AUTOSAR-Sprecher und Hauptreferent Automotive Systems Integration bei Bosch.

Die On-Board-Diagnose wurde Ende der 1980er Jahre erstmals in Kalifornien eingeführt. Ihre Hauptaufgabe ist es, während des Fahrbetriebs alle abgasrelevanten Daten zu überwachen und den Fahrer bei Abweichungen von den Standardwerten zu informieren. Dadurch trägt die On-Board-Diagnose wesentlich dazu bei, die geforderten Abgasgrenzwerte über die gesamte Lebensdauer des Fahrzeugs einzuhalten. Mittlerweile gibt es ähnliche Regelungen auch in Europa und Japan.

Das aktuelle AUTOSAR-Release umfasst 141 Spezifikationen und wird im Internet unter www.autosar.org frei zugänglich sein. Die Mitglieder Initiative dürfen die AUTOSAR-Spezifikationen kostenfrei ihrer Steuergeräte-Software nutzen.

Partnerschaft von Automobilherstellern und -zulieferern
Plattformentwicklung für unterschiedliche Kunden und Modelle



Partnerschaft von Automobilherstellern und -zulieferern
Herausforderungen, Lösungen → Vorteile 1 / 3

- Verbesserungspotential „Prozesssicherheit" durch fehlende Verfolgbarkeit funktionaler Anforderungen und inkompatible Werkzeuge
  Standardisierung von Spezifikationen und Formaten
  → Durchgängige Werkzeuglandschaft und verbessere Spezifikationen

- Unnötiger Aufwand für Implementierung und Optimierung von Funktionen, die vom Kunden nicht als wertvoll erkannt werden
  Vereinheitlichte Basis-Software
  → Verbesserung der Softwarequalität und mehr Konzentration auf kundenwertige Funktionen

Partnerschaft von Automobilherstellern und -zulieferern
Herausforderungen, Lösungen → Vorteile 2 / 3

- Modellwechsel bei Mikroprozessoren zieht hohen Portierungsaufwand nach sich; wachsende Performanzanforderungen bedingen Re-Design

  Hardware-Abstraktion

  → Austausch von Mikroprozessoren ohne Anpassung höherer Softwareebenen möglich

- Hoher Aufwand, Funktionen zwischen Mikroprozessoren zu verschieben (zur Performanz-, Speicher- oder Kommunikationsoptimierung); hoher Aufwand für die Wiederverwertung von Funktionen

  Runtime Environment (RTE)

  → Kapselung von Funktionen unabhängig von Kommunikation

  → Vereinfachte standardisierte Kommunikationsmechanismen

  → Zerlegen und Verschieben von Funktionen vereinfacht

Partnerschaft von Automobilherstellern und -zulieferern
Herausforderungen, Lösungen → Vorteile 3 / 3

- Inkompatible Funktionen müssen in der Hersteller-Umgebung angepasst werden; selbst kleinere Verbesserungen erfordern hohen Aufwand in der Bereitstellung geeigneter Schnittstellen fremder Komponenten; fehlende Schnittstellen zwischen Basis-Software und automatisch generierter Anwendungs-Software

  Standardisierte Schnittstellen

  → Reduktion von Schnittstellen-Wucherungen durch unterschiedliche Hersteller- und Zuliefererinteressen

  → Vereinfachte Implementierung Hardware unabhängiger Funktionen auf Basis generischer Schnittstellen-Kataloge

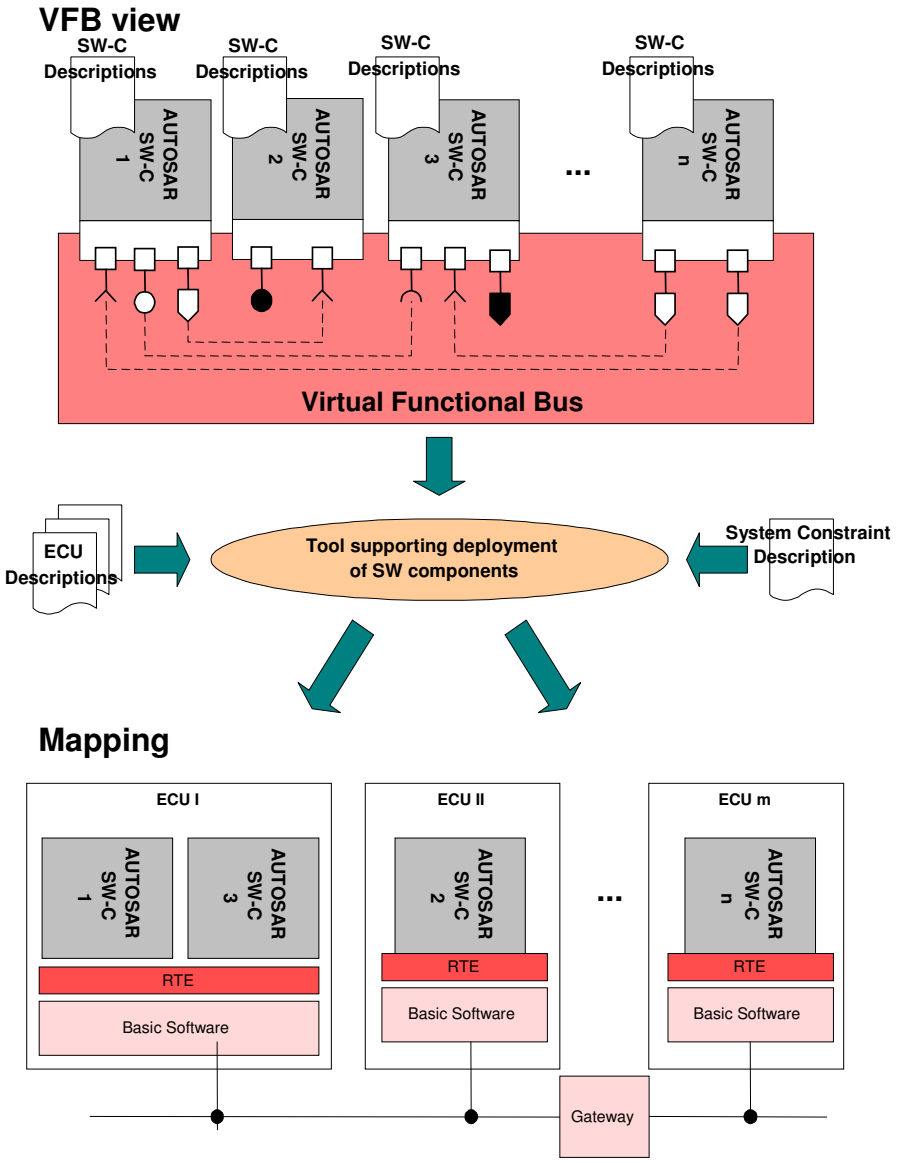
  → Vereinfachung modellbasierter Entwicklung durch standardisierte AUTOSAR Codegenerierungs-Werkzeuge

  → Wiederverwertung von Modulen über Herstellergrenzen hinweg

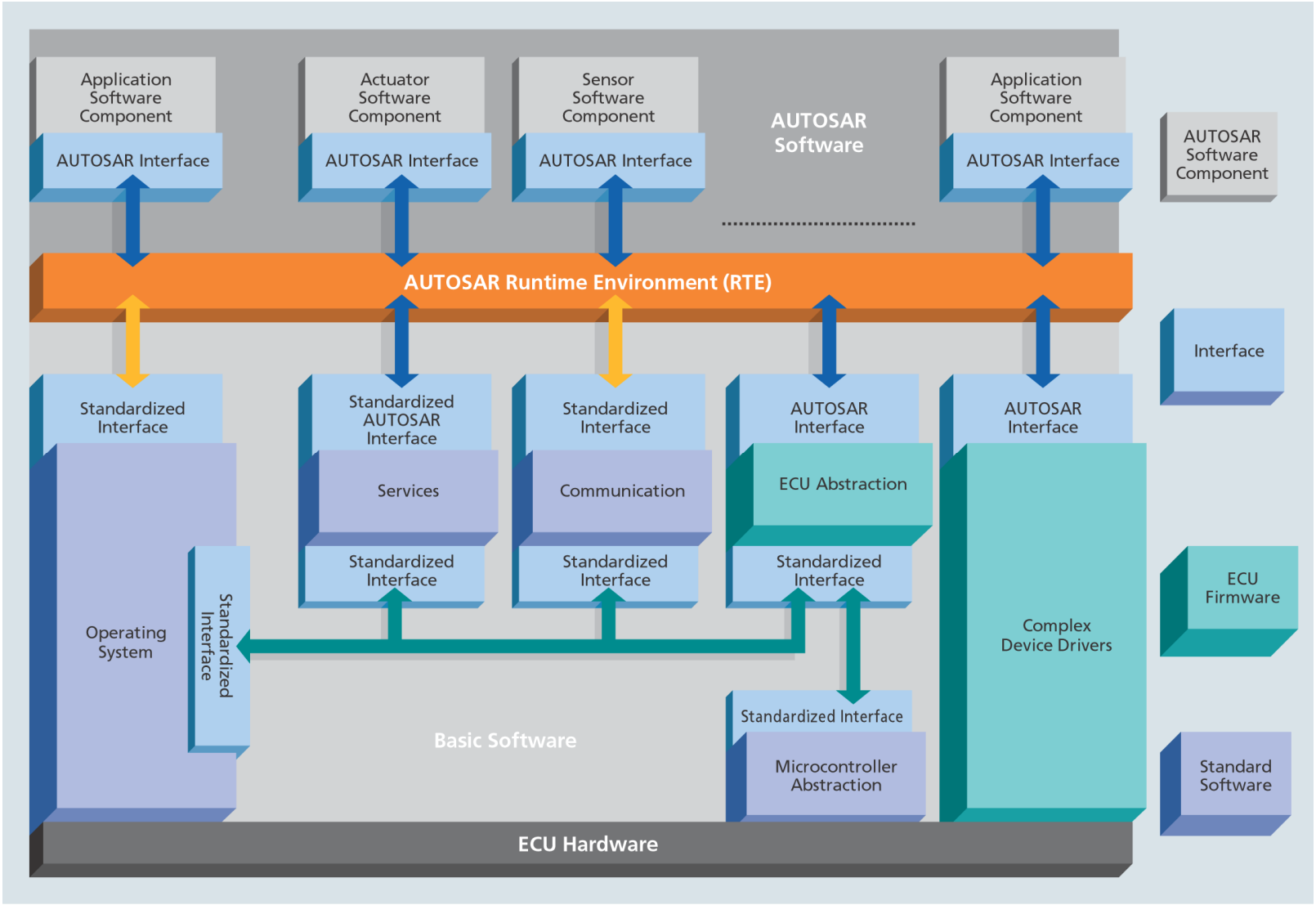
  → Austauschbarkeit von Komponenten unterschiedlicher Zulieferer

Standardisierung von Basis-Softwarearchitekturen und -schnittstellen
Architektur-Elemente

- **SW-C Description**
  - Standardisierte Beschreibung von Software-Komponenten (Application-Layer)
- **AUTOSAR Software Components (SW-C)**
  - Anwendungs-Software mit definierten Schnittstellen
- **Virtual Function Bus (VFB)**
  - Software-Bus für Message-Passing und Client-Server-Kommunikation (nach Bedarf)
- **Electronic Control Unit (ECU) und System Constraint Description**
- **Mapping on ECUs**
  - Betriebssystems-Konfiguration und Build-Prozess
- **Runtime Environment (RTE)**
  - Implementierung des VFB auf einer ECU
- **Basic Software**
  - Echtzeitbetriebssystem
  - Hardware-Treiber
  - Infrastruktur-Funktionen

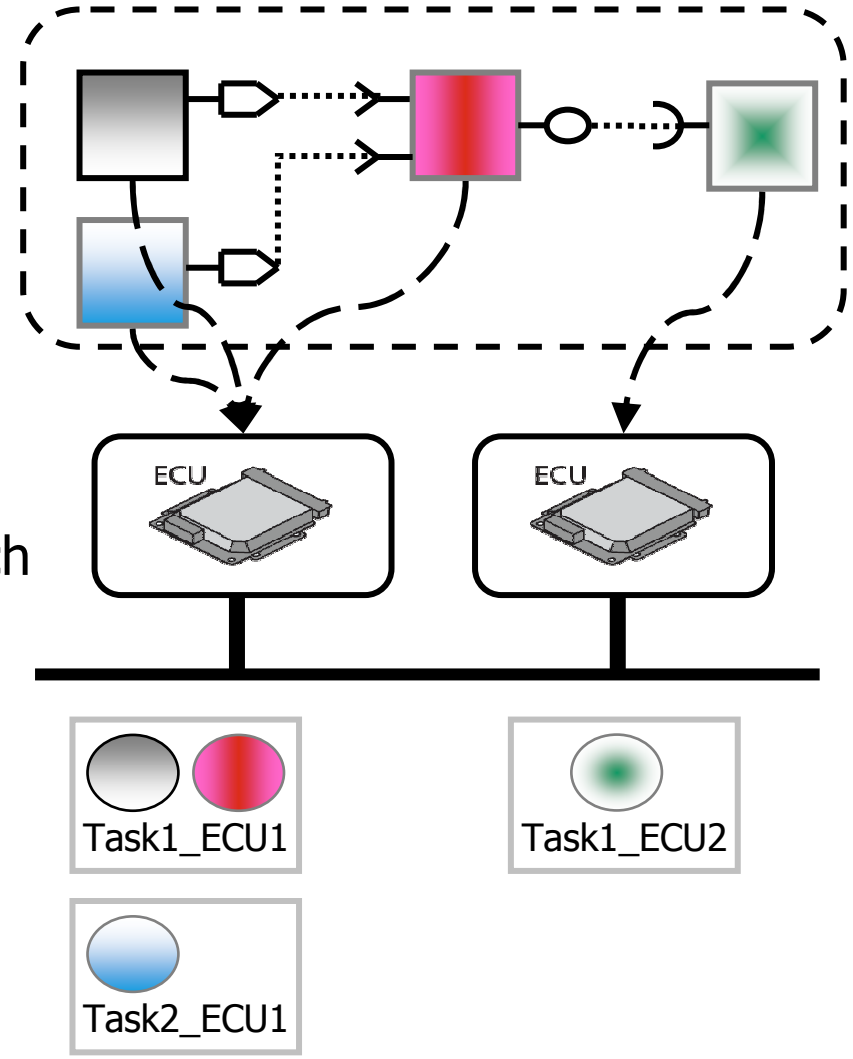


Standardisierung von Basis-Softwarearchitekturen und -schnittstellen
AUTOSAR-Softwarearchitektur

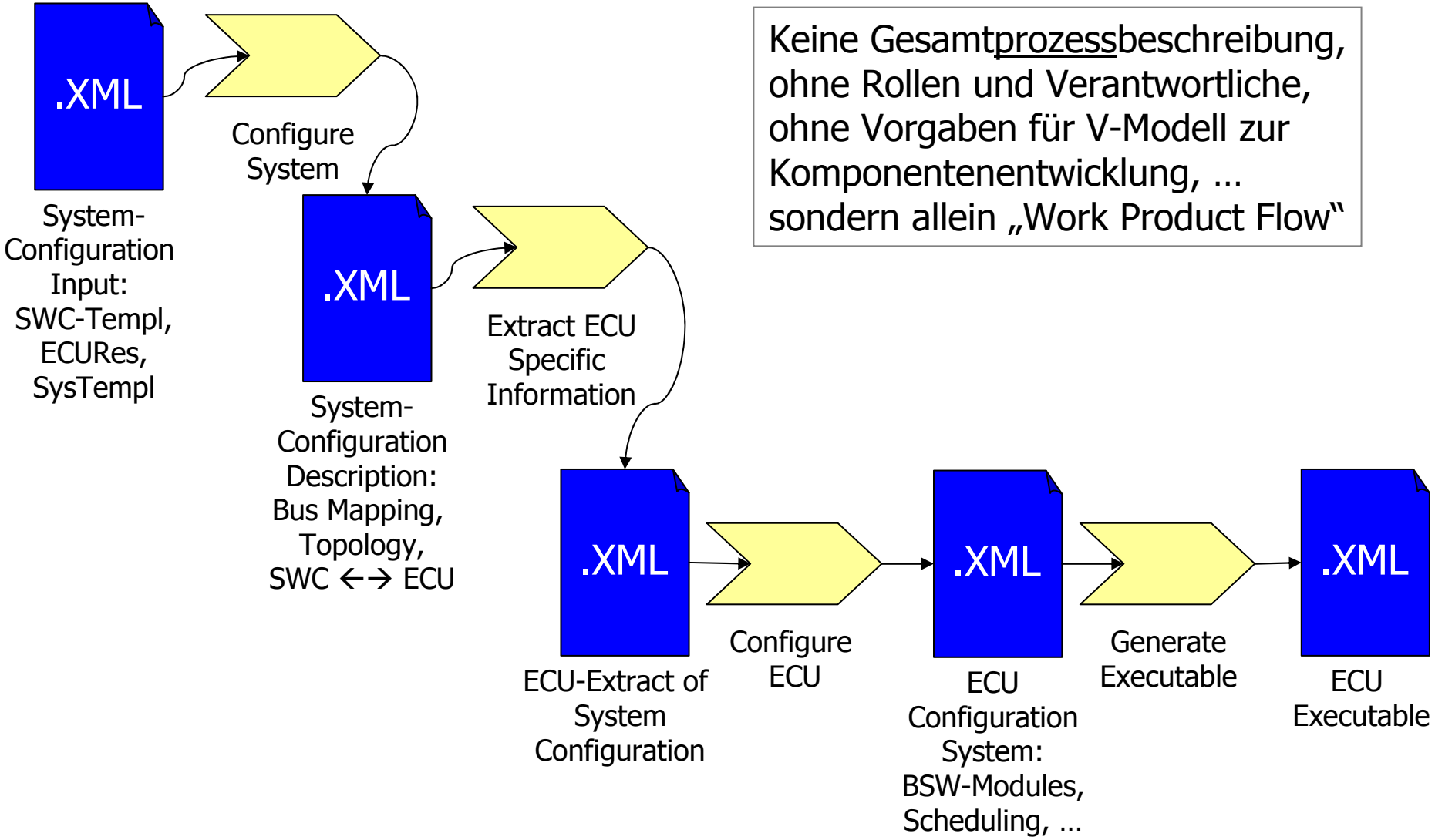


AUTOSAR Kommunikationsmechanismen auf dem VFB
Kommunikation wird durch den VFB abgebildet

- SW-Cs bilden Gesamtsystem
- Systeme implementiert durch
  - Mapping von SW-Cs auf ECUs
  - Mapping von "Runnables" (zuweilen auch als „Prozesse" bezeichnet) in den SW-Cs auf Tasks im Betriebssystem
- Kommunikation von SW-Cs möglich
  - Innerhalb Tasks (intra-task)
  - Zwischen Tasks (inter-task)
  - Zwischen ECUs (inter-ECU)
- Kommunikationsformen
  - Sender-Receiver (Messages)
  - Client-Server

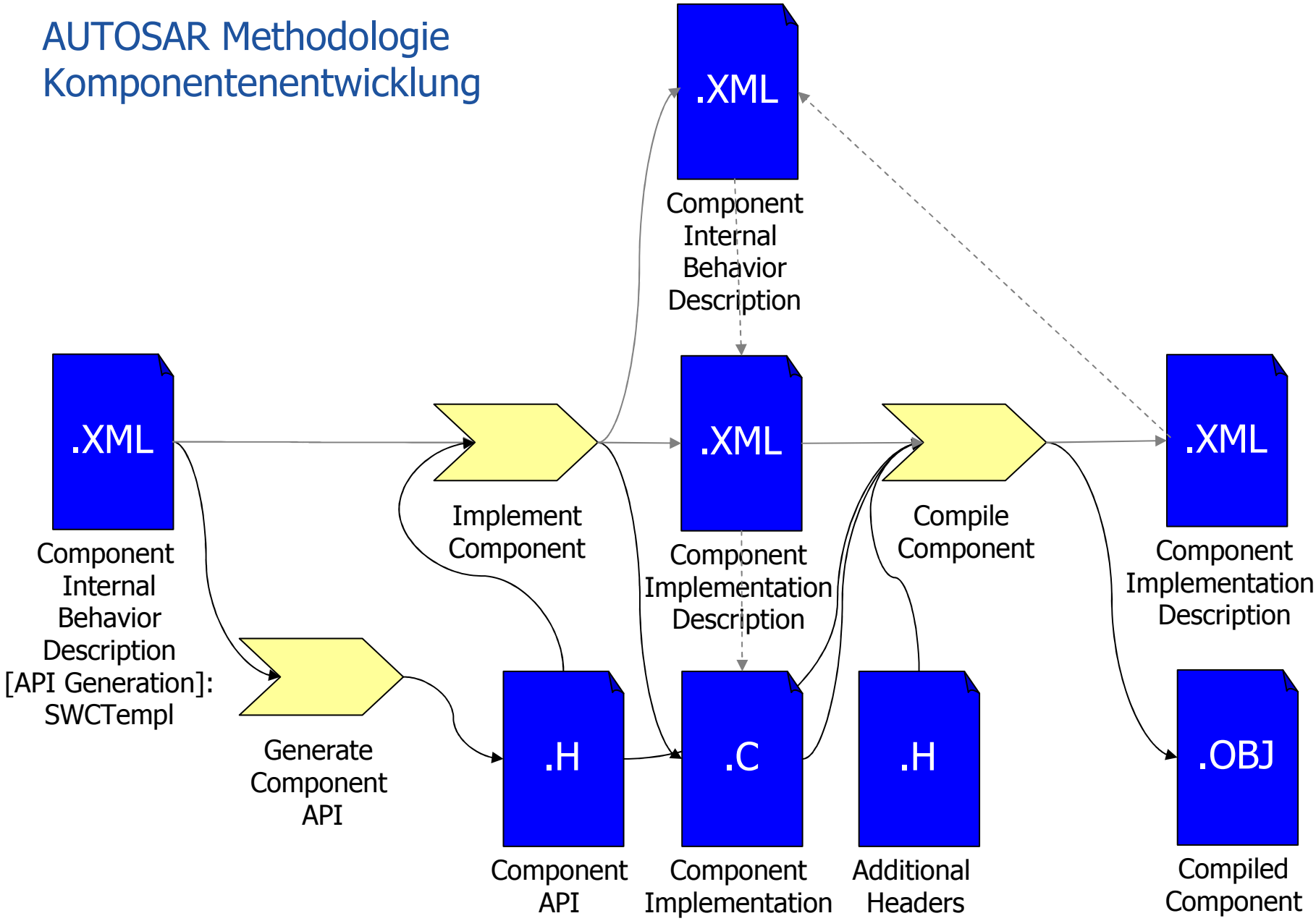


AUTOSAR Methodologie
Entwicklung des Gesamtsystems



Keine Gesamtprozessbeschreibung, ohne Rollen und Verantwortliche, ohne Vorgaben für V-Modell zur Komponentenentwicklung, … sondern allein „Work Product Flow"

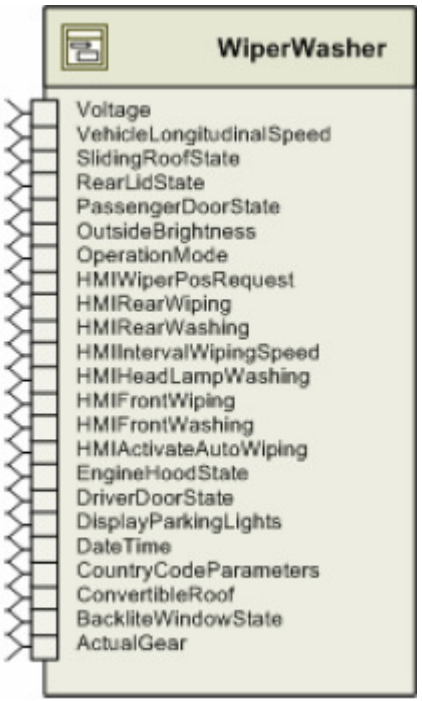
AUTOSAR Methodologie
Komponentenentwicklung



AUTOSAR High-Level Schnittstellen
Beispiel Body- und Komfort-Elektronik: „Wiper and Washer"

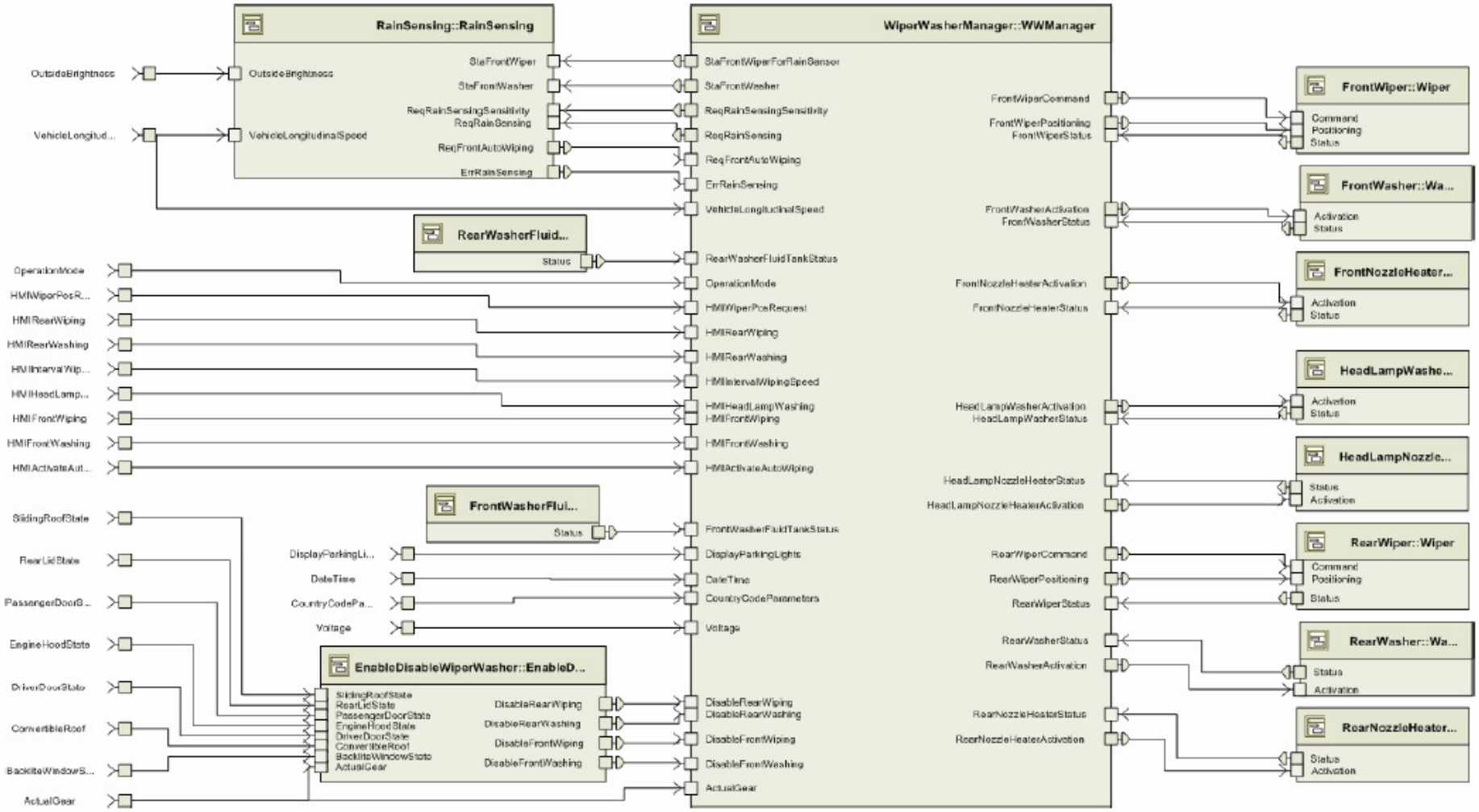
- Aufteilung in Subsysteme
  - Zugangskontrolle
  - Licht und Sicht
  - Akustische Warnungen
  - Komfort (Sitz, Fenster, etc.)
  - Parken, Batteriekontrolle, …

- Namenskonventionen
  - Cmd – Kommando
  - Req – Anfrage
  - Sta – Status
  - Hmi – Benutzeranforderung
  - Dis – Status für Fahrerinformationssystem
  - Err – Fehlerrückmeldung



Quelle: www.autosar.org

AUTOSAR High-Level Schnittstellen
Innere Struktur der Wischersteuerung



Quelle: www.autosar.org

Firmenprofil
Mitglieder der ETAS Group

- ETAS Entwicklungs- und Applikationswerkzeuge für elektronische Systeme GmbH
  - Gründungsjahr: 1994
  - Gesellschafter: 100 % Robert Bosch GmbH
  - Stammsitz: Stuttgart, Deutschland
    13 weitere Standorte weltweit

- Vetronix Corporation
  - Spezialisiert auf Service- und Diagnosewerkzeuge
  - Gründungsjahr: 1984
  - Gesellschafter: 100 % Robert Bosch GmbH
  - Stammsitz: Santa Barbara, USA

Firmenprofil
Weltweite ETAS Group-Standorte

Europa

460 Mitarbeiter

Standorte: Stuttgart/Deutschland, Rungis/ Frankreich, Burton-upon-Trent und York/ Großbritannien, Turin/Italien, Moskau/ Russische Förderation, Stockholm/Schweden

Asien-Pazifik

99 Mitarbeiter

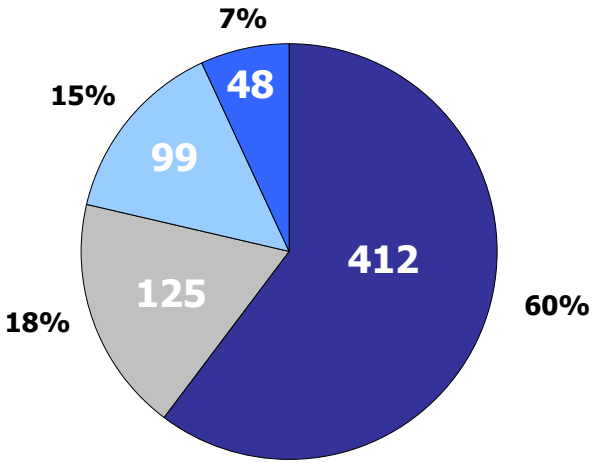
Standorte: Yokohama und Nagoya/ Japan, Seoul/Korea, Shanghai/ Volksrepublik China, Bangalore/Indien

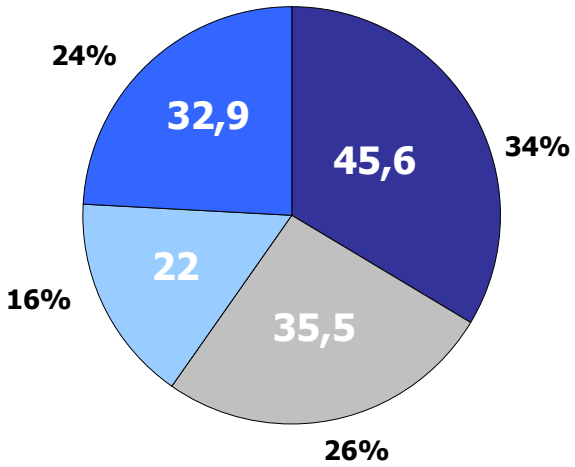
Nord- und Südamerika

125 Mitarbeiter

Standorte: Ann Arbor und Santa Barbara/USA, Sao Paulo/Brasilien

Firmenprofil
Mitarbeiter und Umsatz pro Region





Gesamt: 684 Mitarbeiter

Stand: 1. Mai 2009

Gesamt: 136 Millionen Euro Umsatz

Stand: 31. Dezember 2008

Deutschland | Nord- und Südamerika | Asien-Pazifik | Europa ohne Deutschland

Firmenprofil
Markt und Kunden

- ETAS fokussiert zu 100 % auf die Automobilindustrie
- Zu den ETAS-Kunden gehören
  - Fahrzeughersteller und Steuergerätelieferanten
    - Entwicklungs- und Service-Abteilungen
  - Engineering-Dienstleister

Auszug aus der Kundenliste:

Firmenprofil
Unternehmensstrategie

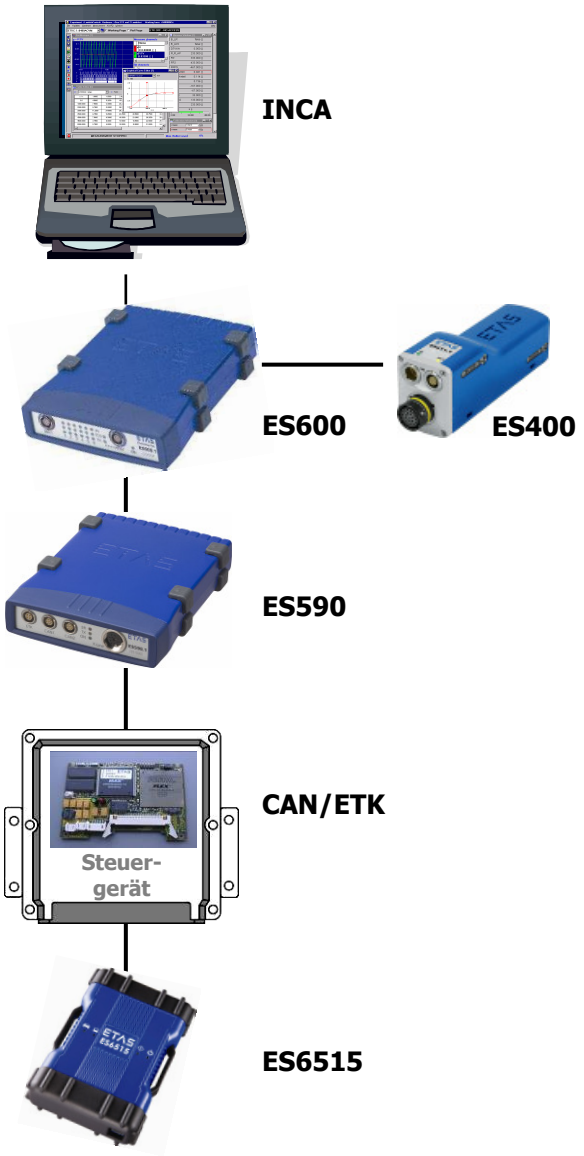
1 **Unsere anerkannte Steuergeräte-Kernkompetenz und die Vertrautheit mit Kundenprozessen werden auch in Zukunft die Basis unseres Handelns bilden.**

2 **ETAS stellt umfassende und durchgängige Werkzeuge sowie Werkzeug-lösungen für die Entwicklung und den Service von automobilen Steuergeräten zur Verfügung.**

3 **Innovative, hochwertige sowie kostenoptimierte Produkte und Lösungen für alle Phasen der Entwicklung und des Service von Steuergeräten zeichnen die ETAS Group als Premium-Toollieferant aus.**

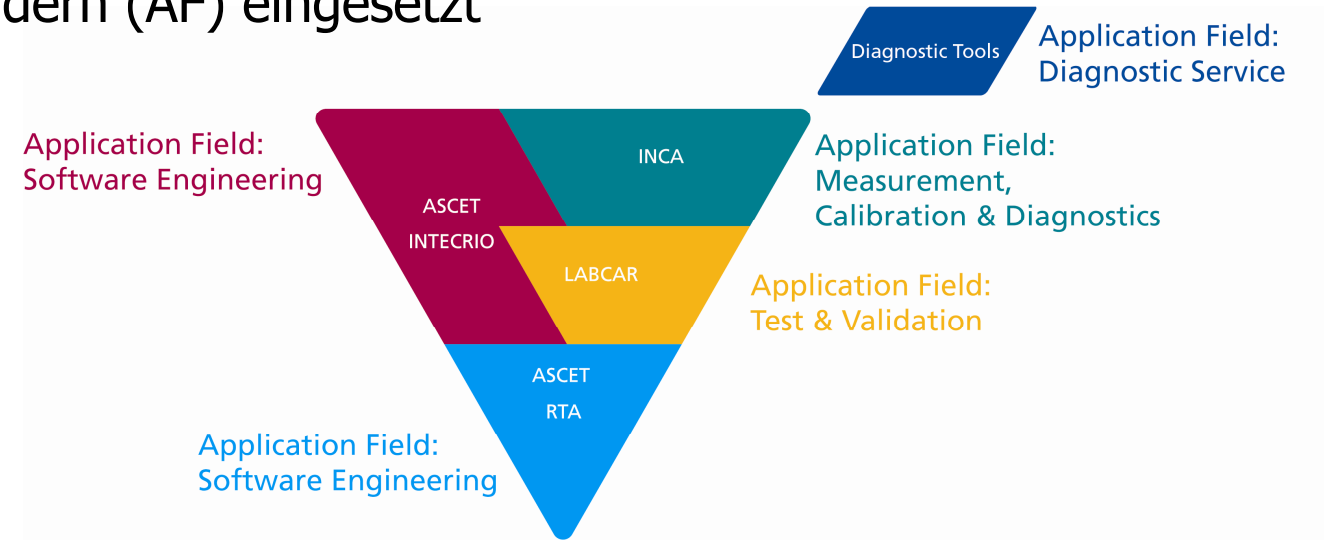
Firmenprofil
Lösungen der ETAS Group

- Durchgängige Hardware- und Software-produkte erlauben eine nahtlose Einbindung in die Software-Entwicklungsumgebung für automobile Steuergeräte
  - Software für Funktions- und Software-Entwicklung, Testen, Validieren sowie Messen und Kalibrieren:
    ASCET, INTECRIO, LABCAR, INCA, RTA
  - Skalierbare Hardware für Prototyping, Steuergerätezugang, Mess-, Verstell- und Validierungsaufgaben sowie Drive Recording – im Fahrzeug, am Prüfstand und im Labor
- Fahrzeugschnittstellenmodule erlauben einen nahtlosen Zugang zu Diagnose-Informationen für Werkstätten



Firmenprofil
Steuergerätesoftware-Entwicklung mit dem V-Modell

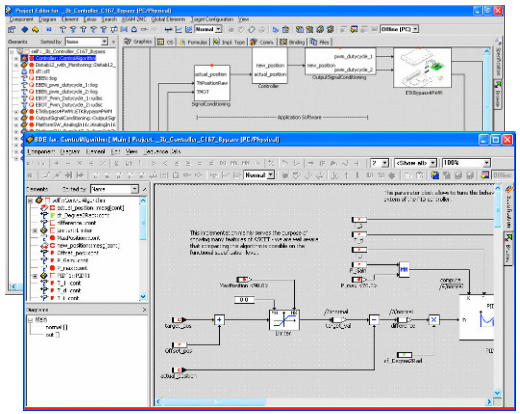
- Der Steuergerätesoftware-Entwicklungsprozess wird durch das so genannte V-Modell dargestellt
- Das V-Modell zeigt die verschiedenen Phasen, die bei der Entwicklung von Steuergerätesoftware durchlaufen werden
- Werkzeuge der ETAS Group begleiten alle Phasen des Steuergeräte-software-Entwicklungsprozesses entlang des V-Modells (bis zur Werkstattdiagnose); sie werden in den entsprechenden Anwen-dungsfeldern (AF) eingesetzt

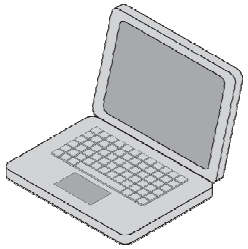


Firmenprofil
Überblick Entwicklungs- und Servicewerkzeuge (1/2)

Modellierung & Simulation

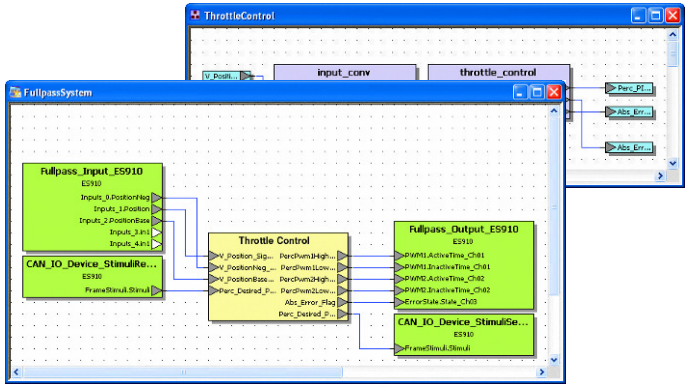
ASCET-MD





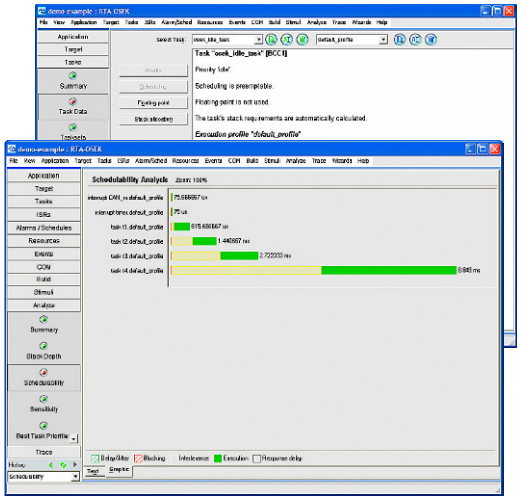
Rapid Prototyping

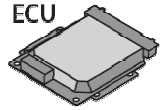
INTECRIO mit ES1000 oder ES900



Steuergeräte-Codegenerierung

ASCET-SE und RTA

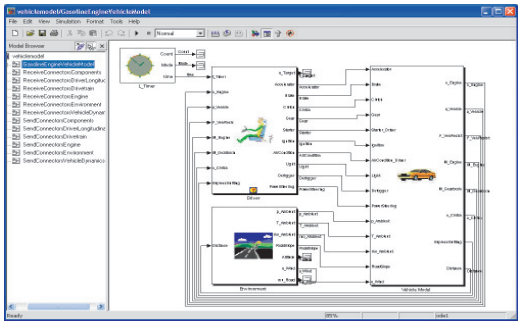


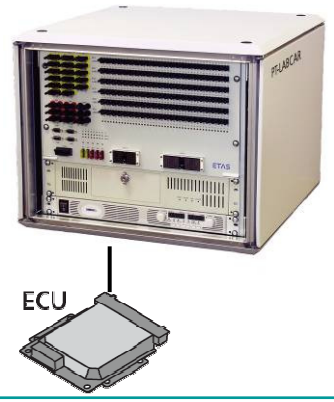


Firmenprofil
Überblick Entwicklungs- und Servicewerkzeuge (2/2)

Steuergeräte-test

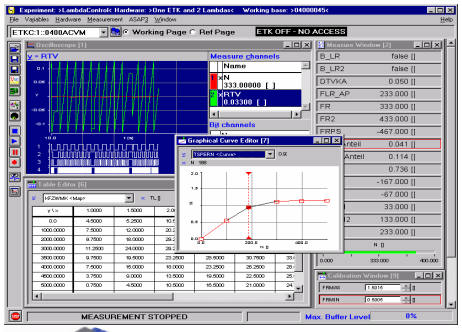
LABCAR und HiL-Prüfstand



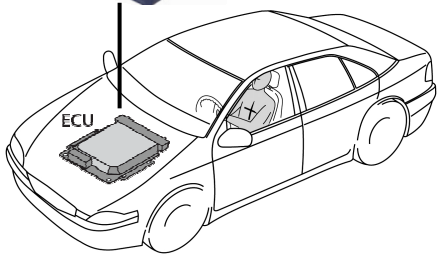


Messen, Applikation und Diagnose (MCD)

INCA und MCD-Module

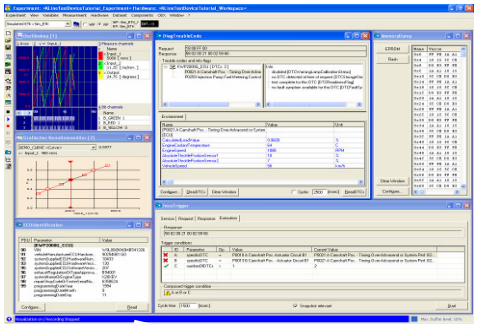


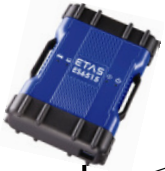
Messungen

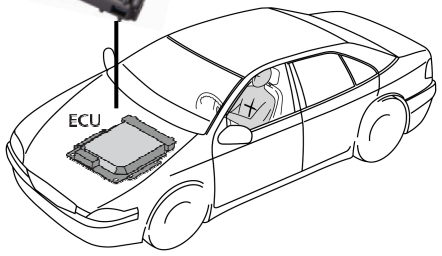


Fahrzeug-diagnose

OEM-Software und Diagnoseschnittstellen



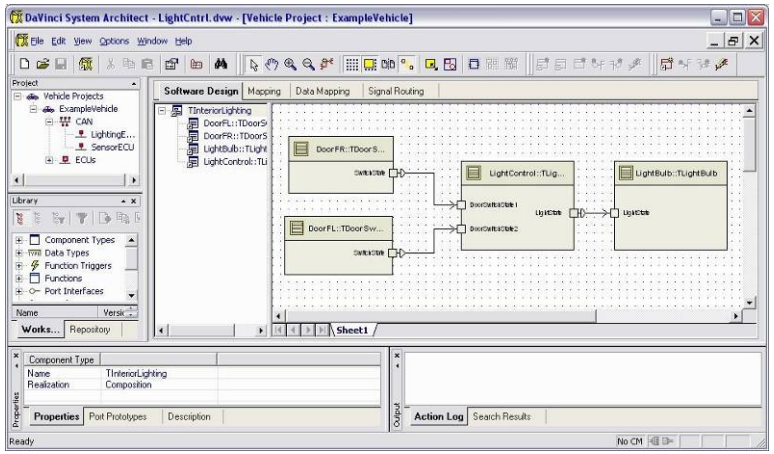




AUTOSAR@ETAS
Übersicht

- ETAS stellt Werkzeuge und Softwaremodule für die modellbasierte Entwicklung von AUTOSAR-Steuergeräten zur Verfügung:
  - Modellbasierter Entwurf und Generierung von AUTOSAR-konformen Komponenten der Anwendungssoftware für Seriensteuergeräte mit **ASCET**
  - **RTA-OS** Echtzeitbetriebssystem und **RTA-RTE** Laufzeitumgebung
  - Integration von AUTOSAR-Softwarekomponenten, RTA-OS und RTE mit **INTECRIO**
  - Validierung von AUTOSAR-konformen Steuerungen und Regelungen am **PC** und mit Prototyping-Hardware **ES900** im Fahrzeug mit INTECRIO
- ETAS-Werkzeuge sind in AUTOSAR-Serienprojekten im Einsatz.
- Durch die Mitarbeit in AUTOSAR-Arbeitsgruppen trägt ETAS mit Embedded-Software Know-how aktiv zum AUTOSAR-Standard bei.

ASCET – Modellbasierte Softwareentwicklung
Schritt 1: Export der AUTOSAR Softwarekomponentebeschreibung (SWC)



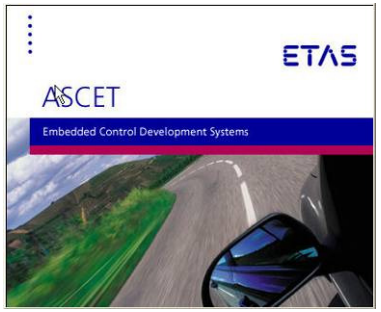
Authoring Tool, e.g.
- DaVinci (Vector)
- SystemDesk (dSPACE)
- AUTOSAR-Builder (Geensys)
- ...

Definierte Schnittstellen

XML

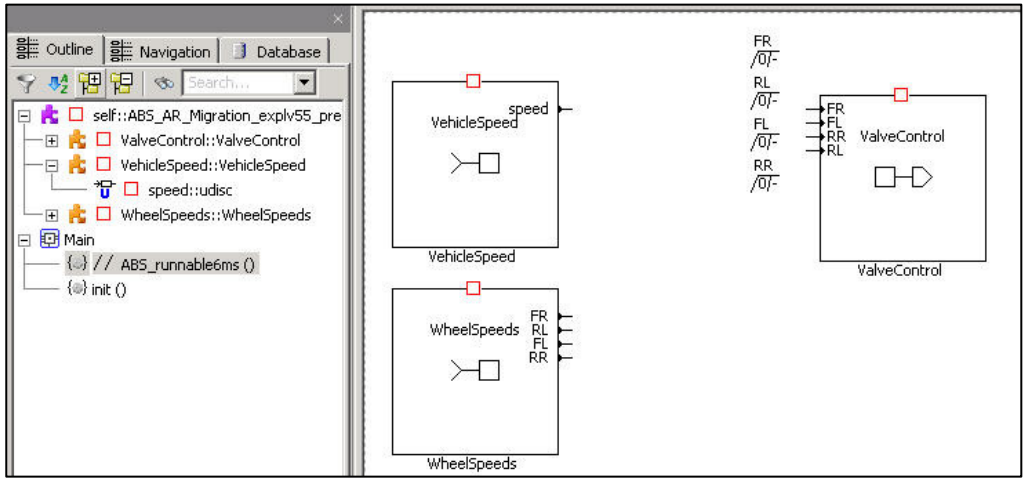
Auswahl einer OEM-Komponente

Typischer Use-Case*:
Hersteller (OEM) definiert Schnittstellen



*an alternative use-case is starting with a common Interface & Algorithm Development in ASCET

ASCET – Modellbasierte Softwareentwicklung
Schritt 2: Import der AUTOSAR Softwarekomponente (SWC)

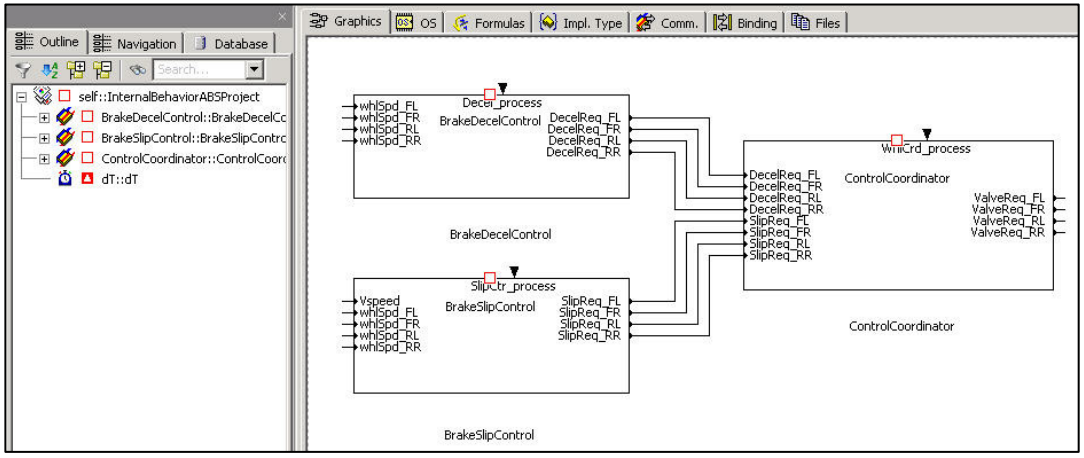


Importierte “Interfaces”:

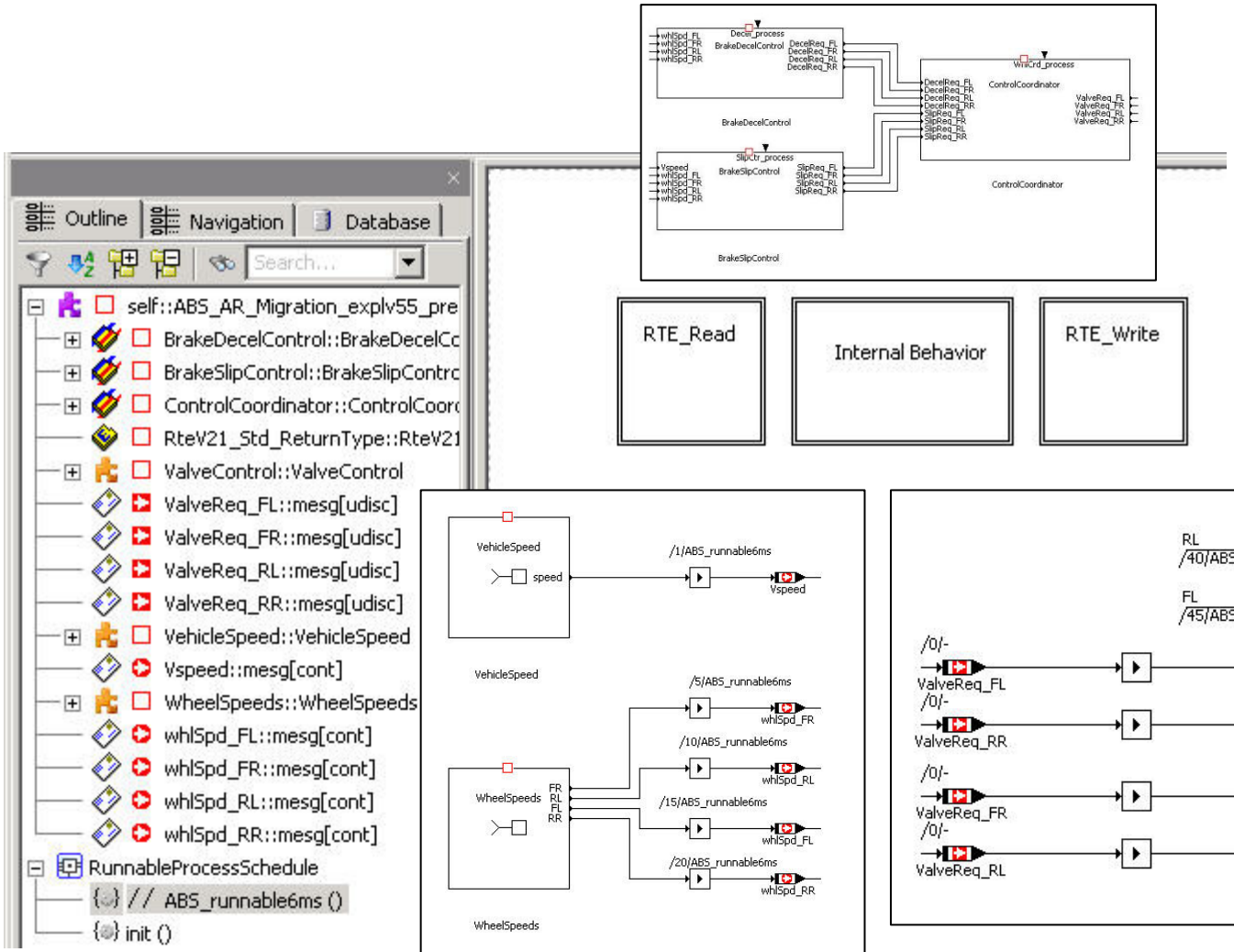
- Atomare SWC
- Port-Prototypes
- Runnables

Internal Behavior:

- Module im ASCET Project
- Message-Kommunication
- Oft durch Rapid Prototyping validiert

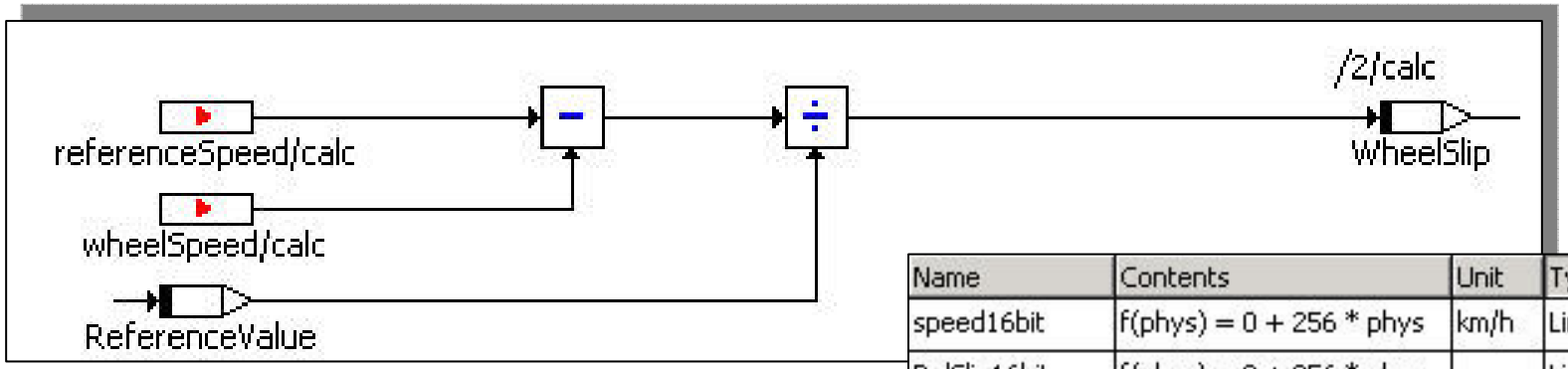


ASCET – Modellbasierte Softwareentwicklung
Schritt 3: Verbindung des Internal Behavior mit den “Interfaces”



Messages für interne Module automatisch erzeugt

ASCET – Modellbasierte Softwareentwicklung
Schritt 4: Implementierungen für Seriencode festlegen

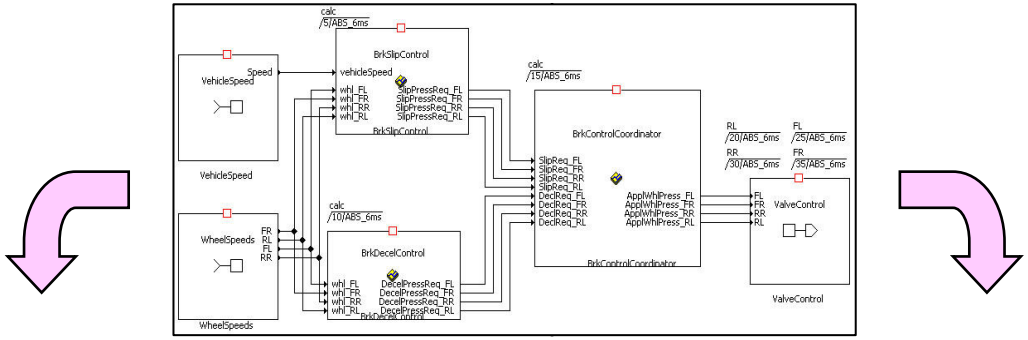


Name	Contents	Unit	Type
speed16bit	f(phys) = 0 + 256 * phys	km/h	Linear
RelSlip16bit	f(phys) = 0 + 256 * phys		Linear

Name	Type	Impl. Type	Impl. Min	Impl. Max	Q	Formula	Limit to ma bit length	Limit Assignmer	Zero not incl.	Min	Max
referenceSpeed/calc	cont	uint16	0	65535	0	speed16bit	Auto	Yes	No	0.0	255.99609
ReferenceValue	cont	int16	-32768	32767	0	RelSlip16bit	Auto	Yes	No	-128.0	127.99609
WheelSlip	cont	int16	-32768	32767	0	RelSlip16bit	Auto	Yes	No	-128.0	127.99609
wheelSpeed/calc	cont	uint16	0	65535	0	speed16bit	Auto	Yes	No	0.0	255.99609

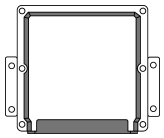
```
_WheelSlip=(sint16)((referenceSpeed-wheelSpeed<<8)/(sint32)_ReferenceValue);
```

ASCET – Modellbasierte Softwareentwicklung
Schritt 5: AUTOSAR Code Generation (.C-File and SWC-Description)




```
_WheelSlip=(sint16)((referenceSpeed-
wheelSpeed<<8)/(sint32)_ReferenceVal
ue);
```

```
Rte_Read_VehicleSpeed_speed(Vspeed_
ABS_runnable6ms)
```

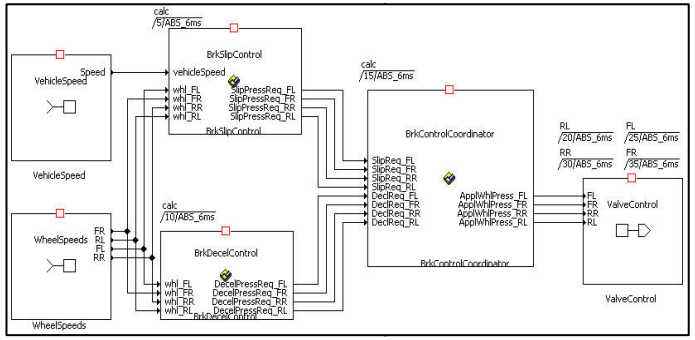


```
...
<ATOMIC-SOFTWARE-COMPONENT-TYPE>
  <SHORT-NAME>ABS_RR</SHORT-NAME>
  <PORTS>
    <R-PORT-PROTOTYPE>
      <SHORT-NAME>WheelSpeeds</SHORT-NAME>
      <REQUIRED-INTERFACE-TREF DEST="SENDER-RECEIVER-INTERFACE">/ASCET_interfaces/AUTOSAR8Bit/WheelSpeeds</REQUIRED-INTERFACE-TR
    </R-PORT-PROTOTYPE>
    ...
    <P-PORT-PROTOTYPE>
    <SHORT-NAME>ValveControl</SHORT-NAME>
    <PROVIDED-INTERFACE-TREF DEST="SENDER-RECEIVER-INTERFACE">/ASCET_interfaces/Valves8Bit/ValveControl</PROVIDED-INTERFACE-TREF>
    </P-PORT-PROTOTYPE>
  </PORTS>
</ATOMIC-SOFTWARE-COMPONENT-TYPE>
<INTERNAL-BEHAVIOR>
  <SHORT-NAME>ABS_RR</SHORT-NAME>
  <COMPONENT-REF DEST="ATOMIC-SOFTWARE-COMPONENT-TYPE">/ASCET_swcomponents/ABS_8bit/ABS_RR</COMPONENT-REF>
  <EVENTS>
   ...
  </EVENTS>
  <RUNNABLES>
    <RUNNABLE-ENTITY>
      <SHORT-NAME>ABS_6ms</SHORT-NAME>
      <SYMBOL>ABS_RR_ABS_8BIT_ABS_6ms</SYMBOL>
    </RUNNABLE-ENTITY>
    <RUNNABLE-ENTITY>
      <SHORT-NAME>cyclic_diag</SHORT-NAME>
      <SYMBOL>ABS_RR_ABS_8BIT_cyclic_diag</SYMBOL>
    </RUNNABLE-ENTITY>
    <RUNNABLE-ENTITY>
      <SHORT-NAME>init_runnable</SHORT-NAME>
      <SYMBOL>ABS_RR_ABS_8BIT_init_runnable</SYMBOL>
    </RUNNABLE-ENTITY>
  </RUNNABLES>
</INTERNAL-BEHAVIOR>
...
```

Electronic
Control Unit

AUTOSAR

ASCET – Modellbasierte Softwareentwicklung
Schritt 6: RTE-Contract Phase (.h File Generation)




```
_WheelSlip=(sint16)((referenceSpeed-
wheelSpeed<<8)/(sint32)_ReferenceVal
ue);
```

```
Rte_Read_VehicleSpeed_speed(Vspeed_
ABS_runnable6ms)
```

<SW-C>.xml

RTA-RTE
rtegen –c swcl

```
#define Rte_Read_VehicleData_...
```

Electronic
Control Unit

AUTOSAR

ETAS

ASCET – Modellbasierte Softwareentwicklung
Schritt 7: ECU-Integration, BSW-Configuration & Generation

ECU-Conf. Tool e.g
iSOLAR
tresos
DaVinci

<ECU>.autosar

<SW-C>.xml

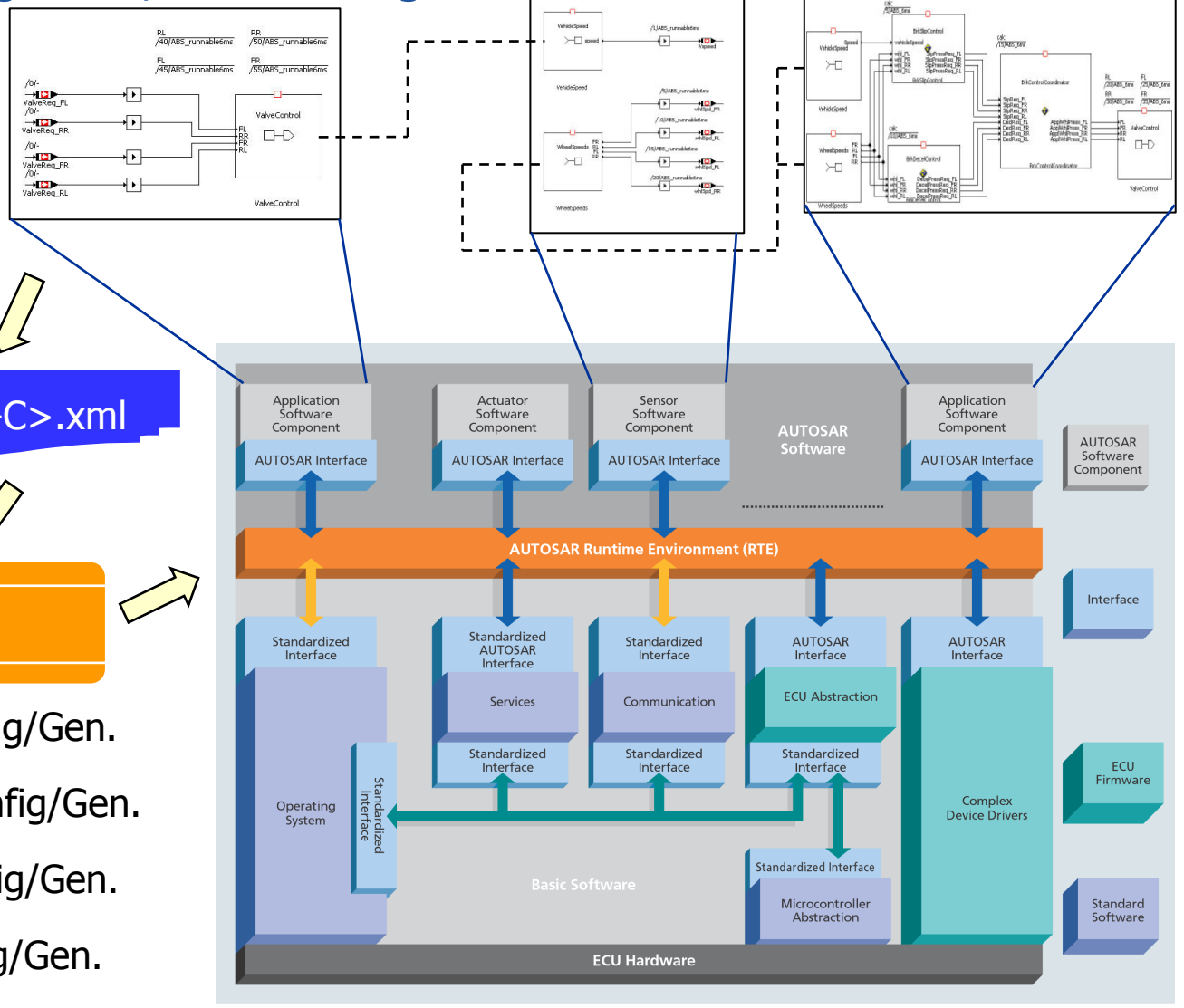
RTA-RTE
rtegen –r ecu

DEM-Config/Gen.

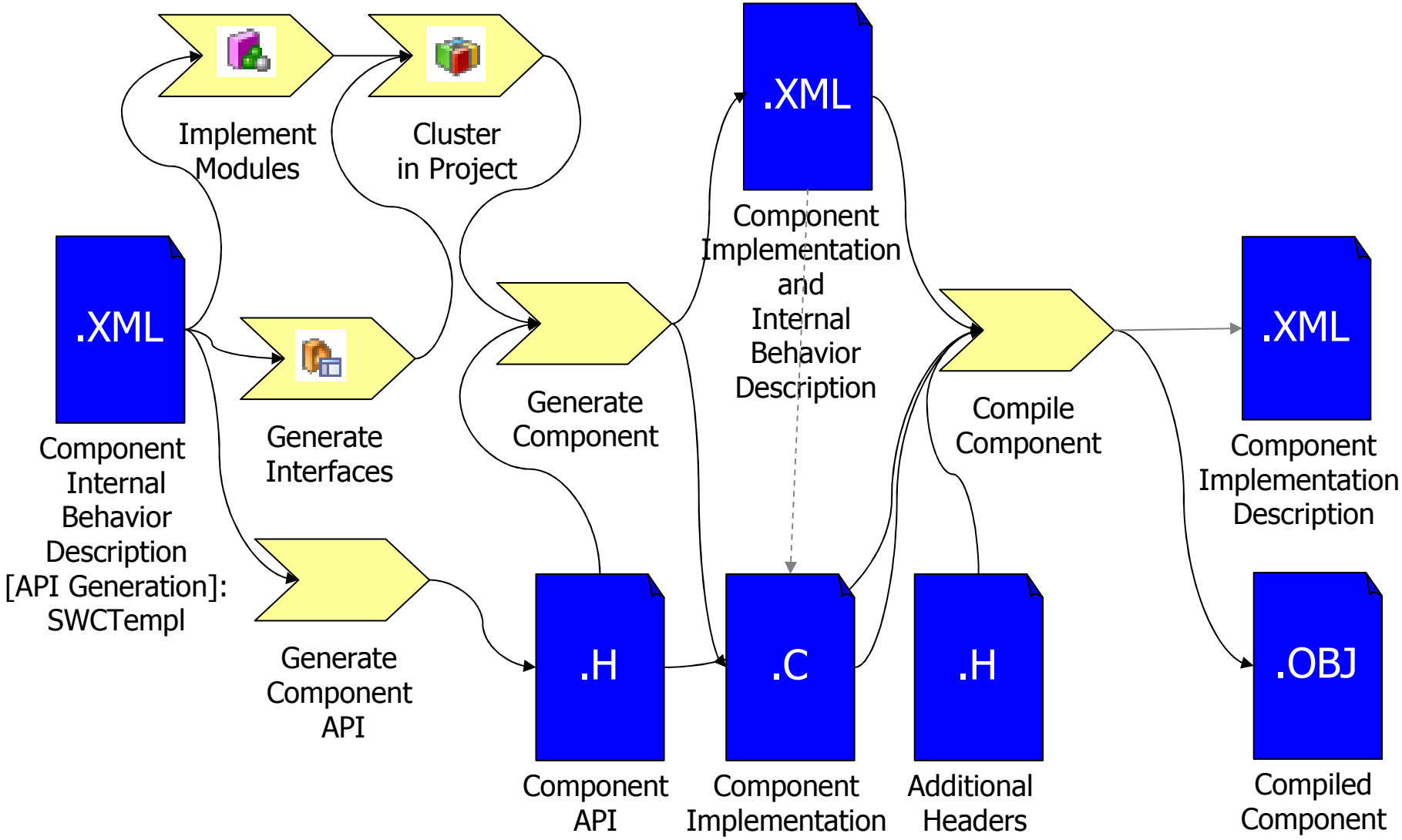
NVRAM-Config/Gen.

COM-Config/Gen.

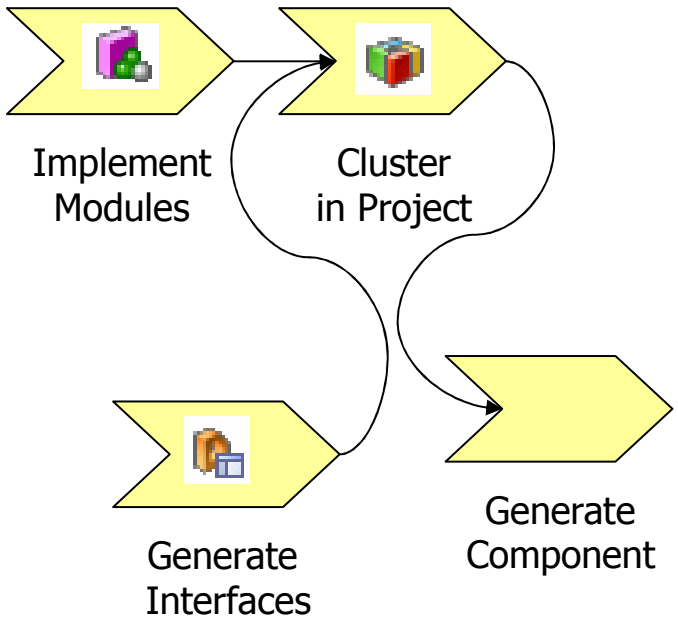
OS-Config/Gen.

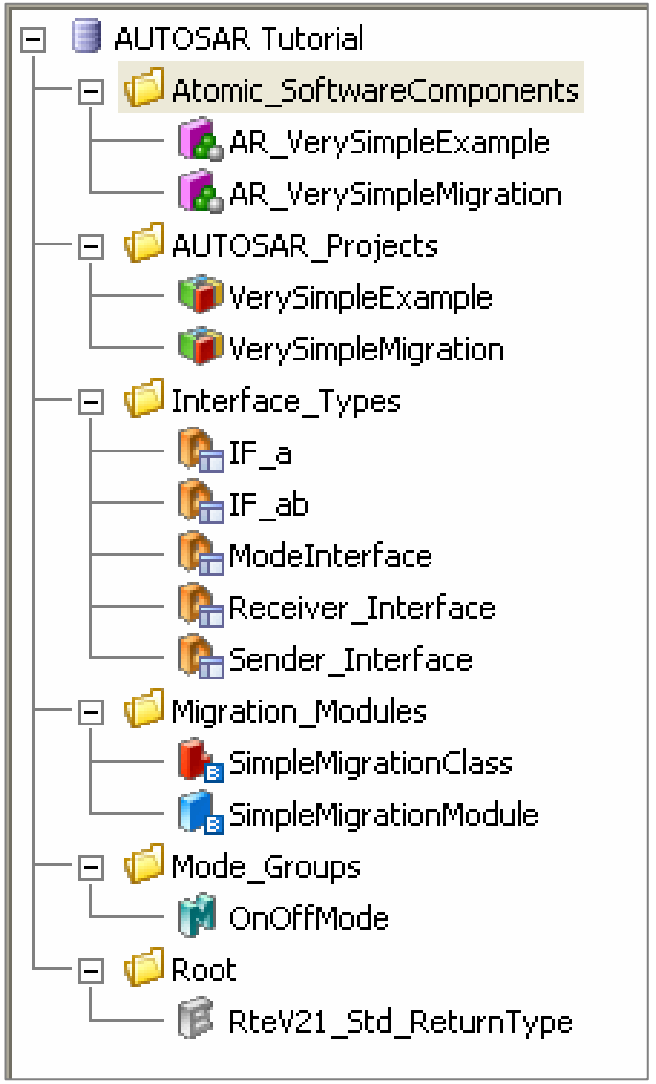


ASCET – AUTOSAR Methodologie
Komponentenentwicklung



ASCET – AUTOSAR Tutorial
Vorgefertigte Elemente





Einführung von AUTOSAR
- "from Scratch"
- Migration

AUTOSAR-Software mit ASCET
Die Zukunft der industriellen Softwareentwicklung hat bereits begonnen.