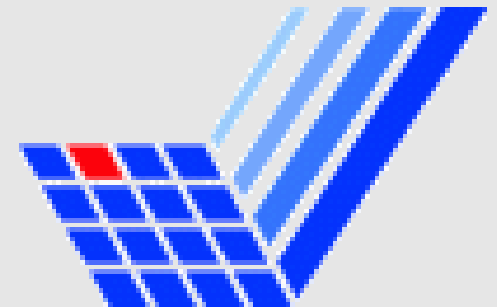


AUTOSAR

Robert Neue

robert.neue@gmx.de

PG AutoLab
Seminarwochenende 21.-23. Oktober 2007



Überblick

- 1. Motivation / Ziele
- 2. Schichtenmodell
- 3. Konzepte
- 4. AUTOSAR-OS vs. OSEK-OS
- 5. Methodik
- 6. Fazit

Gründungsmitglieder

BMW Group

BOSCH 

Continental 

DAIMLERCHRYSLER

SIEMENS VDO
A u t o m o t i v e

VOLKSWAGEN AG



PSA PEUGEOT CITROËN 

TOYOTA



Motivation

“Our vision is an improved complexity management of highly integrated E/E architectures through an increased reuse and exchangeability of SW modules between OEMs and suppliers.”

„Cooperate on Standards

–Compete on Implementation“

Ziele

- Implementierung und Standardisierung von Basisfunktionen
- Skalierbarkeit hinsichtlich verschiedener Fahrzeugtypen
- Möglichkeiten zur redundanten Auslegung
- Einbettung von Modulen anderer Hersteller
- Wartbarkeit während des gesamten Produktlebenszyklus
- Software Updates und Upgrades während des gesamten Fahrzeuglebens

Lösungen von AUTOSAR

- Standardisierung des Austauschformats
- Basic Software
- MicroController Abstraktion
- RunTimeEnvironment
- Standard von Schnittstellen, um Probleme beim Einbinden von Produkten verschiedener Hersteller zu vermeiden

Konzepte

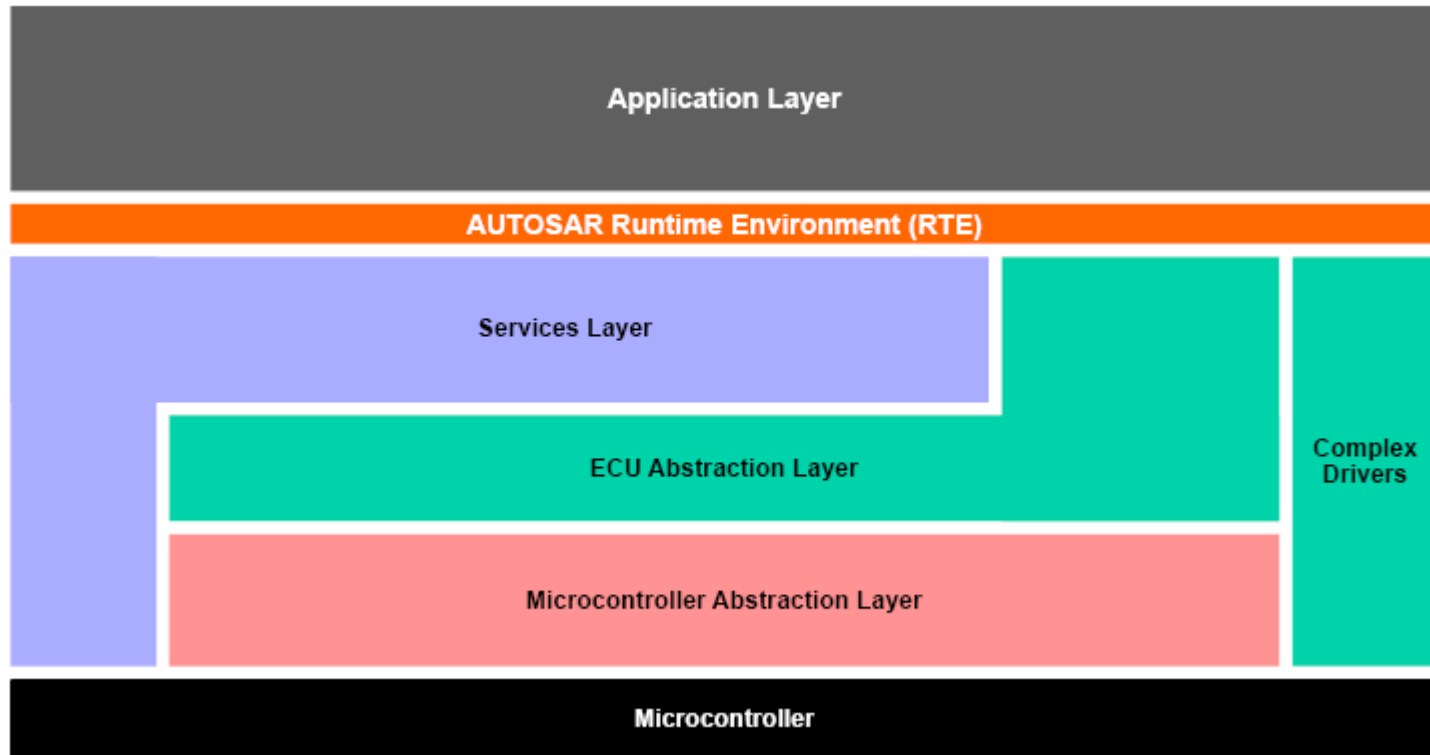
- AUTOSAR SW-C
- SW-C Description
- Virtual Functional Bus (VFB)
- System Constraint und ECU Descriptions
- Mapping on ECUs
- Run Time Environment (RTE)
- Basic Software

Überblick

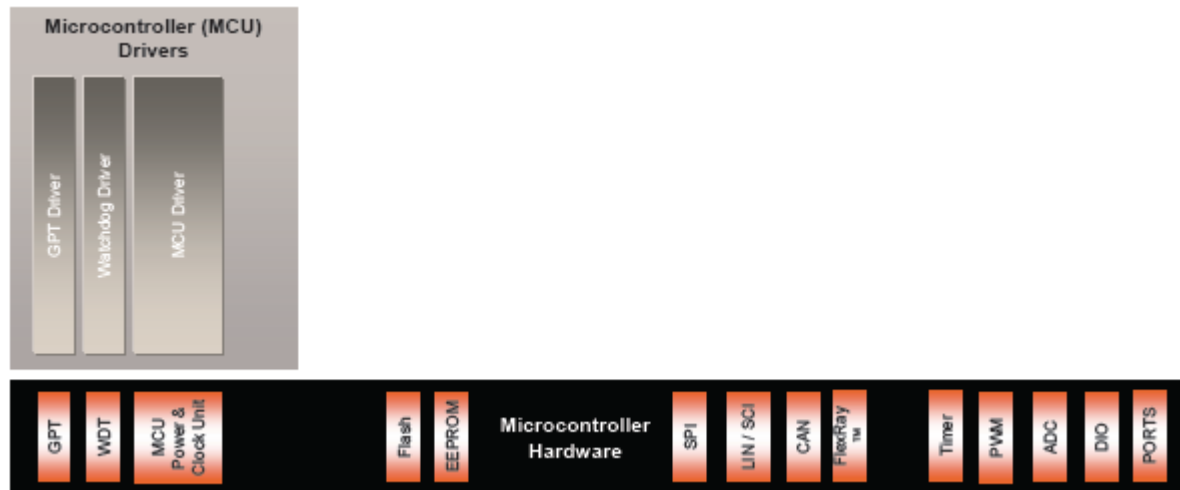
- 1. Motivation / Ziele
- 2. Schichtenmodell
- 3. Konzepte
- 4. AUTOSAR-OS vs. OSEK-OS
- 5. Methodik
- 6. Fazit

Konzepte im Detail 1

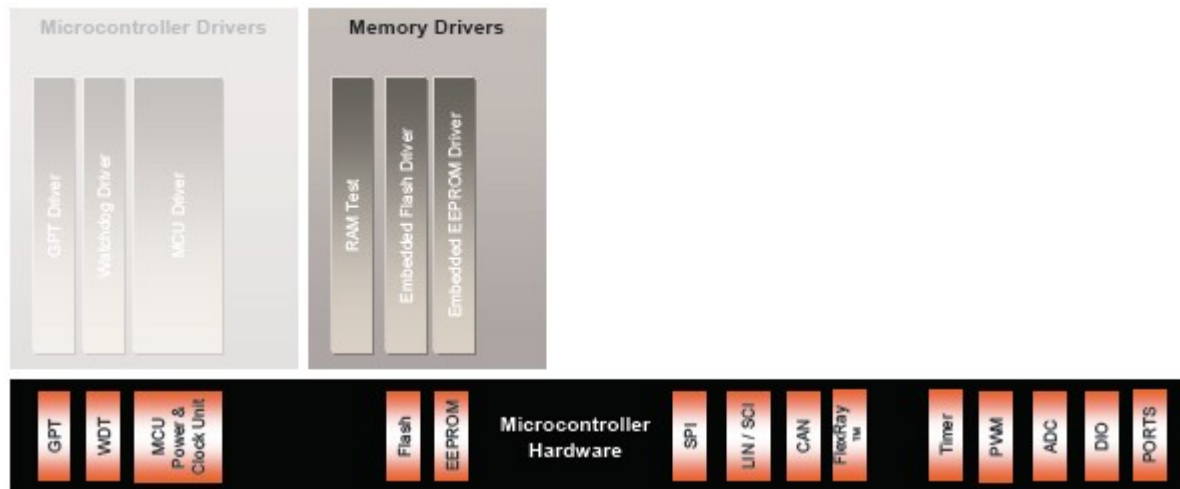
- Run Time Environment
- Basis Software



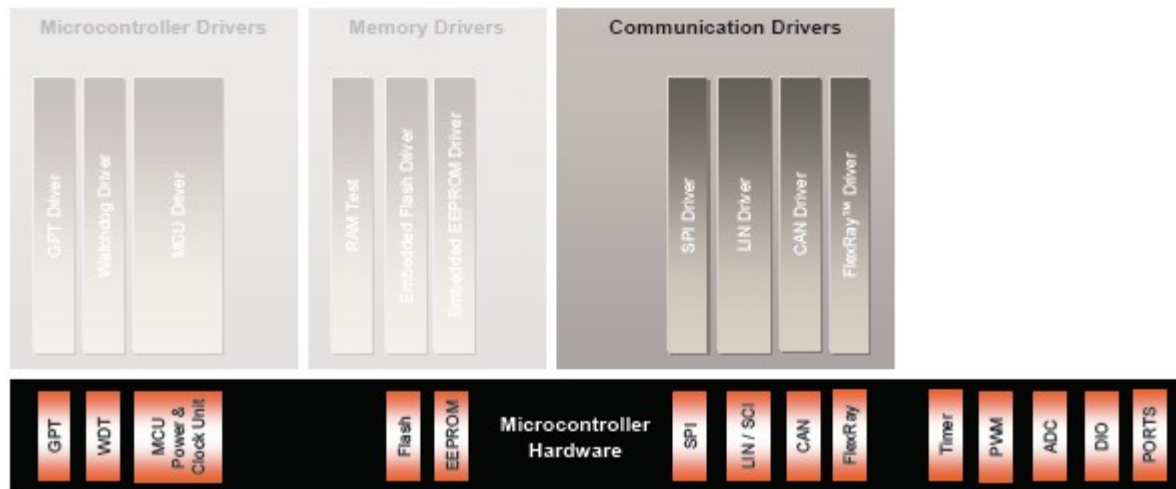
MCAL MicroControllerAbstractionLayer



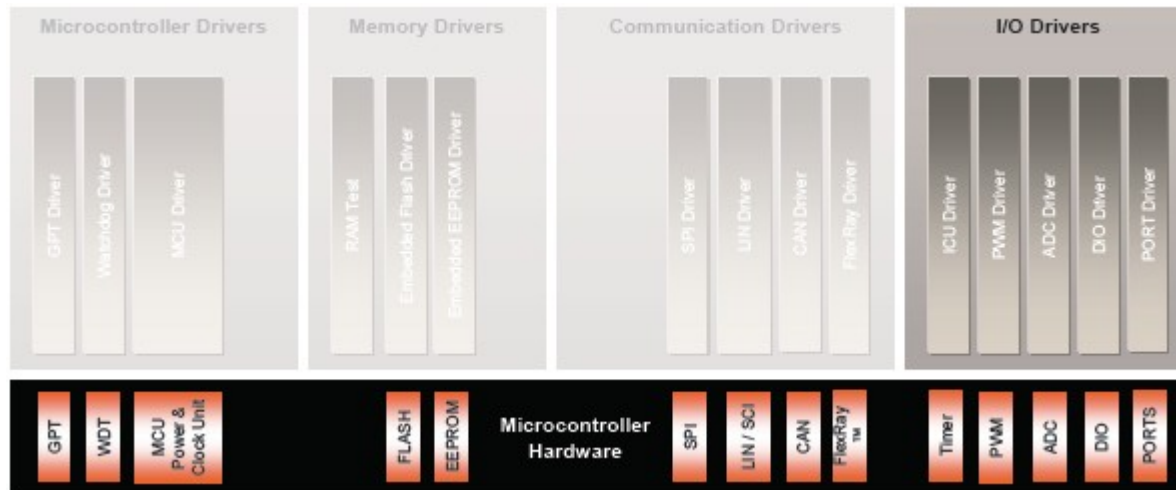
MCAL MicroControllerAbstractionLayer



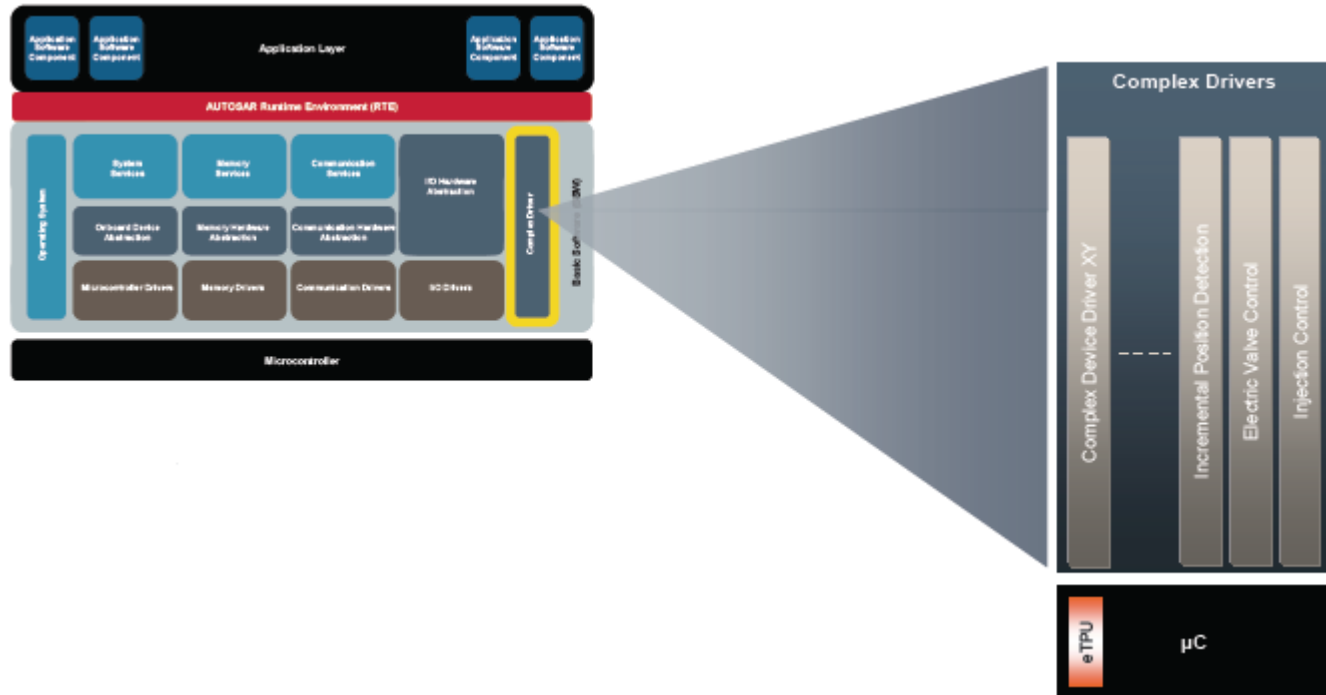
MCAL MicroControllerAbstractionLayer



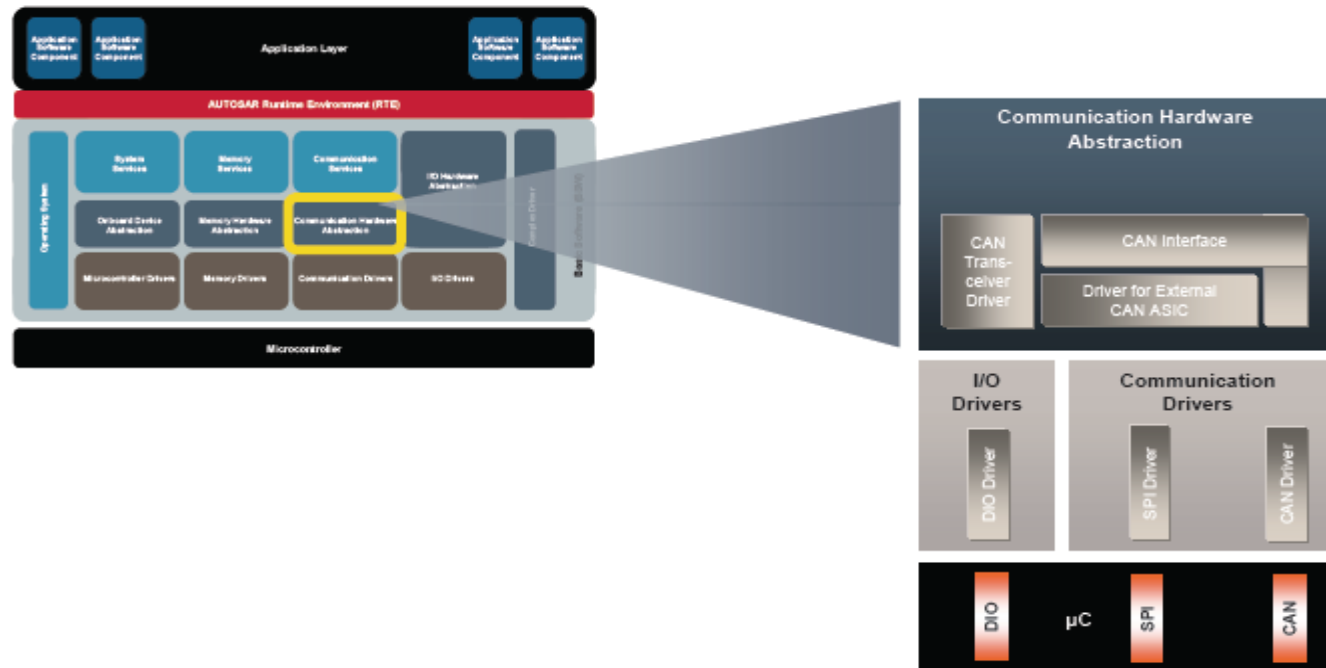
MCAL MicroControllerAbstractionLayer



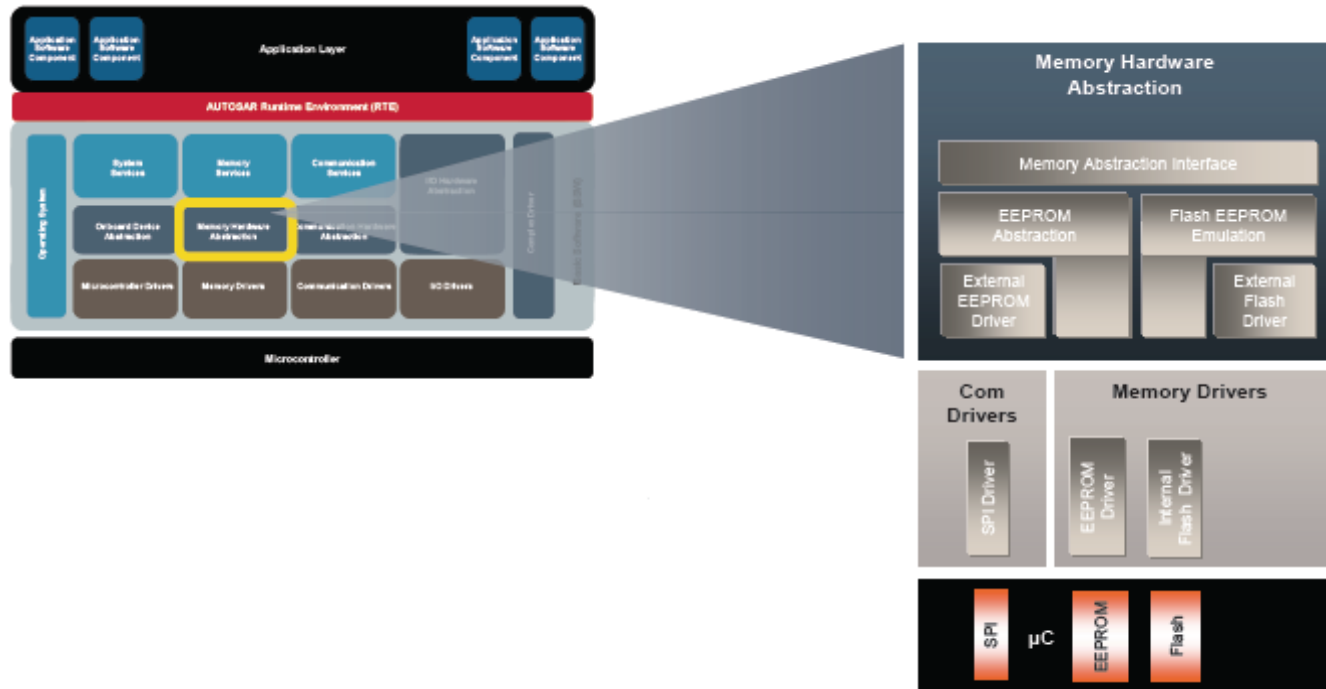
Komplexe Treiber



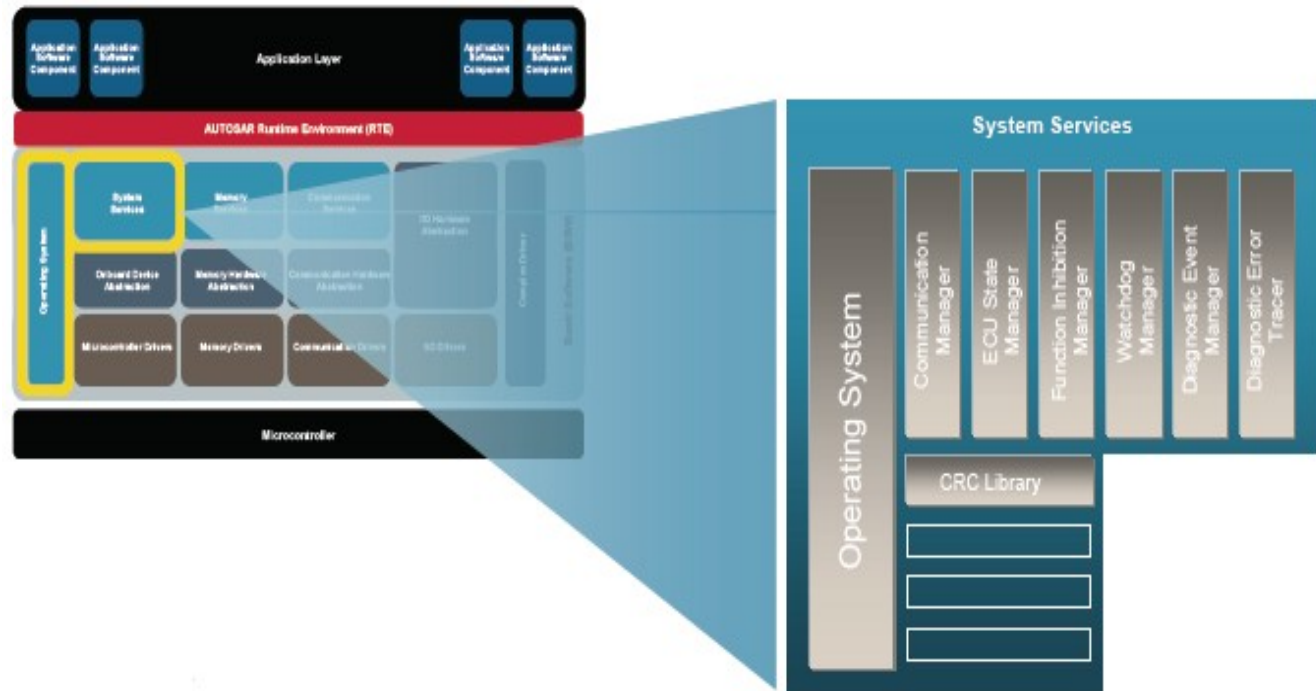
ECU Abstraction Layer



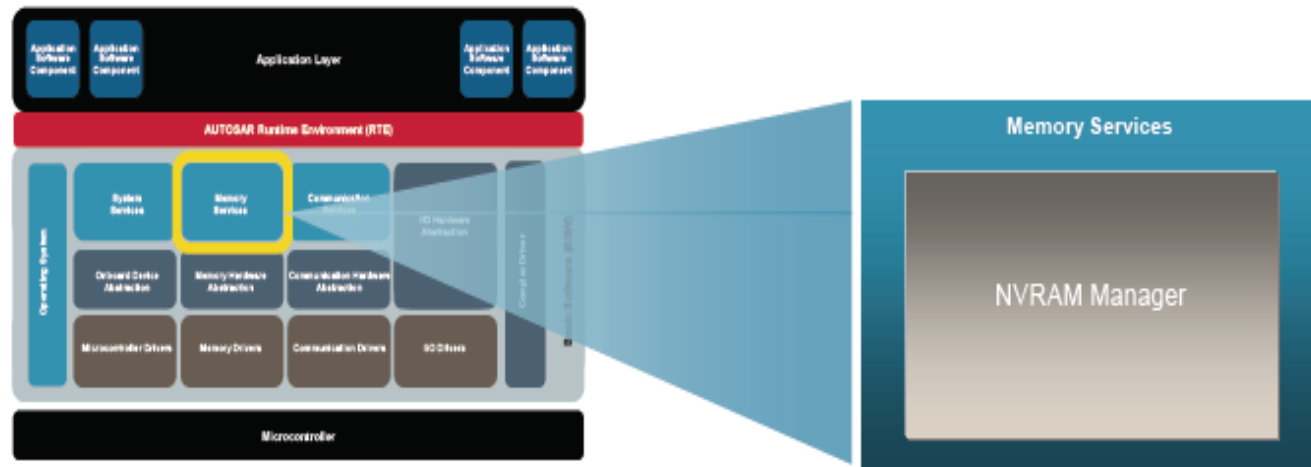
ECU Abstraction Layer



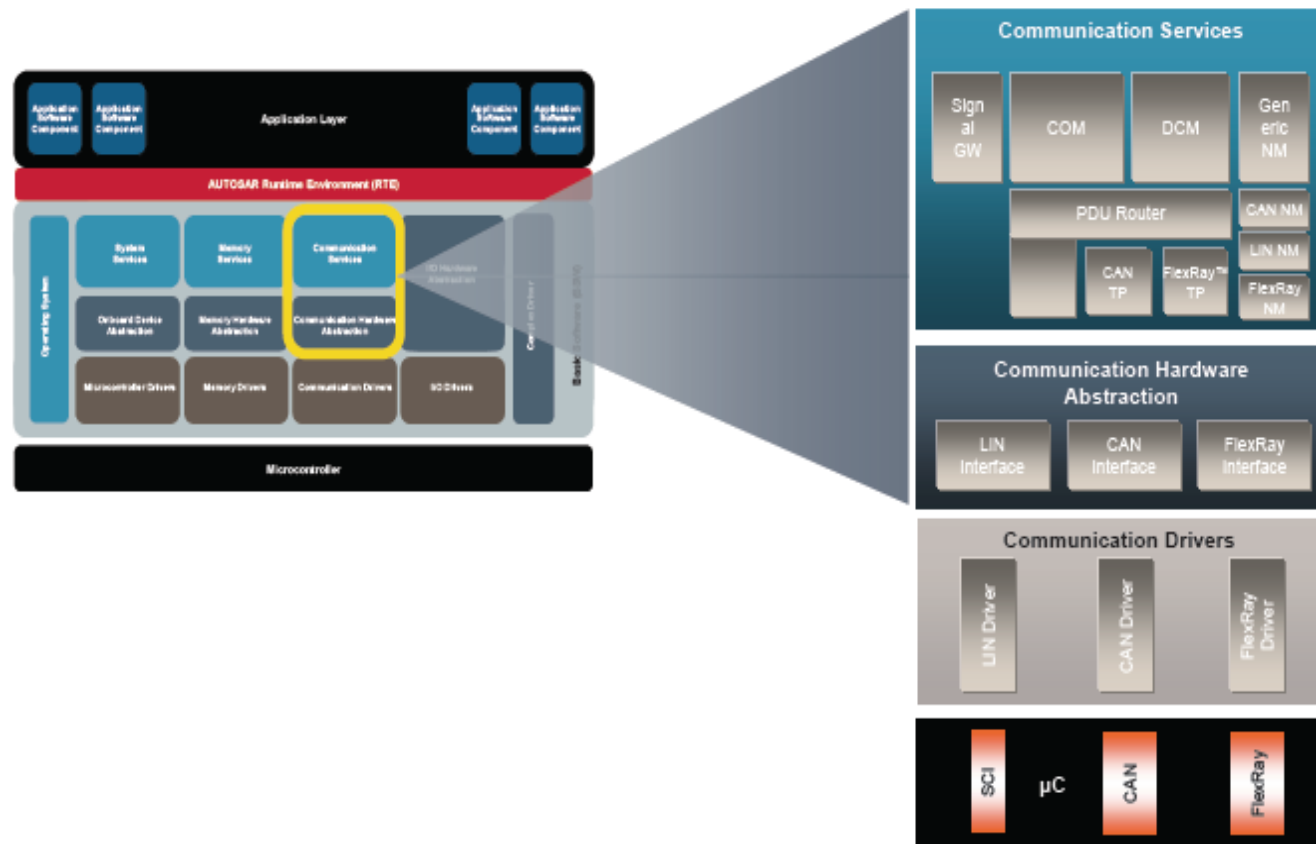
Services Layer



Services Layer



Services Layer



Überblick

- 1. Motivation / Ziele
- 2. Schichtenmodell
- 3. Konzepte
- 4. AUTOSAR-OS vs. OSEK-OS
- 5. Methodik
- 6. Fazit

Konzepte im Detail 1/4

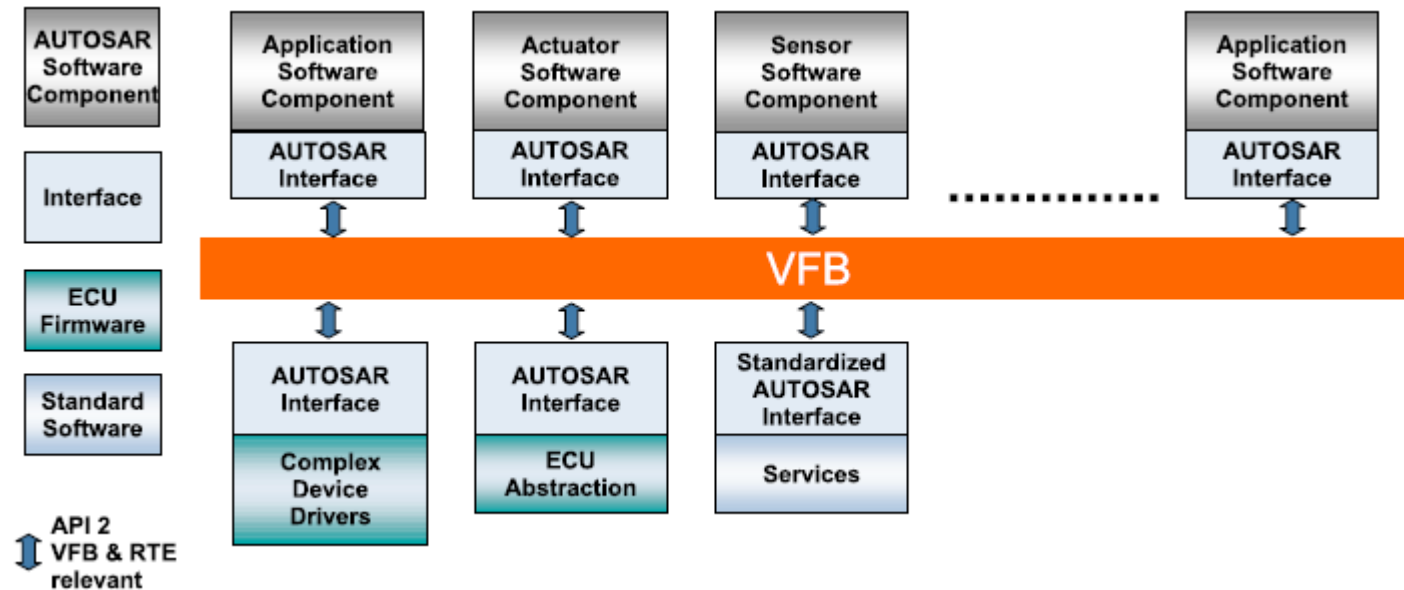
- AUTOSAR Software Component (AUTOSAR SW-C) + SW-C Description
 - Teilfunktionalität
 - Atomic Software Components
 - Composition
 - Sensor / Aktuator SW-Cs

Konzepte im Detail 2/4

- Run Time Environment
 - Ist nicht der abstrakte VFB
 - Implementierung des VFB auf MC
 - RTE muss nicht komplette Funktionalität direkt implementieren
 - Darf soviel wie möglich delegieren
- Basis Software
 - Stellt Funktionalität der ECU unabhängig von der Hardware zur Verfügung
 - Unabhängige Module versch. Hersteller möglich
 - Verschieden Abstraktionsebenen je Einsatzgebiet

Konzepte im Detail 3/4

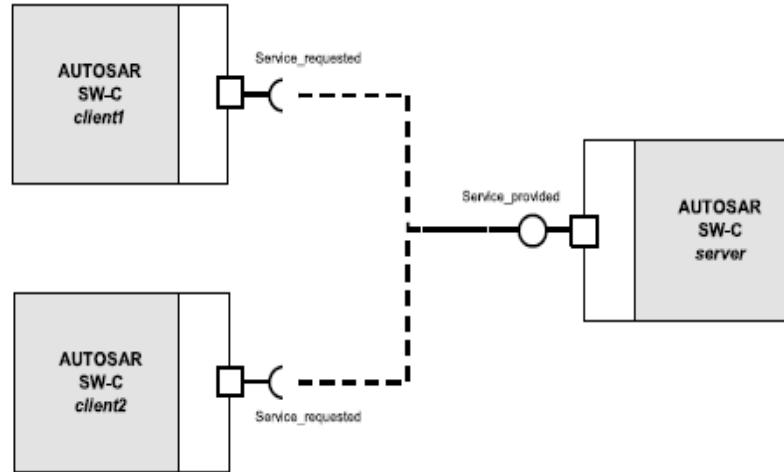
- Virtual Functional Bus
 - Kommunikation auf unabh. abstr. Niveau
 - Komponenten kommunizieren nur über def. Ports



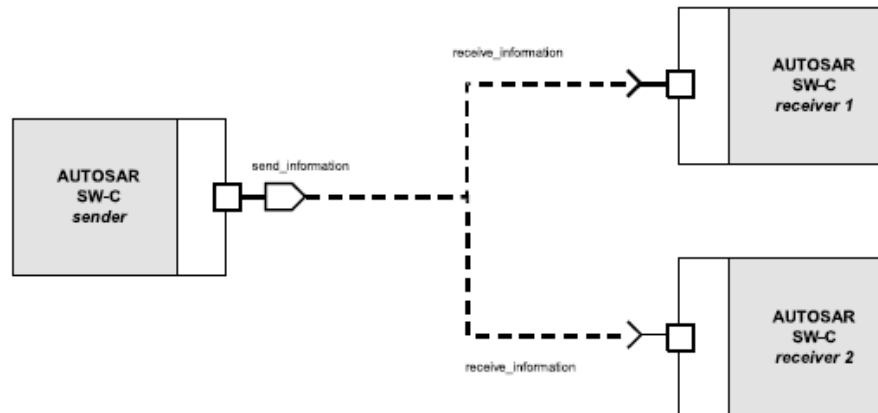
Konzepte im Detail 4/4

- Virtual Functional Bus (Kommunikationsmuster)

- Client-Server



- Sender-Receiver



Überblick

- 1. Motivation / Ziele
- 2. Schichtenmodell
- 3. Konzepte
- 4. AUTOSAR-OS vs. OSEK-OS
- 5. Methodik
- 6. Fazit

AUTOSAR-OS vs. OSEK-OS

- OSEK-OS hat keine Zeitüberwachung
- OSEK-OS hat keine Speicherschutz
- AUTOSAR skaliert dies in den Klassen 1-4

Category	Scalability Class 1	Scalability Class 2	Scalability Class 3	Scalability Class 4
OSEK OS (all conformance classes)	✓	✓	✓	✓
Counter Interface	✓	✓	✓	✓
Schedule Tables	✓	✓	✓	✓
Stack Monitoring	✓	✓	✓	✓
Protection Hook		✓	✓	✓
Timing Protection		✓		✓
Global Time/Synchronization Support		✓		✓
Memory Protection			✓	✓
OS Applications			✓	✓
Service Protection			✓	✓
Call Trusted Function			✓	✓

AUTOSAR-OS vs. OSEK-OS

- Warum spielt das jetzt eine Rolle?
 - Mehrere Anwendungen pro ECU simultan
 - Schon jetzt > 50 ECUs in Luxusautos
- Speicherschutz daher wichtiger denn je

- Execution time enforcement (Budget)
- Arrival rate enforcement

- Trusted applications
- Non-trusted Applications (Blind trust)

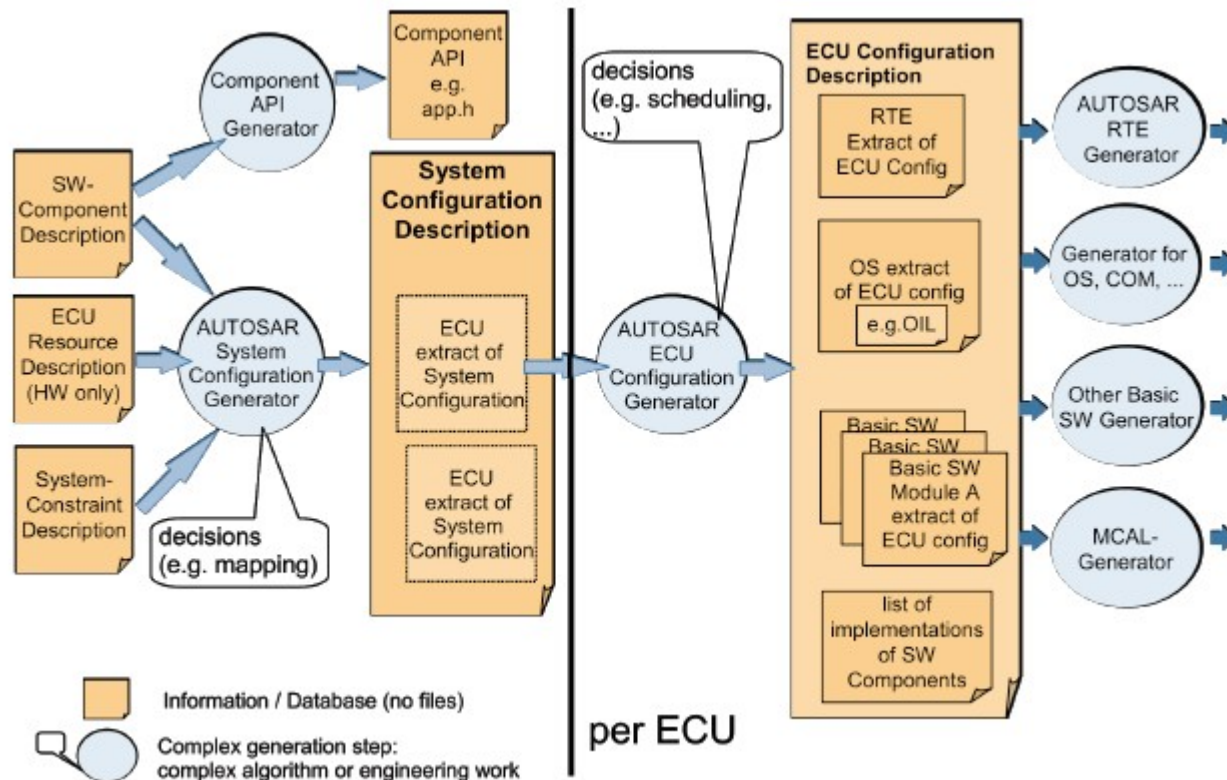
- => Memory protection Units (MPUs)

Überblick

- 1. Motivation / Ziele
- 2. Schichtenmodell
- 3. Konzepte
- 4. AUTOSAR-OS vs. OSEK-OS
- 5. Methodik
- 6. Fazit

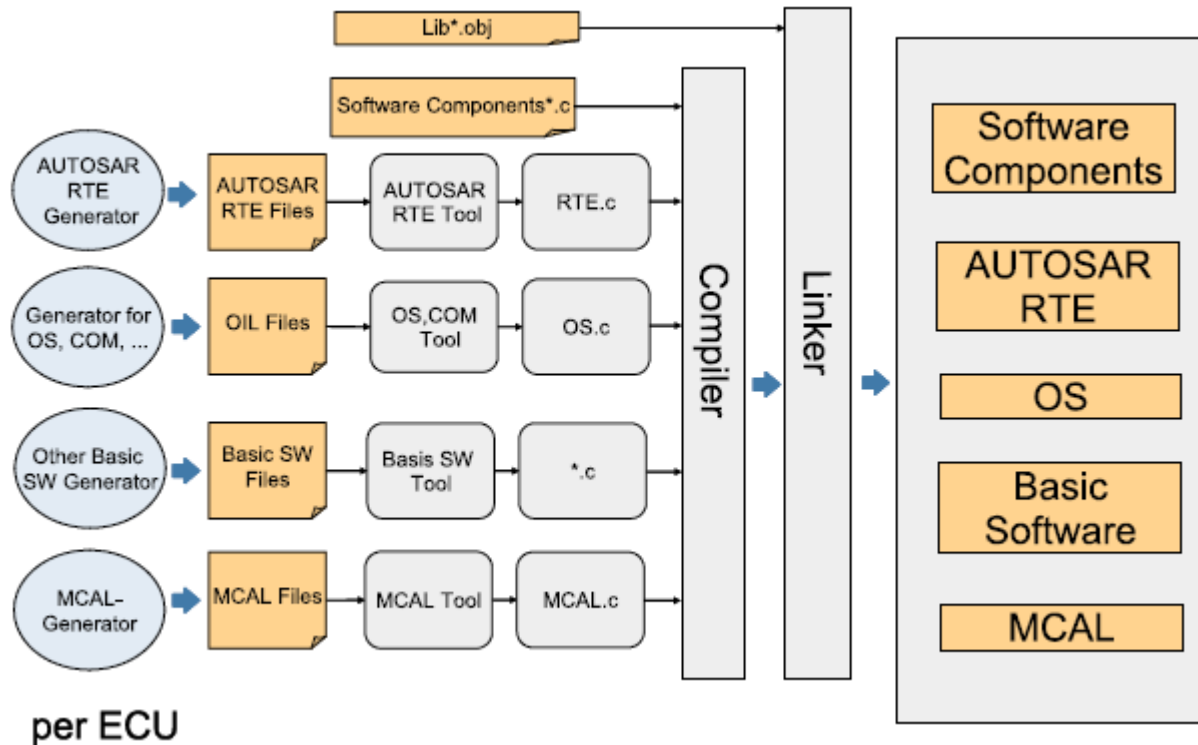
AUTOSAR Methode

- Von der Beschreibung ...



AUTOSAR Methode

- ... zur ausführbaren Executable



Überblick

- 1. Motivation / Ziele
- 2. Schichtenmodell
- 3. Konzepte
- 4. AUTOSAR-OS vs. OSEK-OS
- 5. Methodik
- 6. Fazit

FAZIT

- **AUTOSAR wird von namhaften Lieferanten und Herstellern vorangetrieben => große Unterstützung**
- Einige versuchen ihre Lösungen in den Standard zu drücken
- Austauschbare Software
 - Technische Seite
 - Was passiert auf kaufmännischer Ebene?
- Tool Support noch nicht perfekt, es fehlen Automatismen

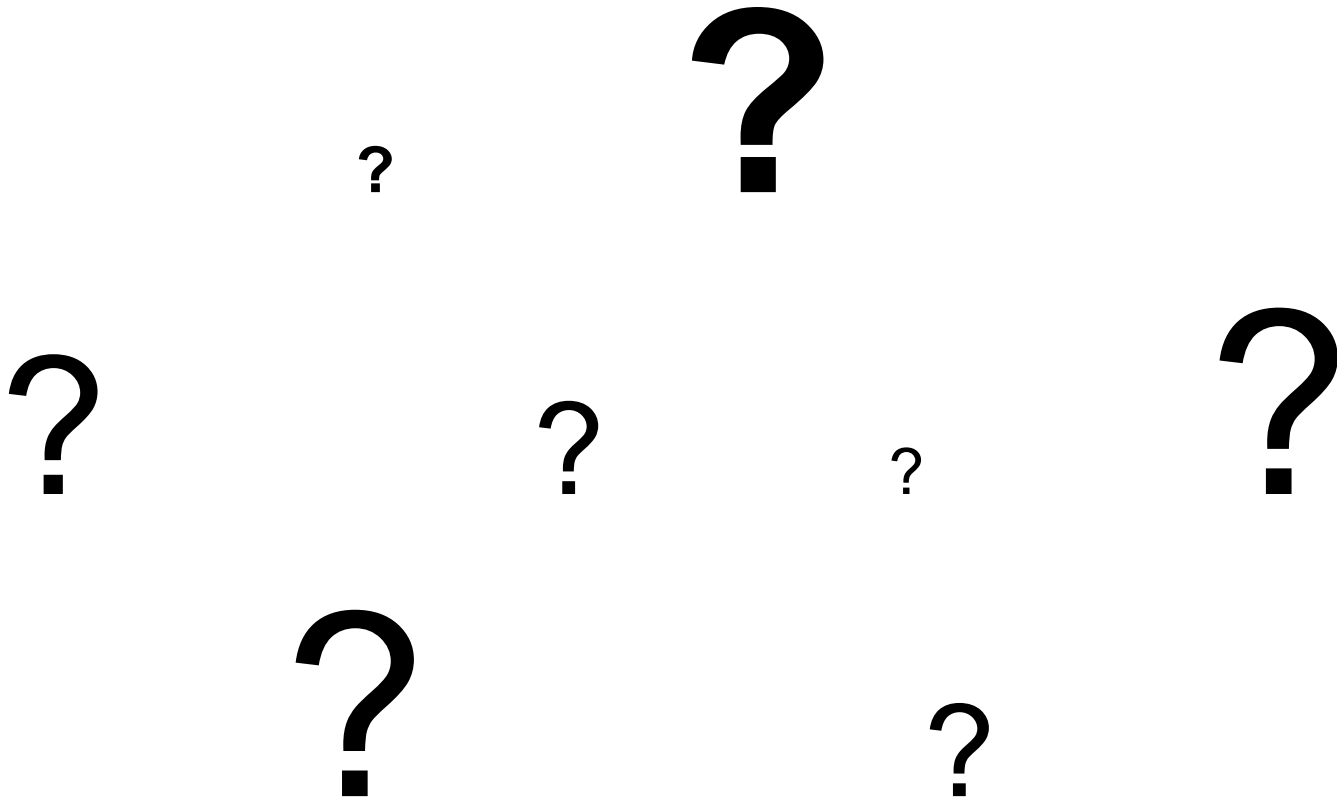
Quellen

www.autosar.org

www.freescale.com

Keywords: How AUTOSAR Will Change the Way You Work with Freescale Products

Fragen?



Ende

**Vielen Dank für eure
Aufmerksamkeit!!!**