IMT SCHOOL

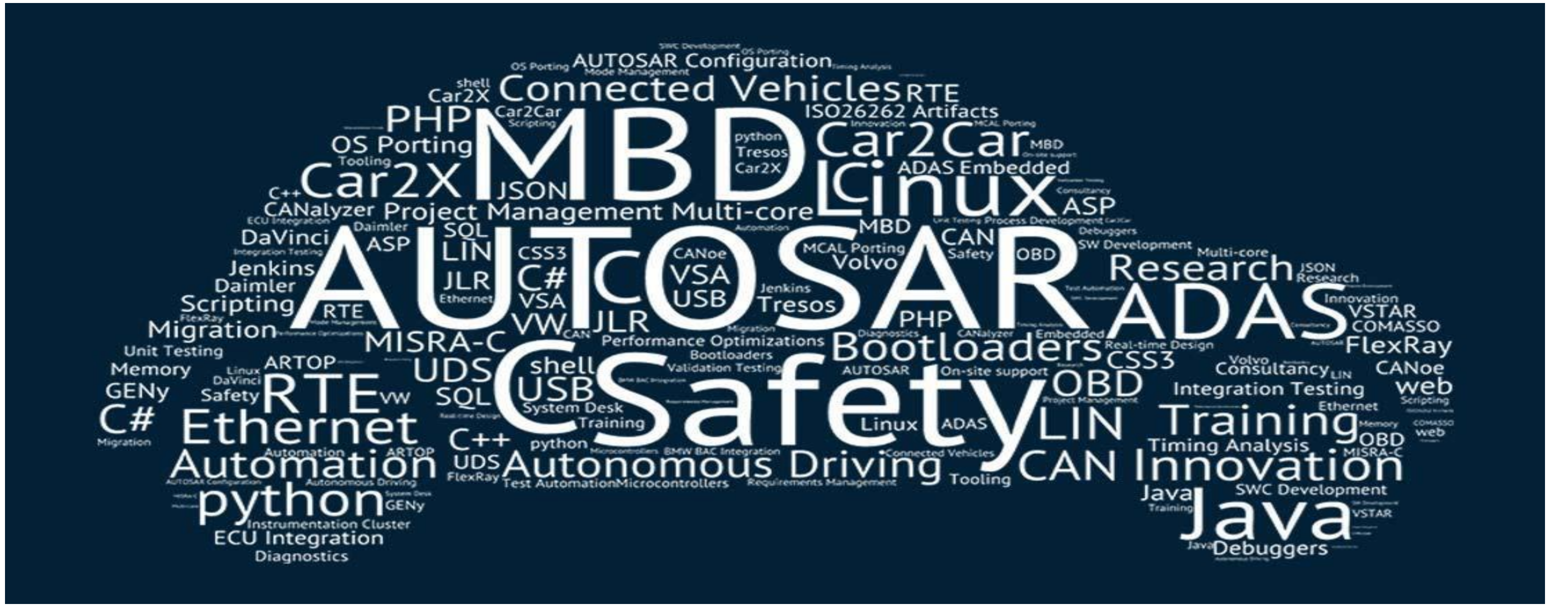
Automotive Industry

Introduction to AUTOSAR

This material is developed by IMTSchool for educational use only

Agenda

- **AUTOSAR Intro**
- **Overview of Software Layers/Stacks**
- **BSW Integration**



AUTOSAR Intro

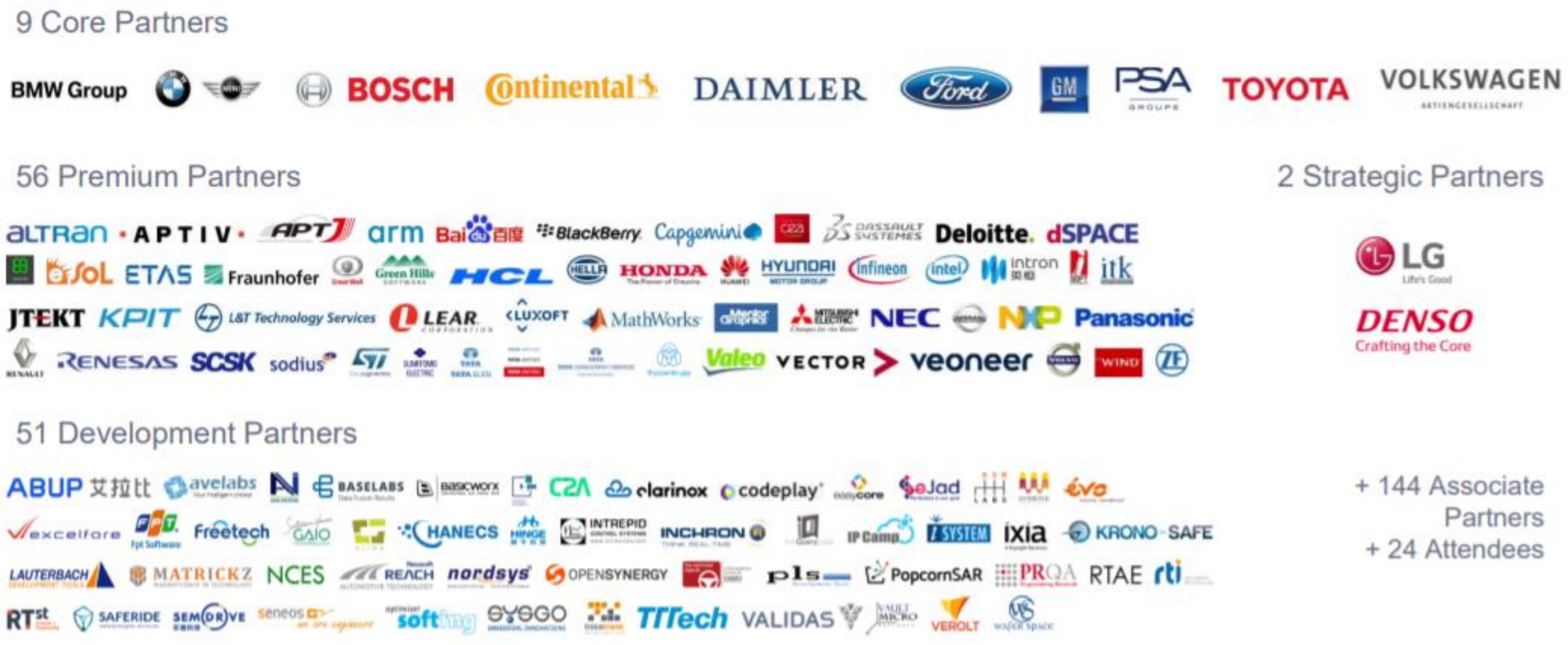
- What is AUTOSAR ?

- **AUT**omotive **O**pen **S**ystem **AR**chitecture

- An open and standardized automotive software architecture, developed by
  - Car manufacturers
  - Suppliers
  - Tool developers

AUTOSAR Intro

- Who participates



9 Core Partners

56 Premium Partners

2 Strategic Partners

51 Development Partners

+ 144 Associate Partners
+ 24 Attendees

AUTOSAR Intro

- AUTOSAR Statistics
  - AUTOSAR has been started in 2003
  - 288 worldwide organizations are involving in AUTOSAR (9 core members, 56 premium members)
  - AUTOSAR has been deployed on millions of ECUs
  - 9 major releases have been published: 2.1, 3.0, 3.1, 3.2, 4.0, 4.1, 4.2, 4.3,4.4

AUTOSAR Intro

AUTOSAR slogan: "Cooperate on standards, compete on implementation"

- Why AUTOSAR ?

- Handle the increasing amount of software

- Technical and commercial benefits:
  - Scalability on different platforms / vehicles to reduce the cost and improve quality
  - Speed up application development to reduce the cost and the risks
  - Standard to reduce amount of bugs and ease the maintenance
  - Improve the software reliability for Quality and Safety
  - Integration of modules of multiple suppliers
  - Separate developing application from the utilities room for innovation

AUTOSAR Intro

- Define a common software architecture for automotive domain

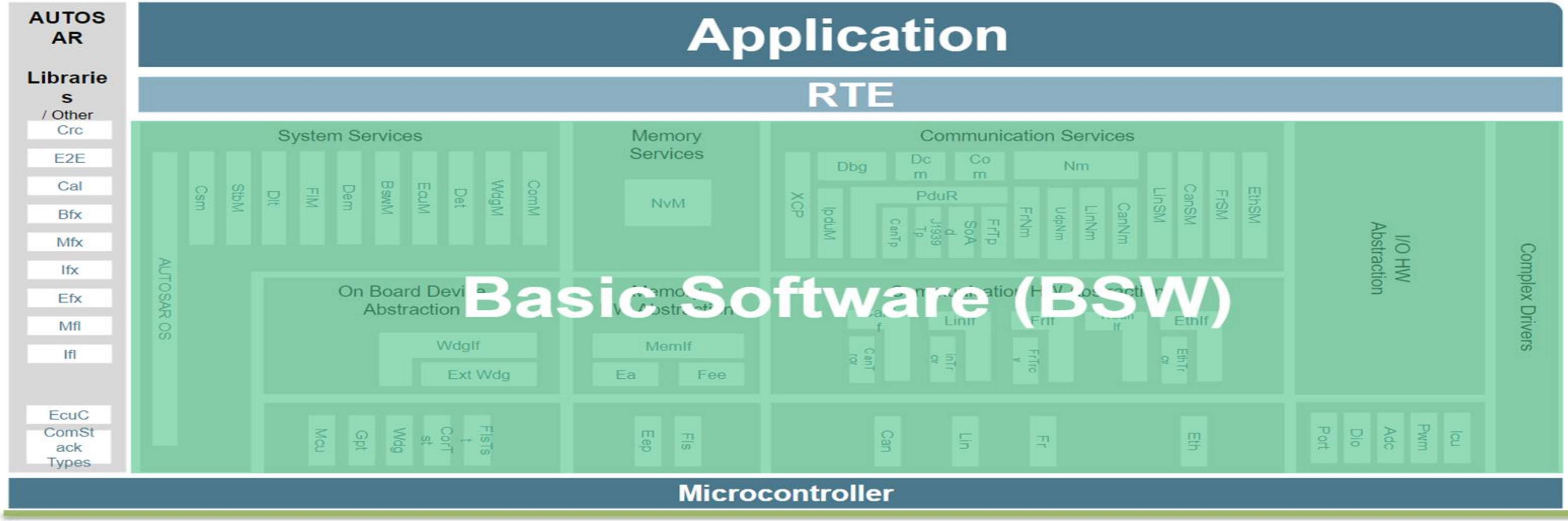
- Define how applications will communicate within AUTOSAR defining some application Interfaces

- Define a methodology to configure, generate and validate functionalities

- AUTOSAR Key Feature
    - Modularity and configurability
    - Standardized interfaces
    - application separated from the utilities (Runtime Environment)

Overview of Software Layers/Stacks

- AUTOSAR architecture consists of three main layers :
  - Basic software (BSW)
  - RTE
  - Application



Overview of Software Layers/Stacks

- Basic software (BSW) :
  - Provides services to the application
  - Responsible for running the functional part of the software
    Ex. Communication (Can , LIN , ETHERNET , …..) , I/O management , network management , memory management , OS
  - Contains standardized and ECU specific components

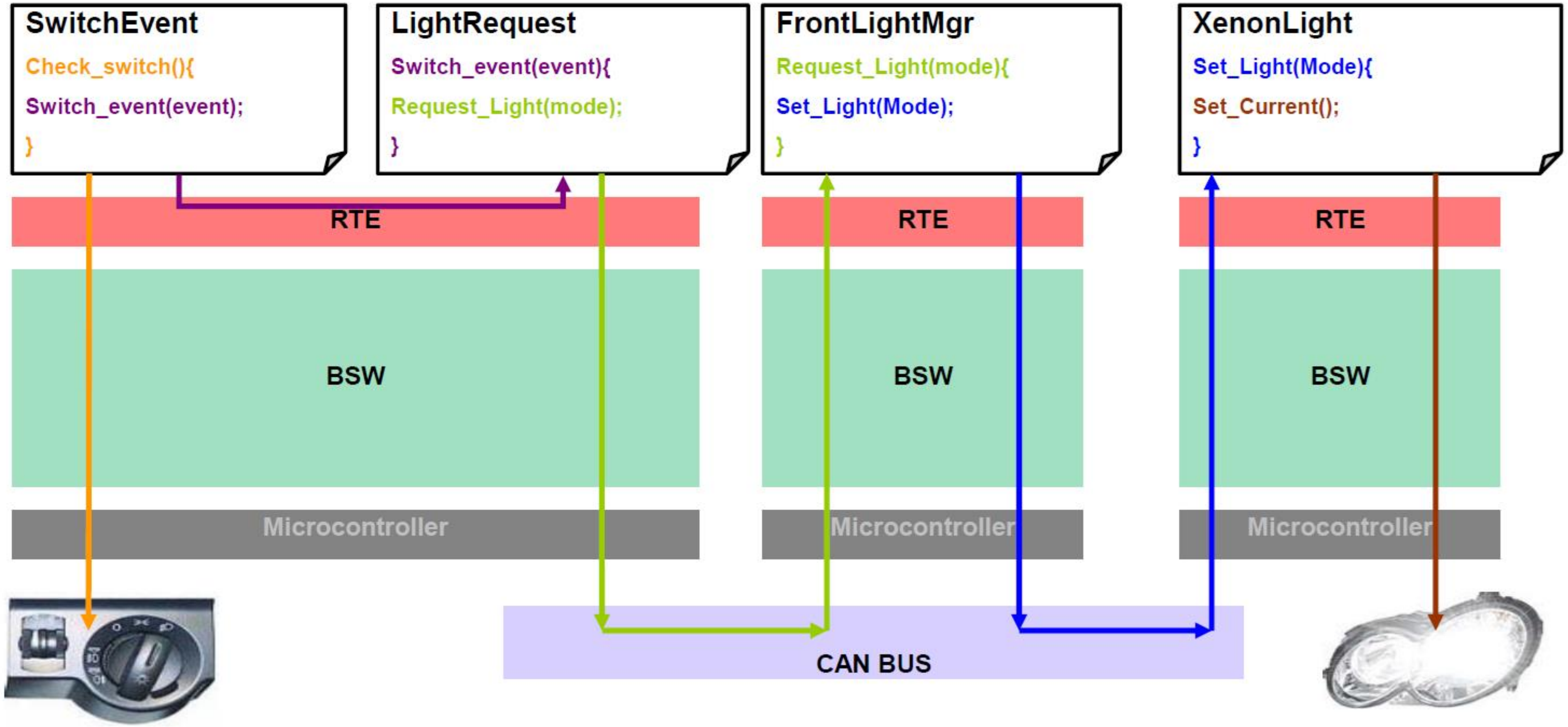
Overview of Software Layers/Stacks

- Application : Software components mapped on the ECU which running the logical part of the software

- RTE :
  - Communication center for
    - intra : SWC/BSW , SWC/SWC
    - inter : ECU information exchange
  - It provides a communication abstraction to Application AUTOSAR Software Components by providing the same interface whether
    - inter-ECU communication channels are used (e.g. CAN , LIN , ..)
    - OR communication stays intra ECU

Overview of Software Layers/Stacks

- Intra/Inter ECU communication



```
SwitchEvent
Check_switch(){
Switch_event(event);
}
```

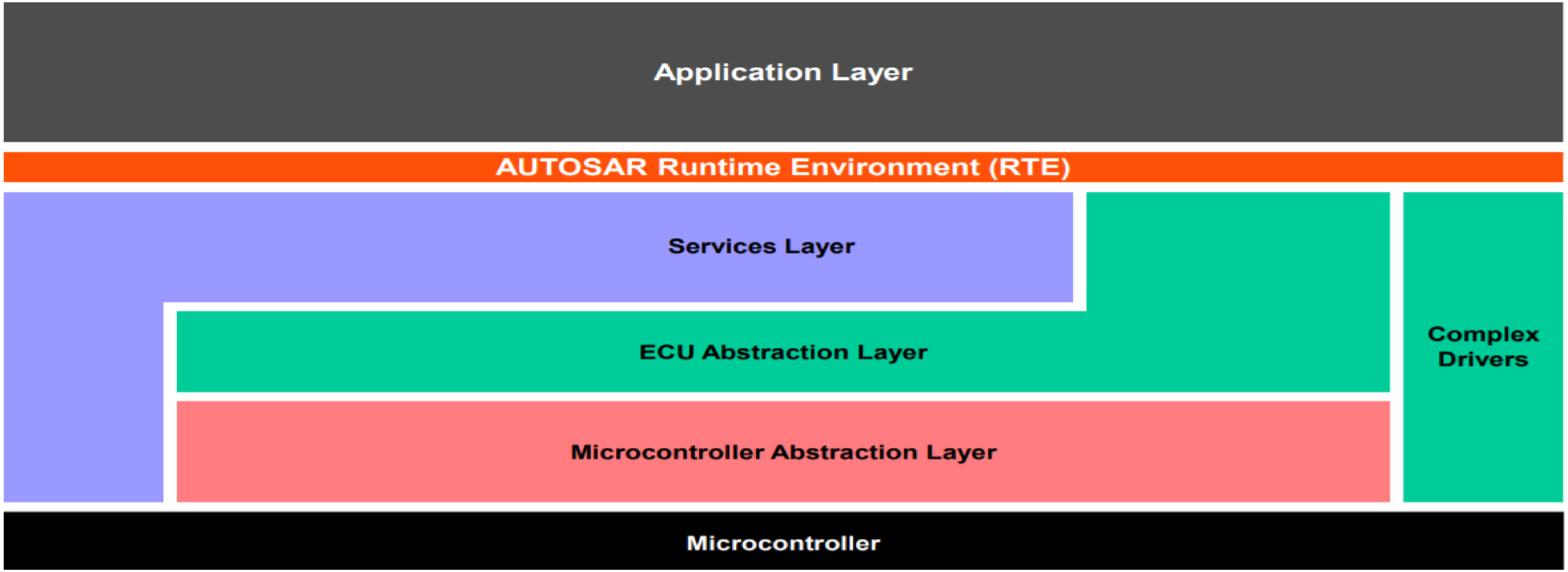
```
LightRequest
Switch_event(event){
Request_Light(mode);
}
```

```
FrontLightMgr
Request_Light(mode){
Set_Light(Mode);
}
```

```
XenonLight
Set_Light(Mode){
Set_Current();
}
```

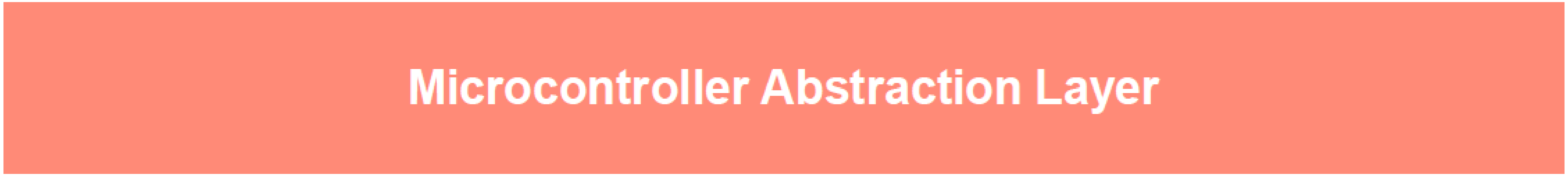
Overview of Software Layers/Stacks

- BSW consists of three layers:
  - Microcontroller Abstraction layer (MCAL)
  - ECU Abstraction layer (ECUAL)
  - Services layer



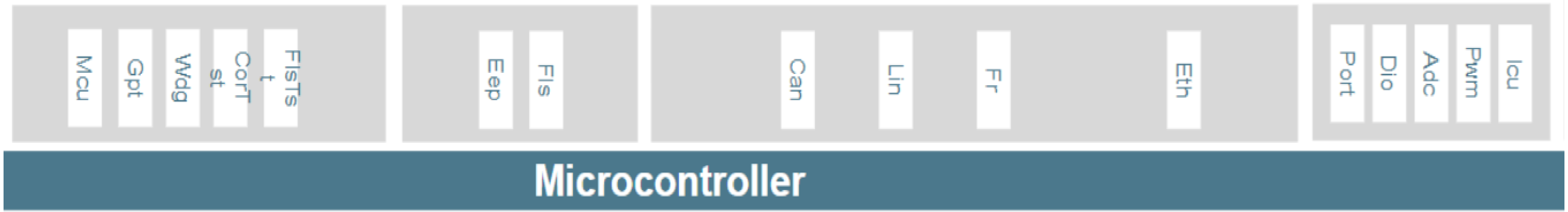
Overview of Software Layers/Stacks

- Microcontroller Abstraction layer (MCAL) :
    - Software layer containing drivers to enable the access of On-Chip peripheral devices of microcontroller
    - It abstracts the microcontroller details
    - MCAL is a hardware specific layer



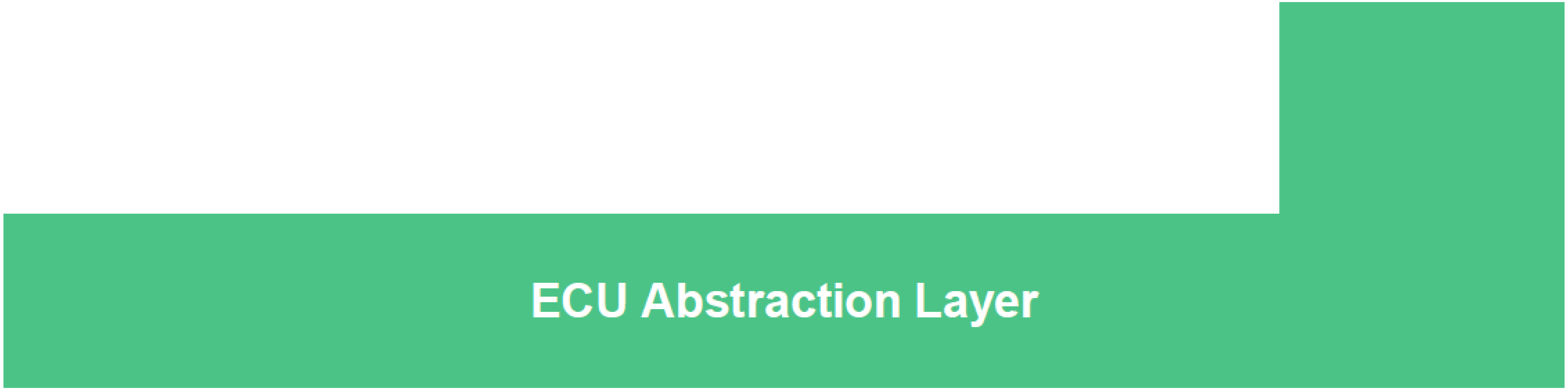
Overview of Software Layers/Stacks

- MCAL Drivers Examples :
  - ADC
  - CAN
  - Internal EEPROM
  - Internal Watchdog
  - ....



Overview of Software Layers/Stacks

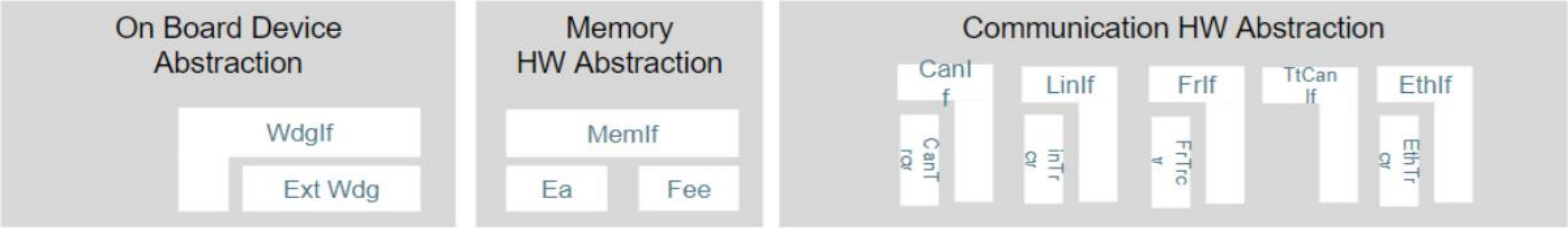
- ECU Abstraction layer (ECUAL) :
    - Offers API’s to access peripherals regardless of
        - their location (internal/external of microcontroller)
        - their connection to microcontroller (port pins , interface type)
    - It abstracts the ECU hardware layout



ECU Abstraction Layer

Overview of Software Layers/Stacks

- ECU Abstraction External Drivers Examples :
  - External EEPROM
  - External Watchdog
  - External Flash
- ECU abstraction Interface Modules Examples :
  - Can Interface (CanIF)
  - Memory Interface (MemIF)



On Board Device Abstraction

WdgIf

Ext Wdg

Memory HW Abstraction

MemIf

Ea

Fee

Communication HW Abstraction

CanIf

CanTrcv

LinIf

LinTrcv

FrIf

FrTrcv

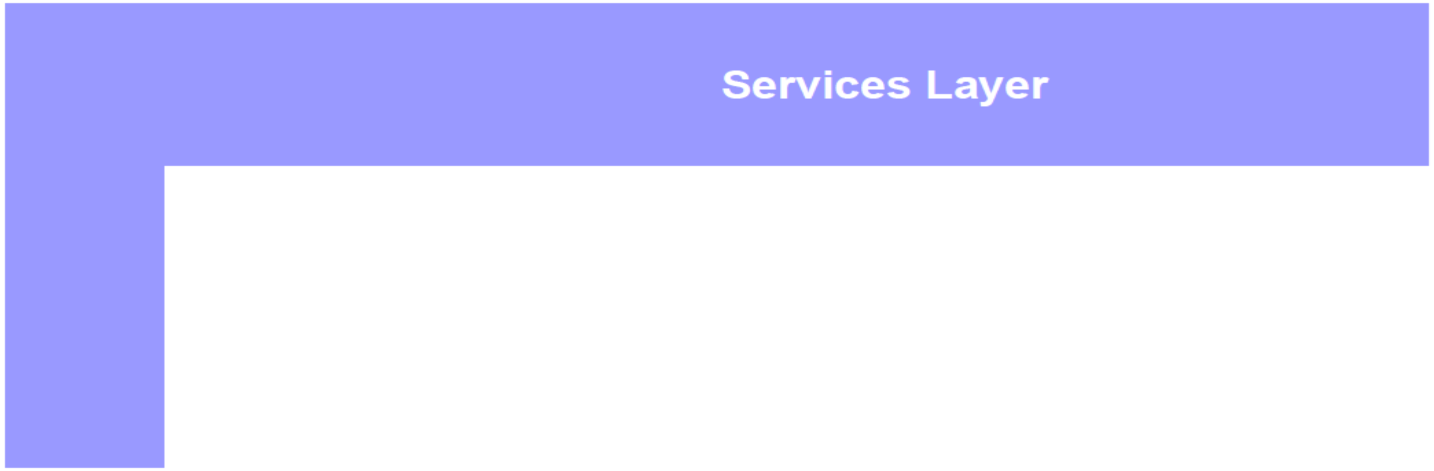
TtCanIf

EthIf

EthTrcv

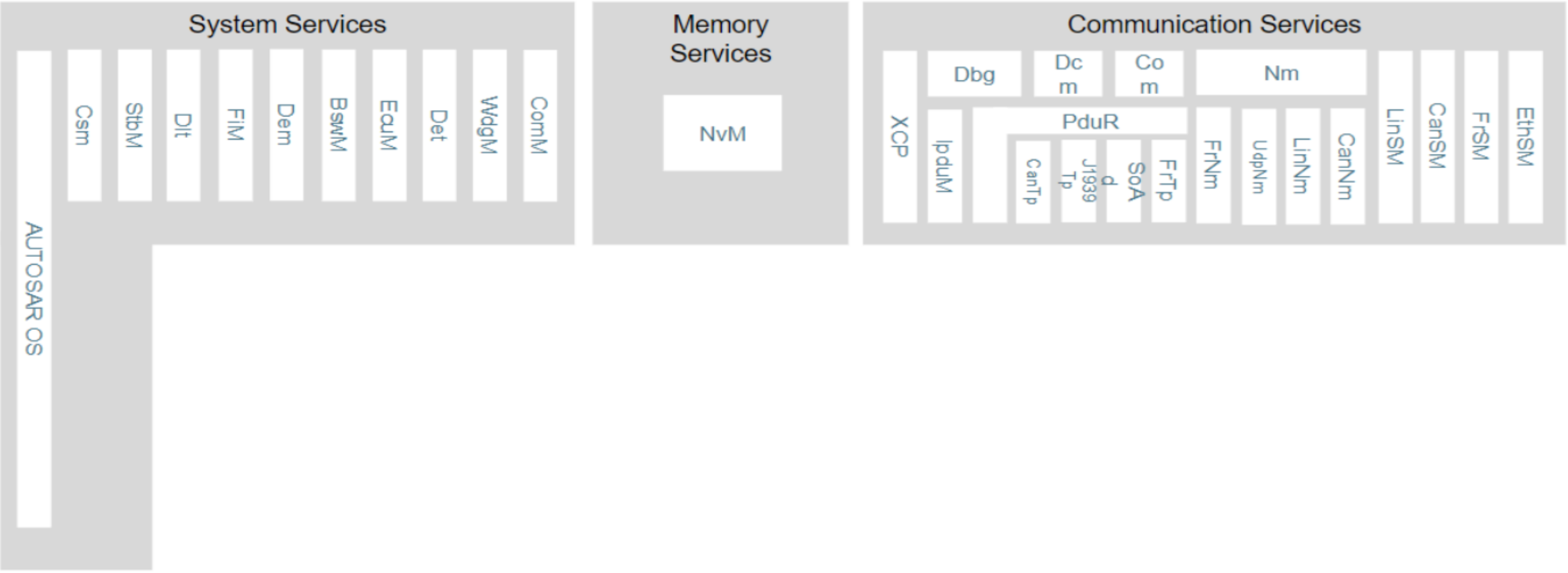
Overview of Software Layers/Stacks

- Services layer :
  - Provides basic services for application and basic software modules :
    - Operating system functionality
    - Vehicle network communication and management services
    - Memory services
    - Diagnostic services
    - Ecu state management and mode management



Overview of Software Layers/Stacks

- Service Managers Examples :
  - COM manager
  - NVRAM manager
  - ECU state manager



Overview of Software Layers/Stacks

- Complex Device Drivers (CDD)
  - Provide the possibility to integrate special purpose functionality
  - EX. driver for devices :
    - which are not specified within AUTOSAR
    - with very high timing constraints

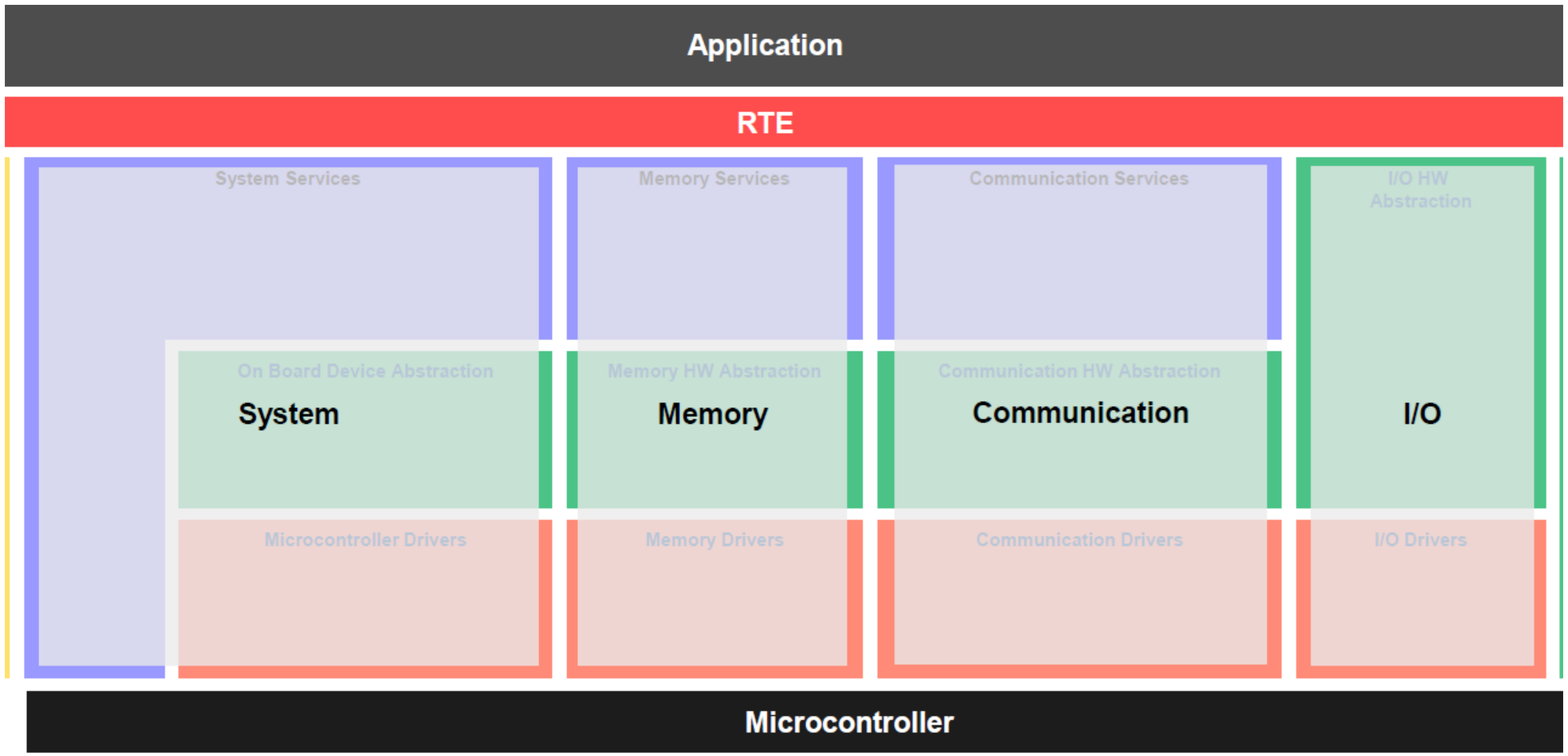
- AUTOSAR libraries :
  - Collection of functions that can be called by BSW , RTE , SWCs
  - EX. CRC , E2E , ..

Complex Drivers

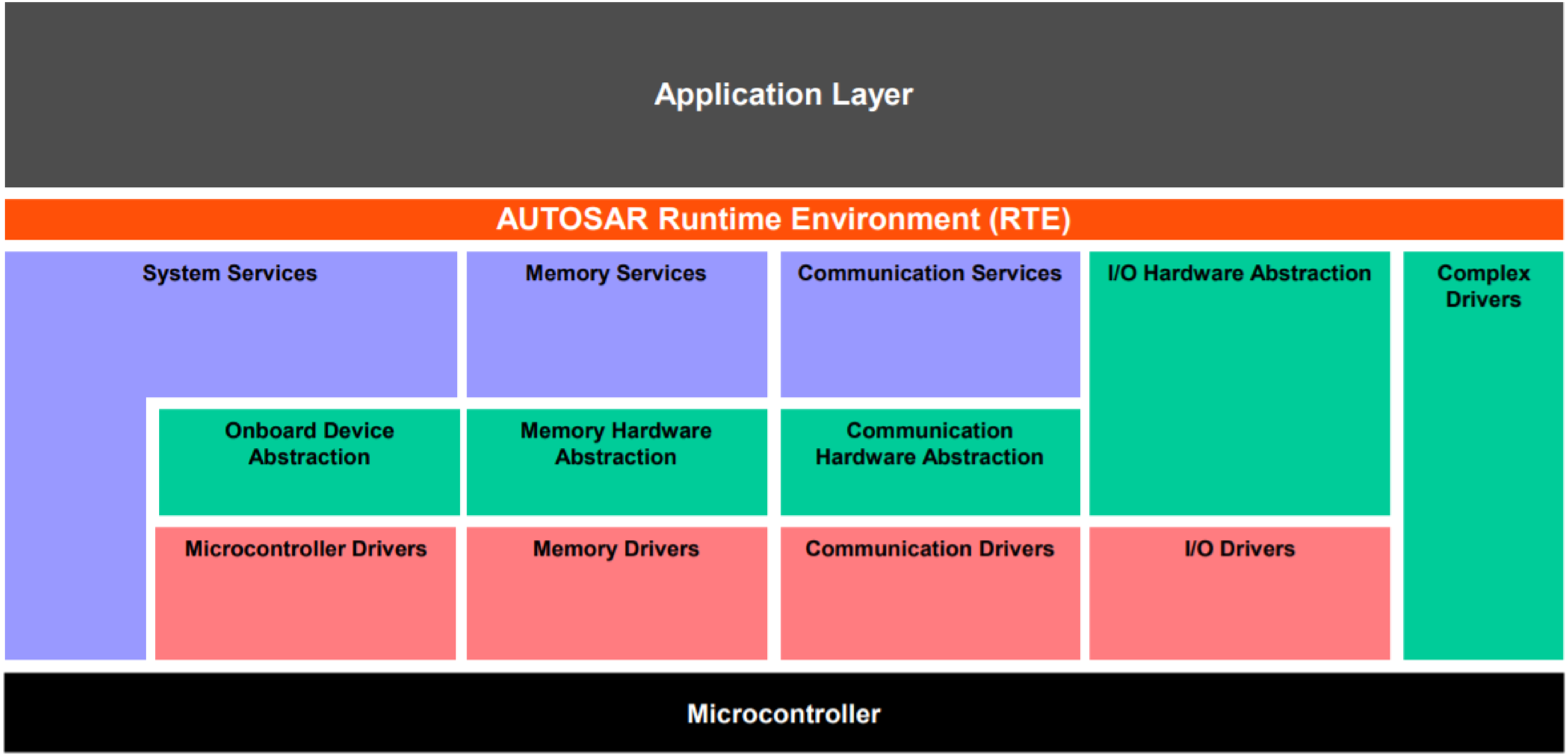
AUTOSAR Libraries

Overview of Software Layers/Stacks

- AUTOSAR architecture consists of four stacks:
  - Communication stack
  - Memory Stack
  - System Stack
  - I/O stack



Overview of Software Layers/Stacks



Application Layer

AUTOSAR Runtime Environment (RTE)

System Services

Memory Services

Communication Services

I/O Hardware Abstraction

Complex Drivers

Onboard Device Abstraction

Memory Hardware Abstraction

Communication Hardware Abstraction

Microcontroller Drivers

Memory Drivers

Communication Drivers

I/O Drivers

Microcontroller

Overview of Software Layers/Stacks

- System Stack :
    - Scheduling management
    - Error management
    - Mode management
    - Watchdog management

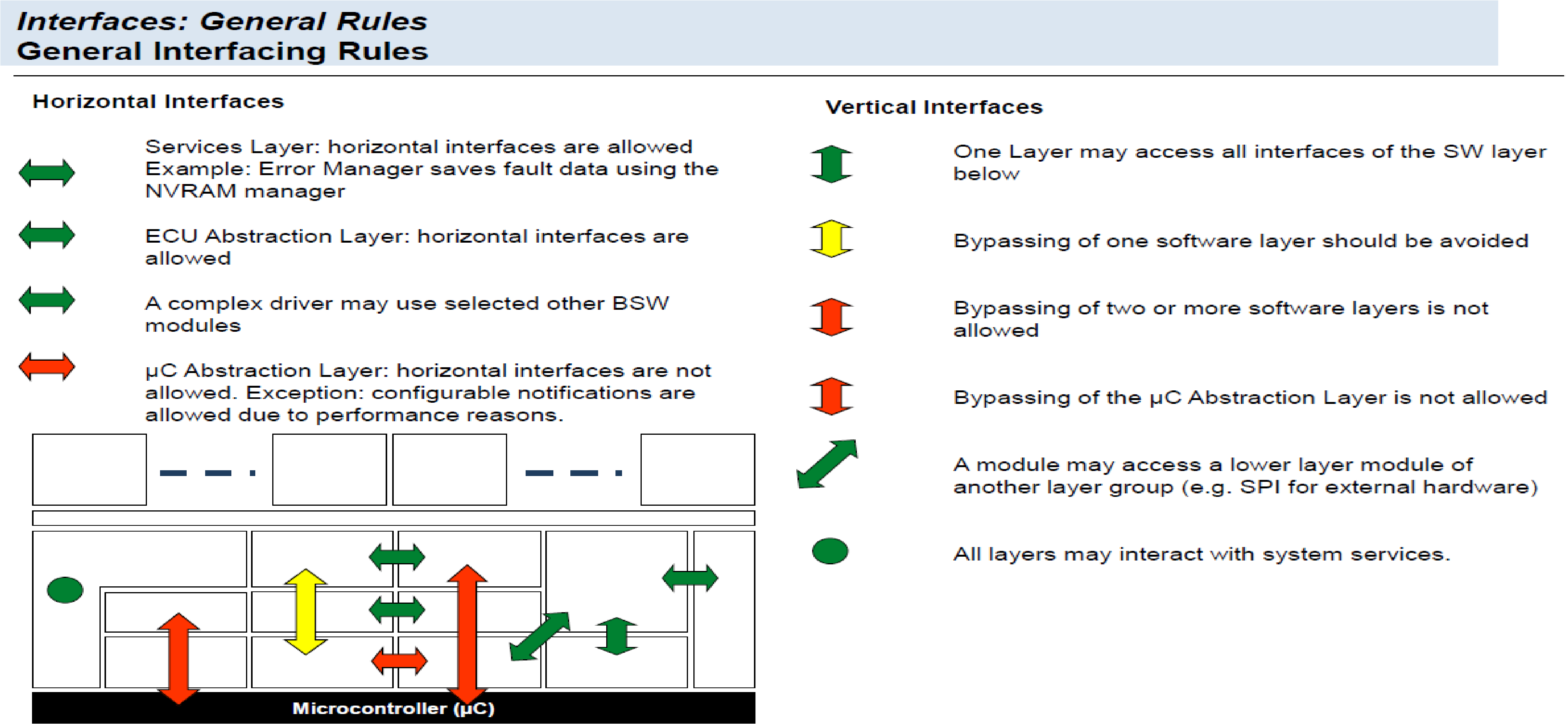
- Memory Stack :
    - Services for reading/writing to NV memory

- Communication Stack :
    - Message transmission/reception through communication protocols (CAN, LIN, …)
    - Diagnostics
    - Network management

- I/O Stack :
    - Management of input/output peripherals

Overview of Software Layers/Stacks

- Layers/Stacks interaction



Interfaces: General Rules

General Interfacing Rules

Horizontal Interfaces

⟷ Services Layer: horizontal interfaces are allowed
Example: Error Manager saves fault data using the NVRAM manager

⟷ ECU Abstraction Layer: horizontal interfaces are allowed

⟷ A complex driver may use selected other BSW modules

⟷ µC Abstraction Layer: horizontal interfaces are not allowed. Exception: configurable notifications are allowed due to performance reasons.

Vertical Interfaces

↕ One Layer may access all interfaces of the SW layer below

↕ Bypassing of one software layer should be avoided

↕ Bypassing of two or more software layers is not allowed

↕ Bypassing of the µC Abstraction Layer is not allowed

⤢ A module may access a lower layer module of another layer group (e.g. SPI for external hardware)

● All layers may interact with system services.

Overview of Software Layers/Stacks

- Interfaces’ Types
  - Provided services
    - The functions a module provides , usually to upper layer modules in the AUTOSAR layer model
  - Callbacks
    - The functions a module provides to receive notifications , usually rom a lower layer module in the AUTOSAR layer model
  - Scheduled functions
    - EX. Main functions

BSW Integration

1. Select your BSW modules :
   - Not all modules are used in all projects
2. Configure the selected modules
   - Using the configuration tool
3. Multiple iterations
   - As the setting of one module may impact other modules
4. Implement BSW Scheduling
   - Not defined by AUTOSAR (ECU integrator role)
5. Fix configuration and generation errors
6. Build generated files with static files

AUTOSAR materials

- Get all AUTOSAR documents from :
  https://www.autosar.org/standards/classic-platform/

- Get free Elektrobit AUTOSAR package from :
  https://www.elektrobit.com/products/ecu/eb-tresos/evaluation-package/

The End ...

Any questions ... ?

IMT SCHOOL

www.imtschool.com

ww.facebook.com/imaketechnologyschool/