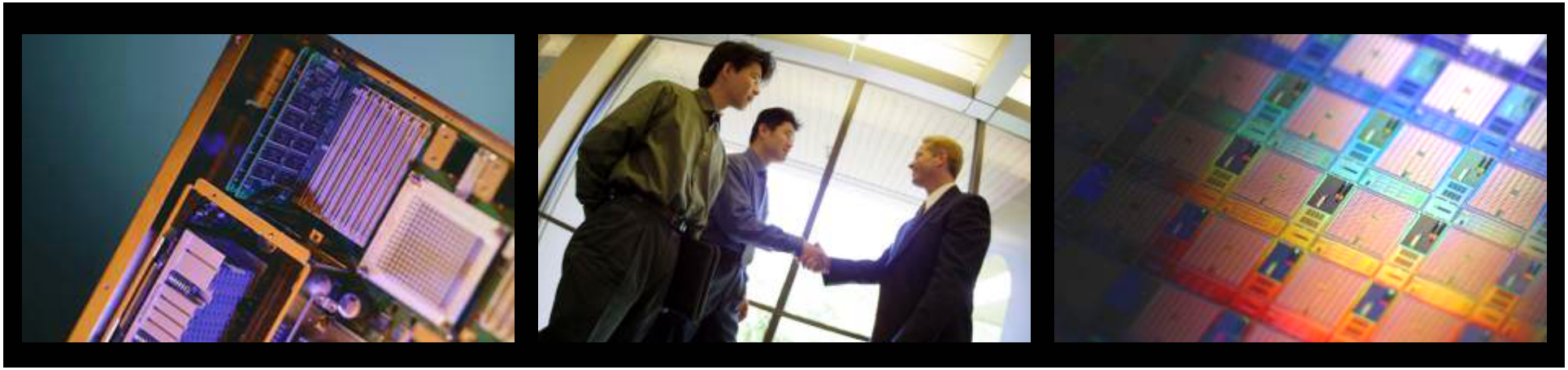
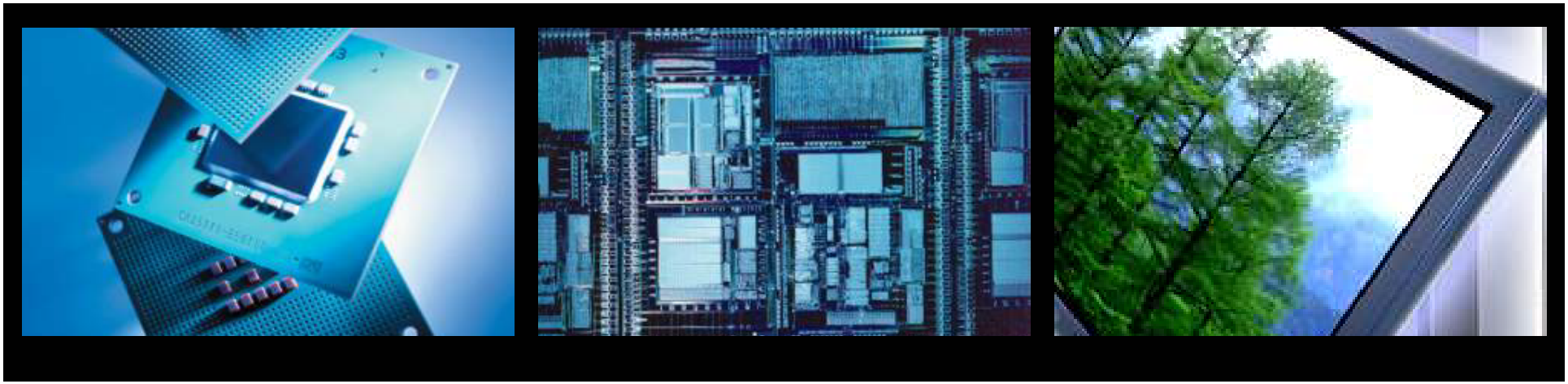


FUJITSU

AUTOSAR - Ein Überblick



AUTOSAR - Ein Überblick

- **Agenda**
  - Fujitsu Microelectronics Europe GmbH
  - Was ist AUTOSAR?
  - Die AUTOSAR Software Architektur
  - Die AUTOSAR Methodik
  - Zusammenfassung und Erfahrungen aus der Praxis

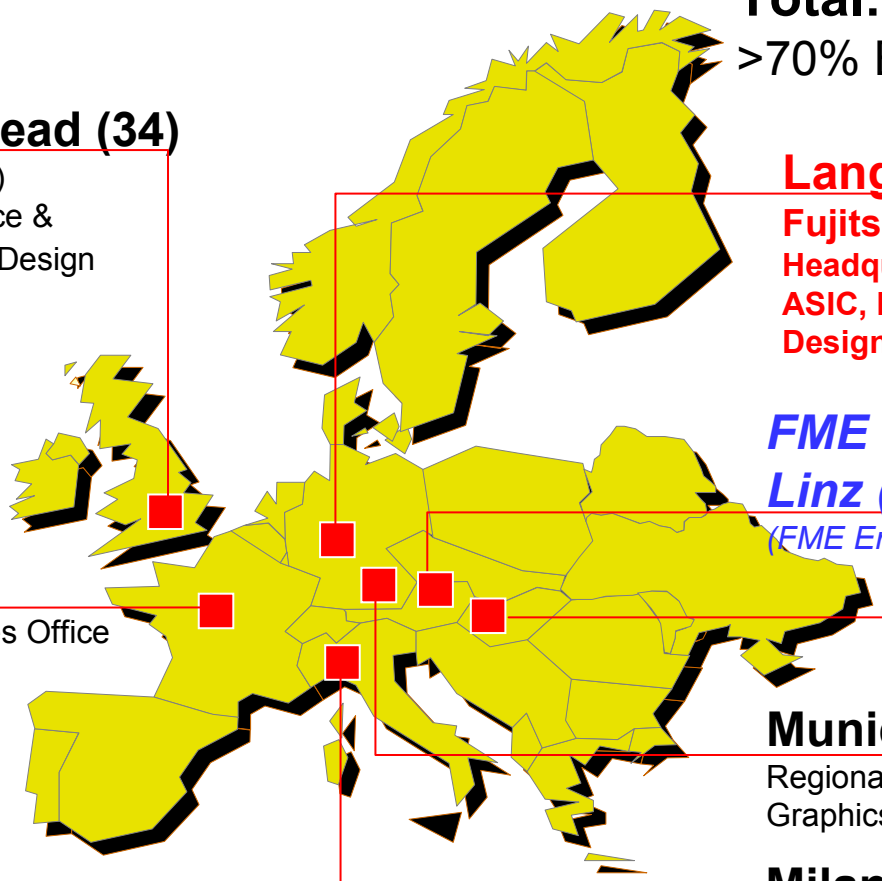
Fujitsu Microelectronics Europe

Total: 304*+45* Employees
>70% Engineers (as of January 2009)

Maidenhead (34)
(near London)
Regional Office &
Mixed Signal Design

Langen (near Frankfurt) (215)
Fujitsu Microelectronics Europe
Headquarters and Warehouse
ASIC, MCU, RF, Multimedia
Design Centres

FME Software Subsidiary, Linz (45)
(FME Embedded Solutions Austria GmbH)



Paris (3)
Regional Sales Office

Budapest (1)

Munich (41)
Regional Sales Office & ASIC Design,
Graphics Competence Centre

Milan (10)
Regional Sales Office

Representatives in:
South Africa

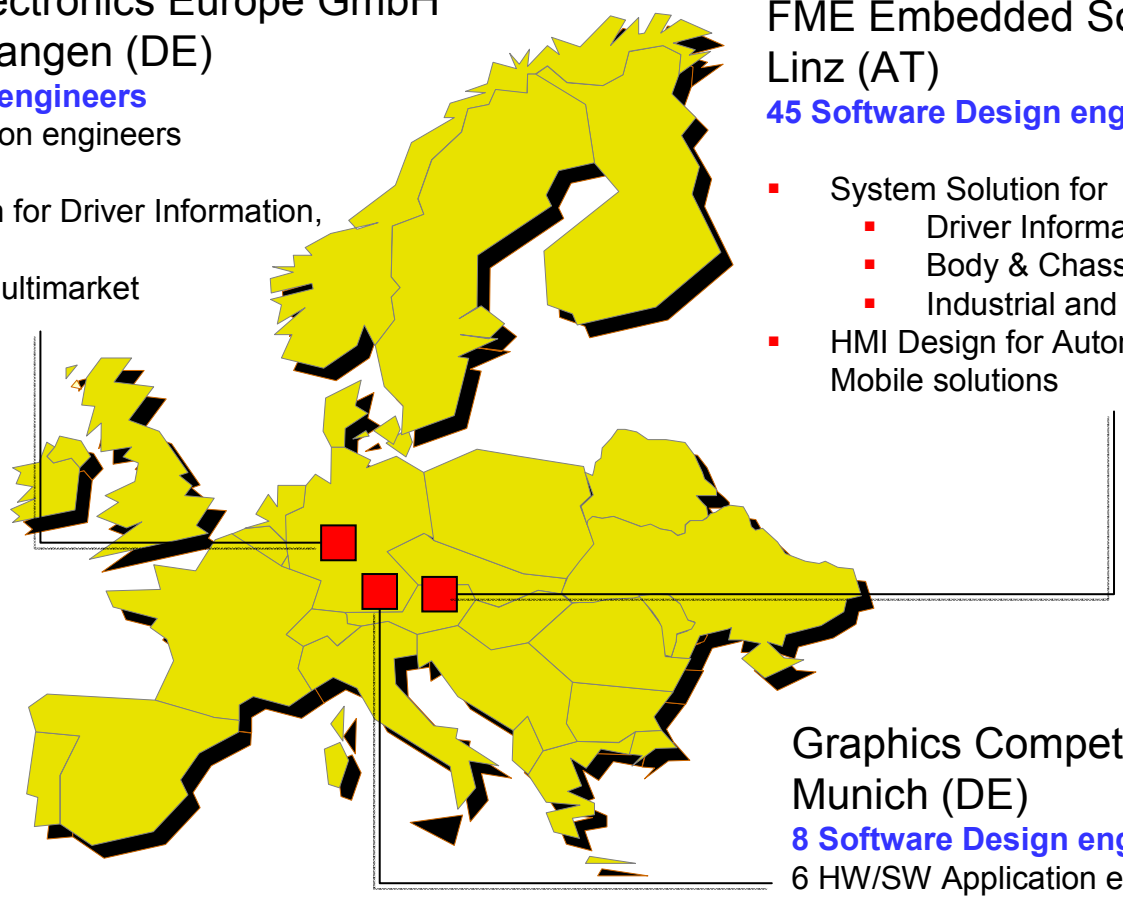
Fujitsu Microelectronics Europe

Fujitsu Microelectronics Europe GmbH
Headquarter, Langen (DE)

8 Software Design engineers

20 HW/SW Application engineers

- System Solution for Driver Information,
- Body & Chassis
- Industrial and Multimarket



FME Embedded Solutions Austria GmbH
Linz (AT)

45 Software Design engineers

- System Solution for
  - Driver Information,
  - Body & Chassis
  - Industrial and Multimarket
- HMI Design for Automotive, Industrial and Mobile solutions

Graphics Competence Center
Munich (DE)

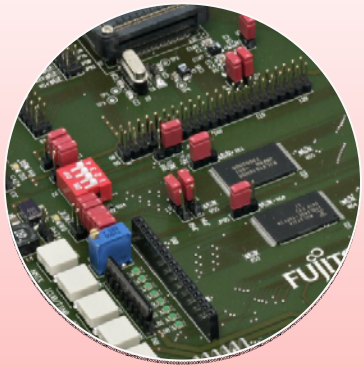
8 Software Design engineers

6 HW/SW Application engineers

- HW, SW and Tool Developments for
- High-End Graphic Solutions

(e.g. based on OpenGL ES 2.0)

System Solution Services



- ✓ Hardware Specification
- ✓ Requirements Screening
- ✓ Engineering Assistance
- ✓ Hardware Configuration
- ✓ Schematics & PCB Design Reviews
- ✓ EMC Characterisation
- ✓ Component Selection
- ✓ Prototyping

- ✓ Software Module Selection
- ✓ Software Development
  - ✓ HMI/GUI Platform
  - ✓ 2D/3D Graphics
  - ✓ Window & Widget Toolkit
  - ✓ Core & Application Modules
  - ✓ Peripheral Components
  - ✓ Complex Drivers
- ✓ Startup Guideance
  - ✓ GUI/HMI Framework
  - ✓ Operating System
  - ✓ Communication Stacks
  - ✓ Peripheral Components
- ✓ Performance Optimization

- ✓ System Conceptioning
- ✓ Partner Coordination
- ✓ 16/32Bit MCU & GDC Platform Training
- ✓ Project Ramp Up Guidance
  - ✓ Intial HW & SW system startup
- ✓ System Definition Workshops
  - ✓ HMI and GUI Graphics
  - ✓ MCAL Software
  - ✓ Basic Software
  - ✓ Hardware Platform

System & Solutions Support Portfolio

AUTOSAR - Ein Überblick

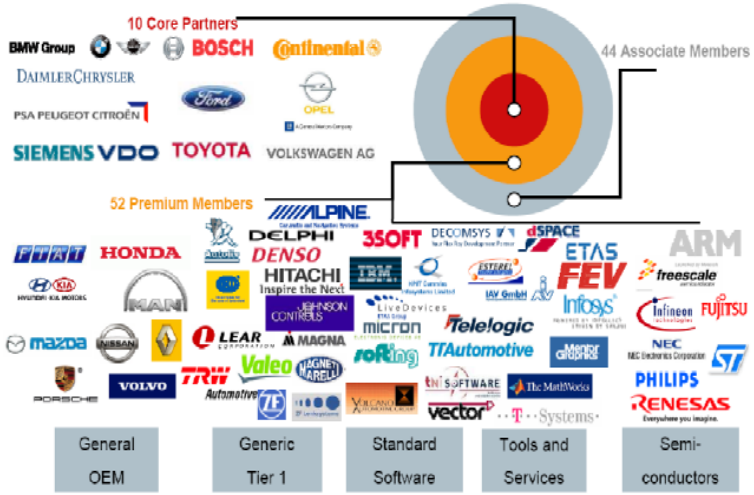
- **Agenda**
  - Fujitsu Microelectronics Europe GmbH
  - Was ist AUTOSAR?
  - Die AUTOSAR Software Architektur
  - Die AUTOSAR Methodik
  - Zusammenfassung und Erfahrungen aus der Praxis

Was ist AUTOSAR?

AUTOSAR

AUTomotive Open System ARchitecture

- **Weltweite Entwicklungspartnerschaft von Automobilherstellern, -zulieferern, und Unternehmen der Software, Halbleiter- und Elektronikindustrie.**
- **Gegründet 2003**
- **Ziel ist die Festlegung eines offenen Standards für die E/E-Architektur im Automobielbereich.**
- **Fujitsu ist Premium Mitglied and arbeitet aktiv an in zwei Arbeitspaketen mit (Kommunikation Stack, MCAL)**



Stand: 4. Dezember 2006, http://www.autosar.org

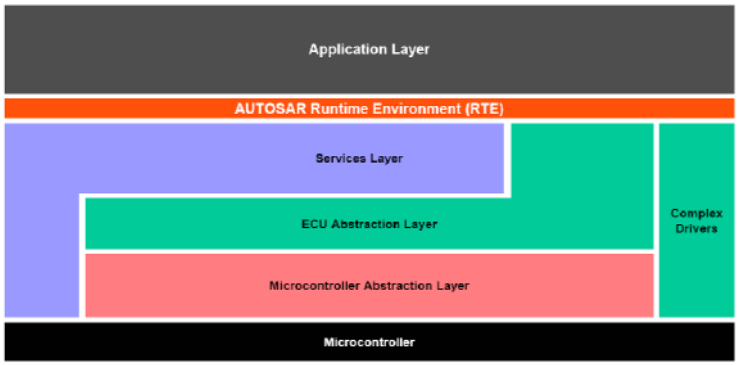
Warum AUTOSAR?

Aktuelle Situation

- **Software** im Automobilen Bereich wächst sehr stark
- **Komplexität** der E/E-Systeme nimmt zu
- Teilweise begrenzt **modularer Aufbau** der Software
- Zum Teil schlechte **Wiederverwendbarkeit**: große Teile der Software müssen neu geschrieben werden, wenn sich der Microkontroller ändert
- Viele verschiedene **Hardware-Plattformen** werden verwendet

Ziele von AUTOSAR

- **Standardisierung**
  - von Schnittstellen
  - der Austauschformate (XML)
  - der Methodik



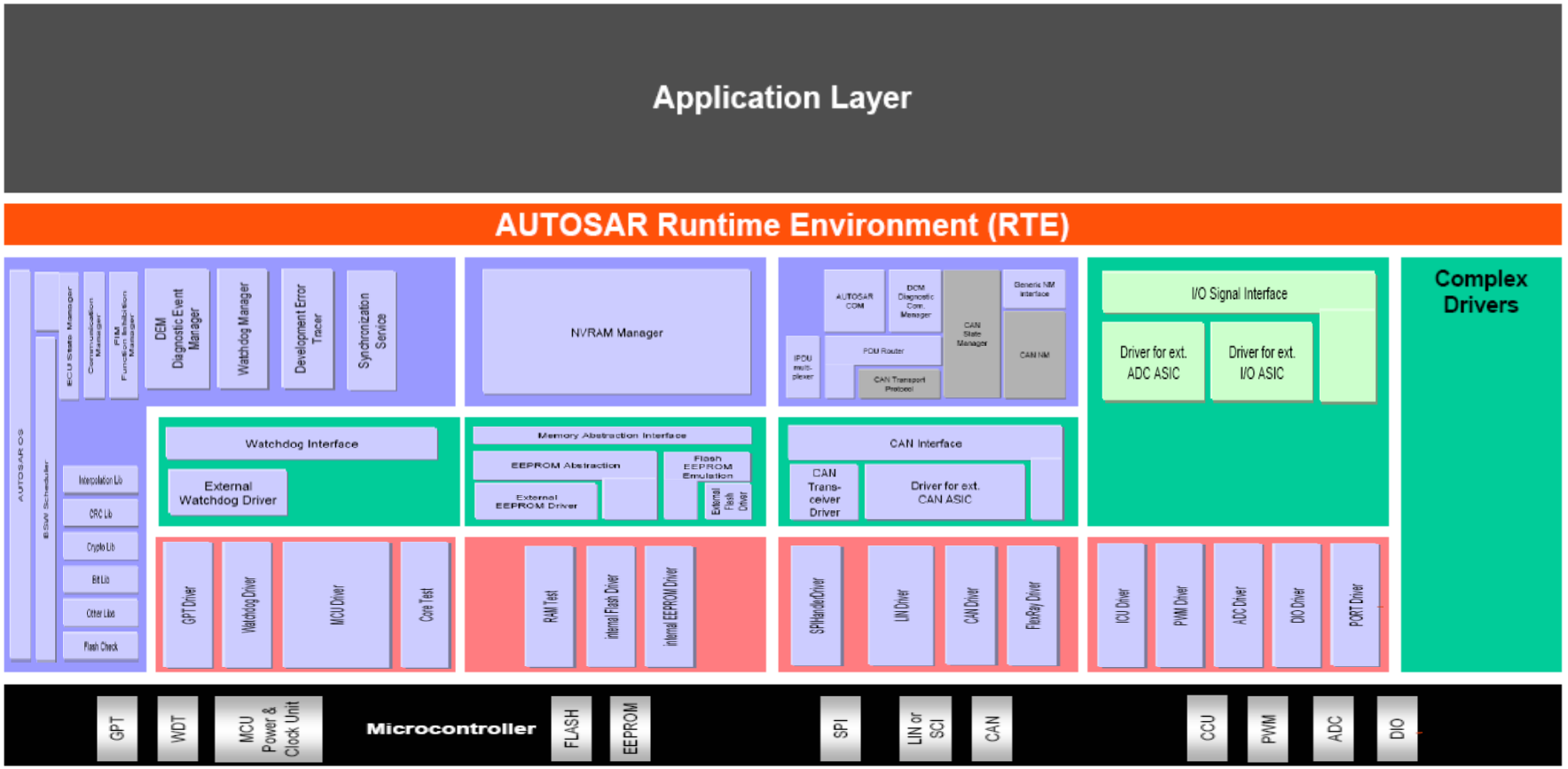
- **Abstraktion** der Hardware
- Verbesserung der **Softwarequalität**
- **Wiederverwendbarkeit** von Funktionen über OEM Grenzen hinweg.
- **Wettbewerb** verstärkt bei applikationsrelevanten Funktionen
- Ermöglichen von standardisierten **Code-Generierungstools**

Agenda

- Fujitsu Microelectronics Europe GmbH
- Was ist AUTOSAR?
- Die AUTOSAR Software Architektur
- Die AUTOSAR Methodik
- Zusammenfassung und Erfahrungen aus der Praxis

Die AUTOSAR Software Architektur

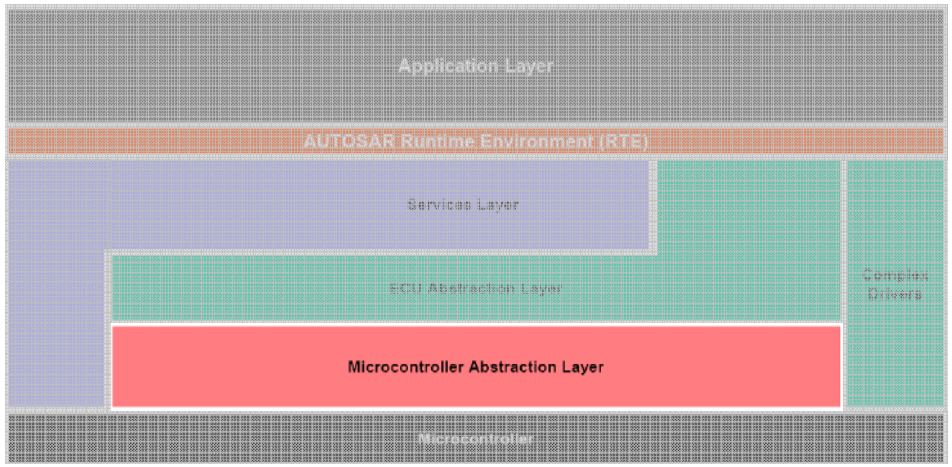
Die AUTOSAR Software Architektur



Source: http://www.autosar.org, „AUTOSAR_LayeredSoftwareArchitecture.pdf“, Version 2.2.1, 23.06.2008

Die AUTOSAR Software Architectur

- **Die Microcontroller Abstraktionsschicht (Mircrocontroller Abstraction Layer)**



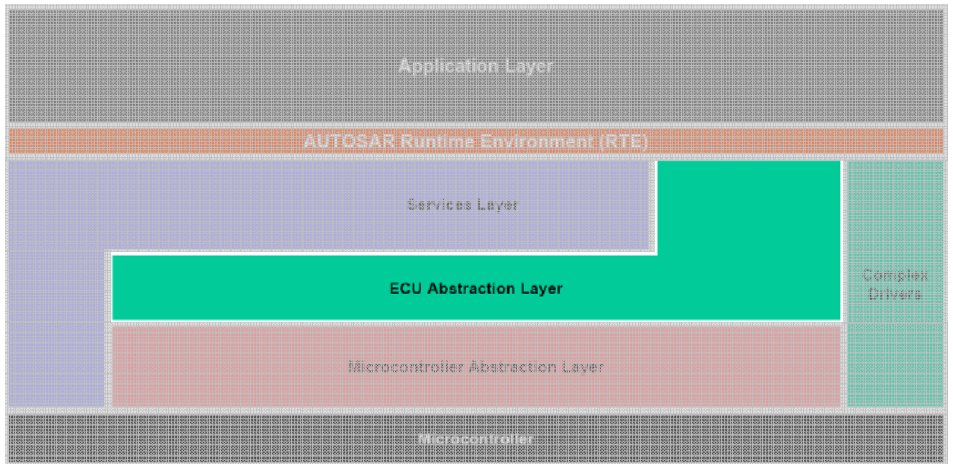
- **Aufgabe:**
  - Höhere Schichten unabhängig von der Microcontroller Hardware zu machen.

- **Funktion:**
  - Treiber für direkten Zugriff auf die interne Peripherie (PWM, DIO, ICU, ...)

Source: http://www.autosar.org, „AUTOSAR_LayeredSoftwareArchitecture.pdf“, Version 2.2.1, 23.06.2008

Die AUTOSAR Software Architectur

- **Die ECU Abstraktionsschicht (ECU Abstraction Layer)**



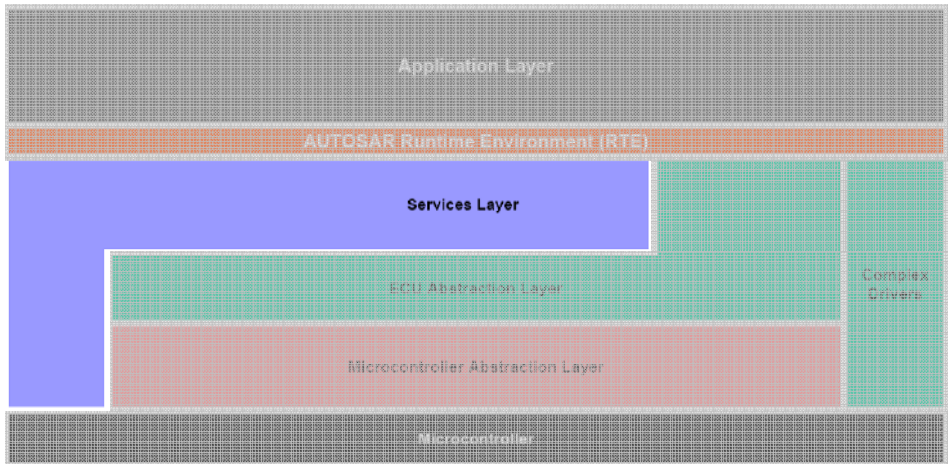
- **Aufgabe:**
  - Höhere Schichten unabhängig von der Steuergeräte Hardware zu machen.

- **Funktion:**
  - Treiber für externe Geräte.
  - Schnittstelle für externe und interne Peripherie.

Source: http://www.autosar.org, „AUTOSAR_LayeredSoftwareArchitecture.pdf“, Version 2.2.1, 23.06.2008

Die AUTOSAR Software Architectur

- **Die Serviceschicht (Service Layer)**



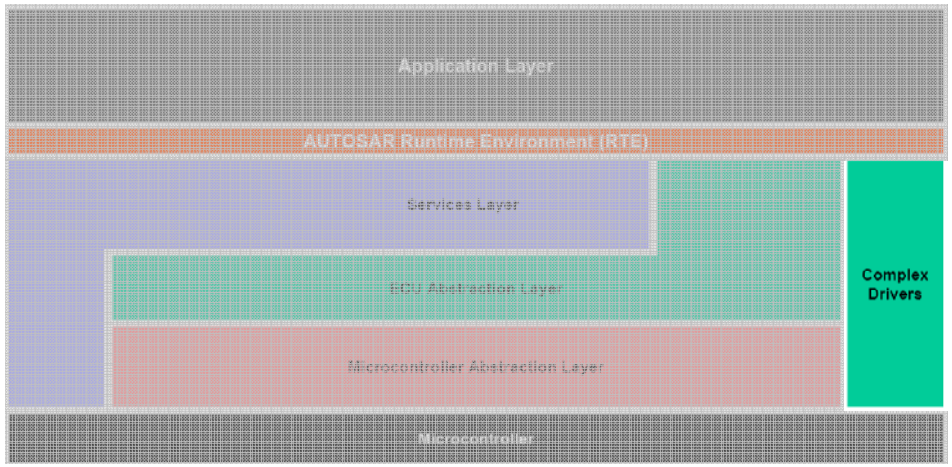
- **Aufgabe:**
  - Bereitstellen von Diensten für die Applikation

- **Funktion:**
  - Diagnose, NVRAM Management, OS, Kommunikation
  - Speicher- und Steuergeräte Management

Source: http://www.autosar.org, „AUTOSAR_LayeredSoftwareArchitecture.pdf“, Version 2.2.1, 23.06.2008

Die AUTOSAR Software Architectur

- **Die Complex Device Drivers**



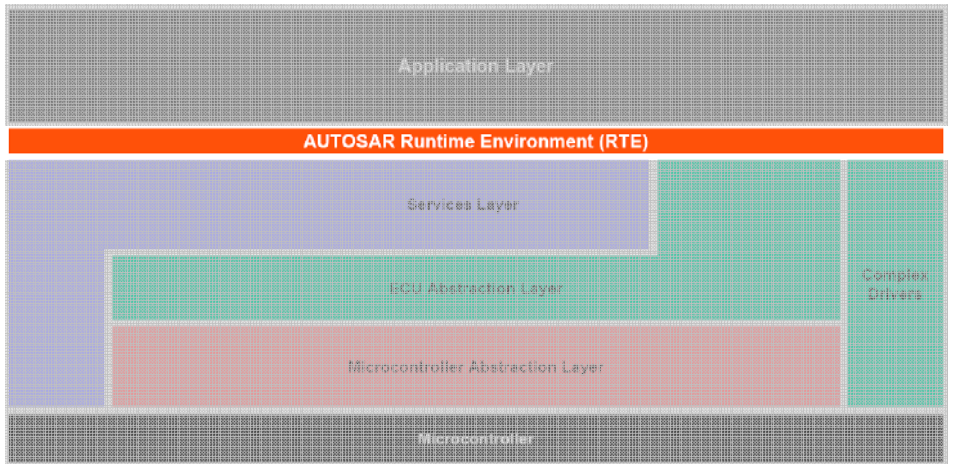
- **Aufgabe:**
  - Ermöglichung der Realisierung von speziellen funktionalen und zeitlichen Anforderungen an komplexe Treiber für Sensoren und Aktoren

- **Funktion:**
  - Direkter Zugriff die Microkontroller Hardware für kritische Anwendungen

Source: http://www.autosar.org, „AUTOSAR_LayeredSoftwareArchitecture.pdf“, Version 2.2.1, 23.06.2008

Die AUTOSAR Software Architectur

- **Die AUTOSAR Runtime Environment (RTE)**



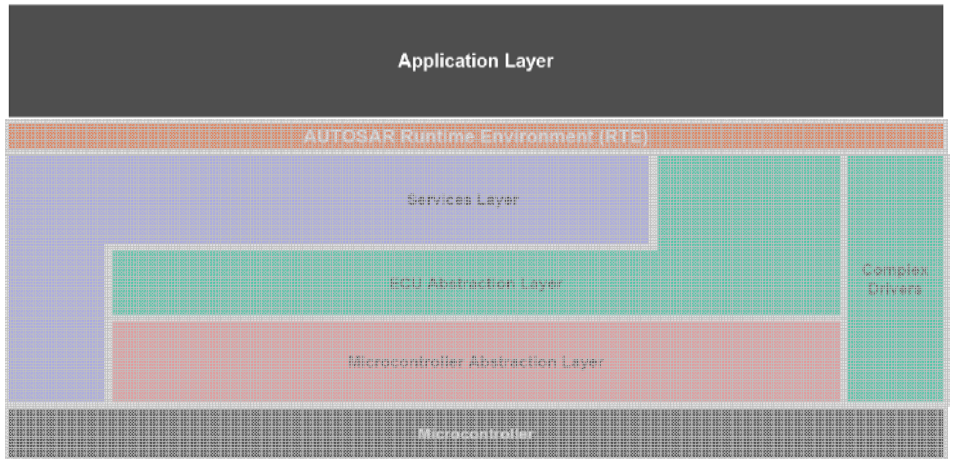
- **Aufgabe:**
  - Softwarekomponenten unabhängig vom Steuergeräte Mapping

- **Funktion:**
  - Middleware, welche Kommunikationsdienste (intra und inter ECU) zur Verfügung stellt.

Source: http://www.autosar.org, „AUTOSAR_LayeredSoftwareArchitecture.pdf“, Version 2.2.1, 23.06.2008

Die AUTOSAR Software Architectur

- **Die Applikationsschicht (Application Layer)**



- **Aufgabe:**
  - Beinhaltet „eigentliche“ Applikation

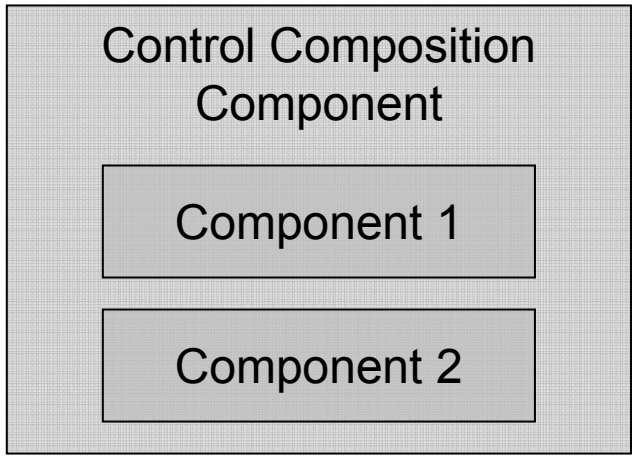
- **Funktion:**
  - Implementierung von Steuerungsalgorithmen, Setzen von Ausgangswerten, ...

Source: http://www.autosar.org, „AUTOSAR_LayeredSoftwareArchitecture.pdf“, Version 2.2.1, 23.06.2008

Die AUTOSAR Software Architectur

Aufbau einer Applikation

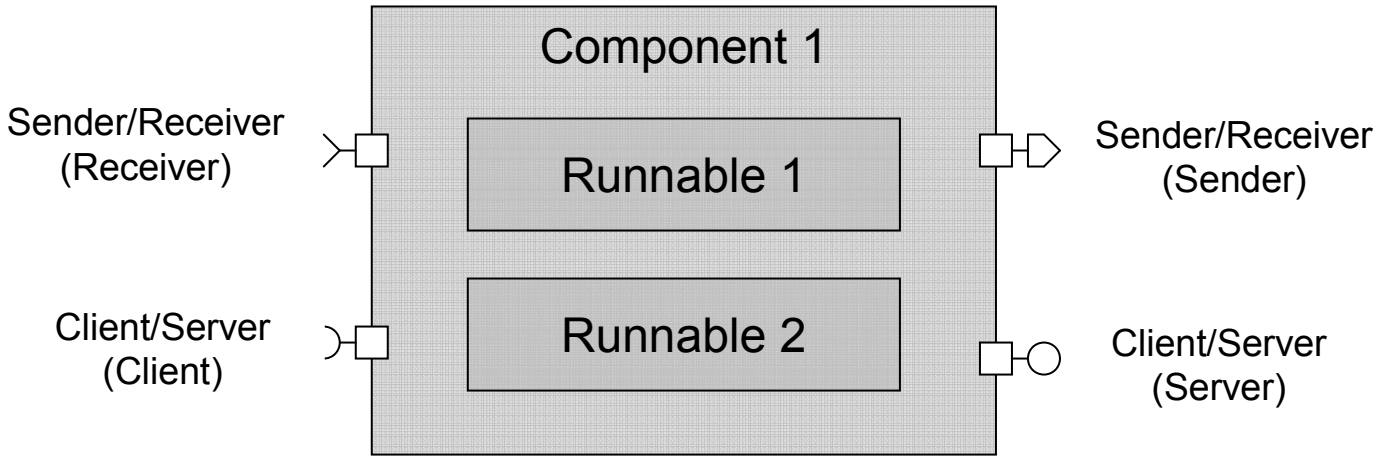
- Eine Applikation besteht aus „Components“.
- „Components“ können logisch zusammengefasst werden in so genannten „Compositions“.



Die AUTOSAR Software Architectur

Aufbau eines Software-Components

- **Ein Software-Components bestehen aus:**
  - Ports
    - Kommunikationsschnittstellen zu anderen Komponenten
  - Runnable Entities (Runnables)
    - Funktionen, die die tatsächliche Implementierung enthalten



Agenda

- Fujitsu Microelectronics Europe GmbH
- Was ist AUTOSAR?
- Die AUTOSAR Software Architektur
- Die AUTOSAR Methodik
- Zusammenfassung und Erfahrungen aus der Praxis

Die AUTOSAR Methodik

Was bedeutet AUTOSAR Methodik?

- Die AUTOSAR Methodik beschreibt die verschiedenen Phasen und Schritte des Entwicklungsprozesses vom Systementwurf bis hin zur Implementierung.
- Die AUTOSAR Methodik beschreibt die Abhängigkeiten von Arbeitsergebnissen
- Die AUTOSAR Methodik ist aber keine komplette Prozessbeschreibung

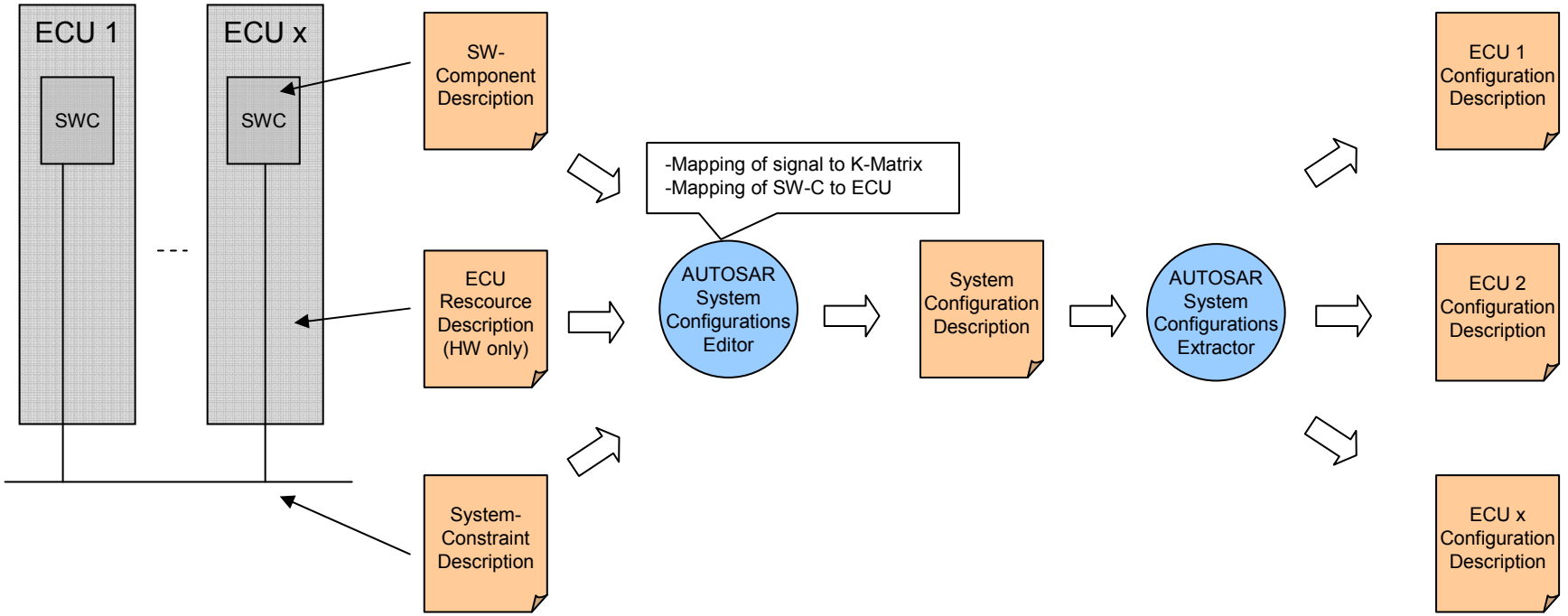
Die AUTOSAR Methodik

Was bedeutet AUTOSAR Methodik?

- Die AUTOSAR Methodik beschreibt die verschiedenen Phasen und Schritte des Entwicklungsprozesses vom Systementwurf bis hin zur Implementierung.
- Die AUTOSAR Methodik beschreibt die Abhängigkeiten von Arbeitsergebnissen
- Die AUTOSAR Methodik ist aber keine komplette Prozessbeschreibung

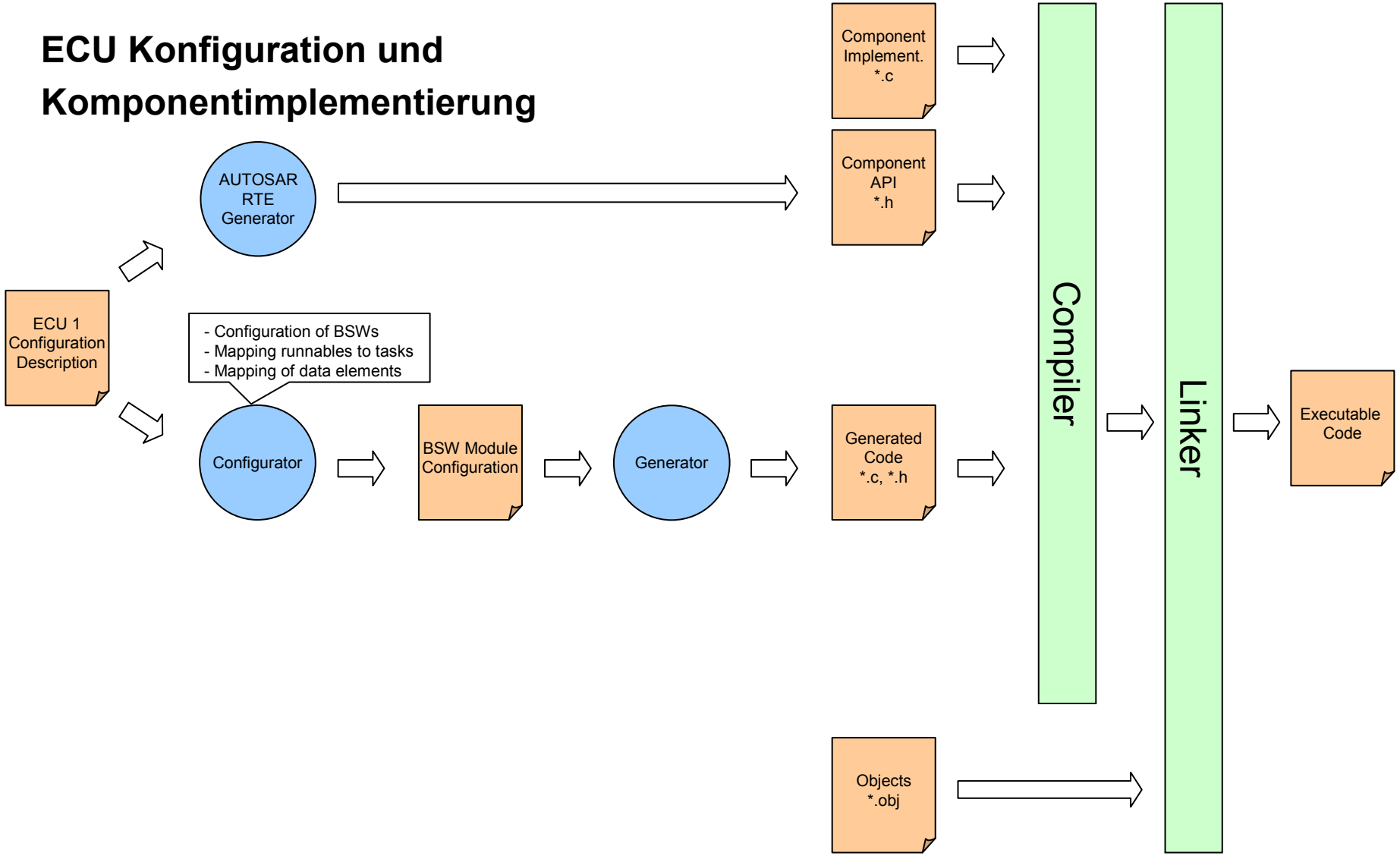
Die AUTOSAR Methodik

System Konfiguration



Die AUTOSAR Methodik

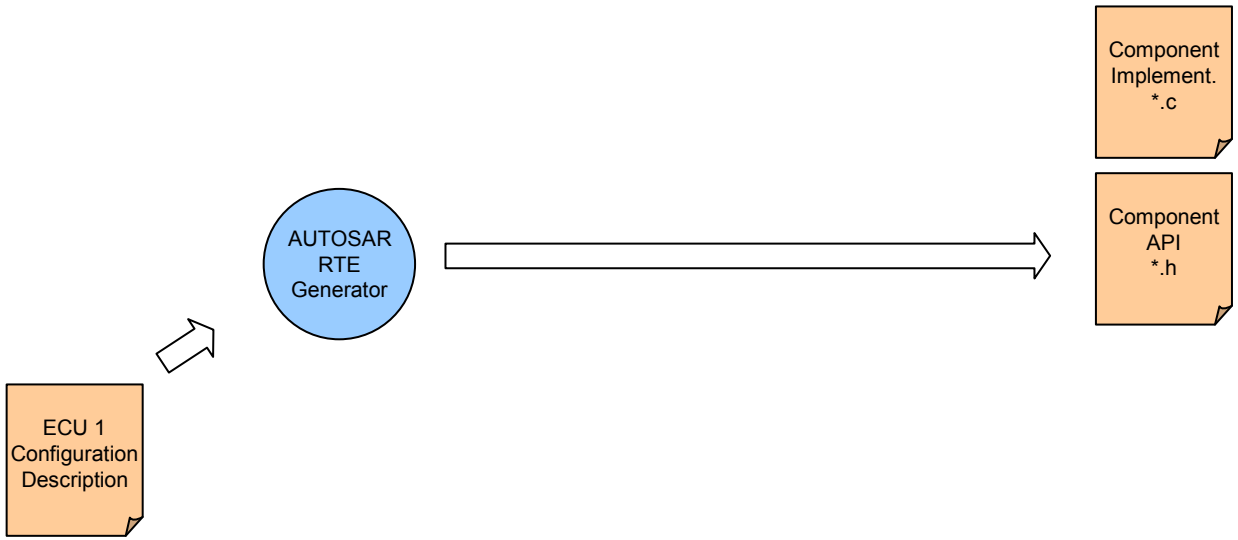
ECU Konfiguration und Komponentimplementierung



Die AUTOSAR Methodik

Komponentimplementierung

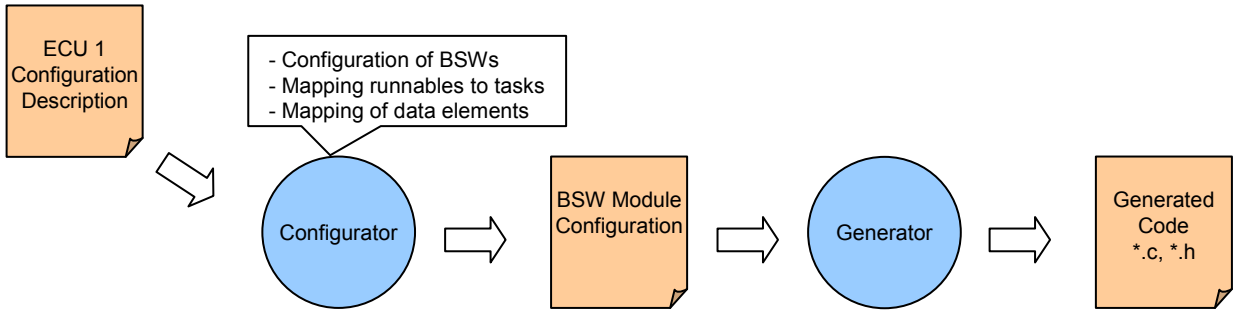
- Entwicklung der Softwarekomponenten unabhängig von der BSW-Konfiguration möglich.
- Softwarekomponenten können simuliert und getestet werden.



Die AUTOSAR Methodik

ECU Konfiguration

- BSW kann unabhängig von der Komponentenentwicklung konfiguriert werden.
- BSW kann unabhängig getestet werden.



Agenda

- Fujitsu Microelectronics Europe GmbH
- Was ist AUTOSAR?
- Die AUTOSAR Software Architektur
- Die AUTOSAR Methodik
- Zusammenfassung und Erfahrungen aus der Praxis

Zusammenfassung

Ziele von AUTOSAR and der Lösung

- **Abstraktion** der Hardware
  - ✓ durch den Microcontroller Abstraction Layer
- Verbesserung der **Softwarequalität**
  - ✓ durch standardisierte Basic Software
- **Wiederverwenbarkeit** von Funktionen über OEM Grenzen hinweg
  - ✓ durch die Runtime Environment
- **Wettbewerb** verstärkt bei applikationsrelevanten Funktionen
  - ✓ durch standardisierte Basic Software
- Ermöglichen von standardisierten **Code-Generierungstools**
  - ✓ durch Standardisierung

Erfahrungen aus der Praxis

Kritik am derzeitigen Stand

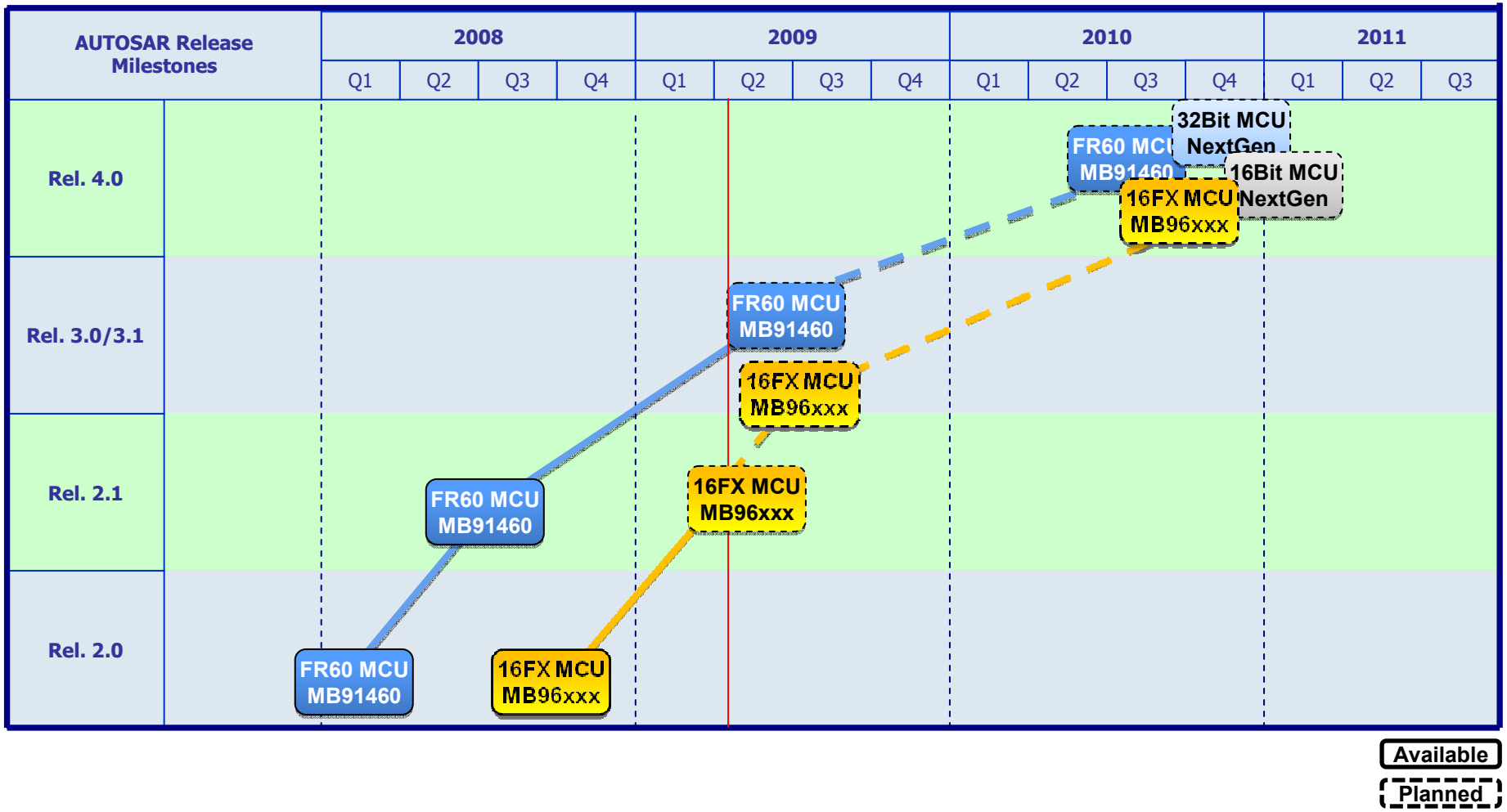
- **Häufige Neuerscheinungen** von AUTOSAR Version verursachen Kosten und erschweren den „Start“ mit AUTOSAR.
- **OEM-spezifische BSW-Module**, die zusätzliche Funktionalitäten abdecken, verursachen derzeit noch zusätzlichen Integrationsaufwand.
- **„Verschiedene Interpretation“** der Software-Spezifikationen erschweren den Austausch von Basic-Softwareschichten bzw. Basic-Softwaremodulen.
- Da AUTOSAR **keine performance-relevanten Anforderungen** enthält, stellt sich das Verschieben von Softwarekomponenten auf verschiedene ECUs als schwierig dar (z. B. Inter- vs. Intra-ECU-Kommunikation).

Erfahrungen aus der Praxis

Lob am derzeitigen Stand

- AUTOSAR-Partnerschaft hält den Standard **aktuell**.
- Standardisierung ermöglicht relativ **einfachen Wechsel** der Hardware.
- **Bewußtsein** für die notwendige Änderung in der E/E-Entwicklung wird geschärft.
- Erste OEMs beginnen mit **Ausschreibung von Projekten** auf Basis von AUTOSAR.

Fujitsu AUTOSAR MCAL - Roadmap



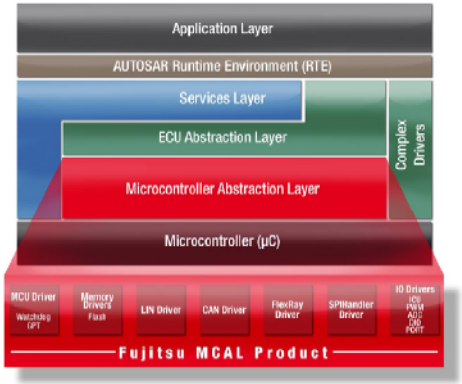
Fujitsu AUTOSAR ECU System

Tresos Studio
Tresos AUTOCORE
R2.1 & R3.0

- ✓ **Partnership** + Distribution Agreement
- ✓ Integrated **Solution** & homogeneous tool environment
- ✓ 32Bit MB91460 Series MCU MCAL in R2.1 supported
- ✓ 32Bit MB91460 Series MCU MCAL in R3.0 on request
- ✓ Dedicated OEM BSW components available

MICROSAR
R3.0 BSW

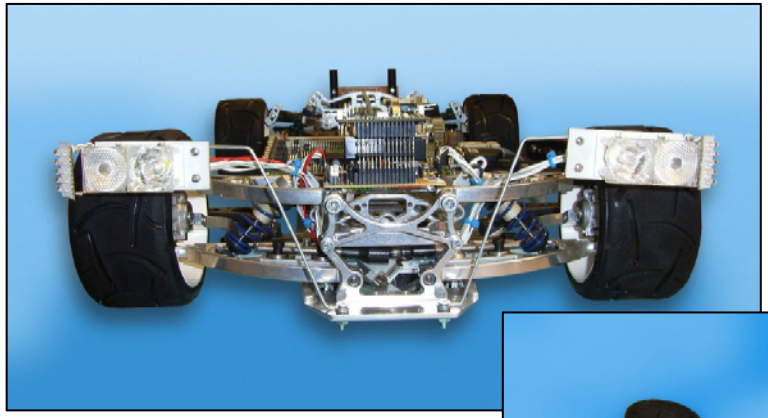
- ✓ 32Bit MB91460 Series MCU MCAL supported
- ✓ Dedicated OEM BSW components available



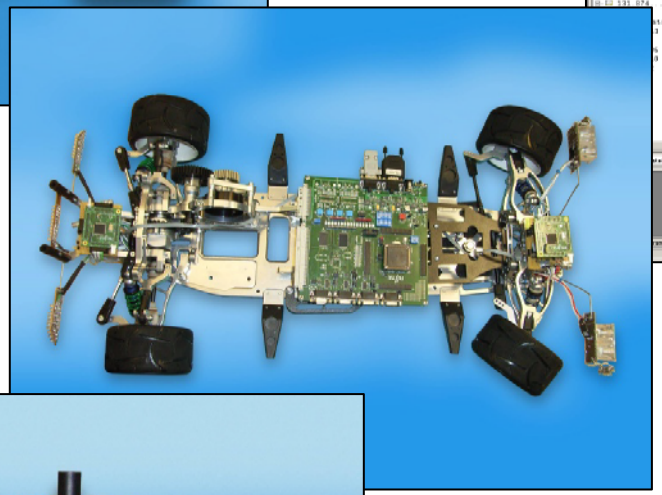
AUTOSAR Builder
ATDK R3.0 BSW

- ✓ 32Bit MB91460 Series MCU MCAL available soon (MCAL 2.1, BSW 3.0)

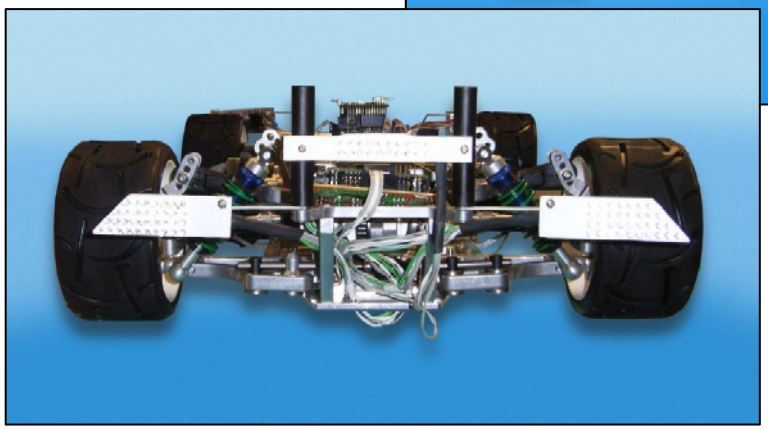
AUTOSAR Roadster



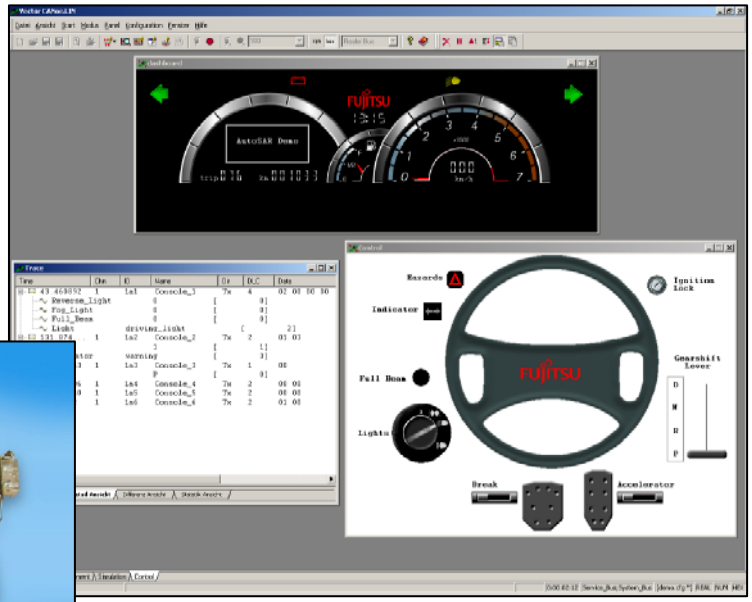
Front View



Top View



Rear View

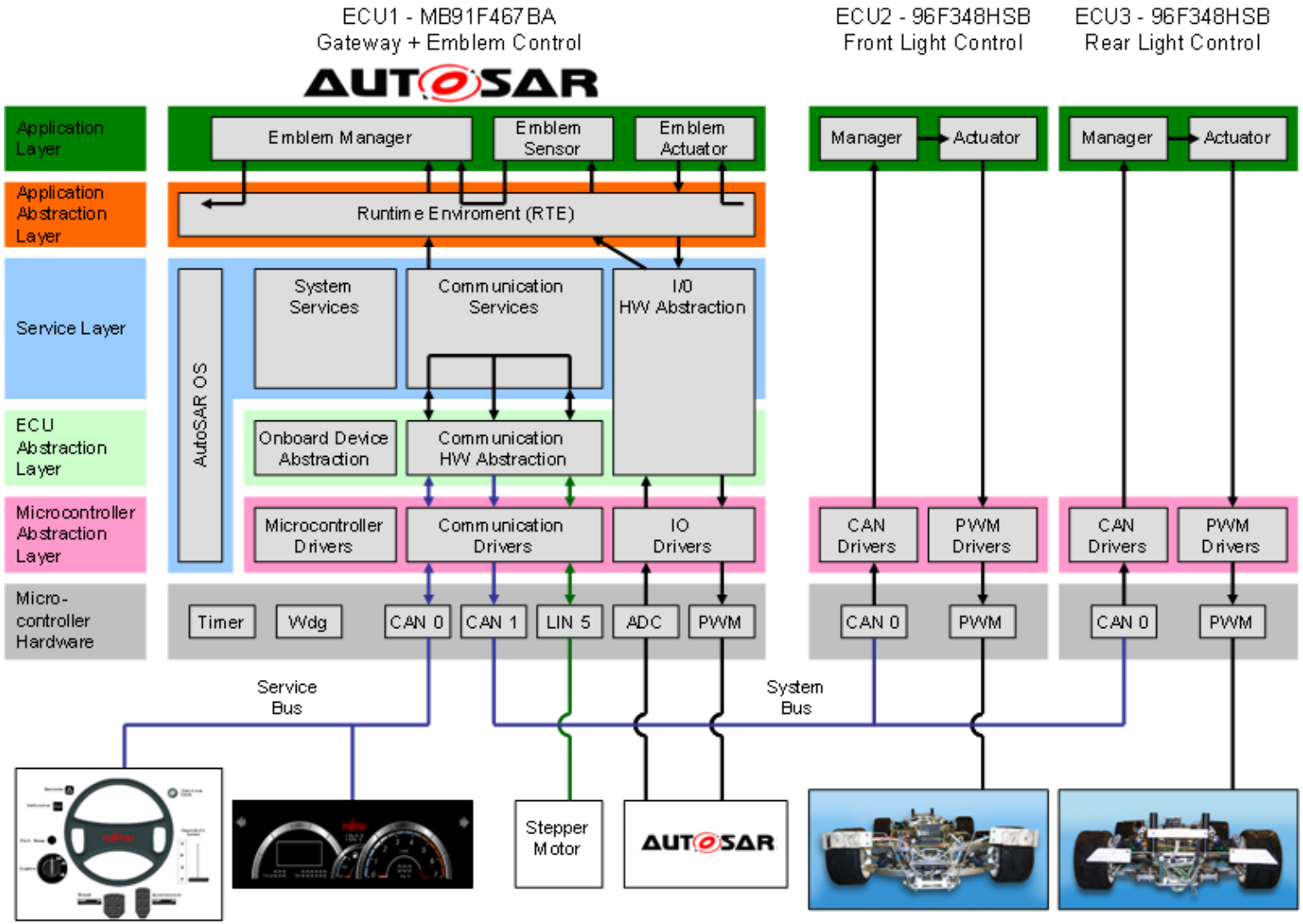


Graphical User Interface

Features of AUTOSAR Roadster

- **The roadster currently consists of 3 ECUs and a graphical user interface**
  - ECU1 – MB91F467BA: CAN Gateway and display controller unit
    - Control via CAN messages
    - Gateway and multicast functionallity (PDU Router)
    - Control the brightness of the AUTOSAR emblem
    - Control of a LIN stepper motor
  - ECU2 – MB96F348HSB: Front light controller unit
    - Control via CAN messages
    - Control of indicators and front lights
  - ECU3 – MB96F348HSB: Back light controller unit
    - Control via CAN messages
    - Control of indicators and back lights
    - Adaptive brake lights
  - Graphical user interface
    - Virtual dashboard
    - Virtual control via CAN

Software Architecture



THE POSSIBILITIES ARE INFINITE