AUTomotive Open System ARchitecture

Source:Google

ANCIT CONSULTING

ABOUT ANCIT

Next Generation Tooling and Engineering Service Provider for Automotive and Aerospace Industry

Product Development

Custom Tool Development

Consulting Services

Training Services

Speaker

Dr.B.Kaarthick,
Director-R&D, ANCIT
16+ years of experience in Research

Awarded as One of the Top 10 Innovators of India in 2016
By DST, Government of India

Website : www.ancitconsulting.com
Email : bk@ancitconsulting.com| Contact : +919787400605

ANCIT CONSULTING

Motive of the WEBINAR

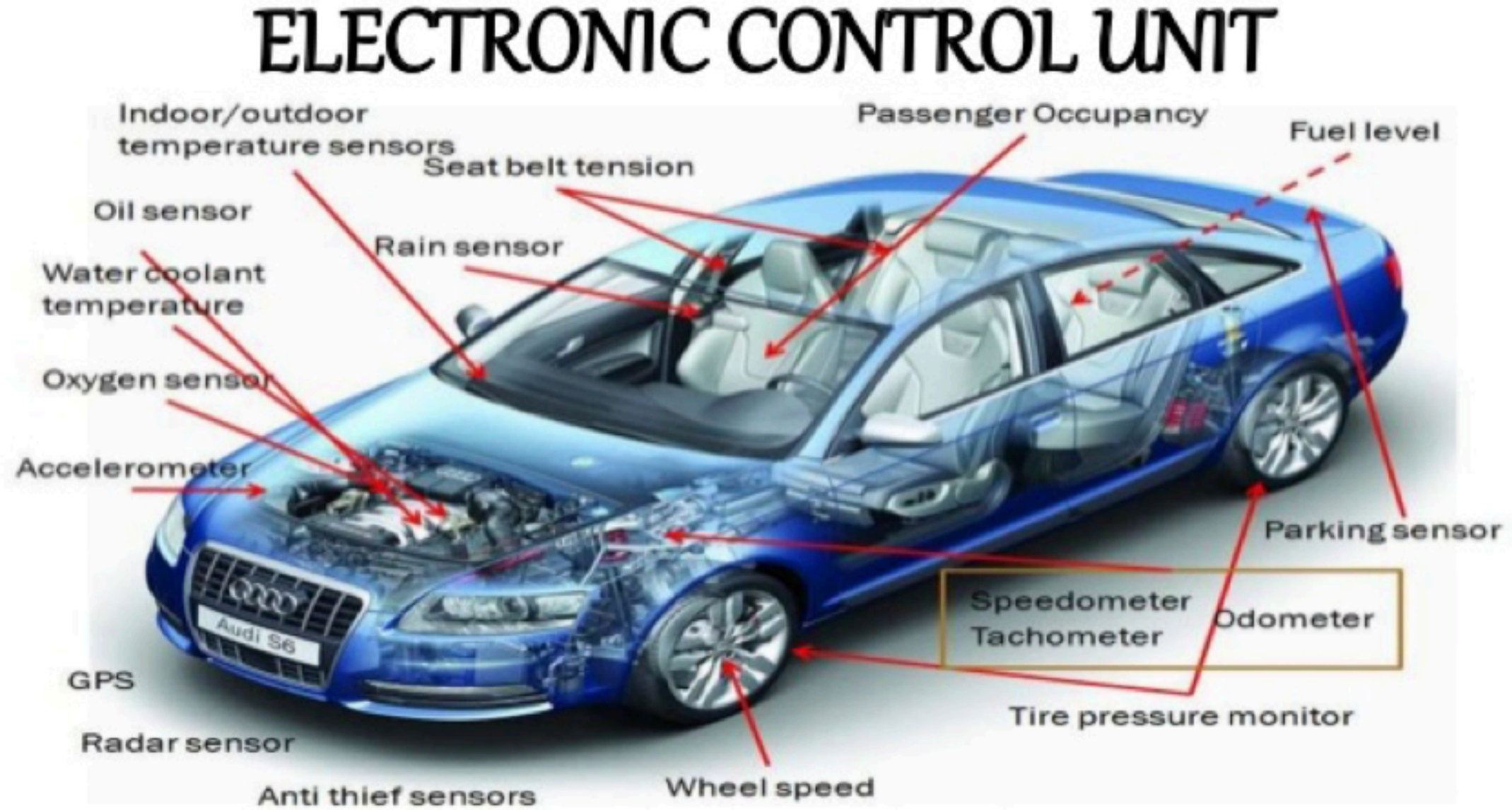
Agenda

- Why AUTOSAR?
- AUTOSAR Layered Architecture
- Design and Communication of AUTOSAR Software Components

Why AUTOSAR ?

Why AUTOSAR ?

.



Source:google

Why AUTOSAR?

Automotive Industry coping up with increasing complexity

- One of the biggest problems faced by the Automotive Companies was that the **software must often be rewritten from scratch when hardware is changed**.

- **BSW Standardisation:** AUTOSAR standardised the BSW system. This provides an opportunity for the software developers to now focus on customer features and functionalities, thereby increasing the competitive value. OEM will prefer to **pay only for Application Software** but **not for the BSW**.

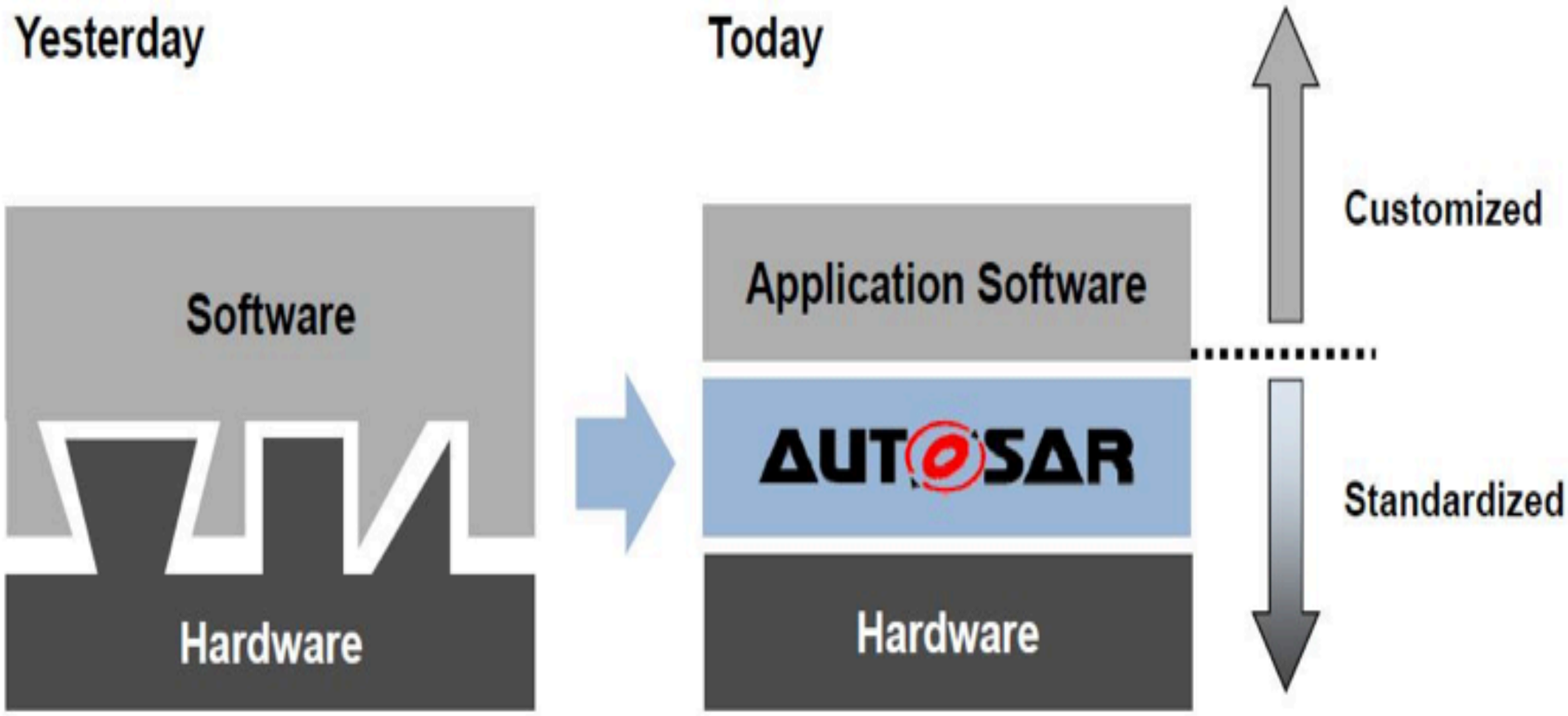
- **Hardware Abstraction:** AUTOSAR provides hardware abstraction i.e., AUTOSAR provides a software module called as Microcontroller Abstraction Layer **(MCAL)** that makes the Basic Software (BSW) and the application layer independent of the the Microcontroller. The **software developer can now focus on building the application than on worrying about configuring the micro controller**.

Why AUTOSAR?

- **Reusability of functions:** across vehicle networks and across OEM boundaries. One of the biggest challenges faced by the OEM was when an OEM wanted to **add a function to an existing ECU it required a lot of effort**. With the introduction of AUTOSAR this large effort when reusing functions has been reduced. Partitioning and relocation of functions has also been made possible with the introduction of AUTOSAR.

- **Standardization of exchange formats:** Before AUTOSAR every supplier to the OEM developed products in an ad-hoc mode. This created a lot a compatibility issues as OEM's work with different suppliers for different products. AUTOSAR is working on standardising the specification of exchange formats. This allows an opportunity for seamless integration among different products from different suppliers. **Interfaces has been standardised.**

Why AUTOSAR?



Organizations can focus on specializations. For instance, ANCIT can focus on building great **Airbag application software**. OEM can buy application from ANCIT, **BSW** from BOSCH and **microcontroller** from Infineon.

AUTOSAR Layered Architecture

ANCIT CONSULTING

APPLICATION LAYER

RUN TIME ENVIRONMENT

BASIC SOFTWARE

SERVICES LAYER

ECU ABSTRACTION LAYER

MICROCONTROLLER ABSTRACTION LAYER

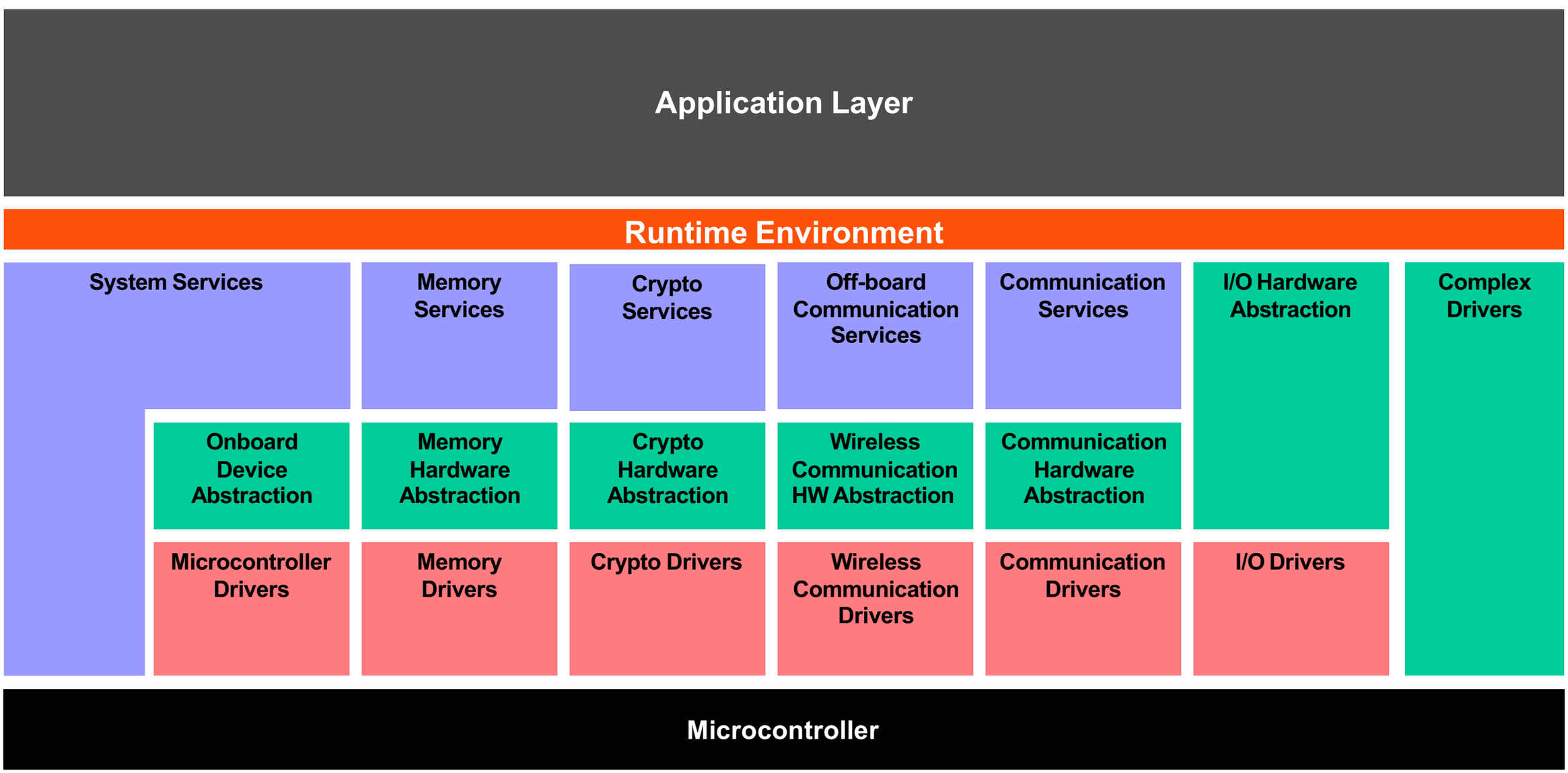
COMPLEX DRIVERS

MICROCONTROLLER

AUTOSAR Layered Architecture

Basic Software

The Basic Software Layers are further divided into functional groups. Examples of Services are System, Memory and Communication Services.



AUTOSAR Basic Software Module

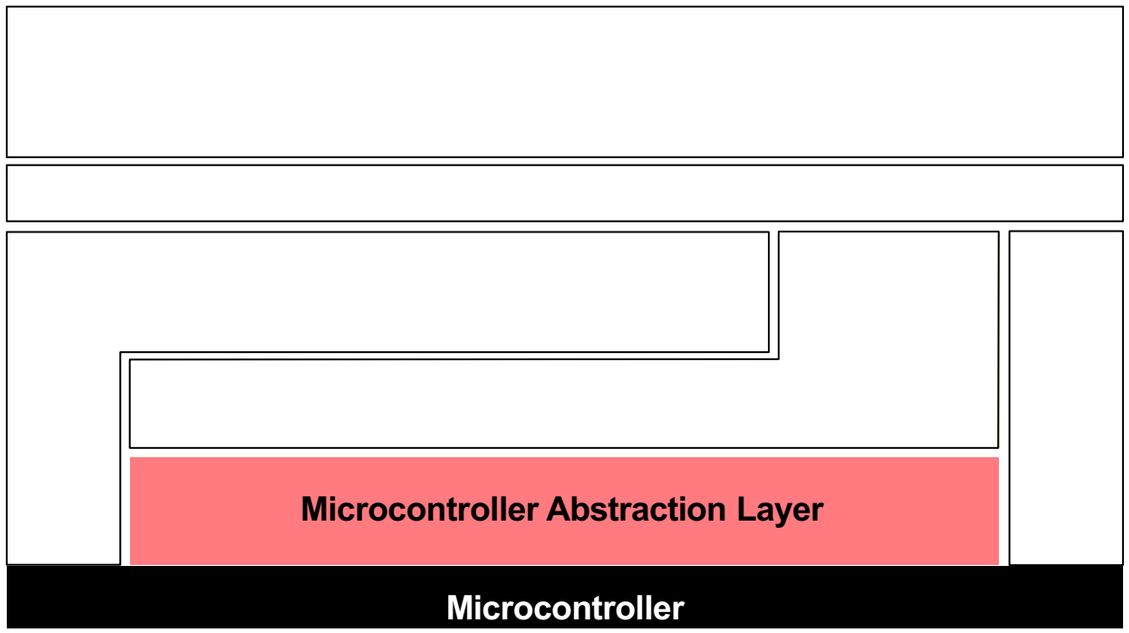
AUTOSAR has defined a set of BSW modules. They are responsible for different tasks:

- Operating System
- Access to non volatile memory
- Communication via CAN, LIN, FlexRay and Ethernet
- Handling the diagnostics
- Access to I/O ports
- System services like ECU state management

In addition, so-called Complex Device Drivers can be integrated into an AUTOSAR ECU. They are used to access the features of the ECU, which are not covered by the standard BSW of AUTOSAR.

Microcontroller Abstraction Layer

- The **Microcontroller Abstraction Layer** is the lowest software layer of the Basic Software.
- It contains internal drivers, which are software modules with direct access to the µC and internal peripherals.



Task

Make higher software layers independent of µC

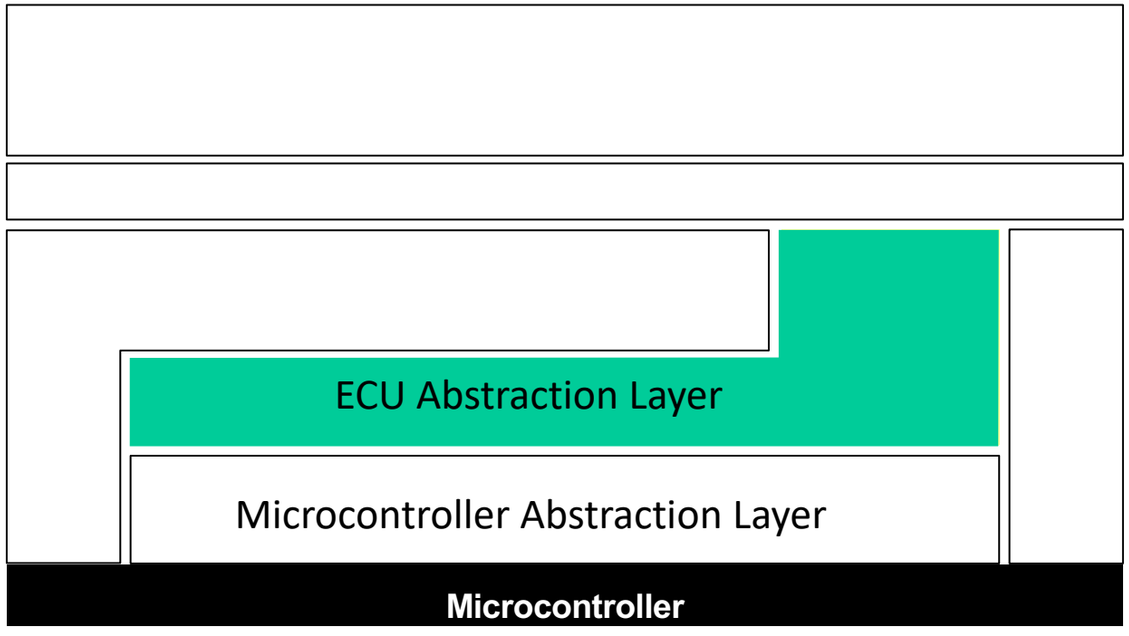
Properties

Implementation: µC dependent

Upper Interface: standardized and µC independent

ECU Abstraction Layer

- The **ECU Abstraction Layer** interfaces the drivers of the Microcontroller Abstraction Layer. It also contains drivers for external devices.
- It offers an API for access to peripherals and devices regardless of their location (µC internal/external) and their connection to the µC (port pins, type of interface)



Task

Make higher software layers independent of ECU hardware layout

Properties

Implementation: µC independent, ECU hardware dependent

Upper Interface: µC and ECU hardware independent

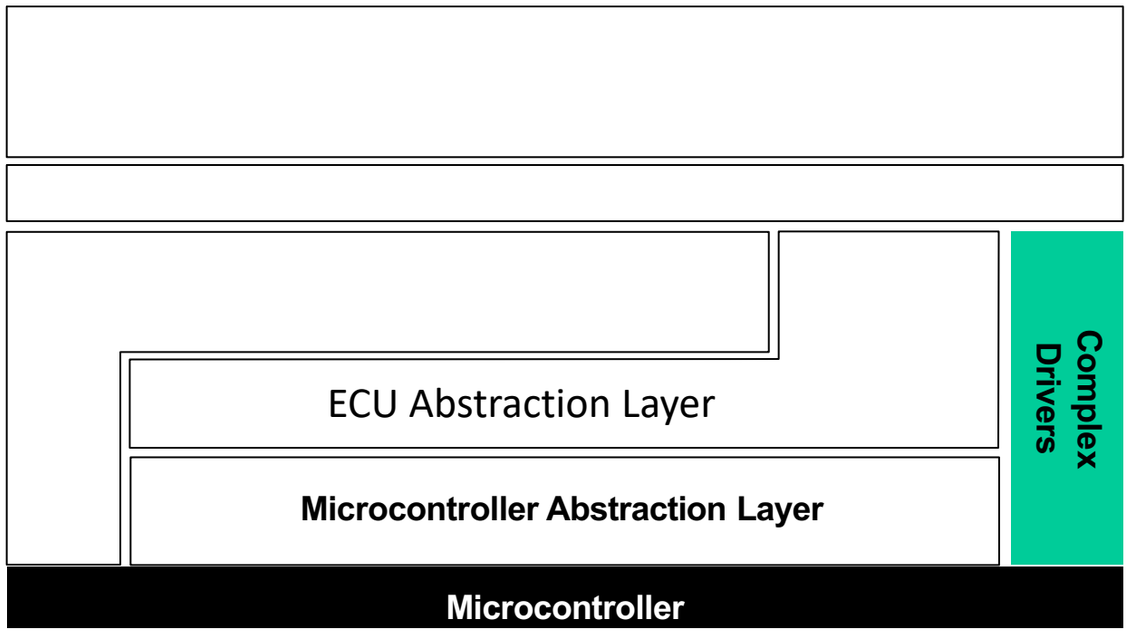
Complex drivers Layer

The **Complex Drivers Layer** spans from the hardware to the RTE.

Task

Provide the possibility to integrate special purpose functionality, e.g. drivers for devices:

- which are not specified within AUTOSAR,
- with very high timing constrains or
- for migration purposes etc.



Properties

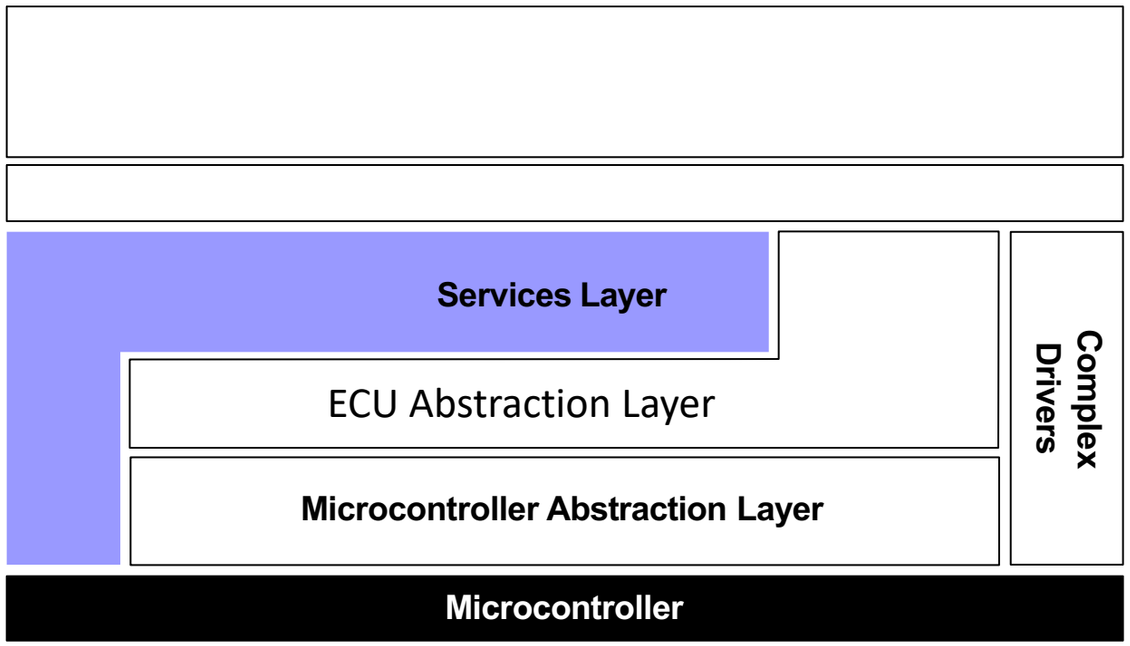
Implementation: might be application, µC and ECU hardware dependent

Upper Interface: might be application, µC and ECU hardware dependent

ECU Abstraction Layer

The **Services Layer** is the highest layer of the Basic Software which also applies for its relevance for the application software: while access to I/O signals is covered by the ECU Abstraction Layer, the Services Layer offers:

- Operating system functionality
- Vehicle network communication and management services
- Memory services (NVRAM management)
- Diagnostic Services (including UDS communication, error memory and fault treatment)
- ECU state management, mode management
- Logical and temporal program flow monitoring (Wdg manager)



Task

Provide basic services for applications, RTE and basic software modules.

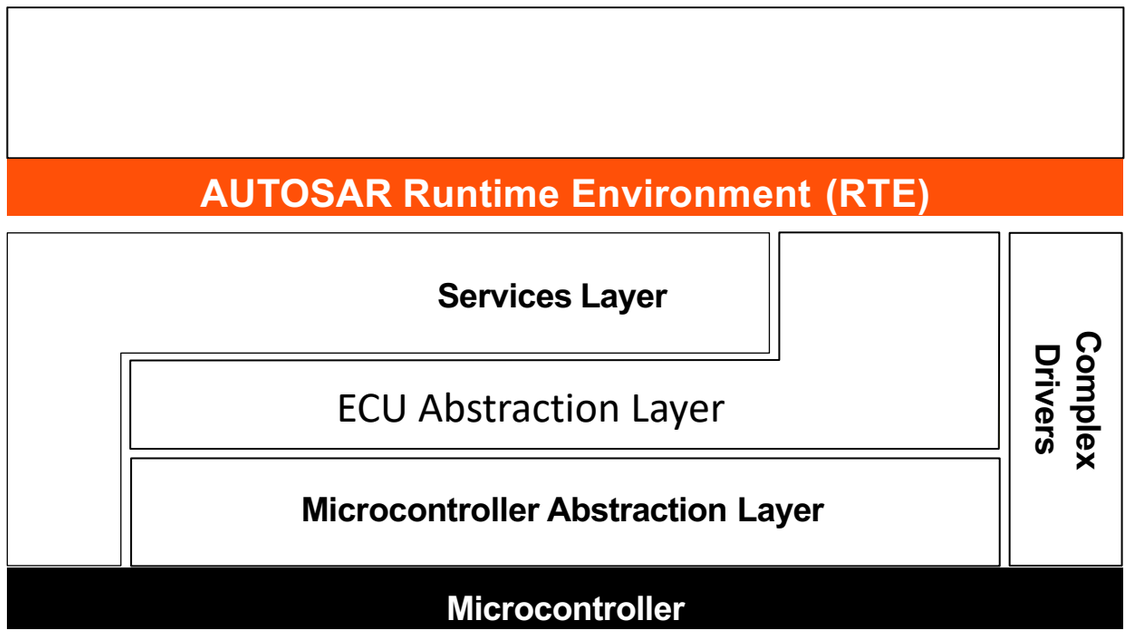
Properties

Implementation: mostly µC and ECU hardware independent

Upper Interface: µC and ECU hardware independent

RTE Layer

- The **RTE** is a layer providing communication services to the application software (AUTOSAR Software Components and/or AUTOSAR Sensor/Actuator components).
- Above the RTE the software architecture style changes from “layered“ to “component style“.
- The AUTOSAR Software Components communicate with other components (inter and/or intra ECU) and/or services via the RTE.



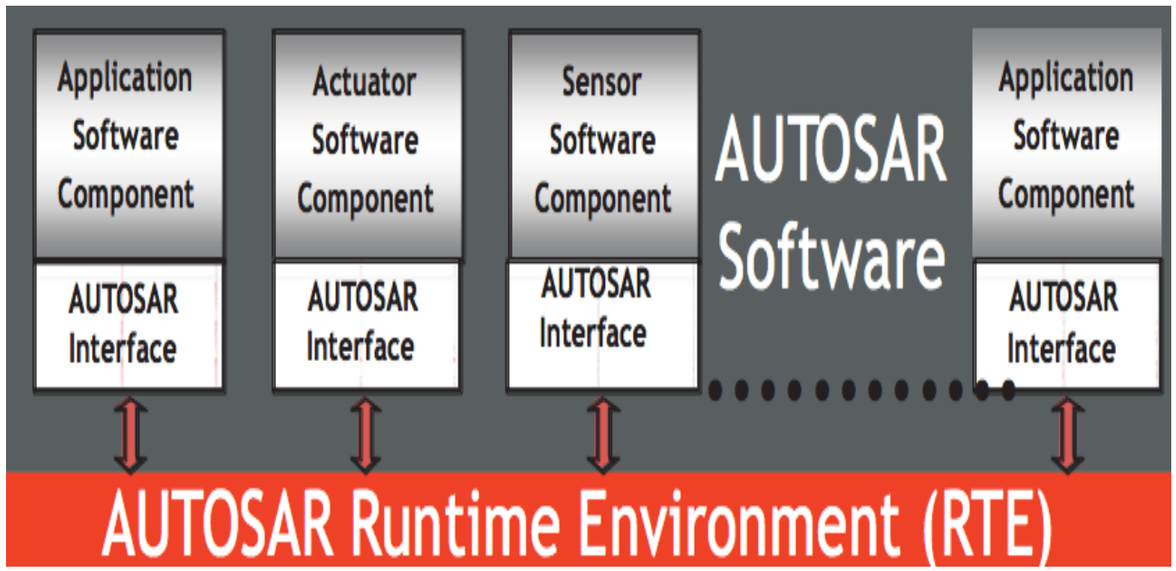
Task

Make AUTOSAR Software Components independent from the mapping to a specific ECU.

Properties

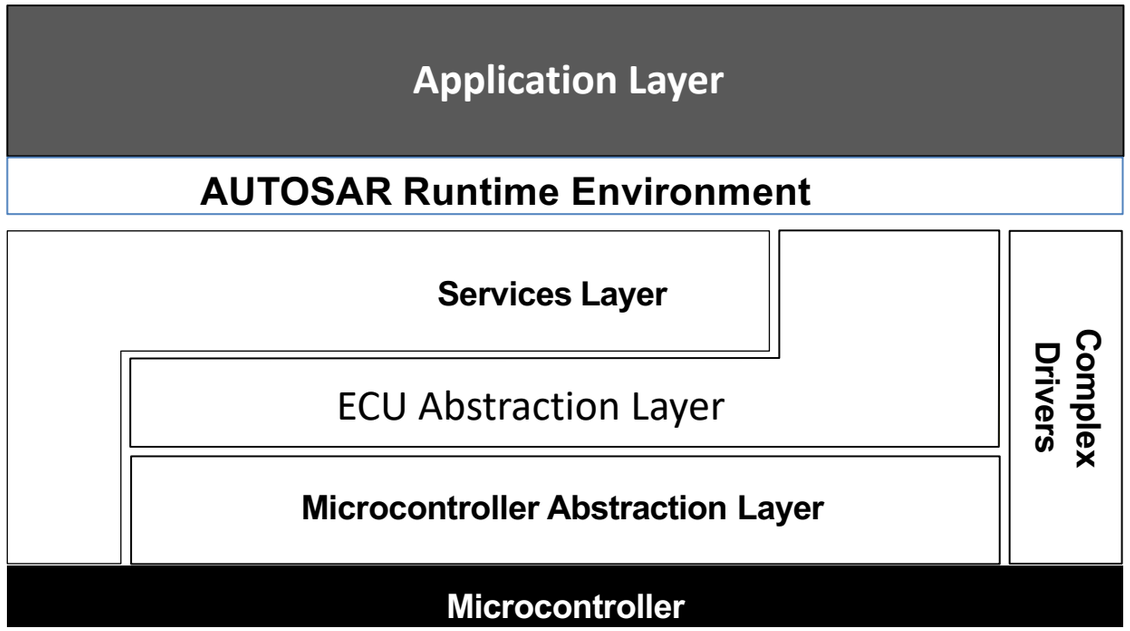
Implementation: ECU and application specific (generated individually for each ECU)

Upper Interface: completely ECU independent

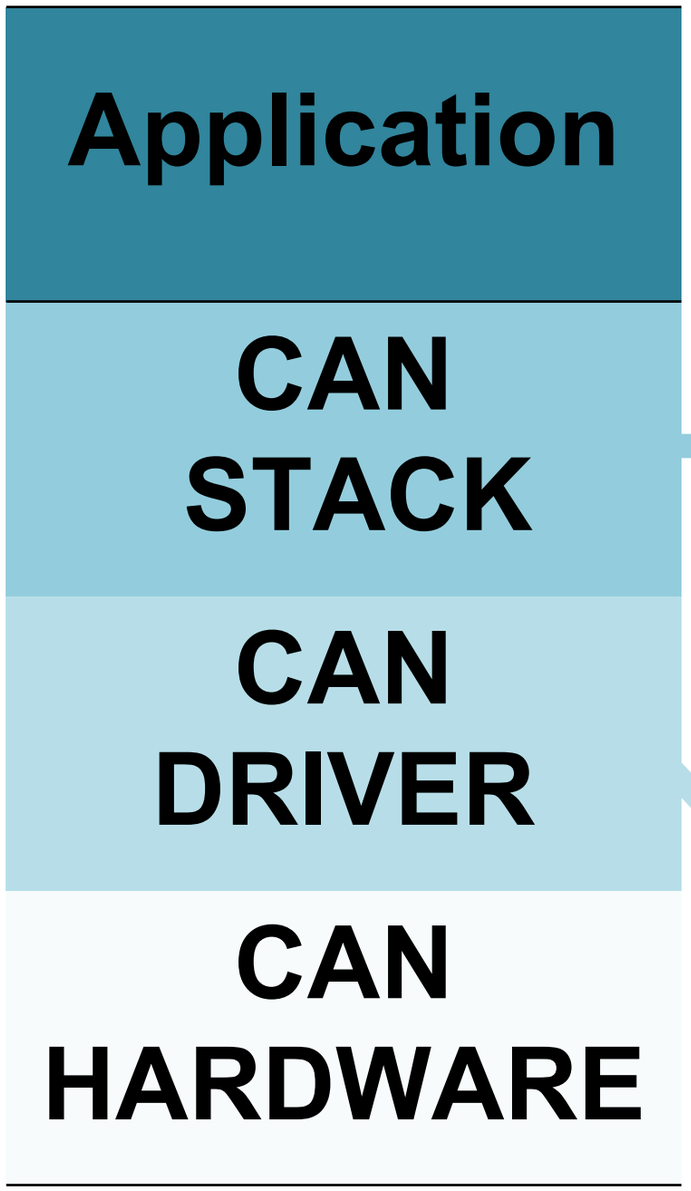


Application Layer and RTE

The **Application Layer** contains the SWCs, which realize the application functionality of the ECU. This is the piece of software that the OEM is ready to pay . Sometimes the OEM build some of the applications to protect their IP.

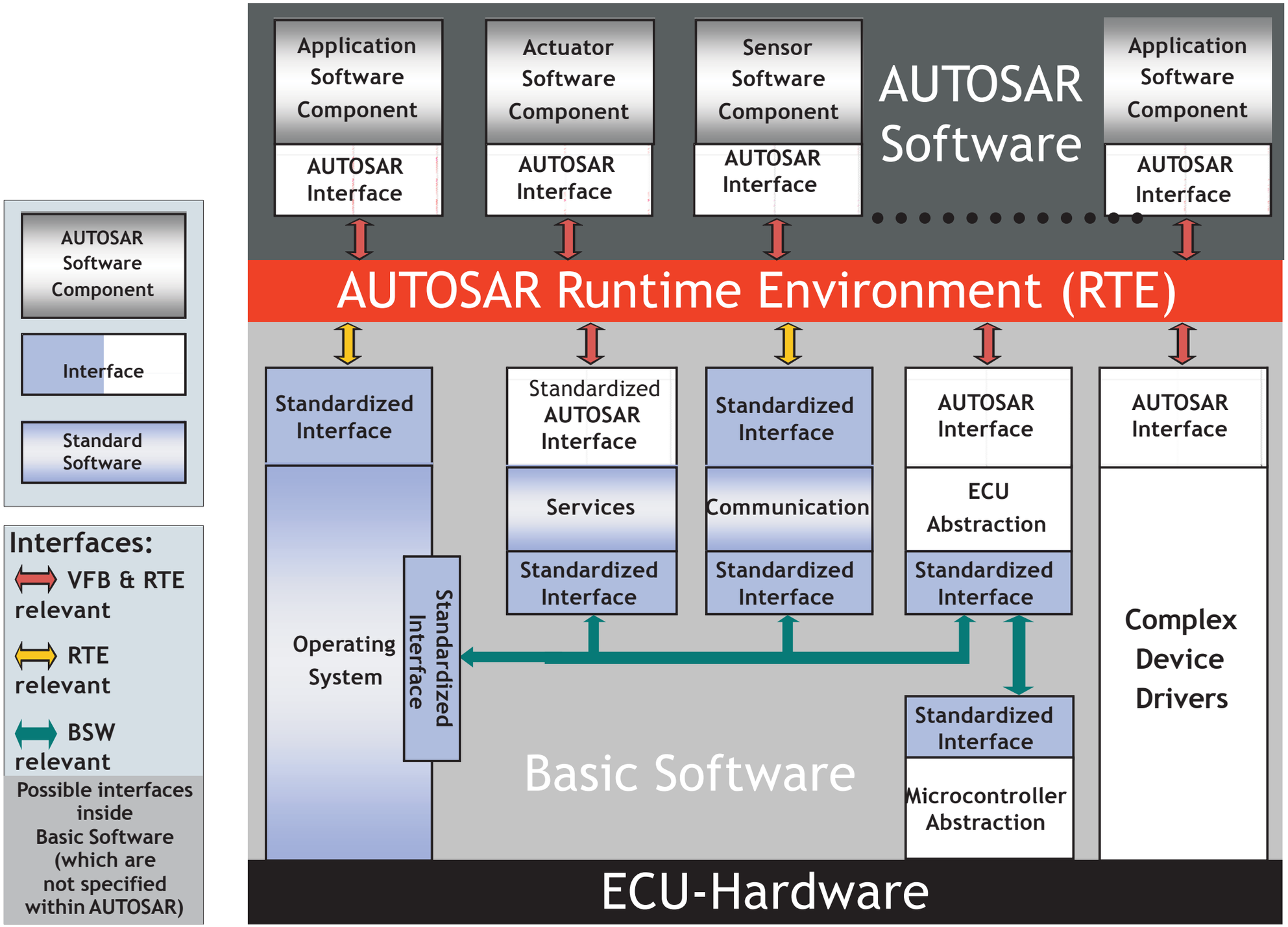


CAN STACK Architecture

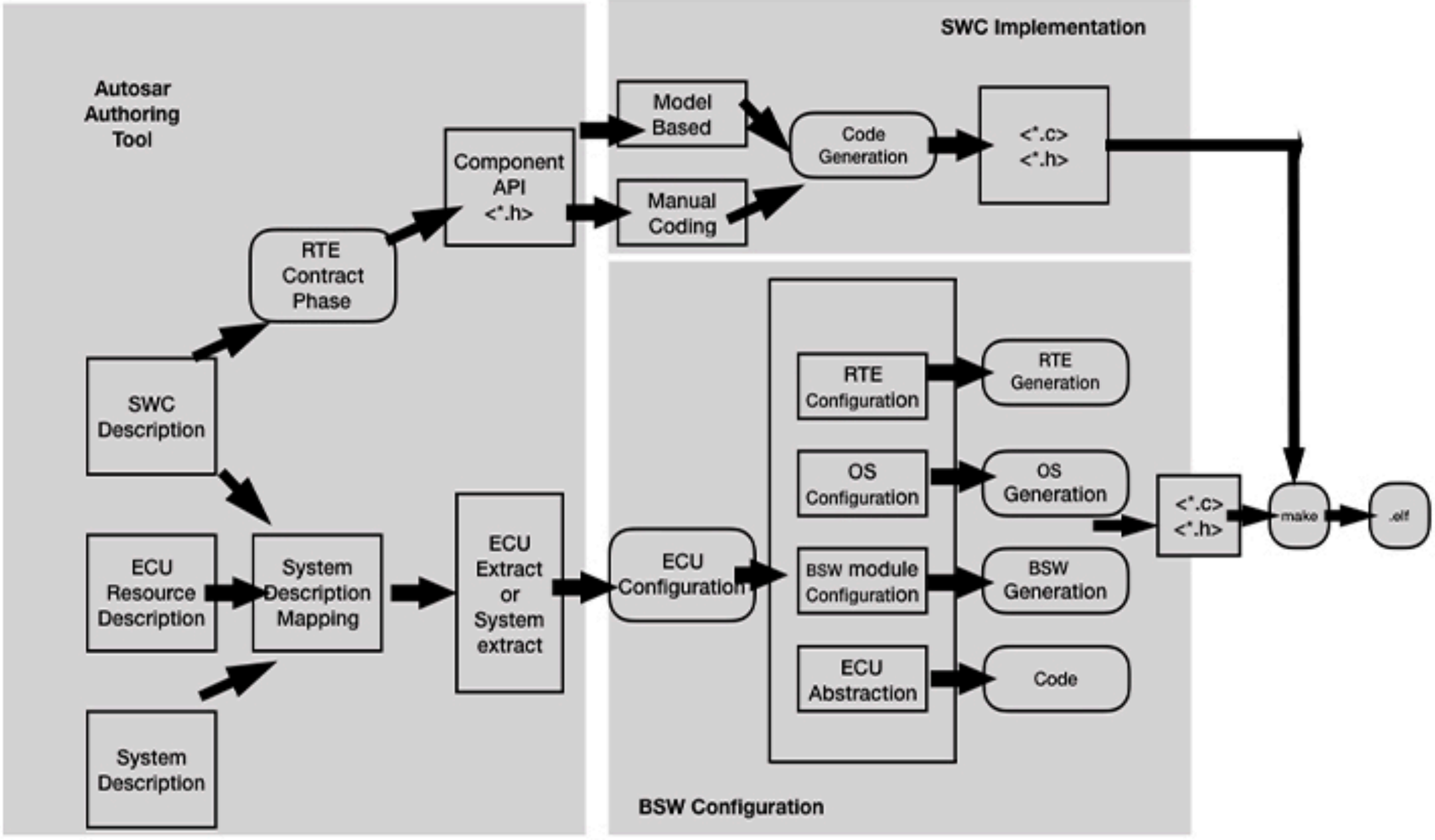


- manages sleep/wake up functionality.
- CAN Bus Interface Layer carries out all hardware tasks, related to the flow of data to the upper layer modules of the CAN software stack.
- UDS Software Stack to manages the diagnostics
- Flash Bootloader Application for ECU Reprogramming..
- segmenting the data, if the size of the data frame is more than 8 Bytes.

The CAN driver enables the access to hardware resources, for the upper software layers. It also offers access to the hardware API, to the application layer.



Simplified Component View

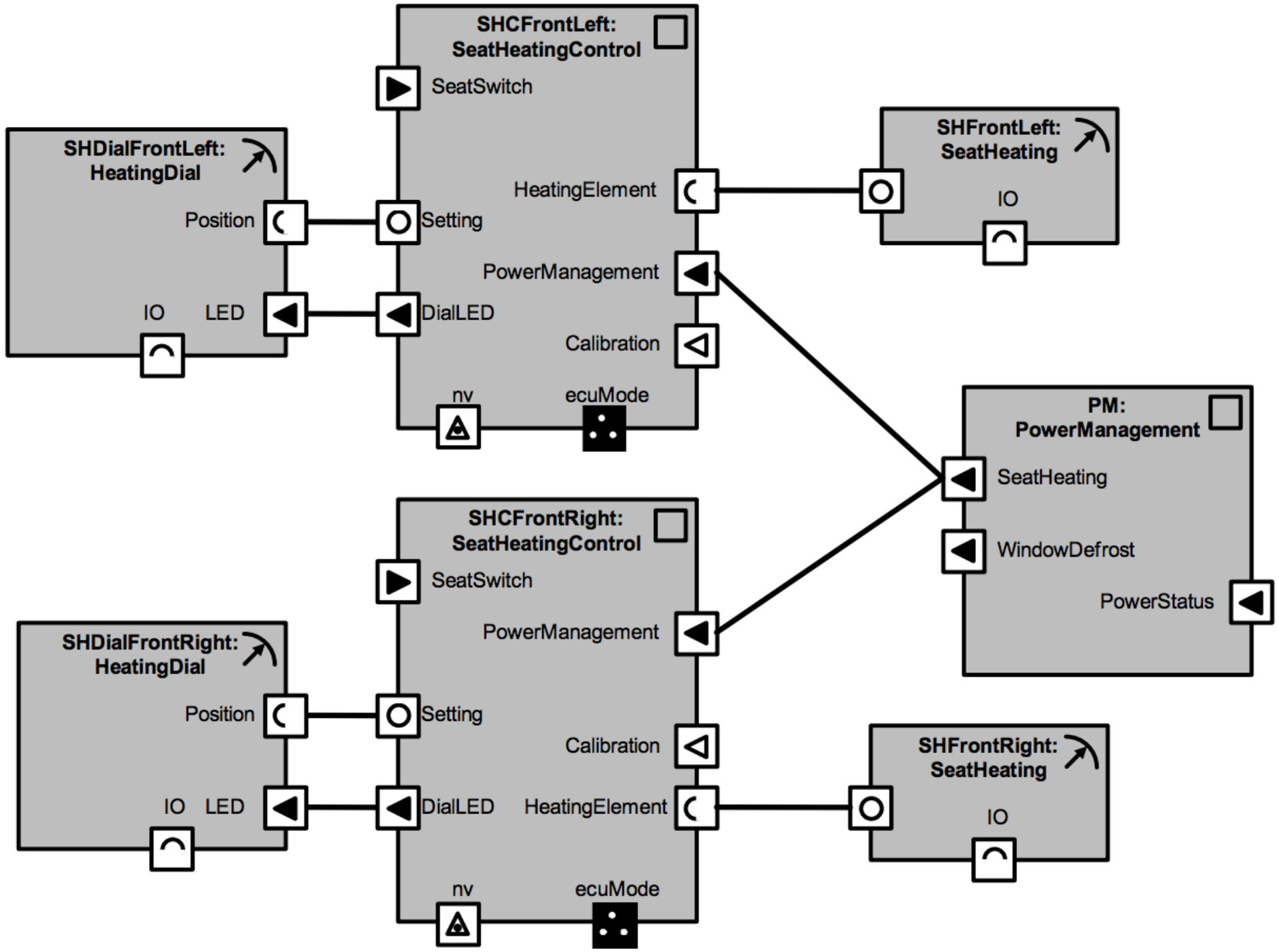


AUTOSAR Methodology

Design and Communication of Software Components

ANCIT CONSULTING

SeatHeatingControl



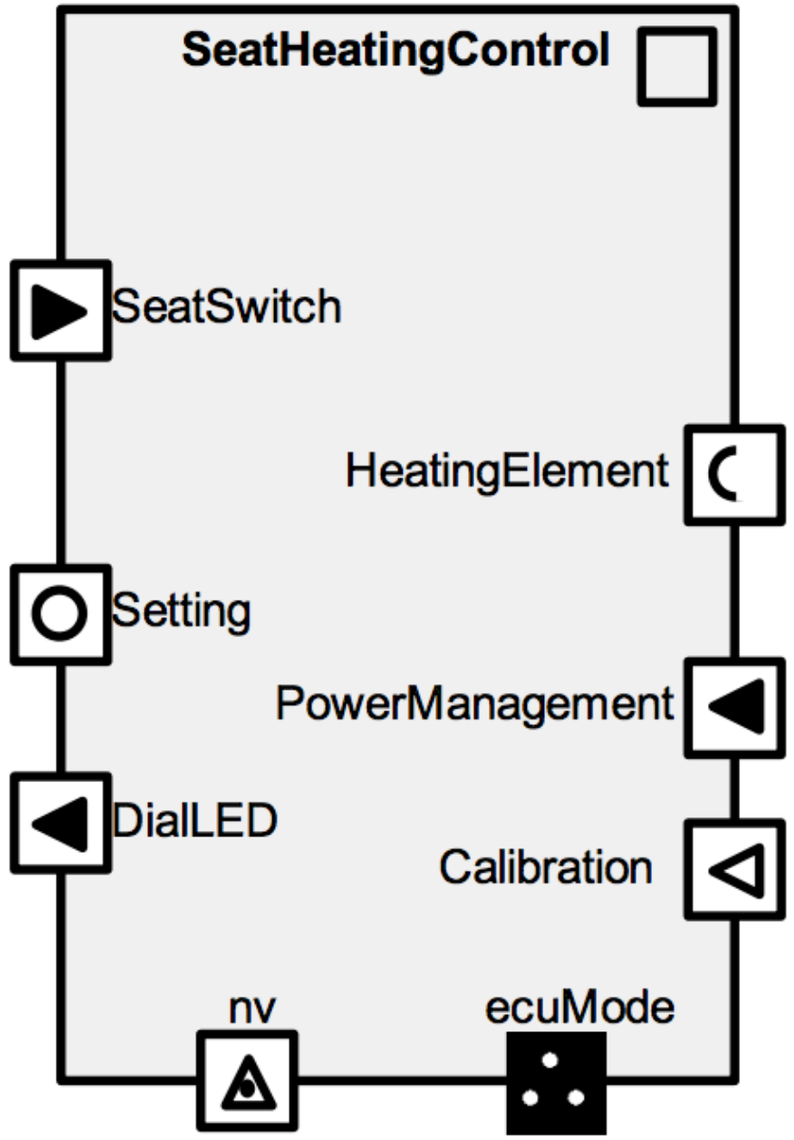
SeatHeatingControl with seven software components

SeatHeatingControl

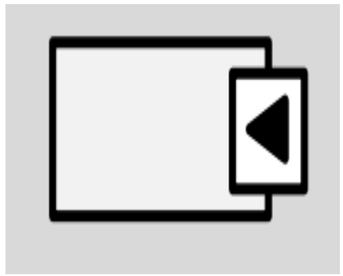
- SeatHeatingControl

- Input:
  - Whether a Passenger is sitting on the seat ”**SeatSwitch**”
  - Setting of the seat temperature dial “**Setting**”
  - Some information from a central power management system “**Power Management**”

- Output:
  - DialLED associated with the seat temperature “**DialLED**”
  - Heating Element to be Turned ON “**HeatingElement**”

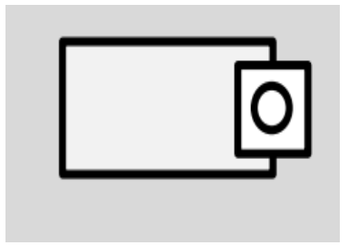


Sender-Receiver RPORT

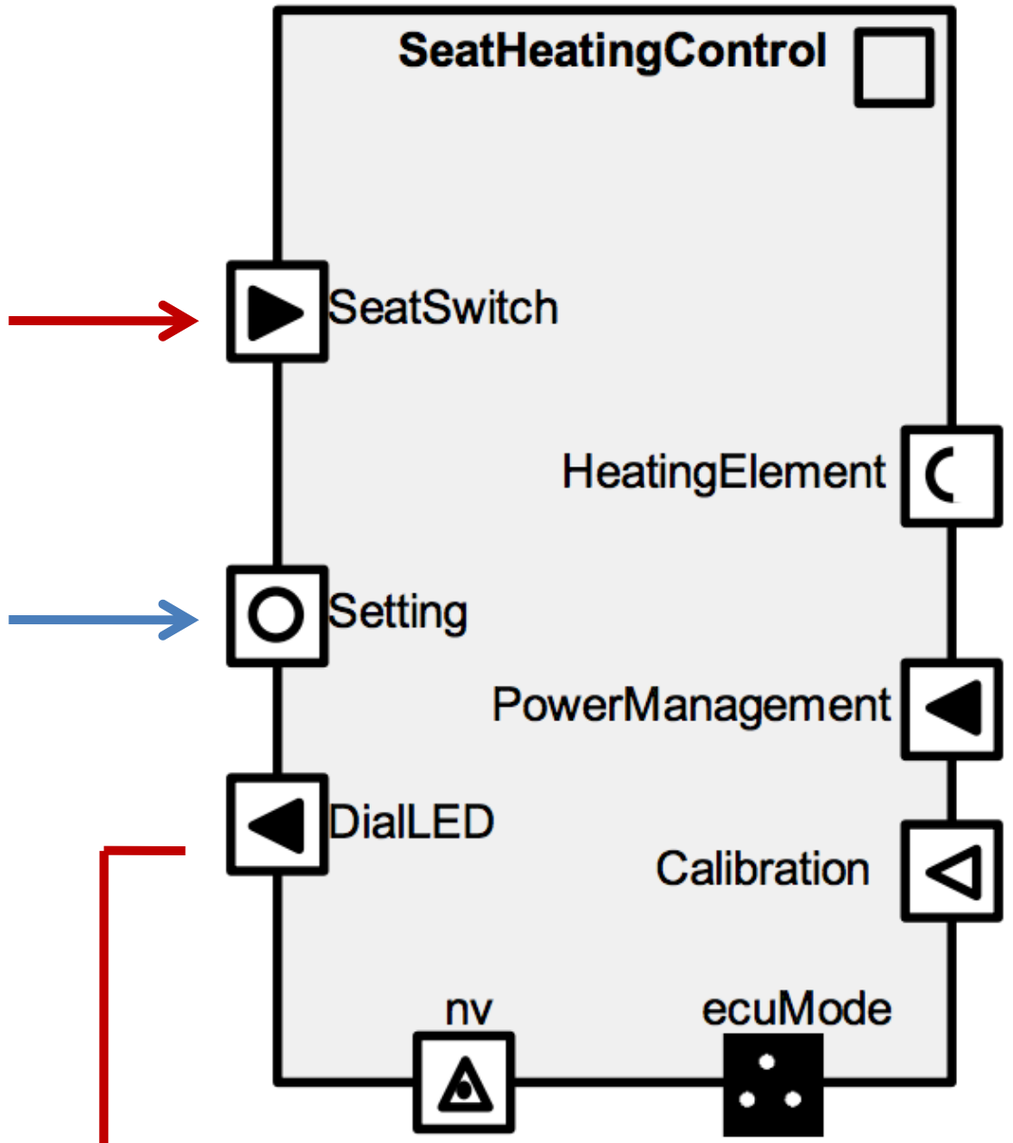


The Component reads/consumes values of data-elements

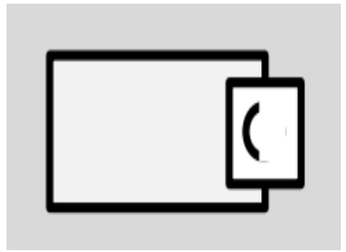
Client - Server PPORT



The Component provides operation defined in the interface

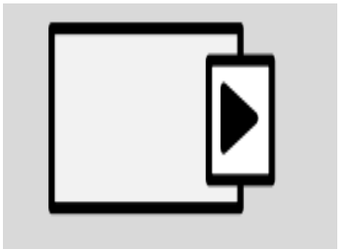


Client – Server RPORT

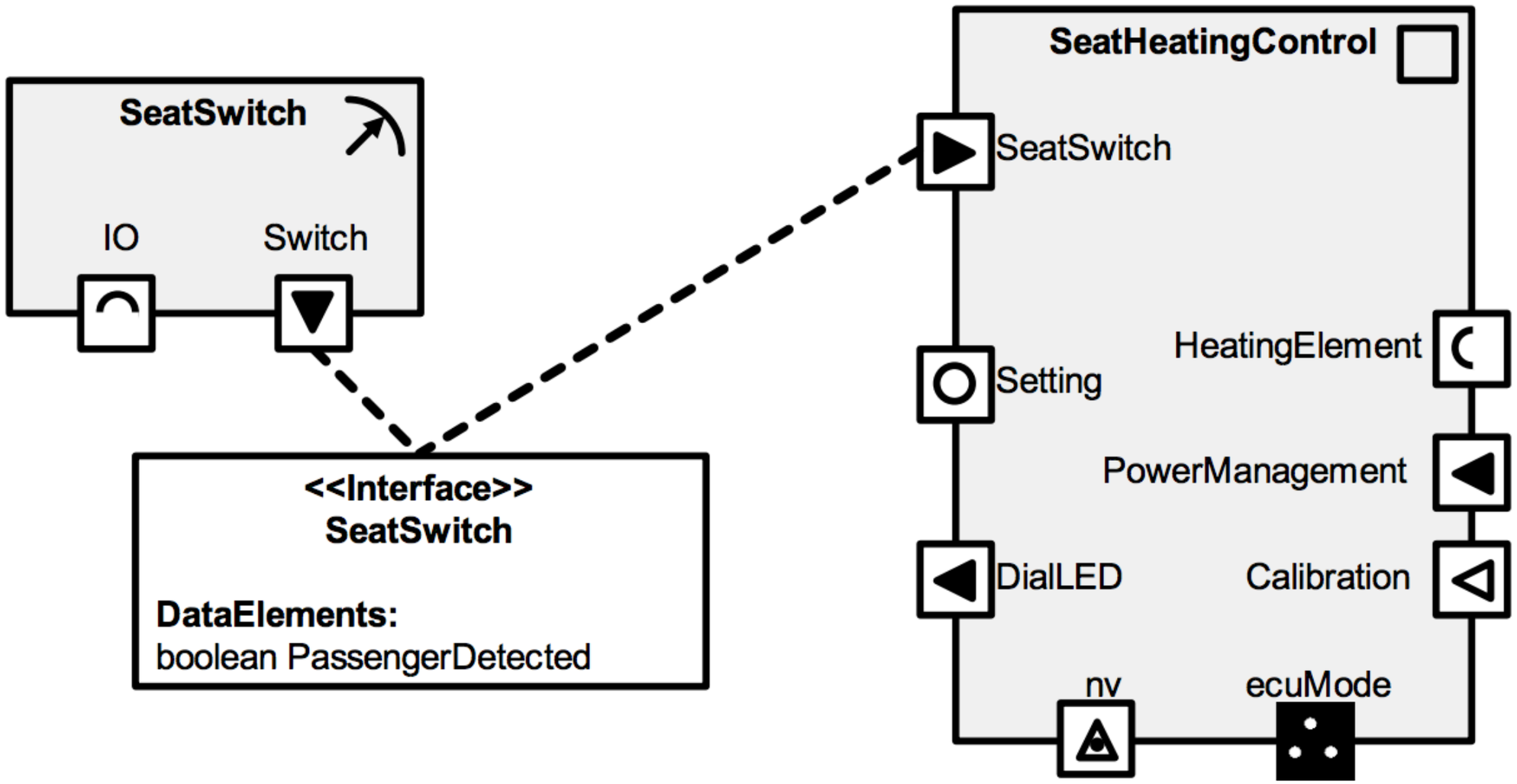


The Component requires operation defined in the interface

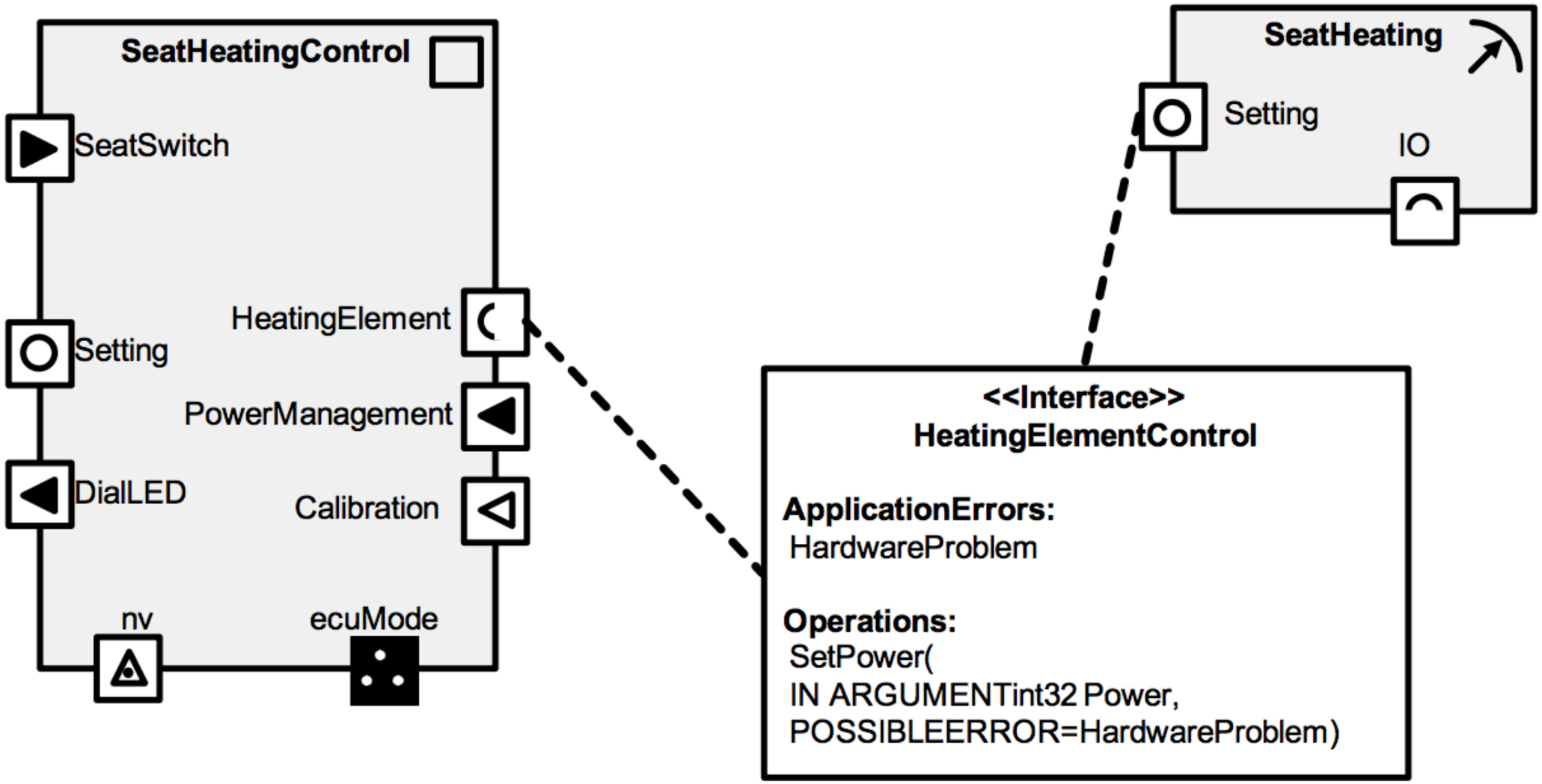
Sender - Receiver PPORT



The Component provides values of data-elements



Sender Receiver Interface



Client Server Interface

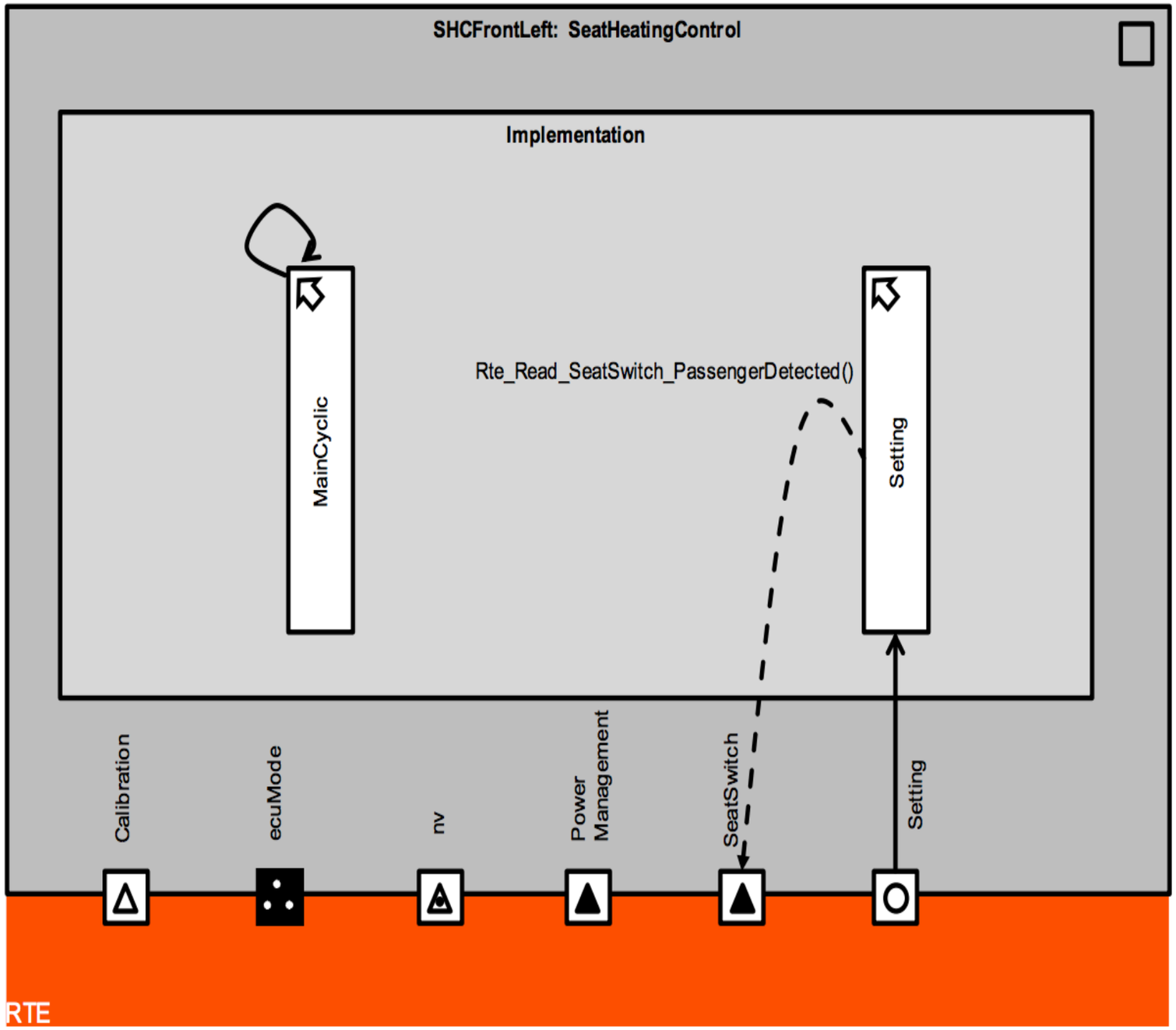
Design and Communication of software Components

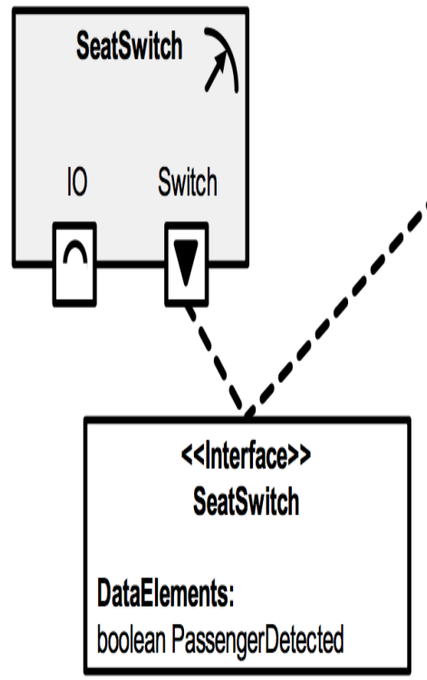
Communication between SWCs takes place chiefly over two kinds of ports,
Client/ Server ports where server is a provider of a service and the client is a user of a service and **Sender/ Receiver ports** where a sender distributes information to one or several receivers in synchronous as well as asynchronous environment.

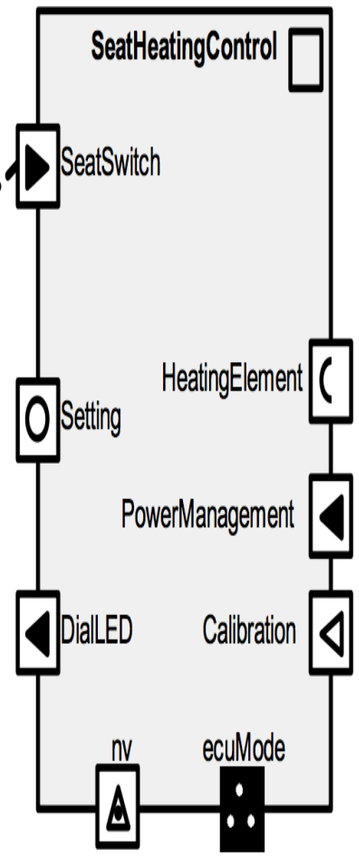
The implementation architecture of SWC is formally defined in terms of so-called **runnable entities**. **They correspond to procedures and are executed on a specific event such as a periodic activation or reception of new input value**.

During system design phase the SWCs can be integrated with their environment (e.g. hardware, driver, OS, etc) based on **Virtual Functional Bus (VFB)**.

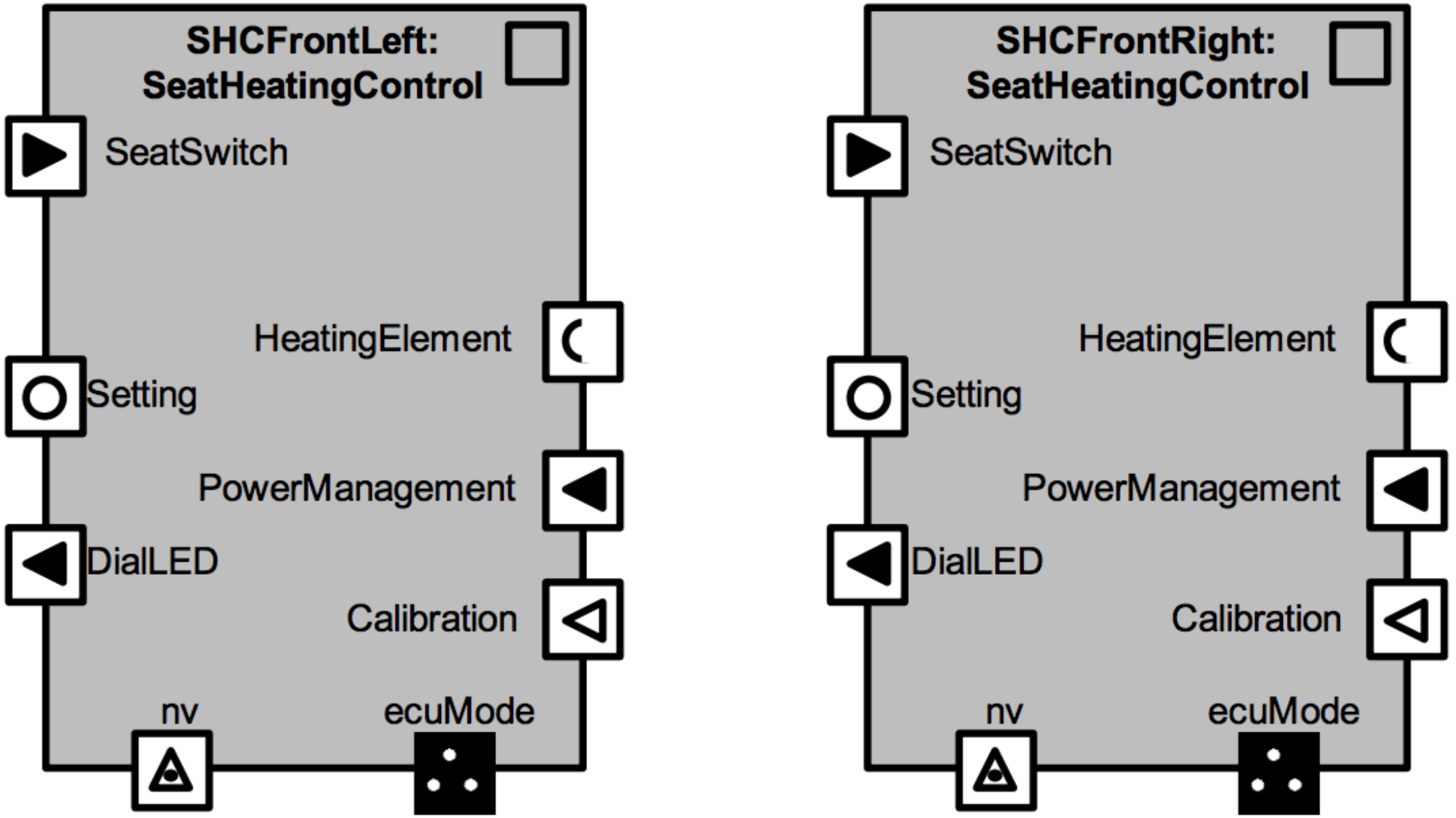
Atomic Software Component- Smallest software component which will remain in one ECU only. Cannot be broken between ECUs.





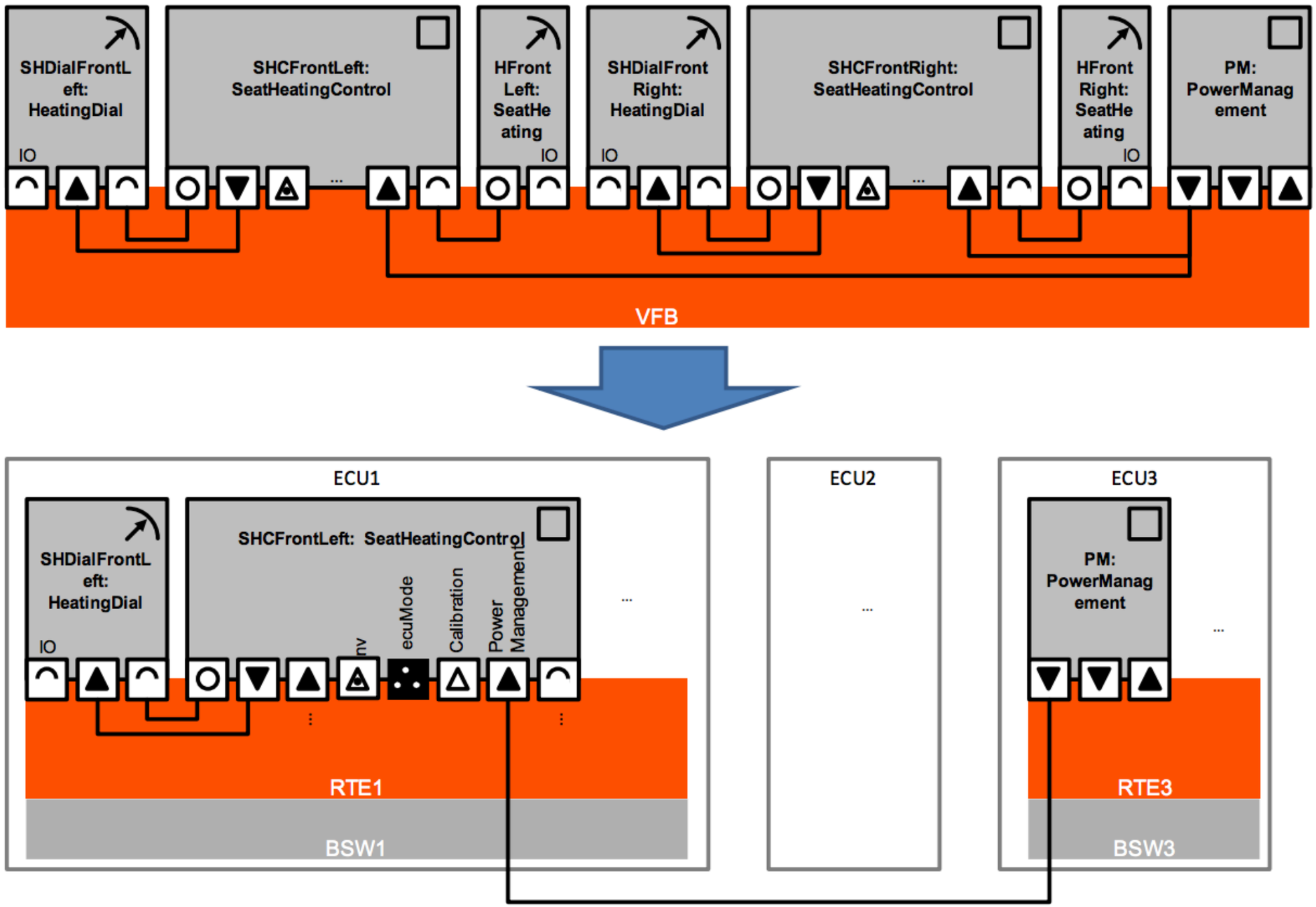


Atomic Component and Runnables



two instances of the “SeatHeatingControl” component-type are used to control the left front seat, respectively the right front seat. These components will typically have their own separate internal state (stored in separate memory locations) but might for example share the same code

AUTOSAR Supports Multiple Instantiation



AUTOSAR System View

Ports

Syntax & Semantics

Application Software Component

AUTOSAR Interface

Actuator Software Component

AUTOSAR Interface

Sensor Software Component

AUTOSAR Interface

AUTOSAR Software

Application Software Component

AUTOSAR Interface

AUTOSAR Runtime Environment (RTE)

Standardized Interface

Standardized AUTOSAR Interface

Services

Standardized Interface

Standardized Interface

Communication

Standardized Interface

AUTOSAR Interface

ECU Abstraction

Standardized Interface

AUTOSAR Interface

API CALLS

Operating System

Standardized Interface

Basic Software

Standardized Interface

Microcontroller Abstraction

Complex Device Drivers

ECU-Hardware

Software Component

Interface

Standard Software

Interfaces:

VFB & RTE relevant

RTE relevant

BSW relevant

Possible interfaces inside Basic Software (which are not specified within AUTOSAR)

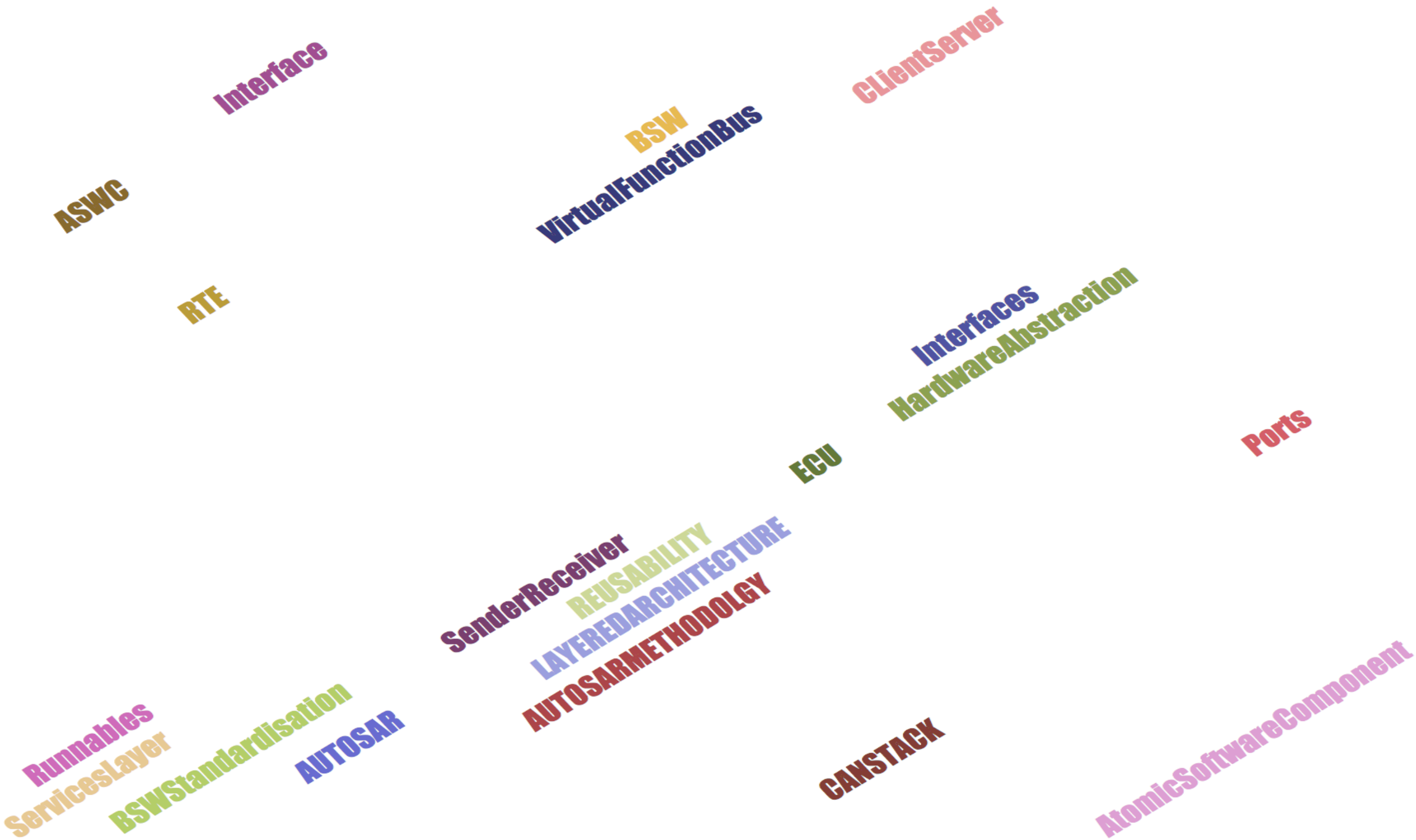
Simplified Component View

AUTOSAR Interfaces

AUTOSAR Interfaces are used in defining the ports of software-components and/or BSW modules. Through these ports, software-components and/or BSW modules can communicate with each other AUTOSAR makes it possible to implement this communication between Software-Components and/or BSW modules either locally or via a network.

- The **AUTOSAR Interface** is a generic interface which is derived from the ports of a SWC.
- AUTOSAR Interfaces are provided by the RTE and serve as interface between SWCs or between SWCs or between a SWC and the ECU firmware (IO HW and Complex Drivers). Via these interfaces, a SWC can e.g. read an input value or write an output value.
- The **Standardized AUTOSAR Interface** is an "AUTOSAR Interface" whose syntax and semantics are standardized in AUTOSAR. Such interfaces are used by the **SWCs to access AUTOSAR Services,** which are provided by BSW modules of the Service Layer like the **ECU manager or the diagnostic event manager.**
- The **Standardized Interface** is an interface, which is predefined by the AUTOSAR standard as API in C language. It is used between BSW module within an ECU, between RTE and Operating System (OS), or between RTE and the Communication Layer.

AUTOSAR Summary



Future Plans

To create an AUTOSAR Learning Group in India
[in collaboration with SemiCon, Tier 1, OEM , Service Providers and Universities]

ANCIT will continue to deliver Series of Webinars
Create AUTOSAR Learning Kit & Easy to Learn AUTOSAR Stack
Tools and Automation Support for AUTOSAR Development

Next Webinar:
Application Software Component Development

You will learn:
- How to create an system file (arxml)
- How to build a system level application software component
- Generate <*.c> and <*.h> file
- Build the ASWC <*.c>

Date: Sep 3, 2020
Time: 3pm

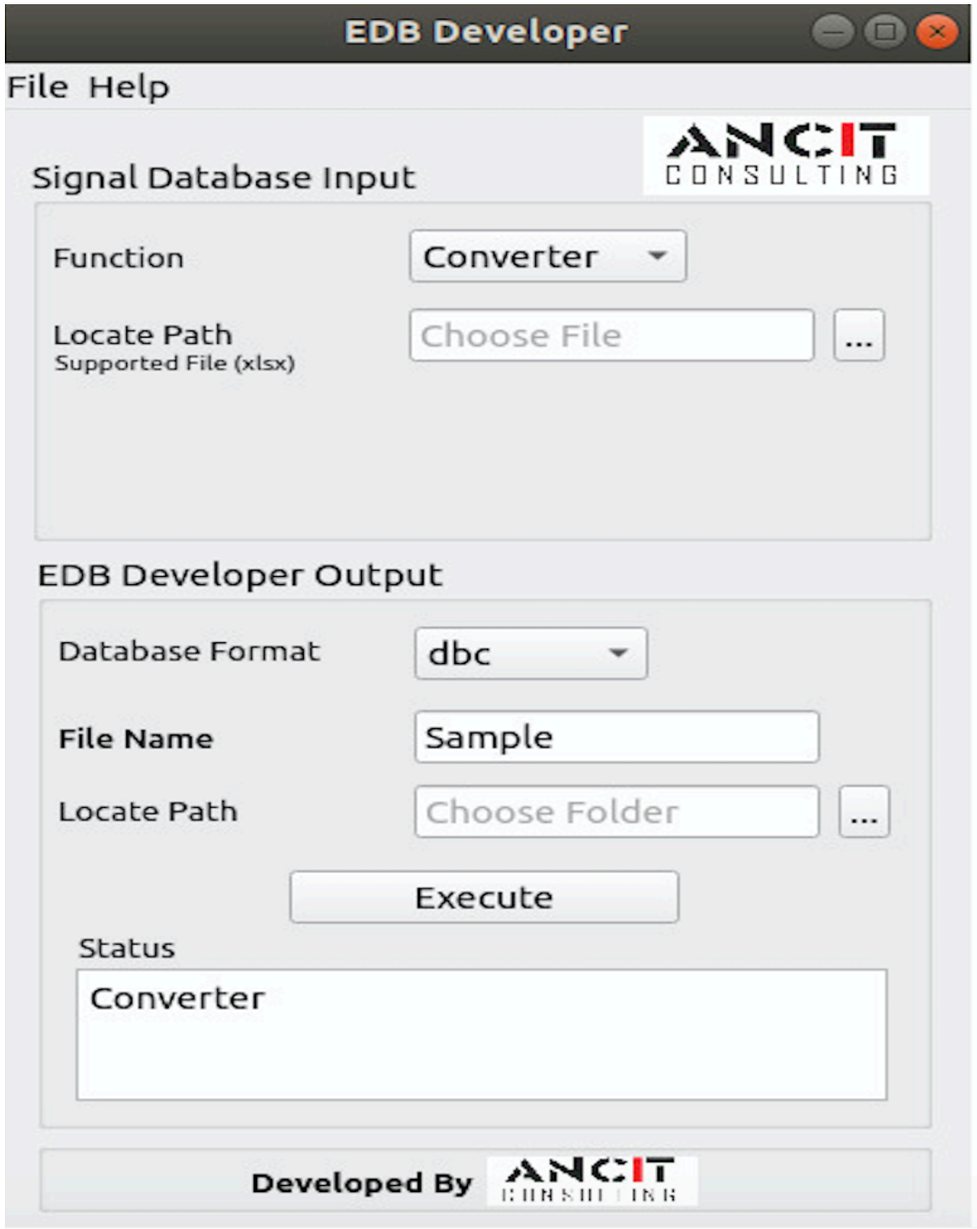
EDB EDITOR

ANCIT
CONSULTING

EDB Editor

Features:

- Create the electrical database in the form of excel file.
- Signals and node addition or removal.
- Message event or cyclic time conversion.
- Add or remove message in to the database file(excel)
- Edit existing database files.
- Convert the one format file(excel) to other(vector dbc/arxml). ex: vector database to .arxml standard & vice versa. Conversation made as per new file conversion standard.



Available for download from: August 15,2020 Its FREE!!!!

ANCIT CONSULTING

EDB EDITOR FEATURES

1. easness to create the electrical database in the form of excel file.
2. easiness on signals and node addition or removal.
3. easiness on message event or cyclic time conversion.
4. easiness to add or remove message in to the database file(excel)
5. the above mention the feature can be added or removed on the already exist database file.
6. easiness to convert the one format file(excel) to other(vector dbc/arxml). example: vector database to arxml standard and vice versa. it's required becuse of due to huge autosar adaptation is taken place in the automotive industry and development and testing tools are made and understandable by old format.
7. Conversation will be made as per new file conversion standard and quality will be assured.

Our Locations

Registered Office
4/1, Sri Krishna Villas,
1st Cross, Kongu Nagar,
Ramanathapuram
Coimbatore,TN, India
Pin : 641045
Ph : +91-422-6461778

Consulting Office
#28, Shankar Mutt Road,
Bangalore, KA, India.
Pin : 560004
Ph : +91-80-41244277/+91-80-23163244

In Germany
PW-tech
Unterhaunstädter Weg 3
85055 Ingolstadt, Germany

In US
Four Colors Technology LLC
3828 Tonsley Pl
High Point, NC 27265

| Website : www.ancitconsulting.com
Email : bk@ancitconsulting.com Contact : +919787400605