Software Platforms for Automotive Systems

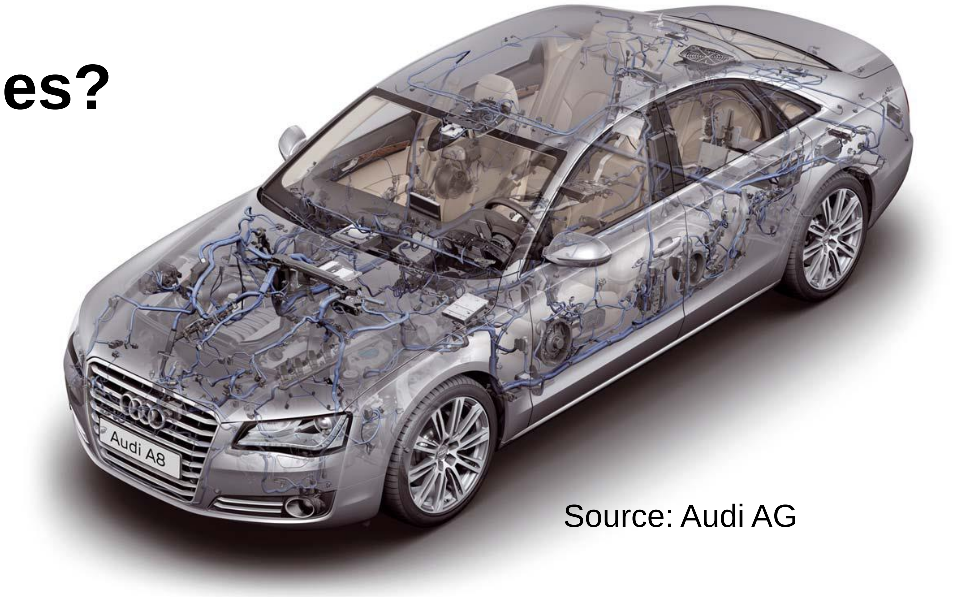
Lecture 1: Introduction

Alejandro Masrur

15th October 2015, TU Chemnitz

What do you know about cars?

- How many processors are there in a modern car?
- What is an ECU and an OEM?
- How many kilometers of cable are there?
- How long is the product life cycle of a car?
- How many lines of codes?

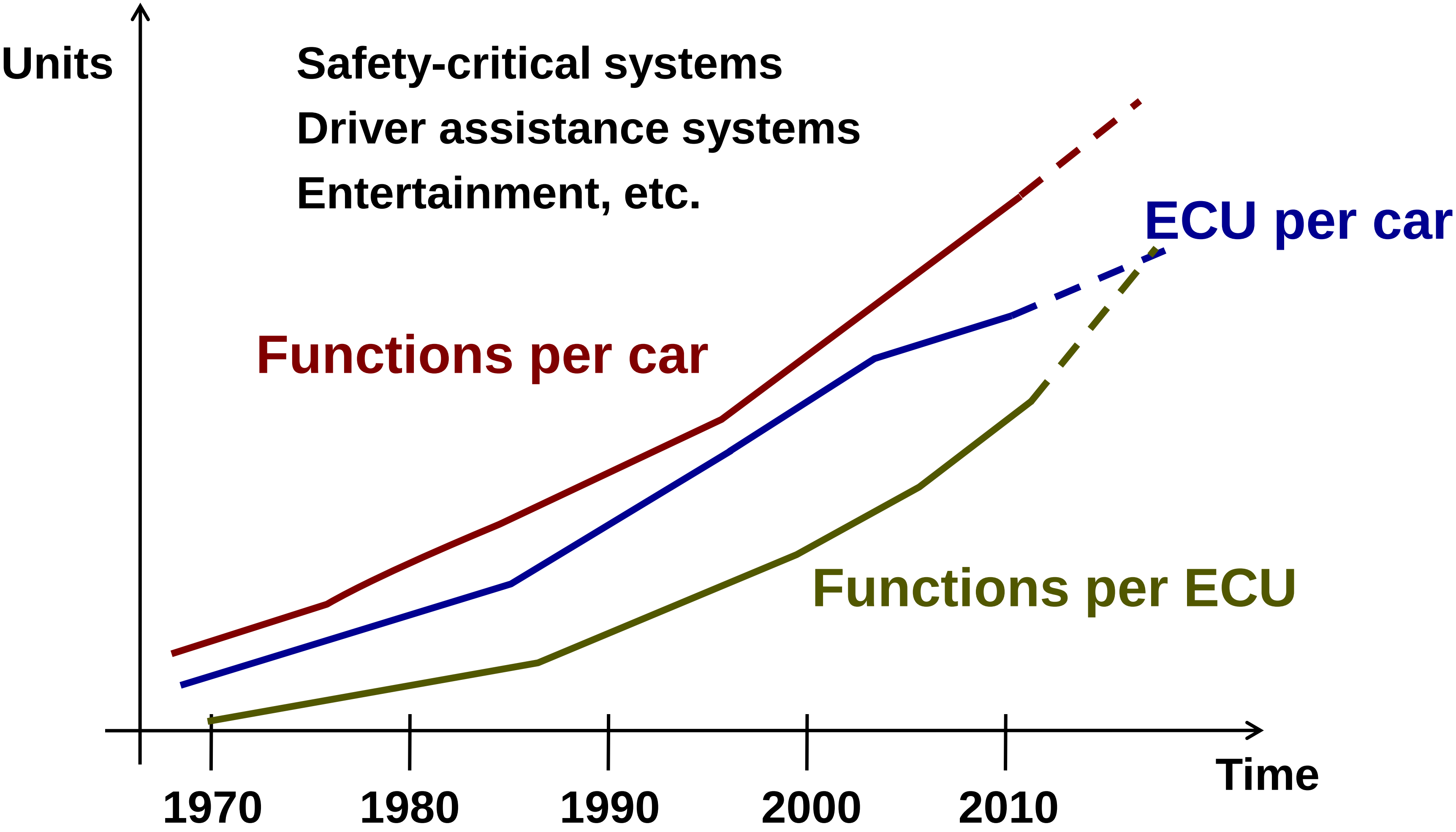


Source: Audi AG

Automotive Companies

- OEM: Original Equipment Manufacturer
  - Companies which develop cars
  - BMW, Audi, Toyota, Ford, etc.
- Tier 1 providers (develop automotive systems)
  - Bosch, Continental, Delphi, etc.
- Tier 2 providers (develop processors chips, etc.)
  - Freescale, Infineon, etc.
- Tool providers (develop automotive software tools)
  - dSpace, Vektor, Mentor Graphics, etc.

Trend in Automotive Systems



Safety-critical systems

Driver assistance systems

Entertainment, etc.

The Automotive E/E Architecture

- All electrical/electronic (E/E) components in a car
  - Currently it has a high complexity and a high cost
    - Around 100 electronic control units (ECUs)
    - Around 10 buses and over 4 Km cable
      - Domain-specific technologies, e.g., LIN, FlexRay, etc.
    - Over 100 million lines of code
- Carrier of most innovation in today’s cars
- Software has a crucial role
  - Trend to even more functionality and complexity
  - Need for portability and reusability of software

Requirements on Car’s E/E Systems

- Harsh environmental conditions
  - Wide temperature range, high humidity and vibration
  - Electromagnetic interference (EMI)
- Reliability and availability
  - Providing required functions in specified time interval
- Functional Safety
  - Avoiding potential hazards for passengers and others
- Relatively long product life cycles
  - Around 20 to 25 years

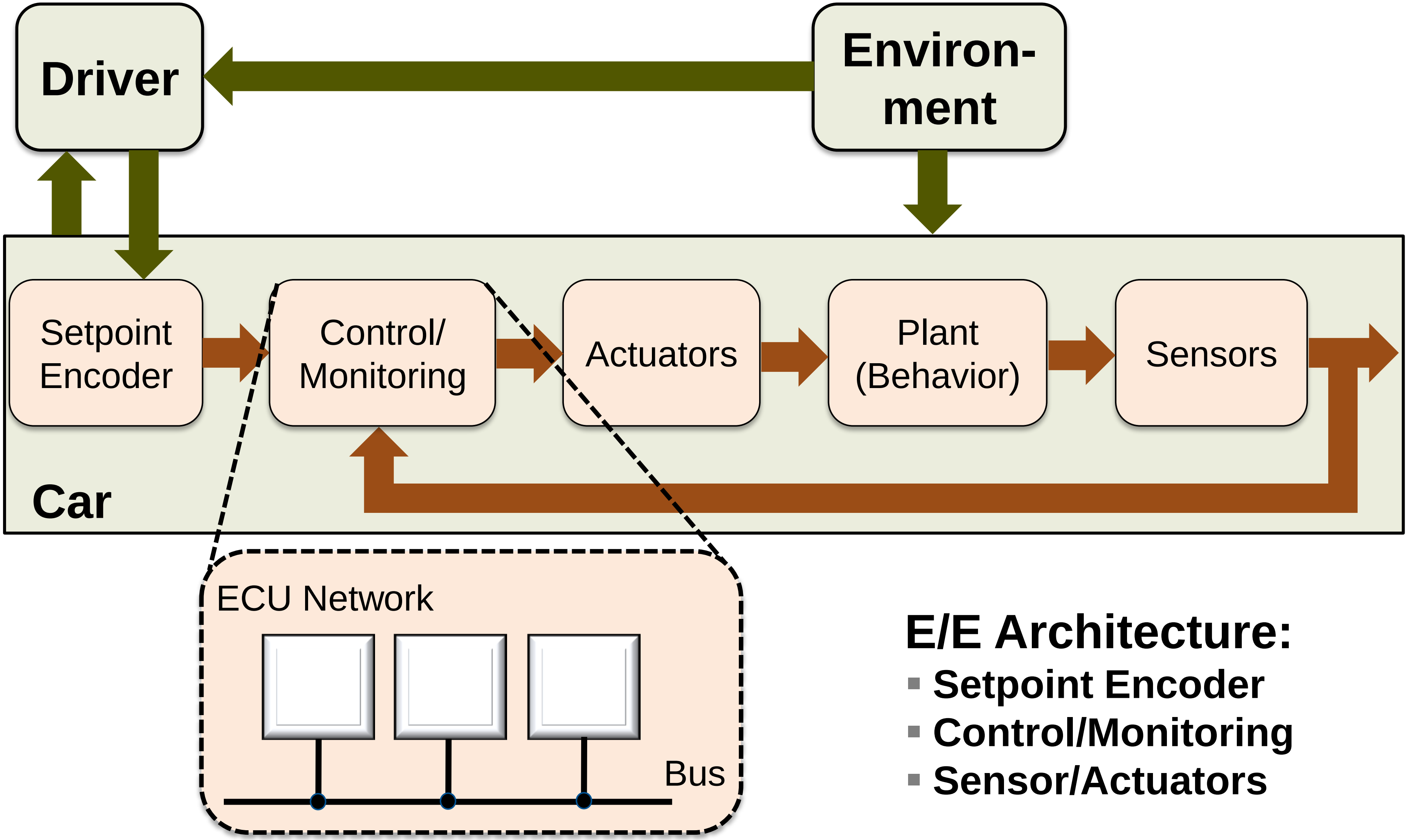
Automotive Software Engineering

- An attempt of definition:

"Systematic use of techniques and tools that support the design and development process with the aim of mastering complexity and achieving efficient and reliable software for the automotive domain."

- Efficient in the sense of resources and reusability
  - ECUs need to be cost-effective and have limited resources
  - A high reusability of software saves costs
- Reliable because of safety-critical applications
  - Brakes, airbags, electronic stability program (ESP), etc.
  - Real-time behavior is required most of the time

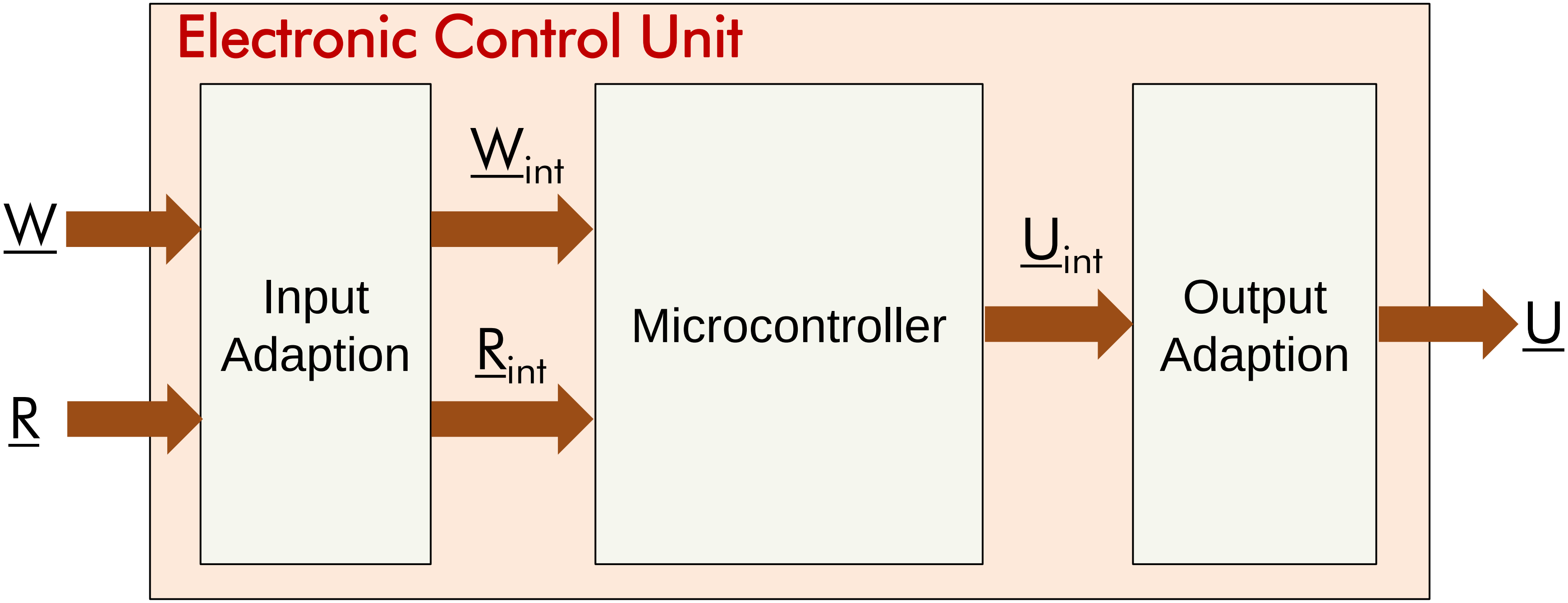
The Driver-Car-Environment System



E/E Architecture:

- Setpoint Encoder
- Control/Monitoring
- Sensor/Actuators

What's an ECU?



- **Input/output adaption**
  - **Translate voltage and current levels**

Automotive Functional Domains

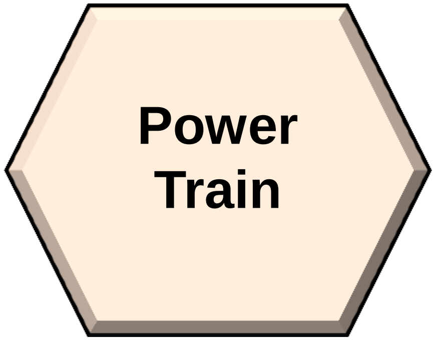
- Functions are divided into 6 main domains
  - Manage complexity of the overall system
  - Different requirements and different technologies
- Power Train
- Chassis
- Safety (passive)

Vehicle centric (more critical)

- Body/Comfort
- Multimedia/Telematics
- Man Machine Interface

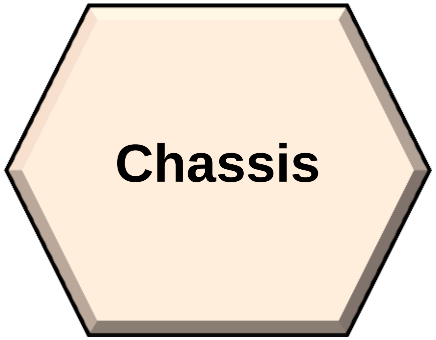
Passenger centric (less critical)

Power Train


Power Train

- Concerns the following
  - Engine, gearbox, driveshaft, crankshaft, etc.
- Sensors
  - Throttle position, battery voltage, etc.
- Actuators
  - Throttle, ignition plugs, fuel injectors, etc.
- On-board communication
  - HS-CAN (High Speed-Controller Area Network)
- Engine management, gearbox control, etc.

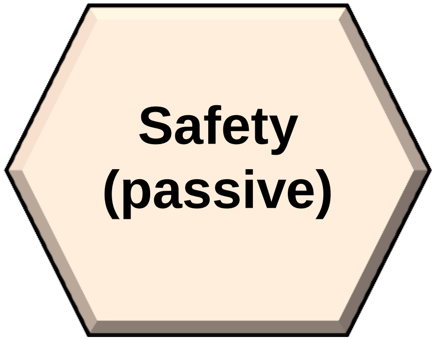
Chassis



Chassis

- Concerns the following
  - Front and back axles, brakes, wheels, etc.
- Sensors
  - Wheels’ rotational speed, yaw and steer angle, etc.
- Actuators
  - Individual brakes at wheels, etc.
- On-board communication
  - HS-CAN (High Speed-Controller Area Network)
- Anti-blocking system, electronic stability program, etc.

Safety (Passive)



- Concerns the following
  - Airbags, seatbelt, etc.
- Sensors
  - Collision detectors, seat occupancy, etc.
- Actuators
  - Airbag inflation, etc.
- On-board communication
  - LS-CAN (Low Speed-Controller Area Network)
- Airbag control, seatbelt tightener, etc.

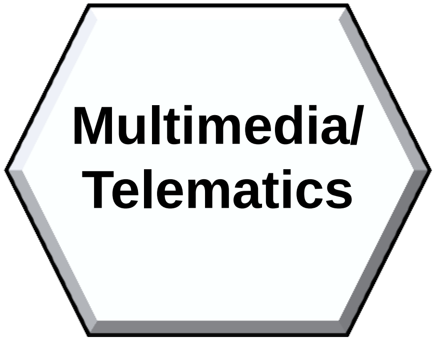
Body/Comfort

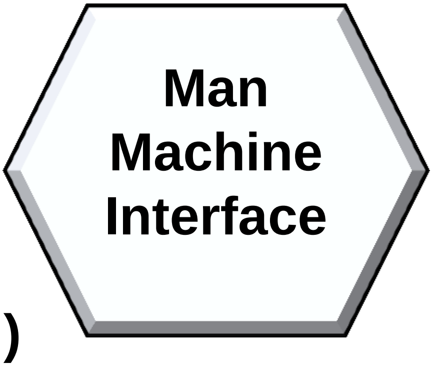
Body/ Comfort

- Concerns the following
  - Side mirrors, seats, steering wheel, etc.
- Sensors
  - Rain sensor, temperature sensor, etc.
- Actuators
  - DC motor actuator for windows and mirrors, heating, etc.
- On-board communication
  - LS-CAN (Low Speed-Controller Area Network)
- Parking assistant, electric windows, etc.

Remaining Domains

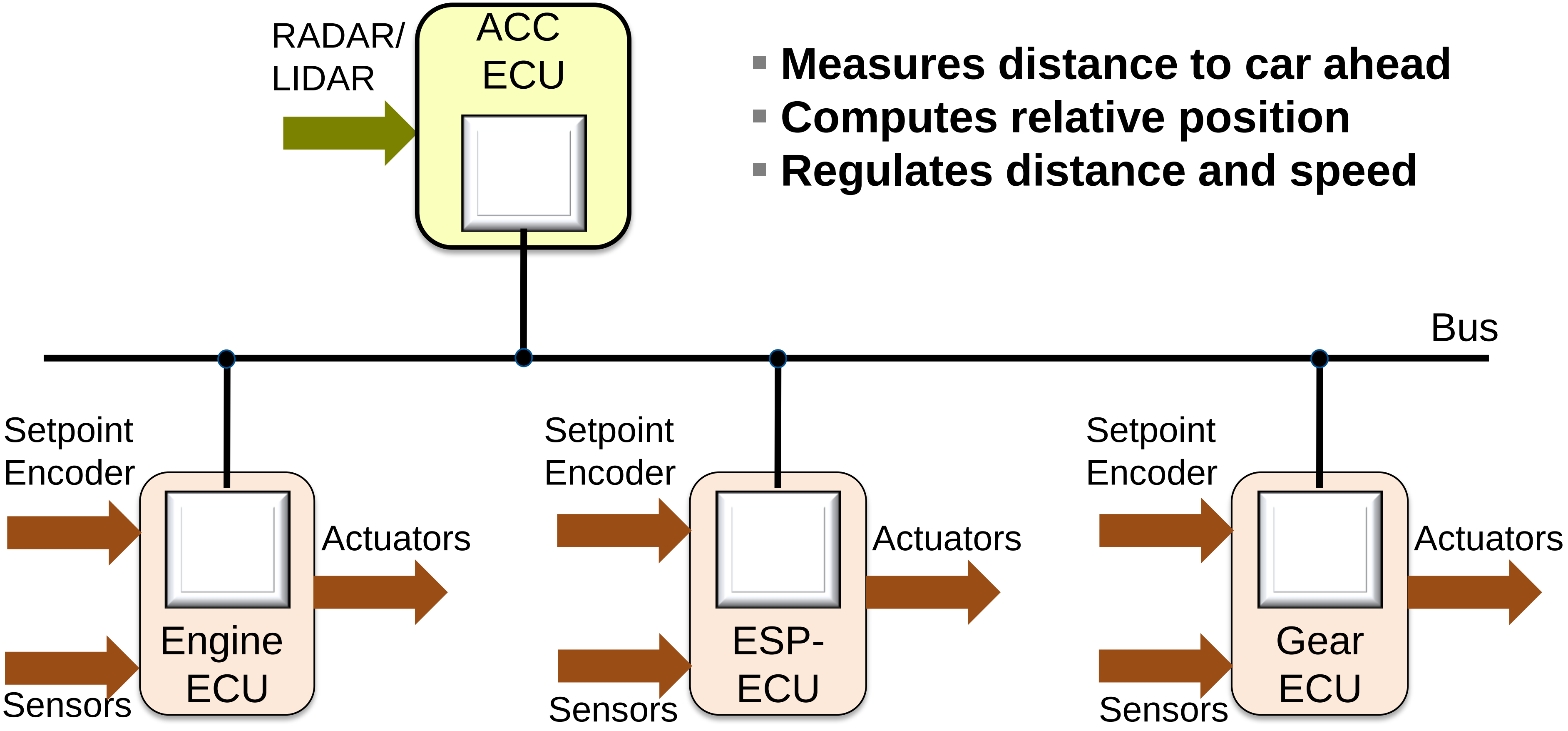
- Multimedia/Telematics
  - Entertainment: Radio, Video, etc.
  - Internet access
  - Navigation
- Man Machine Interface
  - Instrument panel
  - Control panel (i.e., buttons, turn-switches, etc.)
  - Voice command
- On-board communication
  - Mainly MOST (Media Oriented Systems Transport)


Multimedia/ Telematics


Man Machine Interface

Cross-Domain Function

Adaptive cruise control (ACC):

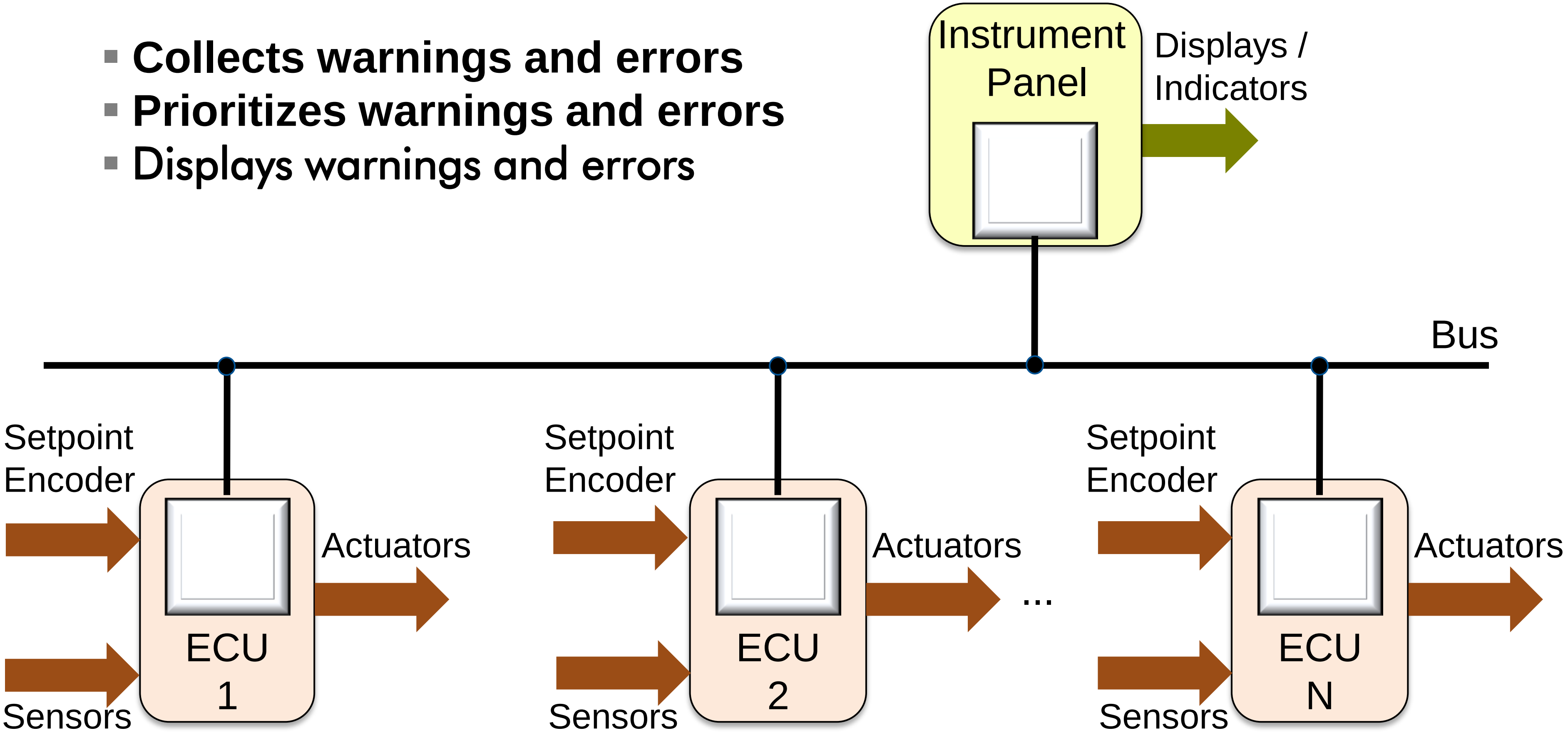


- **Measures distance to car ahead**
- **Computes relative position**
- **Regulates distance and speed**

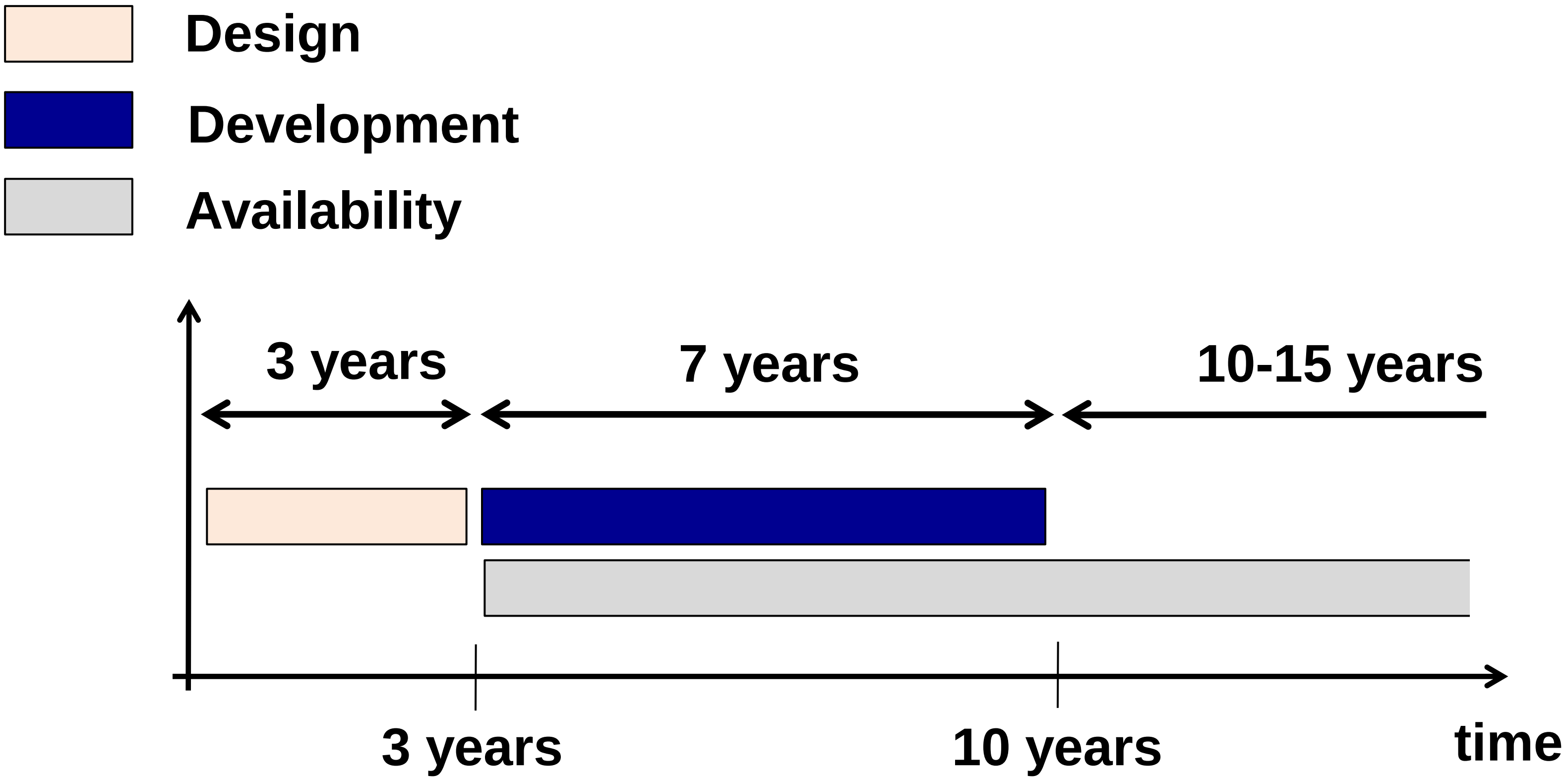
Cross-Domain Function

Instrument Panel Control

- **Collects warnings and errors**
- **Prioritizes warnings and errors**
- Displays warnings and errors

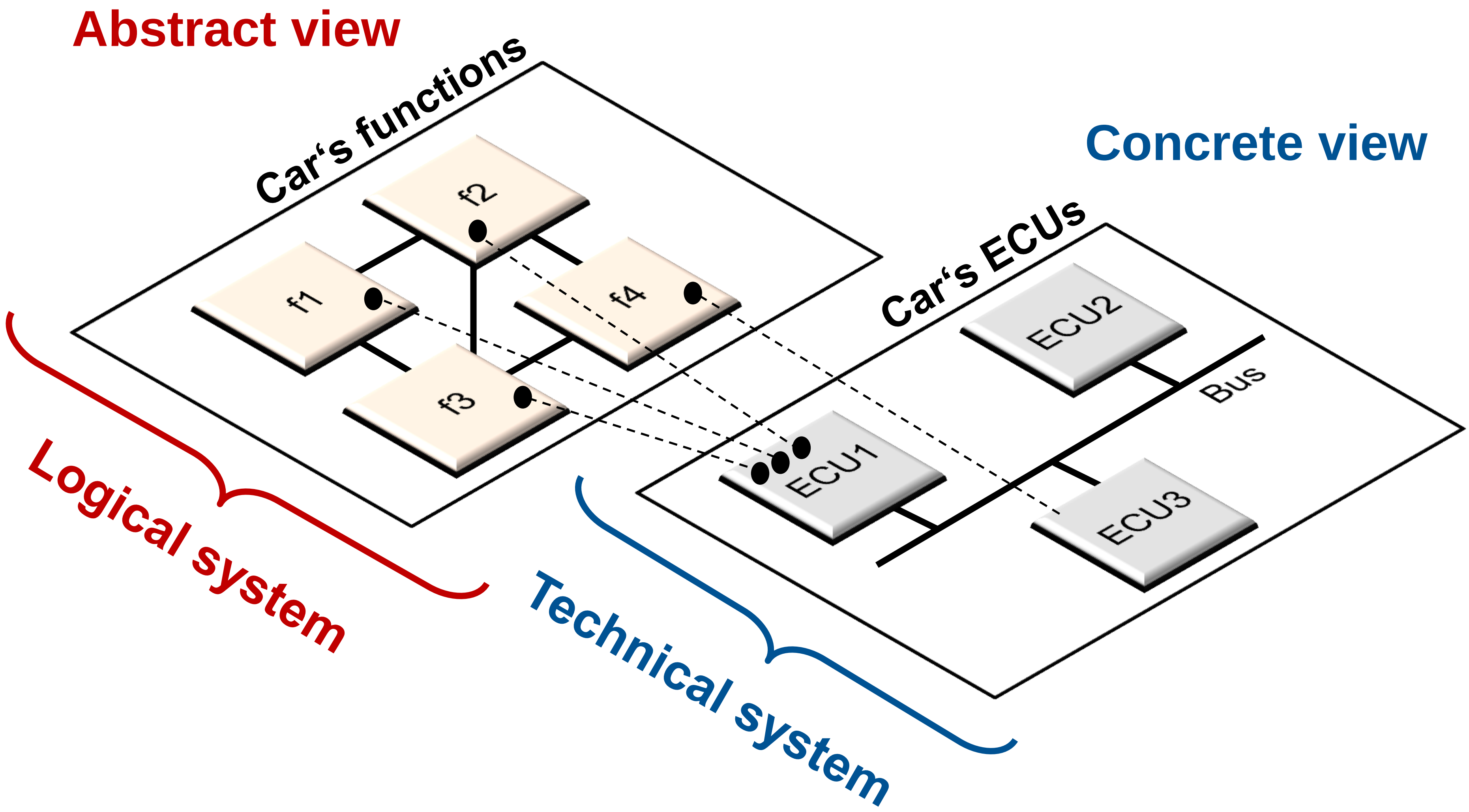


Product Life Cycle of a Car

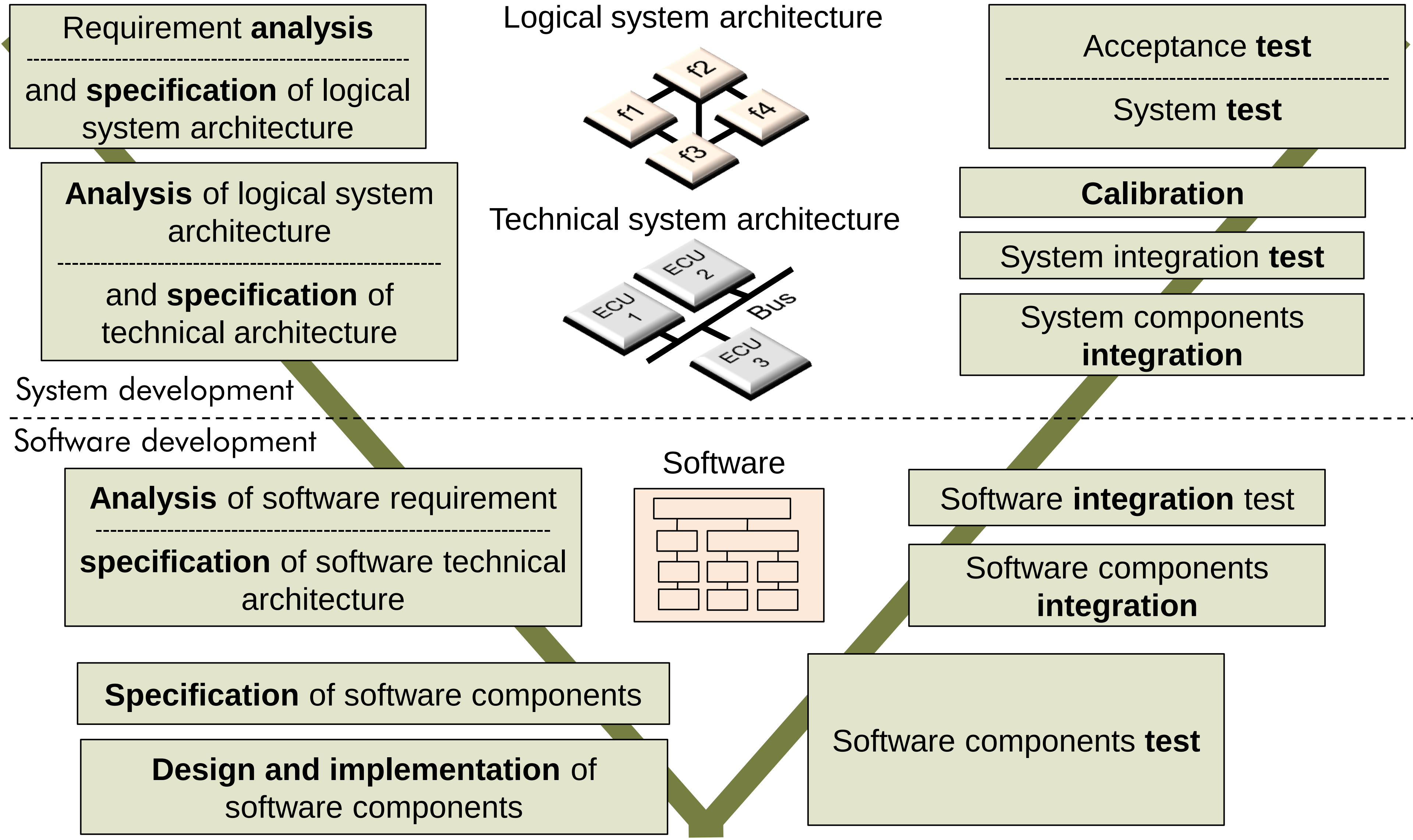


- Puts emphasis on the design
- Technology should be available
- Puts focus on software upgrades

Functions and their Implementation



Main Development Process



Requirement **analysis**

and **specification** of logical system architecture

Analysis of logical system architecture

and **specification** of technical architecture

System development

Software development

Analysis of software requirement

specification of software technical architecture

Specification of software components

Design and implementation of software components

Logical system architecture

Technical system architecture

Software

Acceptance **test**

System **test**

Calibration

System integration **test**

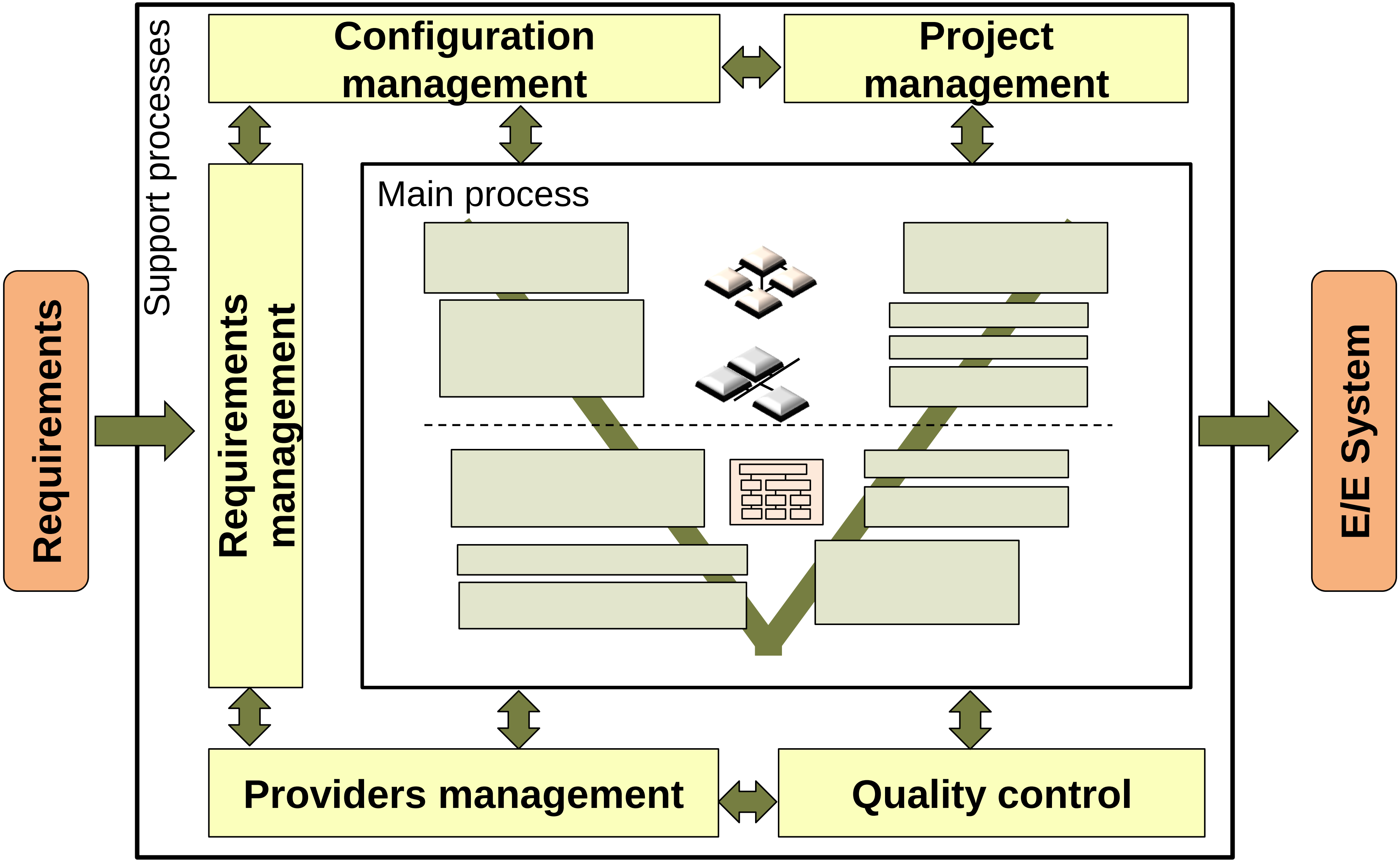
System components **integration**

Software **integration** test

Software components **integration**

Software components **test**

Support Development Process



Requirements

Support processes

Configuration management

Project management

Requirements management

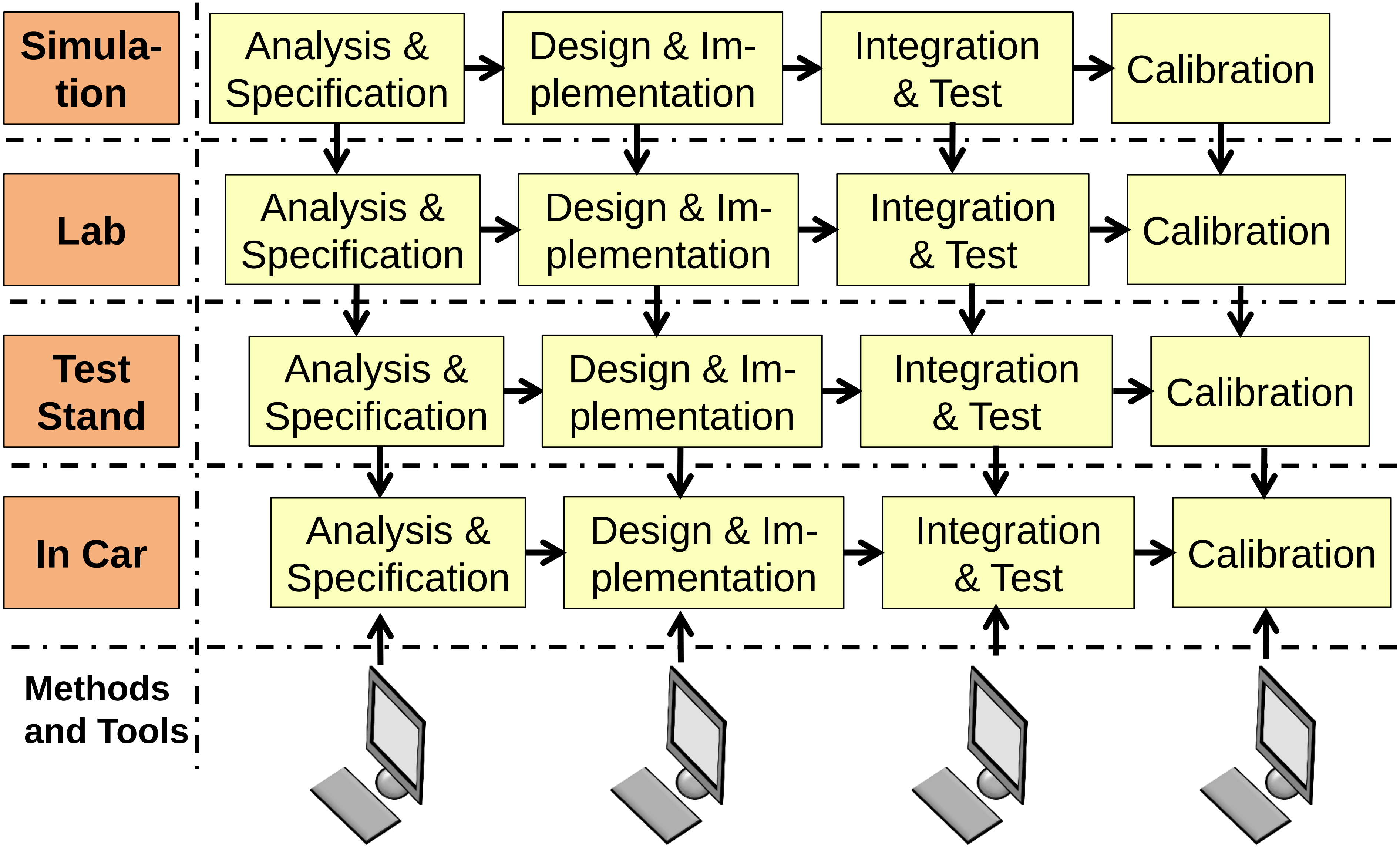
Main process

Providers management

Quality control

E/E System

Simultaneous Engineering



Simulation
Analysis & Specification → Design & Implementation → Integration & Test → Calibration

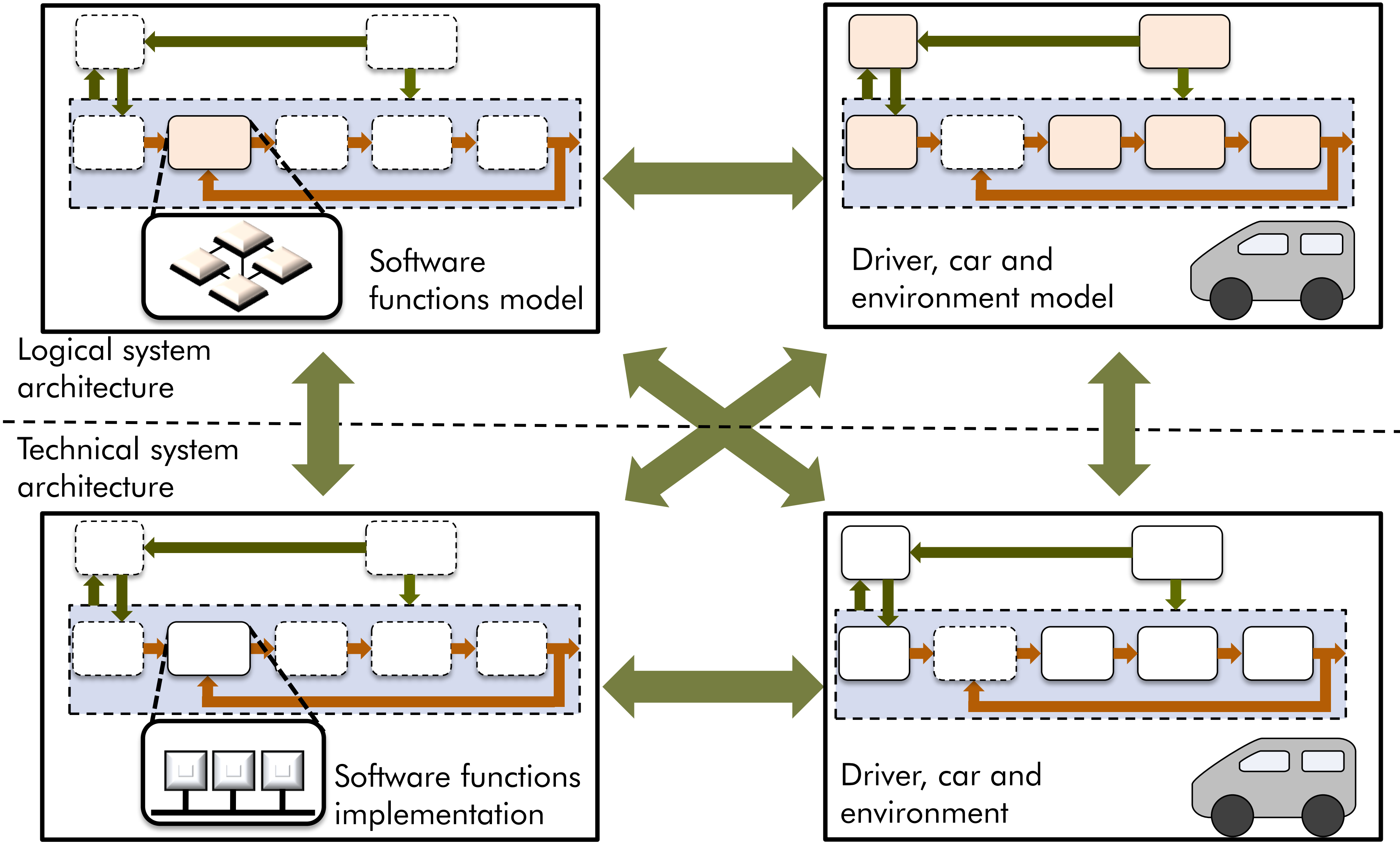
Lab
Analysis & Specification → Design & Implementation → Integration & Test → Calibration

Test Stand
Analysis & Specification → Design & Implementation → Integration & Test → Calibration

In Car
Analysis & Specification → Design & Implementation → Integration & Test → Calibration

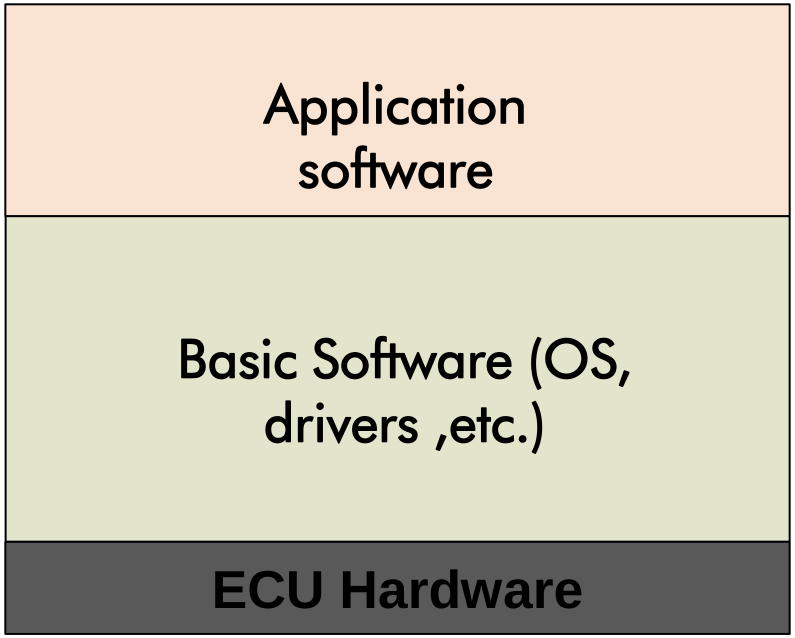
Methods and Tools

Model-Based Design & Development


Software functions model
Driver, car and environment model
Logical system architecture
Technical system architecture
Software functions implementation
Driver, car and environment

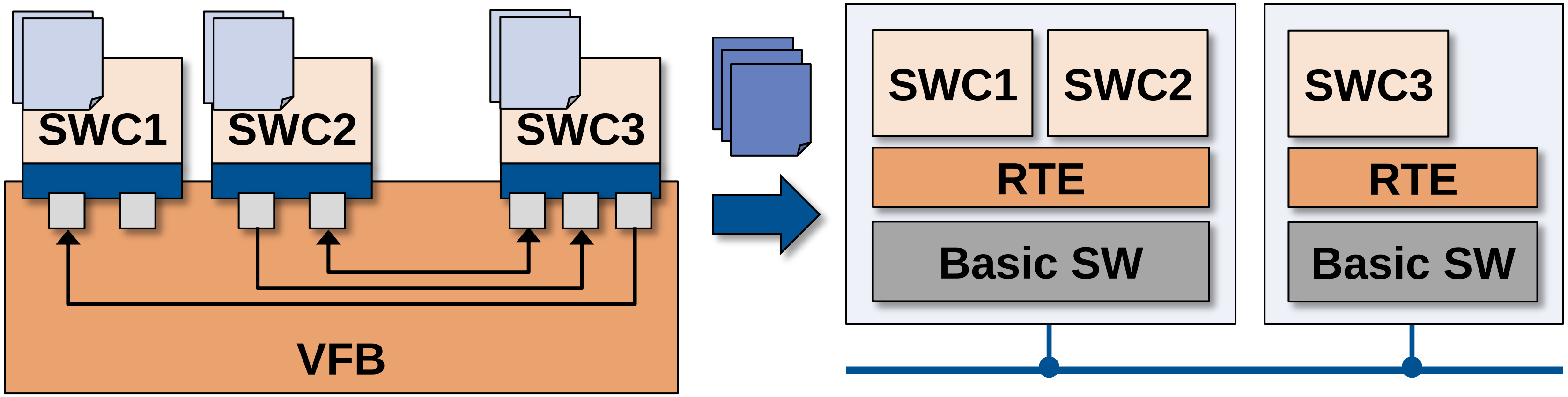
Traditional Software Architecture

- Application software written for a given platform
  - Given operating system (OS), e.g., OSEK-OS
  - Given architecture, e.g., PowerPC, ARM, etc.
  - It is not portable to other ECUs

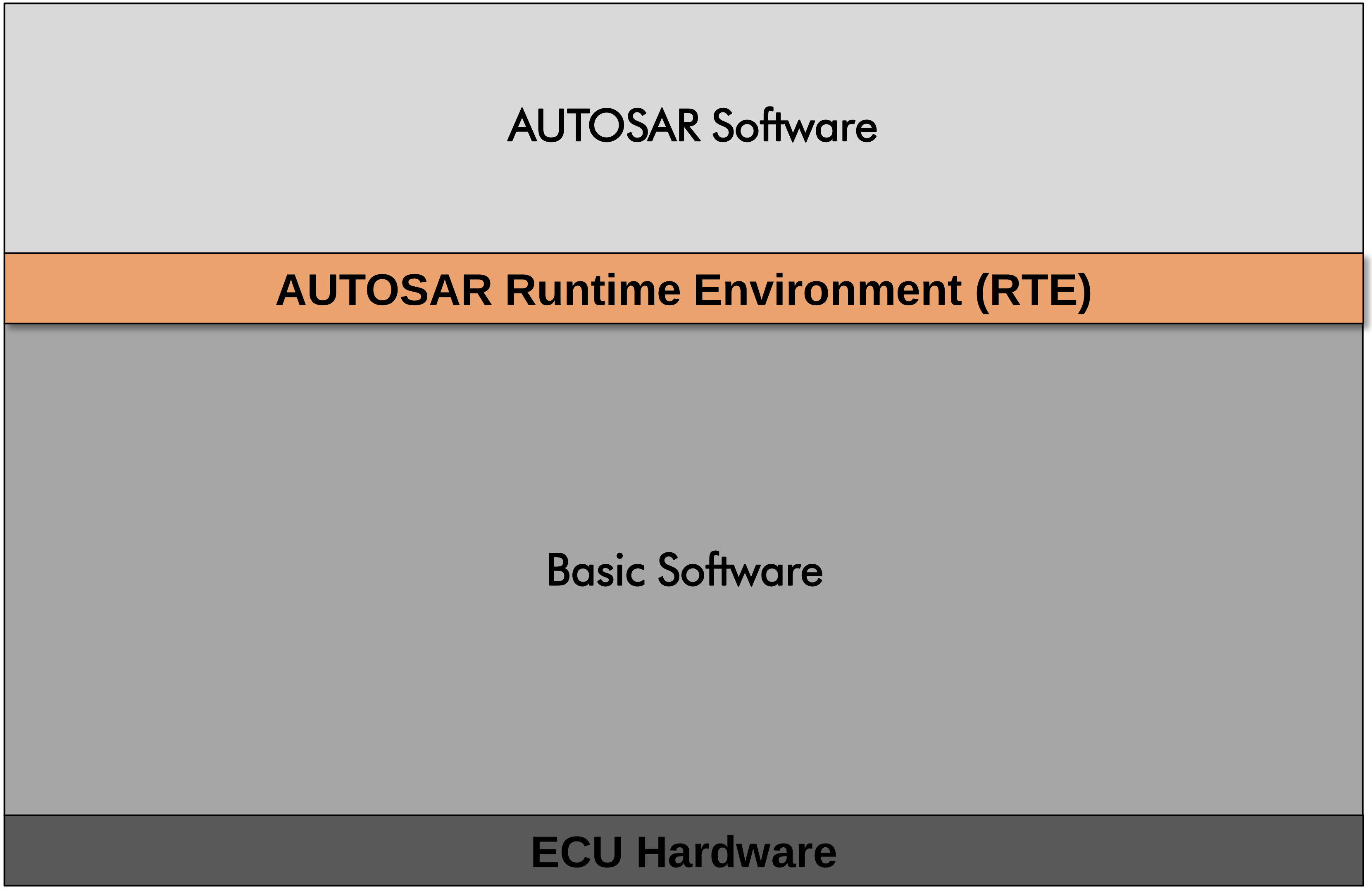


The AUTOSAR Standard

- AUTomotive Open System Architecture
  - Manage increasing E/E complexity
  - Improve flexibility and scalability
  - Improve quality and reliability of E/E systems
- Software Component (SWC)
- Virtual Functional Bus (VFB)



AUTOSAR Software Architecture



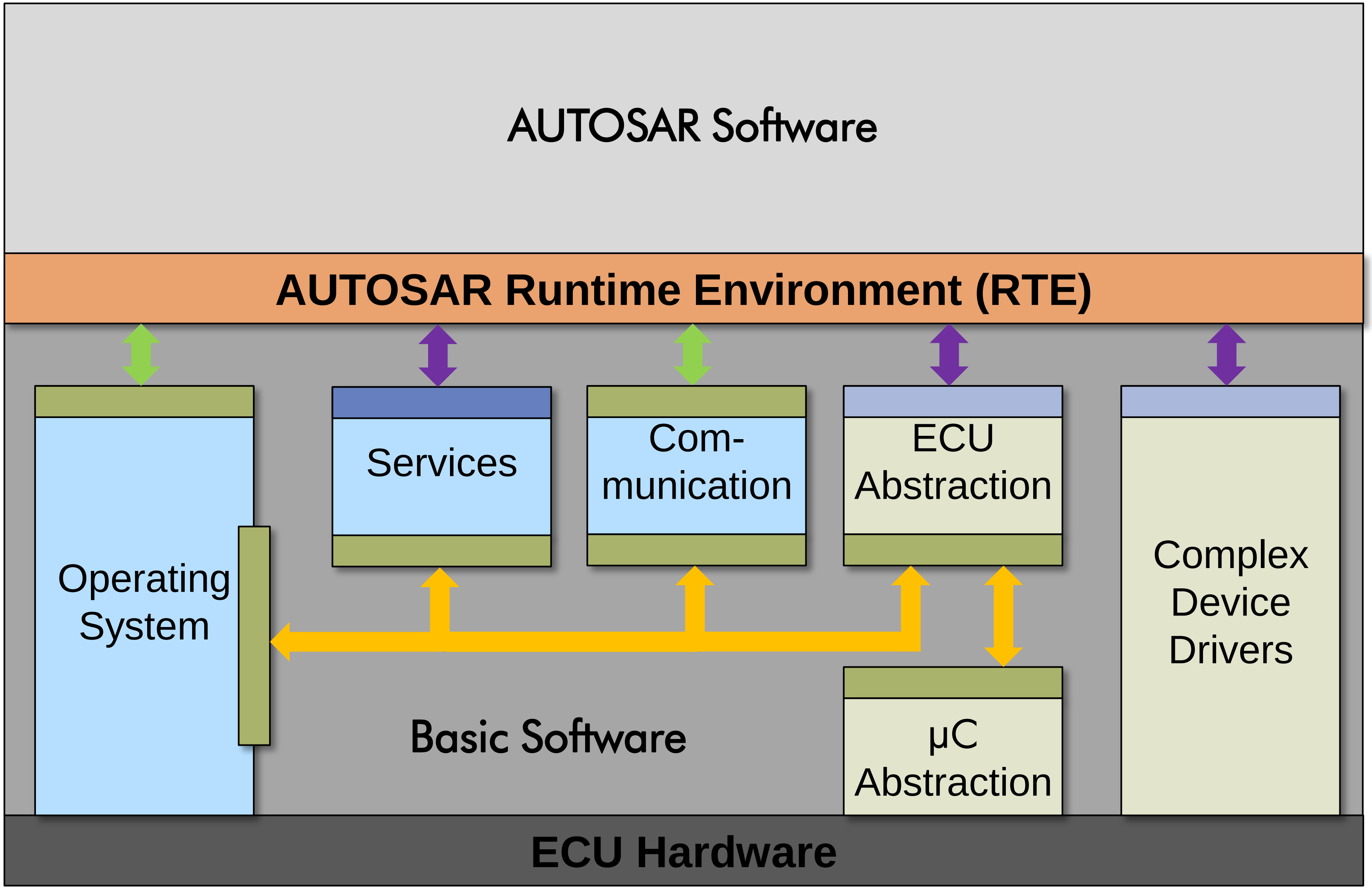
AUTOSAR Software

AUTOSAR Runtime Environment (RTE)

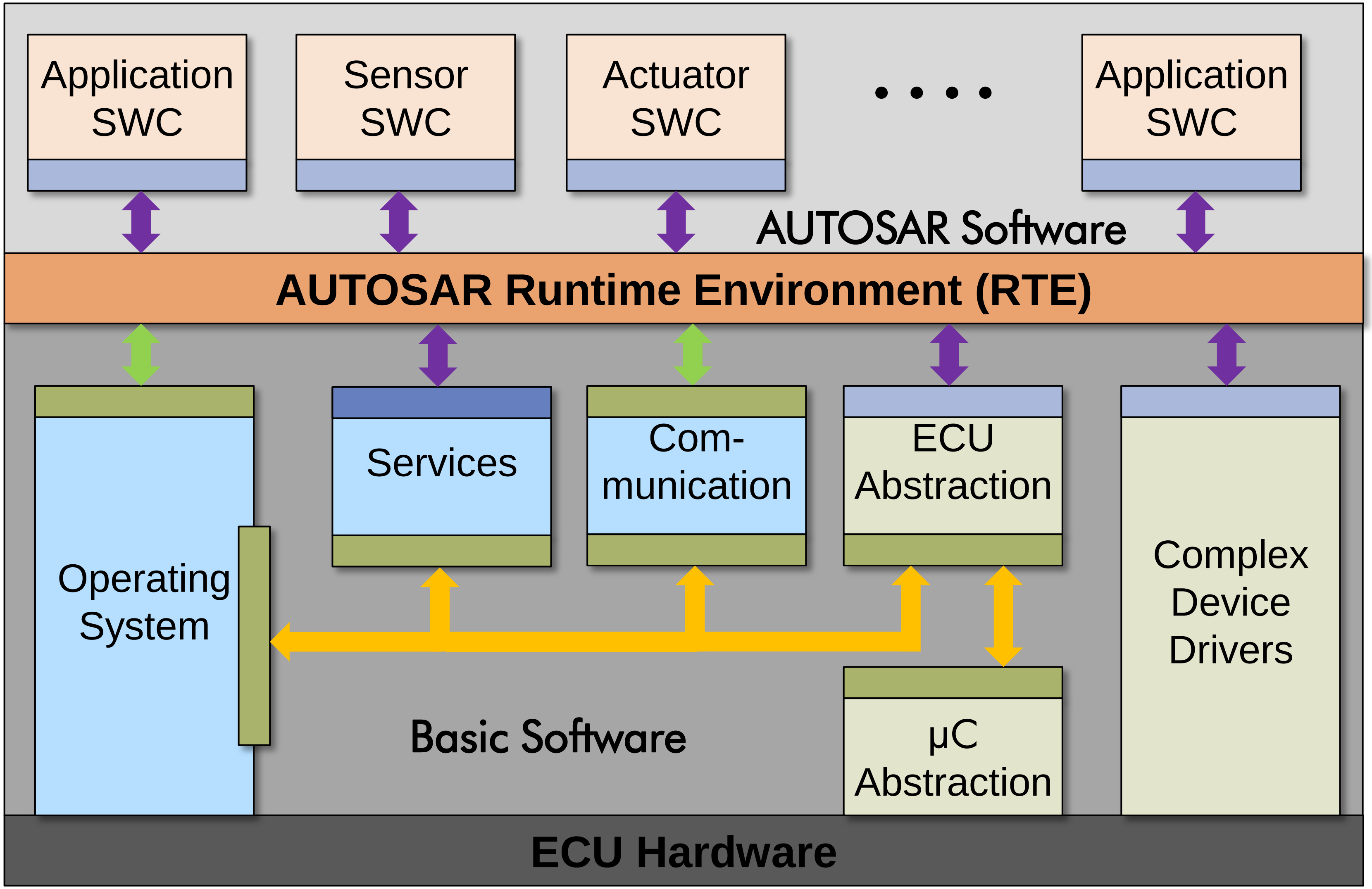
Basic Software

ECU Hardware

AUTOSAR Software Architecture



AUTOSAR Software Architecture



Summary

- E/E architectures are highly complex
  - Trend towards more functionality on fewer ECUs
  - Automotive Software need to be efficient and reliable
- Functional Domains
  - Vehicle centric: Power Train, Chassis and Safety
  - Passenger centric: Multimedia, Body/Comfort and MMI
- Long product life cycles of around 25 years
- Design and development techniques
- The AUTOSAR standard

Challenges in Automotive Systems

- Migration to multi-core architectures
  - Traditionally single core → big issue for software
- Migration from distributed to centralized systems
  - Need to reduce complexity → big issue for software
- Variant-oriented applications
  - Sport, comfort variants, etc. → big issue for software
- Car-to-X communication → big issue for software
- Electric vehicles, alternative power-train solutions
- Autonomous cars → very big issue for software

What did you learn about cars?

- How many processors are there in a modern car?
  - Around 100 different processors
- What is an ECU and an OEM?
  - ECU= Electronic Control Unit
  - OEM= Original Equipment Manufacturer
- How many kilometers of cable are there? → 4 Km
- How long is the product life cycle of a car?
  - Around 20 to 25 years
- How many lines of codes? → Around 100 Million

Organizational Issues

- **Lecture: every Thursday starting today**
  - **From 15:30 hs until 17:00 hs**
  - **This room: 1/201**
- **Different timings for tutorials (see OPAL)**
- **Slides and reading materials available over OPAL**
- **Accompanying book (available in the library):**
  - **Title: “Automotive Software Engineering”**
  - **By J. Schäuffele and T. Zurawka, 4th Edition**