

KPIT

AUTOSAR Overview

KPIT

AUTOSAR Introduction

AUTOSAR (**AUT**motive **O**pen **S**ystem **AR**chitecture) is a revolutionary milestone in the development of Automotive E/E Systems. The development partnership established by leading OEM's and Suppliers had defined an open reference architecture for ECU software.

The idea behind AUTOSAR initiative is found in 2003 by Automotive OEM's and Suppliers to avoid re-developing of similar software components.

Uniform Producer Independent System Platform which enables Shorter time to market, Less development effort, More products out of one set of components, Improved quality

KPIT – AUTOSAR Premium Member

9 Core Partners



12 Development Members



48 Premium Members



- 66 Associate Members
- 6 Attendees

General OEM's

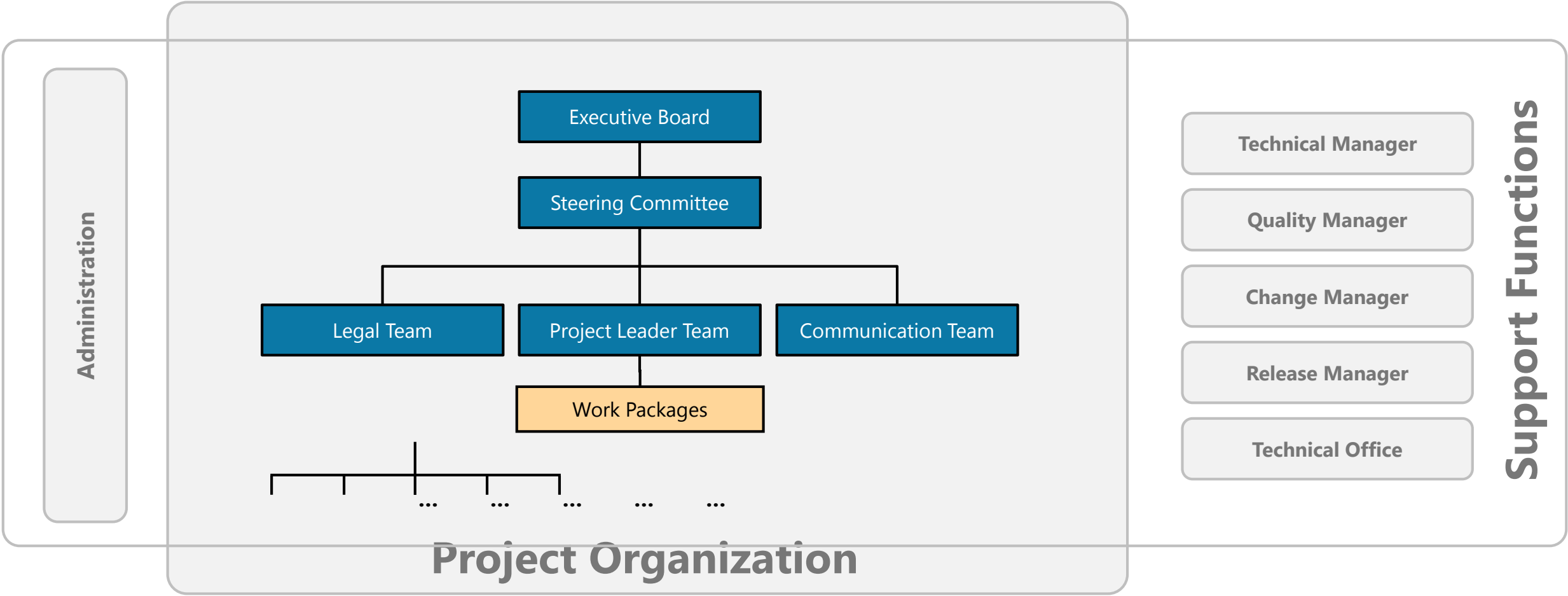
Generics Tier1's

General Software

Tools & Services

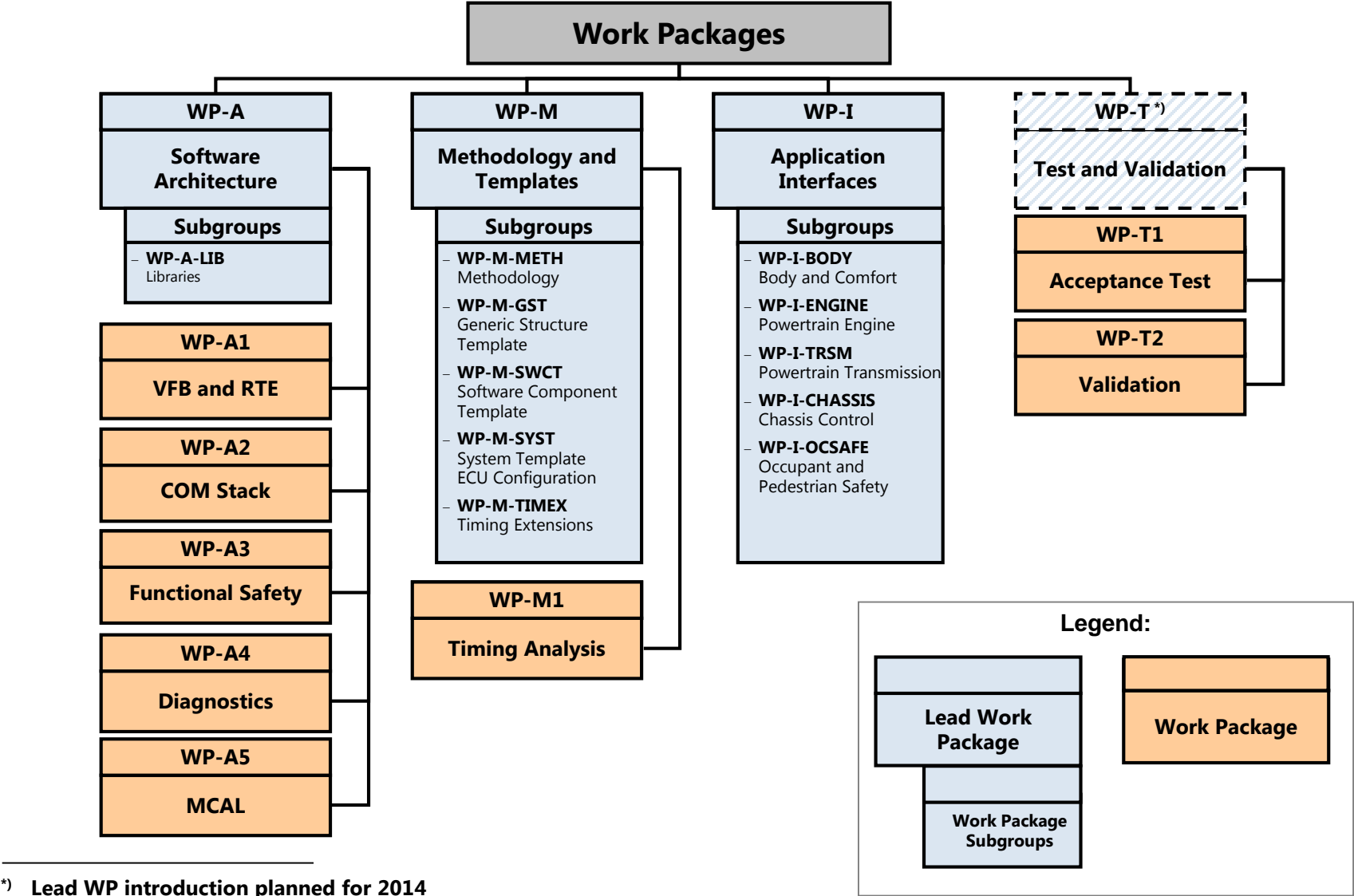
Semicon's

AUTOSAR– Organizational Structure



Core Partner
Core Partner, Premium and Development Member
Subcontractor

WP Structure



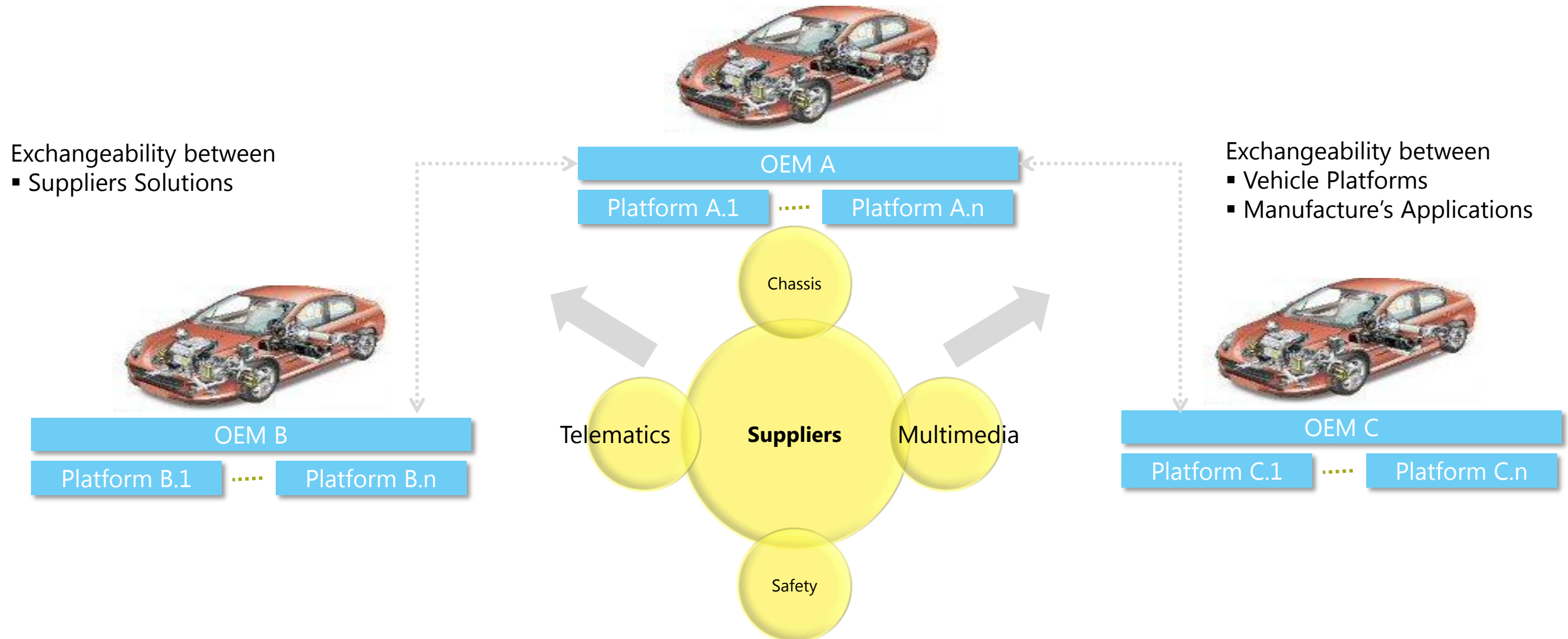
Agenda

- 1** AUTOSAR– Vision
- 2** AUTOSAR Concepts
- 3** AUTOSAR – ECU Software Architecture
- 4** Use Cases view
- 5** AUTOSAR Roll Out Plan
- 6** AUTOSAR Benefits



AUTOSAR– Vision

AUTOSAR 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.



Agenda

- 1** AUTOSAR– Vision
- 2** AUTOSAR Concepts
- 3** AUTOSAR – ECU Software Architecture
- 4** Use Cases view
- 5** AUTOSAR Roll Out Plan
- 6** AUTOSAR Benefits



AUTOSAR Concepts

Architecture

Software architecture including a complete **Basic** or Environmental **Software Stack** for **ECUs** –**AUTOSAR Basic Software**

Methodology

Exchange formats or description templates to enable a **Seamless Configuration** process of **BSW stack** and the **Integration** of application software in **ECU's** and it includes even the methodology how to use this framework.

Application Interfaces

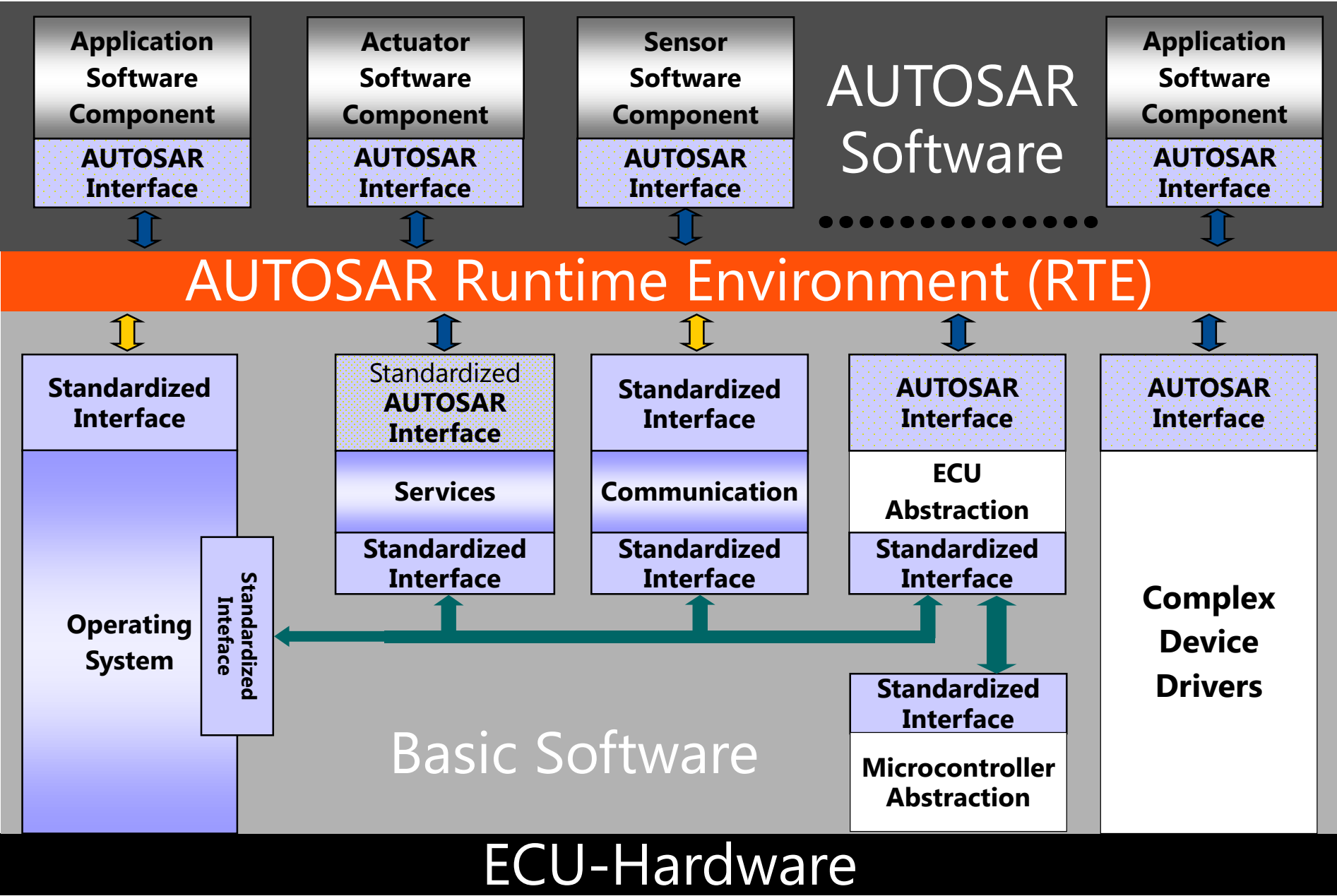
Specification of **Interfaces** of typical **Automotive Applications** from all domains in terms of **syntax and semantics**

Agenda

- 1** AUTOSAR– Vision
- 2** AUTOSAR Concepts
- 3** AUTOSAR – ECU Software Architecture
- 4** Use Cases view
- 5** AUTOSAR Roll Out Plan
- 6** AUTOSAR Benefits



AUTOSAR – ECU Software Architecture



AUTOSAR layer model offers...

Modular Components with defined interfaces

Hardware Independent Software layer

Transferability of modules

Decoupled Basic Software and Application Components using RTE

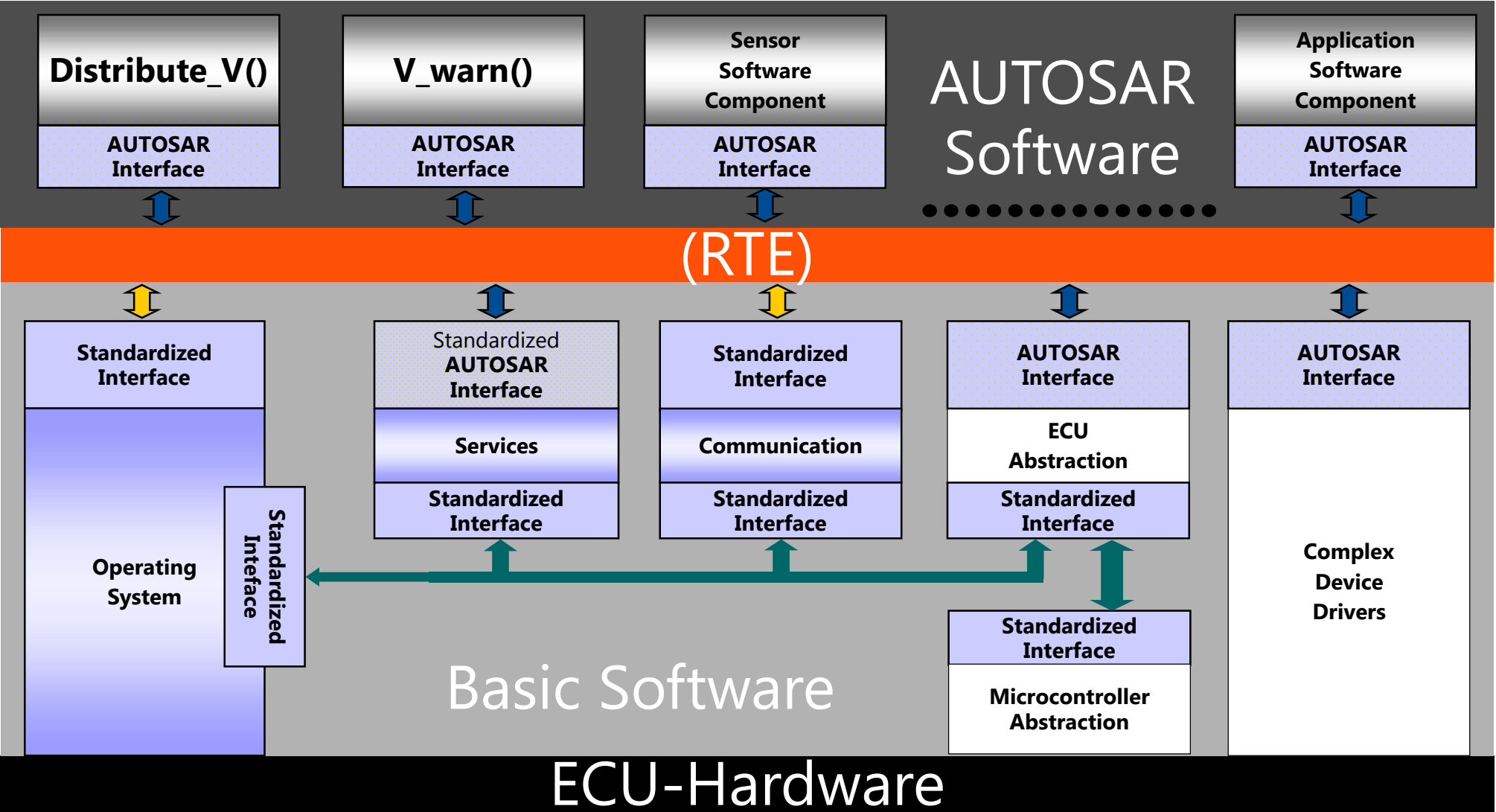
Agenda

- 1** AUTOSAR– Vision
- 2** AUTOSAR Concepts
- 3** AUTOSAR – ECU Software Architecture
- 4** Use Cases view
- 5** AUTOSAR Roll Out Plan
- 6** AUTOSAR Benefits



Use Case 'Pedal Management' view for one ECU

RTE Implements data exchange between Application Software Components & BSW



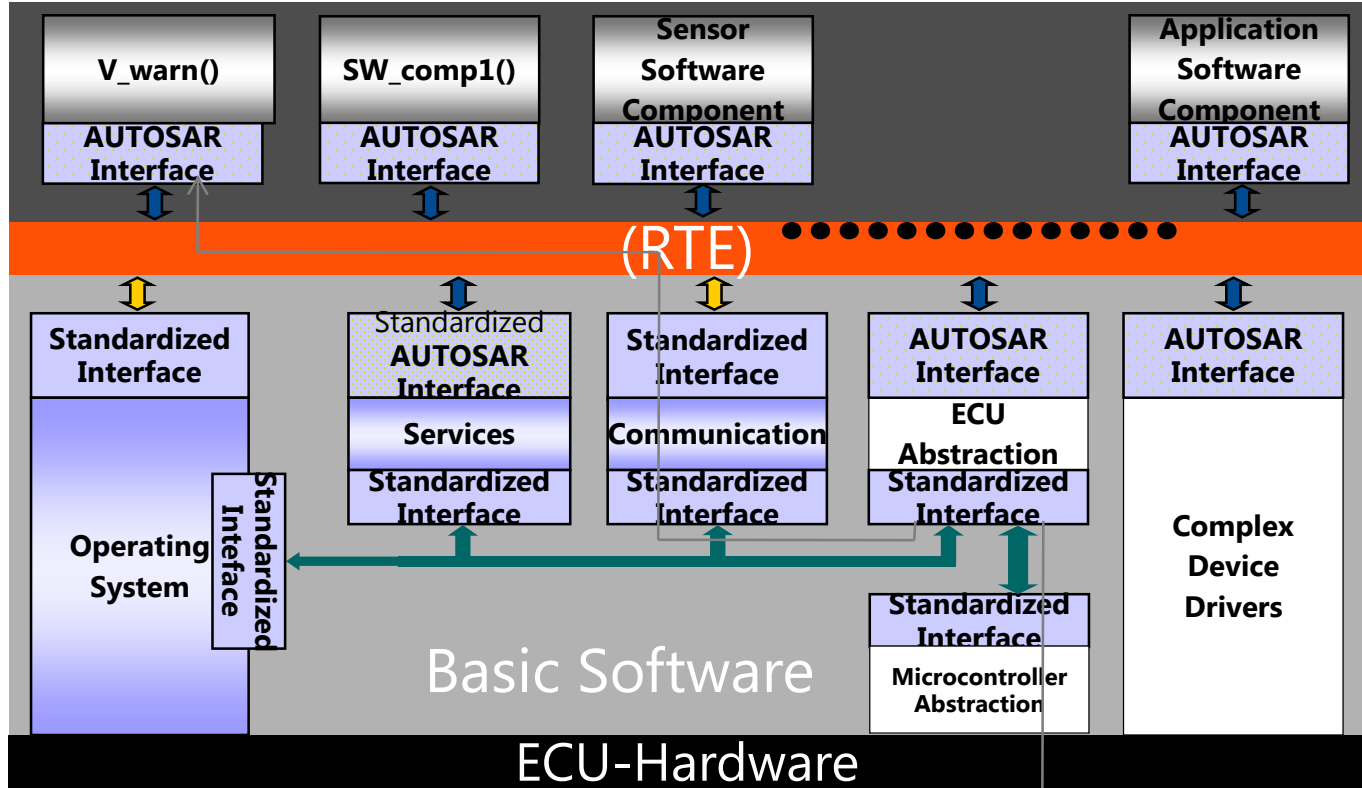
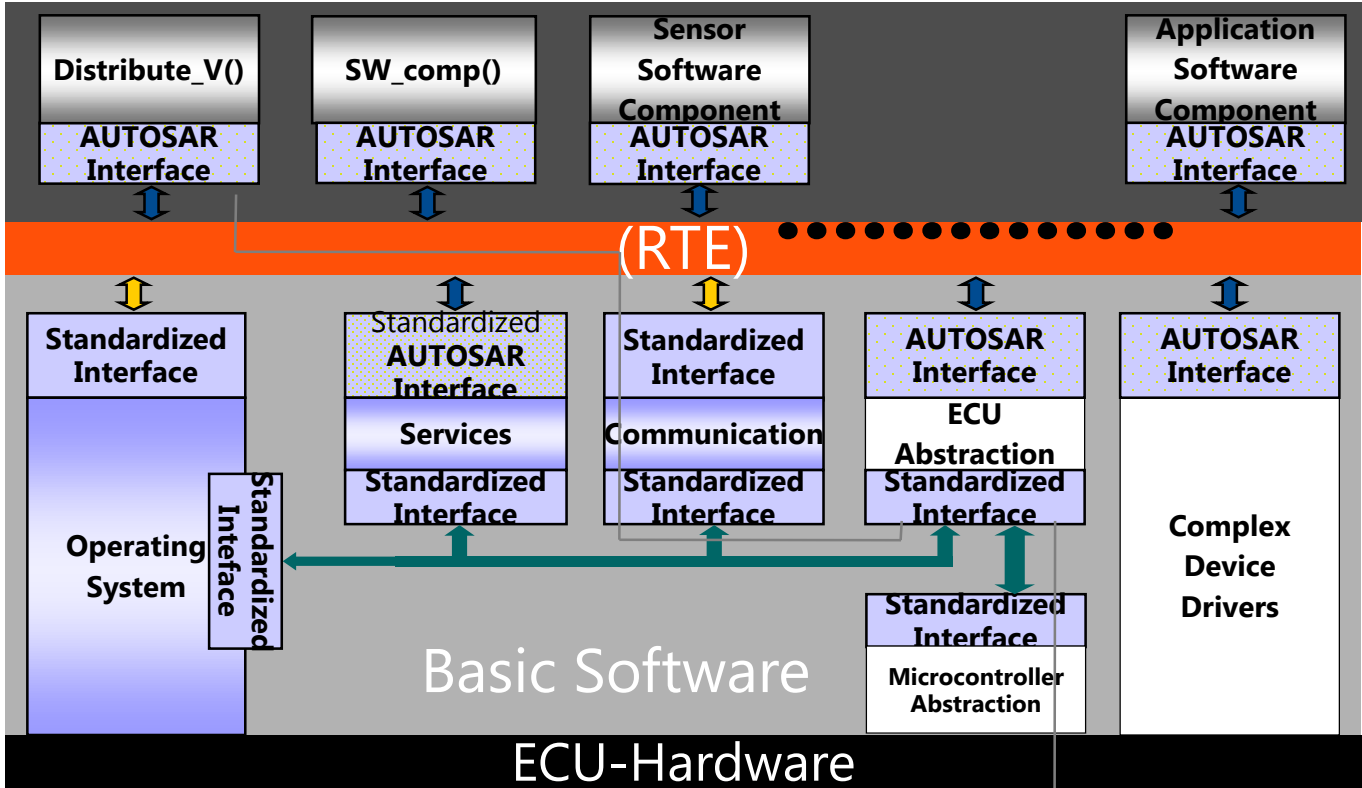
```
Void Distribute_V(void)
{
  ---
  RTE_Write_p_V(rte_i,v);
  --
}
```

```
Void V_warn(void)
{
  ---
  RTE_Read_p_V(rte_i,v);
  --
}
```

Use Case 'Pedal Management' view for two ECU

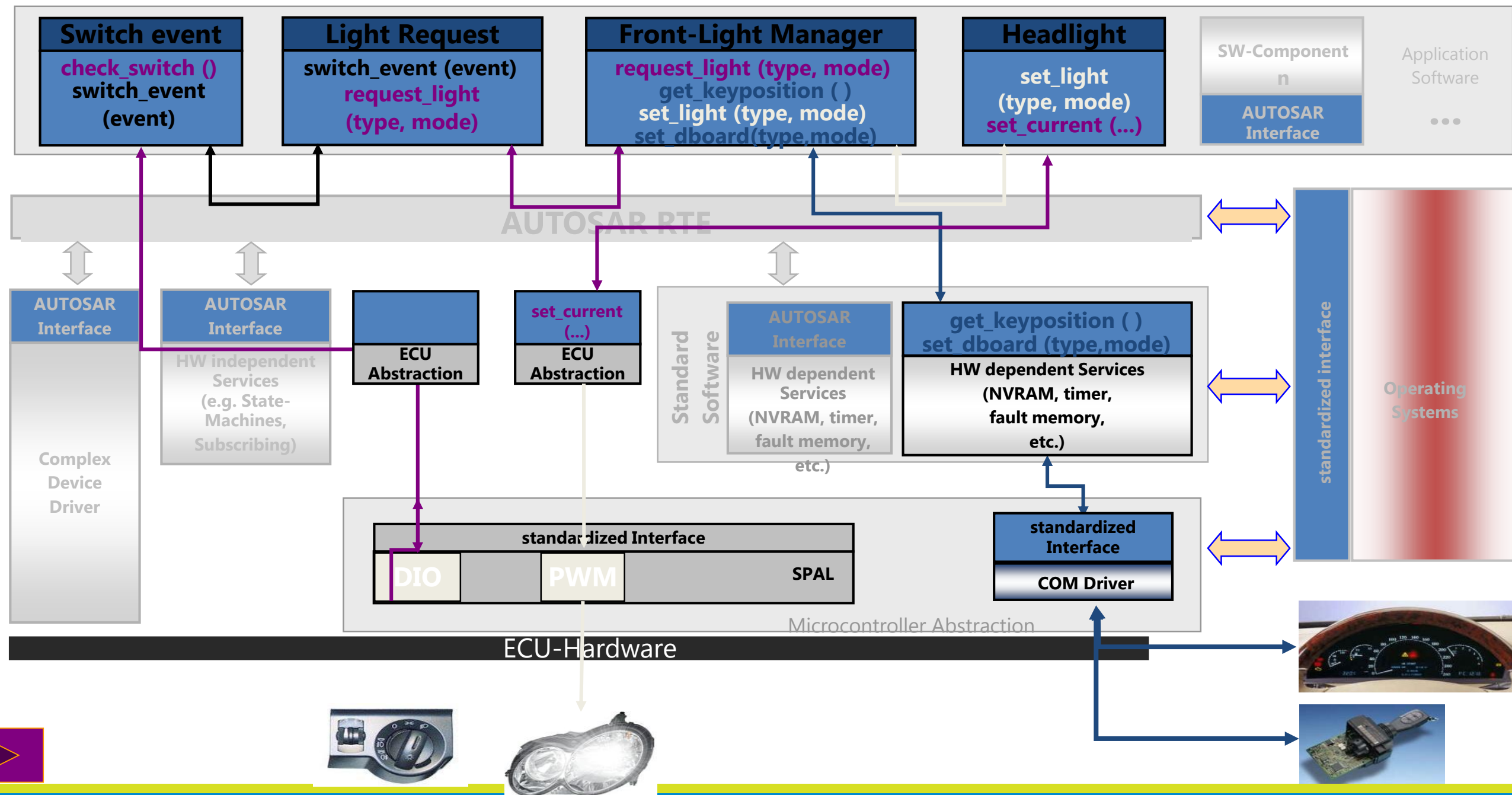
```
Void Distribute_V(void)
{
  -----
  RTE_Write_p_V(rte_i,v);
  ----
}
```

```
Void V_warn(void)
{
  ----
  RTE_Read_p_V(rte_i,v);
  ----
}
```

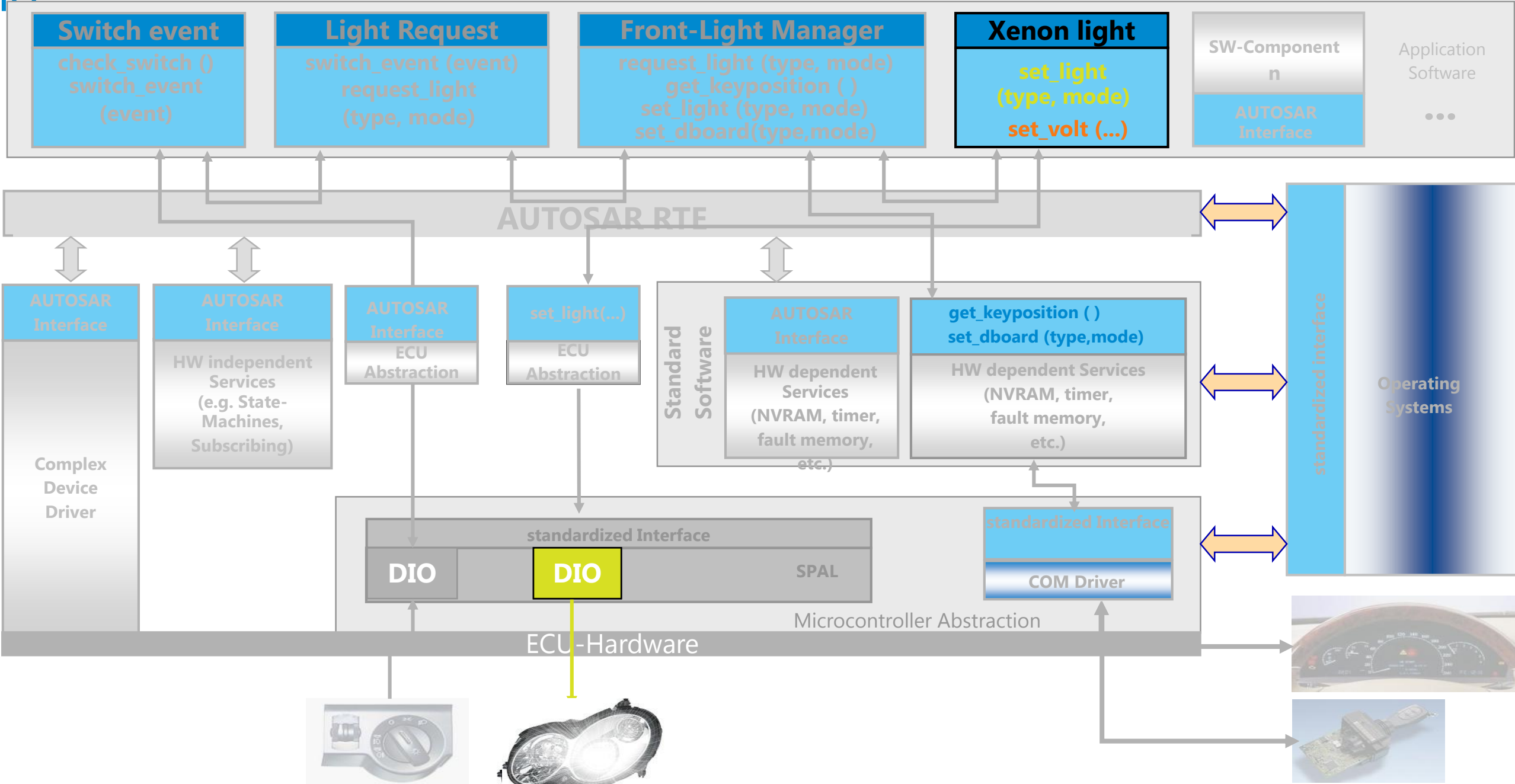


CAN

Use Case 'Front Light Management' mapped to AUTOSAR architecture



Exchange of type of front light and consequences to the Use Case mapping

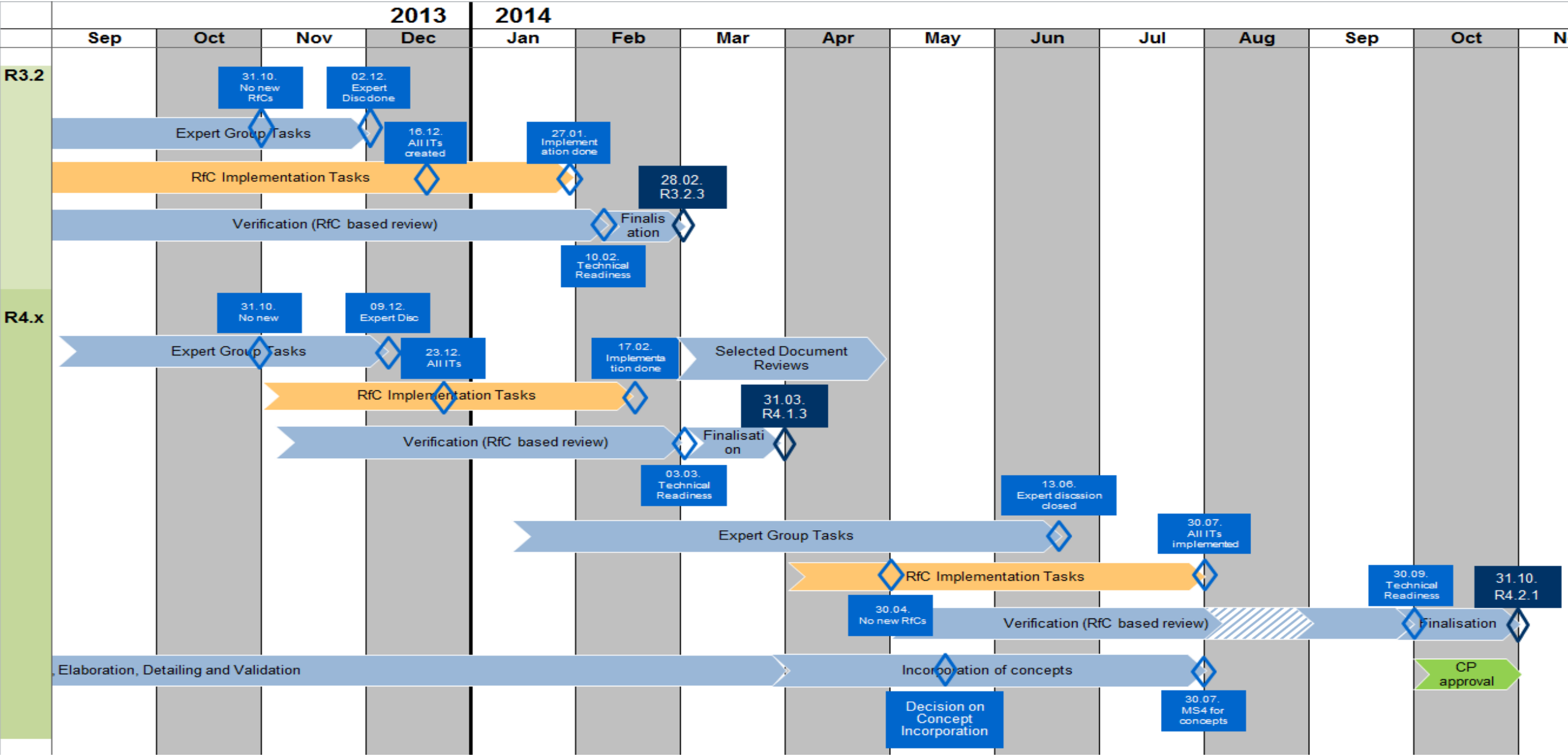


Agenda

- 1** AUTOSAR– Vision
- 2** AUTOSAR Concepts
- 3** AUTOSAR – ECU Software Architecture
- 4** Use Cases view
- 5** AUTOSAR Roll Out Plan
- 6** AUTOSAR Benefits



Release Schedule for R3.2, R4.1 / R4.2



Agenda

- 1** AUTOSAR– Vision
- 2** AUTOSAR Concepts
- 3** AUTOSAR – ECU Software Architecture
- 4** Use Cases view
- 5** AUTOSAR Roll Out Plan
- 6** AUTOSAR Benefits



Platform and Network Architecture Standardization

Challenges	Solutions	Benefits	KPIT Experience
<ul style="list-style-type: none">▪ OEM has to maintain their own Network platform specification	AUTOSAR Standard	<ul style="list-style-type: none">▪ AUTOSAR maintains the specification	<ul style="list-style-type: none">▪ R3.x and R4.0 considered being introduced in Production
<ul style="list-style-type: none">▪ Effort wasted on maintaining proprietary Network Platform Solution which add no value recognized by the customer	Standard Software (Basic Software Core)	<ul style="list-style-type: none">▪ Enhancement of software quality▪ Concentration on functions with competitive value	<ul style="list-style-type: none">▪ KPIT Solution is being used / evaluated by several OEMs / Tier1s

Platform and Network Architecture Standardization

Challenges	Solutions	Benefits	KPIT Experience
<ul style="list-style-type: none">• OEM specific platform has to be integrated with application for every OEM by Tier1 and this costs overhead for OEMs	Standardized components / Interfaces	<ul style="list-style-type: none">• Reusability of platform components (Basic SW modules) across-OEMs	<ul style="list-style-type: none">• KPIT stack being used by Several tier1s in Production Programs and pilots. Could be used for multiple OEMs
<ul style="list-style-type: none">• Lack of guidelines for use/ buy of software components	Conformance Test Process	<ul style="list-style-type: none">• Conformance Tests and Acceptance Tests by AUTOSAR	<ul style="list-style-type: none">• 70% coverage for Services modules (e.g., COM) and 50 % coverage for HW dependent comm. Drivers in conformance tests.

Microcontroller Abstraction Standardization

Challenges	Solutions	Benefits	KPIT Experience
<ul style="list-style-type: none">• Obsolescence of hardware (µC, circuits,. ...) causes huge efforts in adapting existing software• Extended needs for microcontroller performance (caused by new functions) cause need for upgrade, i.e. re-design effort	Microcontroller Abstraction (MCAL)	<ul style="list-style-type: none">• Part of the hardware can be exchanged / upgraded without need for adaptation of higher software/ functions/ applications	<ul style="list-style-type: none">• Micro vendors provide MCAL• Micro change for middleware was possible in the middle of the project without change in schedule.

Architecture Tool Interface Standardization

Challenges	Solutions	Benefits	KPIT Experience
<ul style="list-style-type: none">• Different Tool suppliers /Network SW suppliers use their own proprietary formats and hence the OEMs stuck with such tools forever <p>Interoperability with other formats / tools not possible.</p>	Standard format - System Configuration Description	<ul style="list-style-type: none">• Need not get locked with a single tool with proprietary interfaces.	<ul style="list-style-type: none">• KPIT has partnerships with Application/ Architecture tool vendors such as Mathworks and ETAS• Mix and match possible.

Application Interface Standardization

Challenges	Solutions	Benefits
<ul style="list-style-type: none">• OEM buys black-box & is not able to extend/ integrate new functionality in an ECU (e.g. integration of tire guard functionality)	Technical Integration of Software of Multiple Suppliers	<ul style="list-style-type: none">• Eased process of integration of different software components allows optimization of hardware costs (No. of additional ECUs required for new functionality could be reduced)
<ul style="list-style-type: none">• Large effort when relocating functions between ECUs• Large effort when reusing functions	Runtime Environment (RTE)	<ul style="list-style-type: none">• OEM can effectively relocate a function among ECUs.

✓ **Architecture:**

Software architecture including a complete basic (environmental) software stack for an ECU as an integration platform for hardware independent SW applications

✓ **Methodology:**

Exchange formats (templates) to enable a seamless configuration process of the basic software stack and the integration of application software in ECUs

✓ **Application Interfaces:**

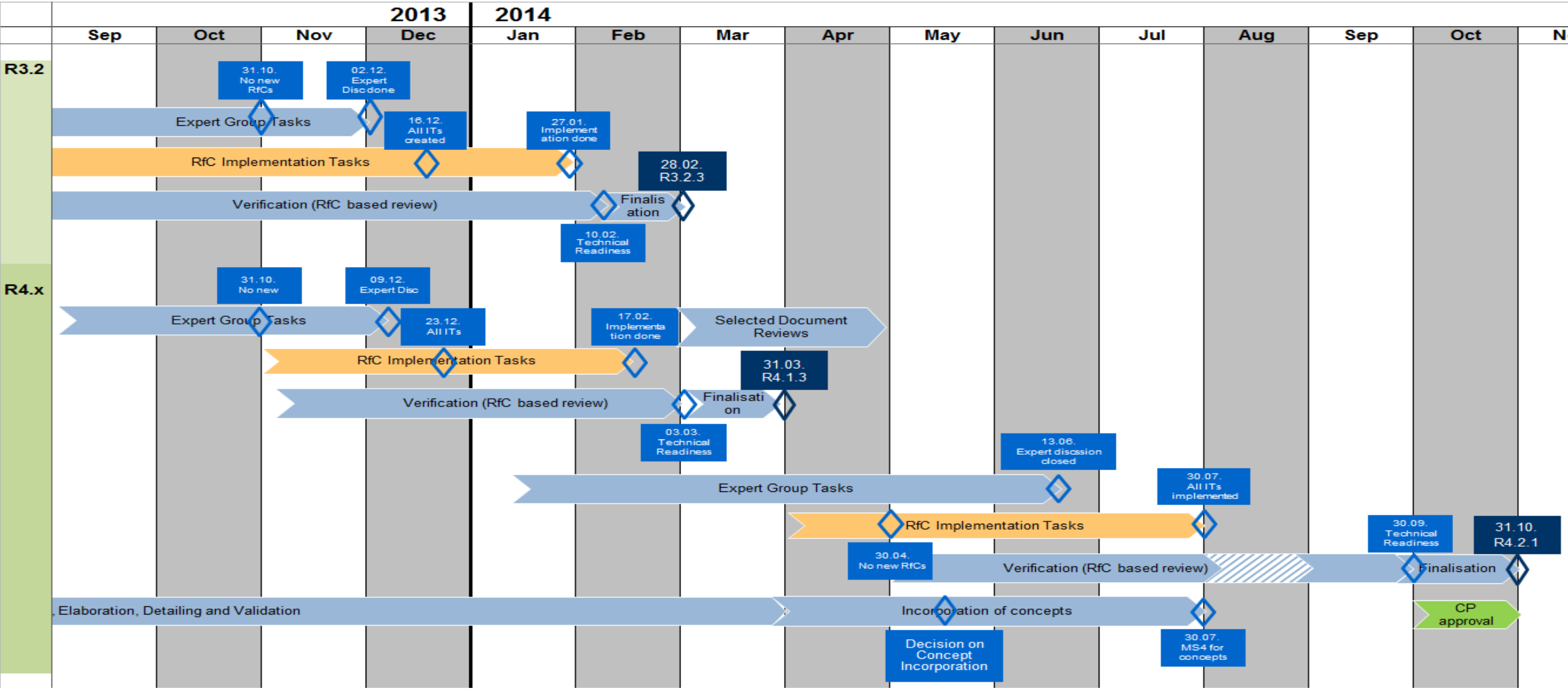
Specification of application interfaces as a standard for application software modules

Change in Automotive Development



- ❖ **Hardware- and software will be widely independent of each other.**
- ❖ **Development processes will be simplified.**
This **reduces development time and costs.**
- ❖ **Reuse of software increases at OEM as well as at suppliers.**
This **enhances also quality and efficiency.**

Release Schedule for R3.2, R4.1 / R4.2





Questions

Thank You

www.kpit.com

