TODAY’S TOPIC

AUTOSAR Introduction

AUTOSAR Architecture& Layers

AUTOSAR Methodology

1. AUTOSAR Introduction

Q. What is AUTOSAR?

AUTOmotive Open System Architecture

Q. Why Autosar?

Managing the growing complexity of automotive E/E systems.

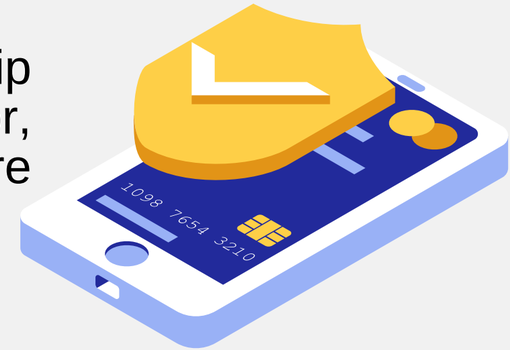
Flexibility for product modification, upgrade and update.

Scalability of solutions within and across product lines.

Quality and reliability of E/E system

Q. What is Autosar?

AUTOSAR (AUTomotive Open System ARchitecture) is a worldwide development partnership of vehicle manufacturers, suppliers, and other companies from the electronics, semiconductor, and software industries. The AUTOSAR standard is designed to enable software standardization, reuse, and interoperability.



1. AUTOSAR Introduction

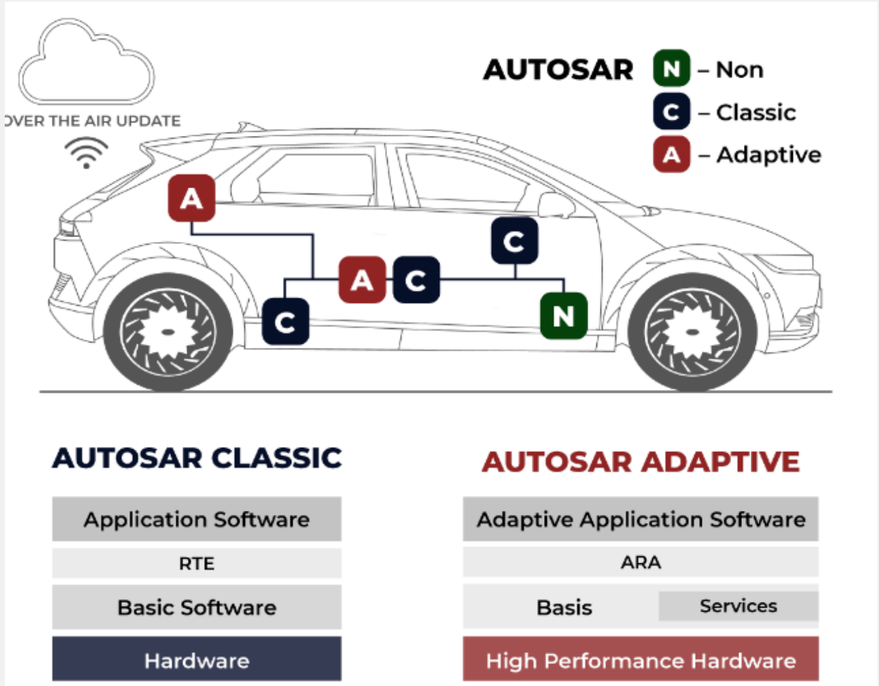
Q. Benefits?

OEMs: With AUROSAR you can use the same software code again and again for different OEMs. It is more flexible to adapt for different designs and also reduces the time and cost of production.

Suppliers: Suppliers can increase their efficiency of functional development and create their own business model that is suitable for them.

Tool Provider: AUTOSAR has a common interface that helps the tools provider to standardize their development process.

New Market Entrant: For the new entrants AUTOSAR acts as a transparent and defined interface that can help them understand the industry standards and also to create their own business models.



Q. Type of AUTOSAR?

The classic have all the modules which are generally needed for a application whereas the Adaptive can be configured and adapted according to application by removing unnecessary modules.

2. AUTOSAR Architecture && Layers

Q. Various Layers?

Application Layer: This layer has the application code which resides in top. It can have different application blocks called as Software Components(SWCs) for each feature which the ECU needs to support according to application.

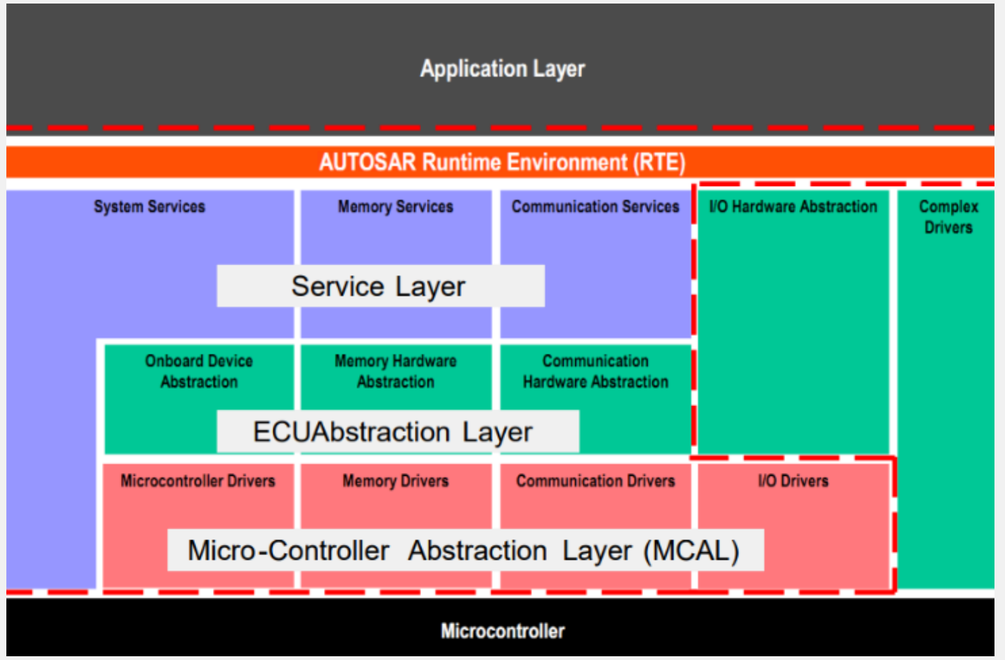
AUTOSAR RTE: This is one of the important layers of AUTOSAR, it provides communication between different SWCs and also between ECUs.

Services Layer: This layer provides different services for application to use.

ECU Abstraction Layer: This layer provides ECU related abstractions.

MCAL: This is Micro Controller Abstraction Layer it has drivers using which the above layers communicates with Micro controller hardware peripherals.

Complex Device Driver: This layer has special timing and functional requirement for dealing with complex sensors and actuators.



3. AUTOSAR MEHTODOLOGY

Q. What is it about?

The AUTOSAR Methodology outlines a sequence of steps for developing an AUTOSAR system, but the implementation of software components is relatively independent of ECU configuration.
This is an important feature of the methodology.

Q. Development Stage?

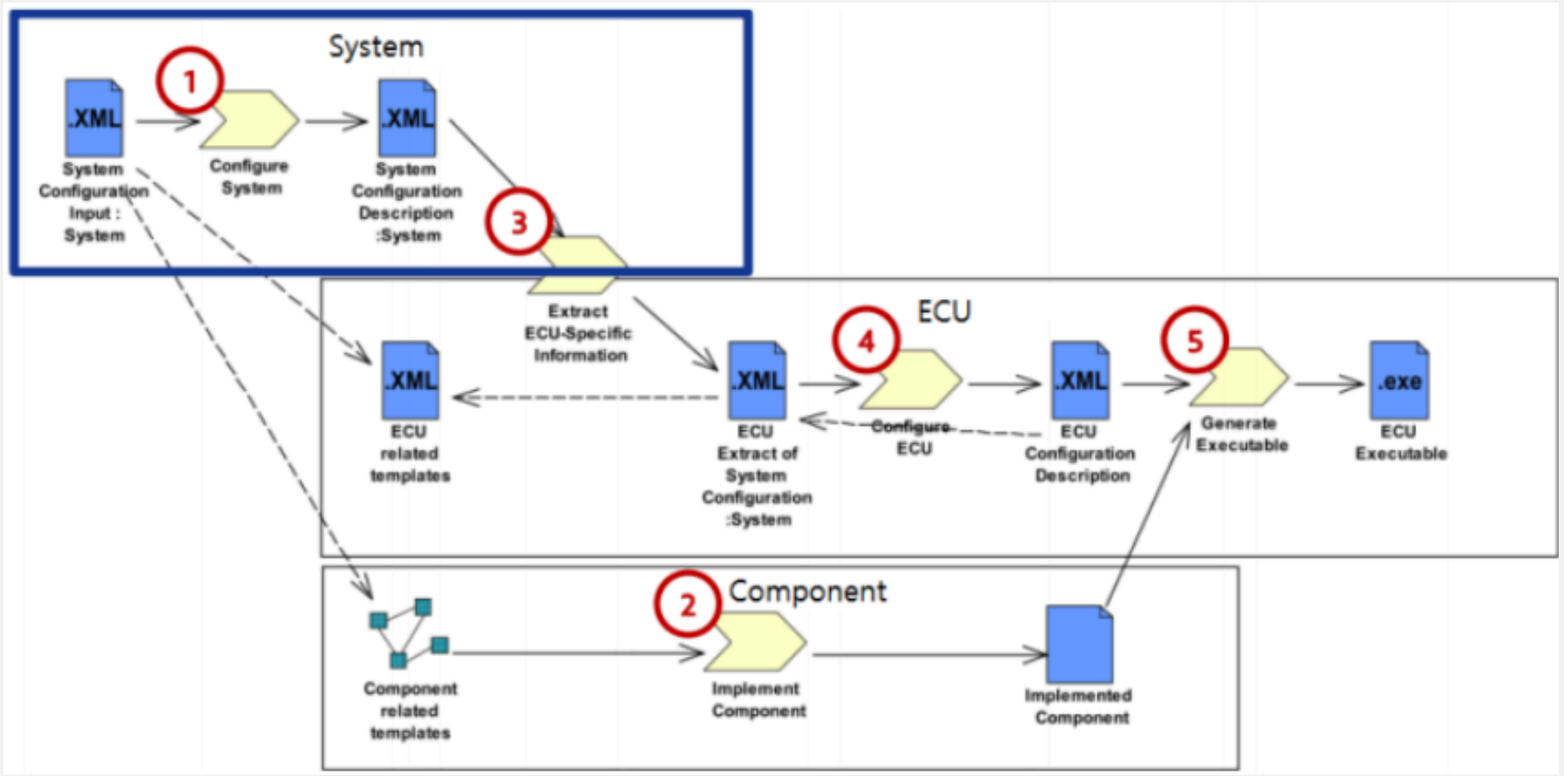
1. Configure System

2. Implement Component

3. Extract ECU-specific Information

4. Configure ECU

5. Generate Executable



3. AUTOSAR MEHTODOLOGY

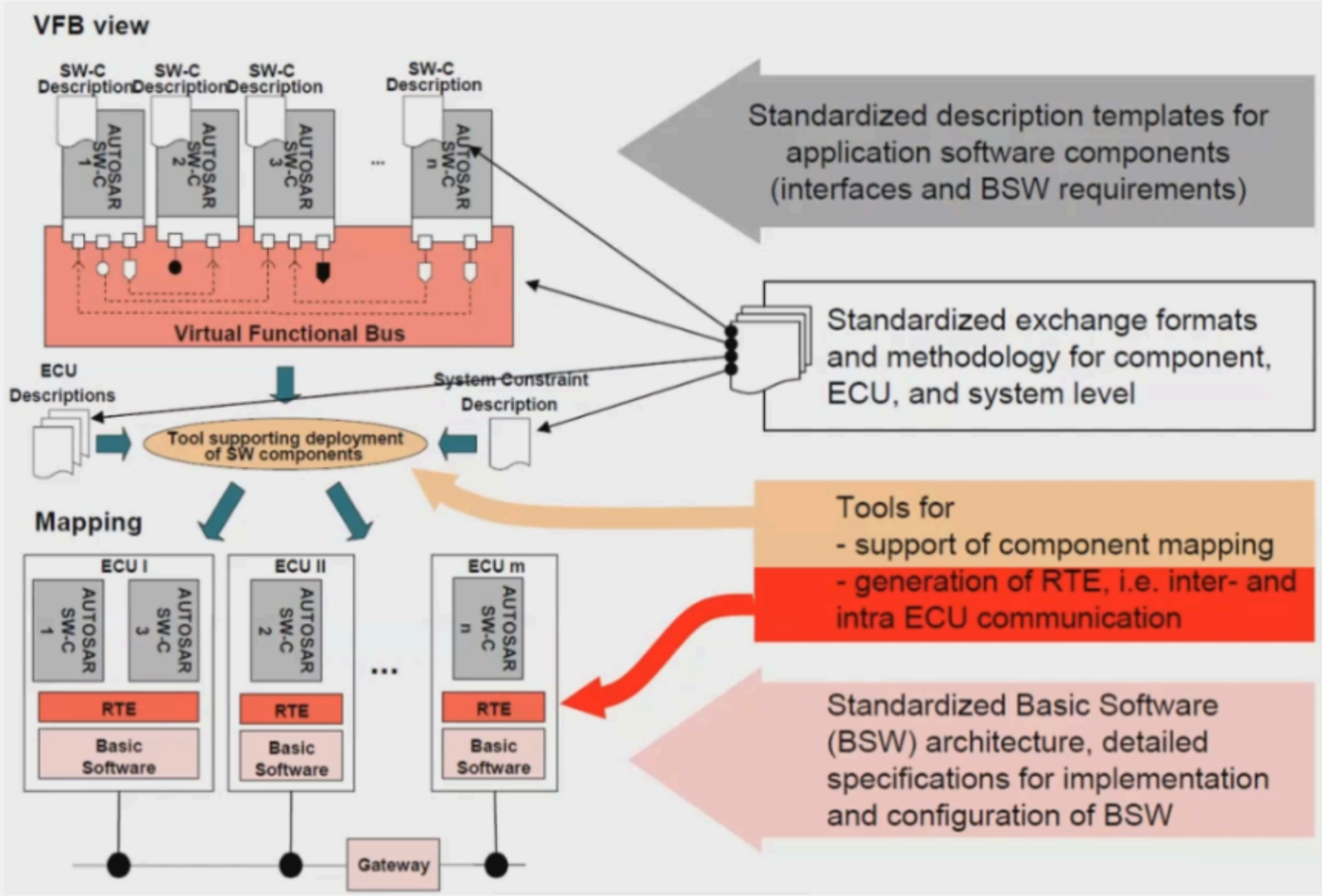
Q. How to operate Configure System?

1. Application SW
   - Design applications using interconnected software components (SW-C) without considering controllers.

2. Virtual Functional Bus
   - virtually connect communication between software components (SWC)

3. ECU Mapping
   - The process of mapping software components to specific ECUs in the system.

4. Run-Time Environment (RTE)
   - Provides concrete interfaces for component-to-component or component-to-BSW (Basic Software) interactions

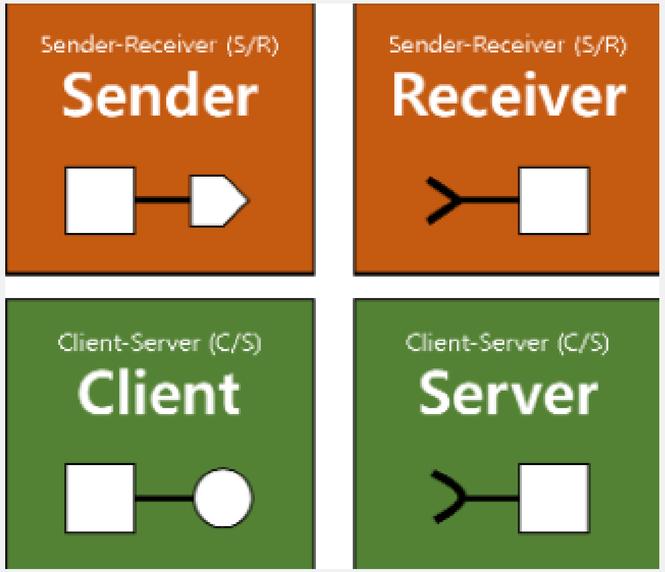


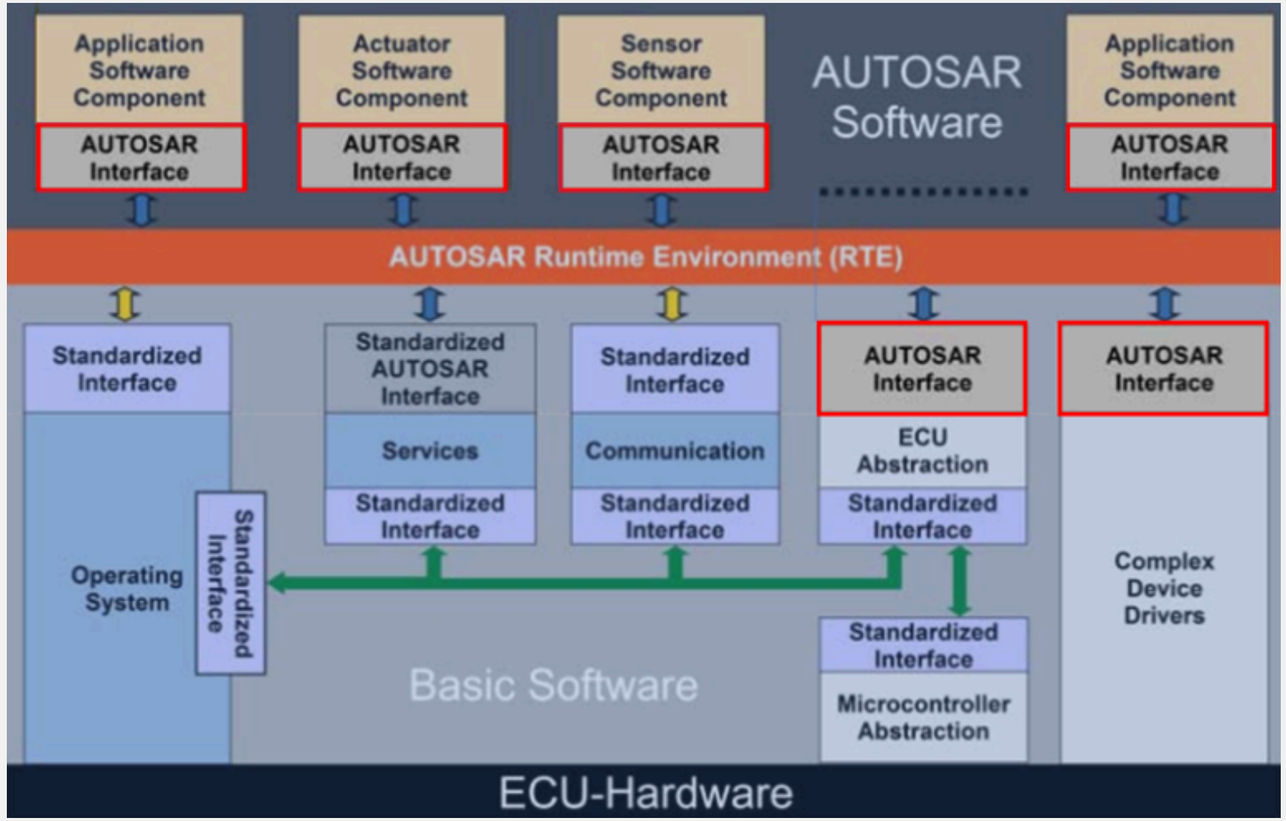
3. AUTOSAR MEHTODOLOGY

Q. Type of Interfaces?

1. AUTOSAR Interface

- abstracted by sender-receiver or client-server
- provided by the RTE and serve as interfaces between SWCs or between an SWC and the ECU firmware





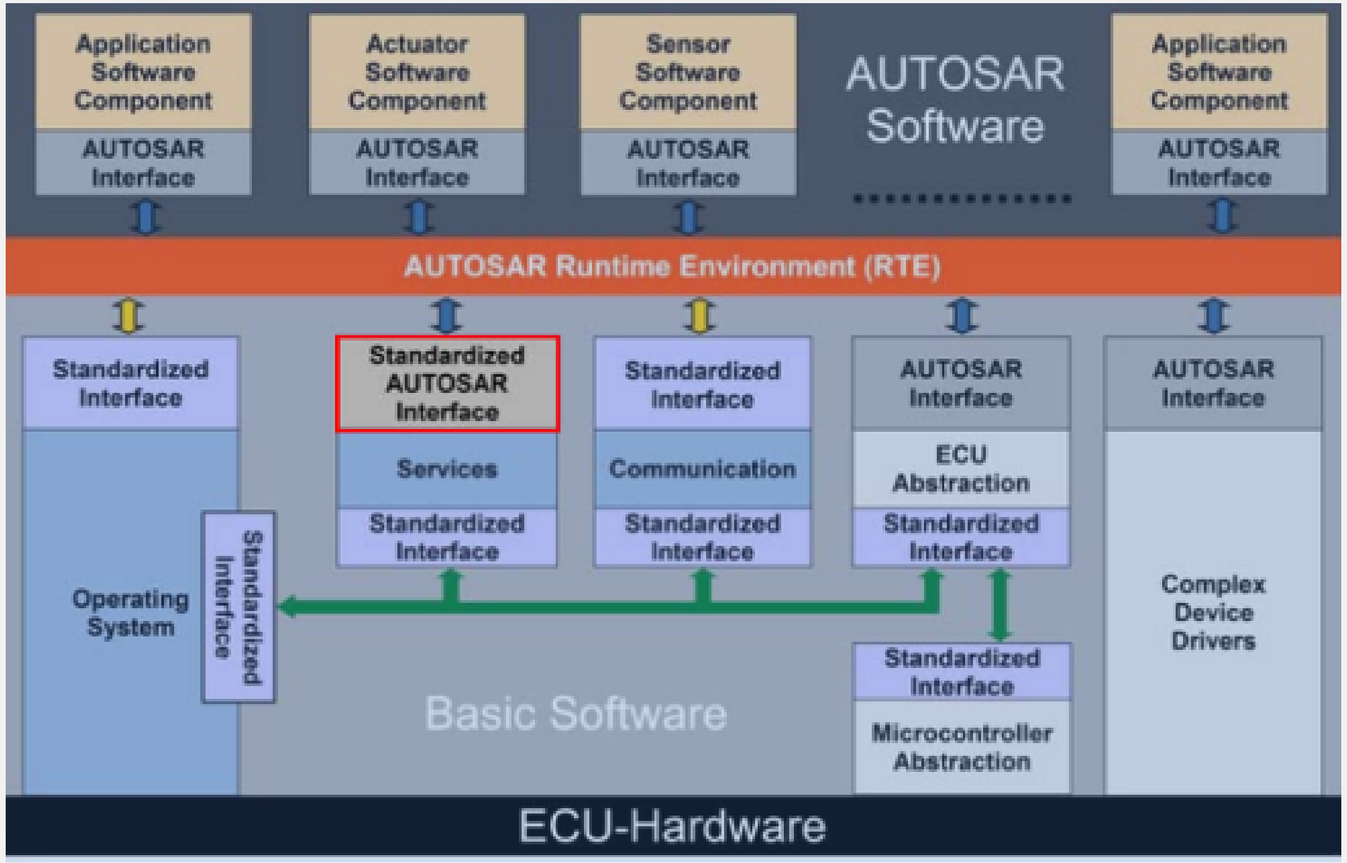
- sender - receiver : for data transmission
- client-server : for instruction conduction

3. AUTOSAR MEHTODOLOGY

Q. Type of Interfaces?

2. Standardized AUTOSAR Interface

- special AUTOSAR predefined by the AUTOSAR standard
- SWCs for access to AUTOSAR services, which are provided by BSW modules of the Service Layer, such as the ECU State Manger or the Diagnostic Event Manager.

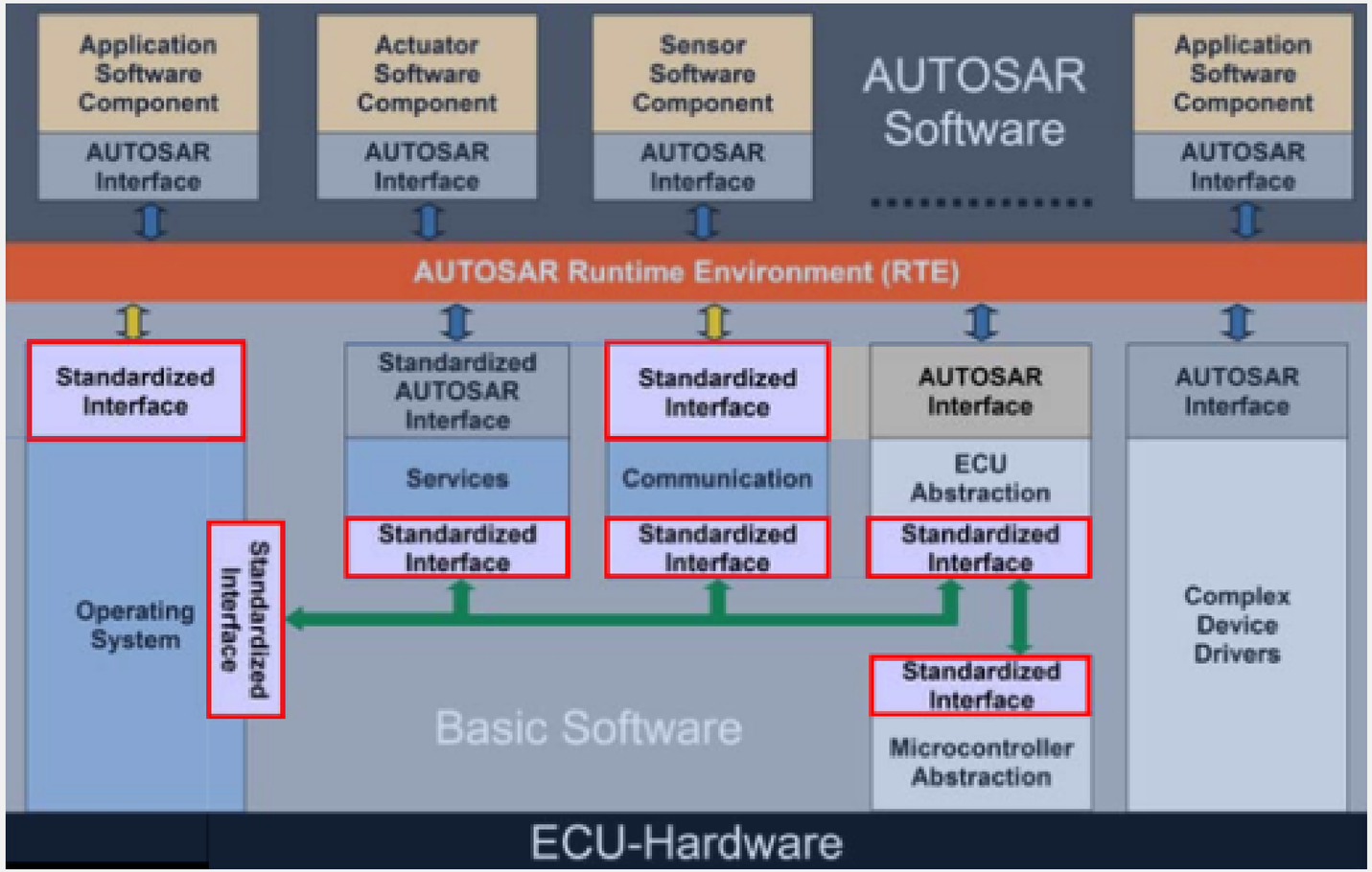


3. AUTOSAR MEHTODOLOGY

Q. Type of Interfaces?

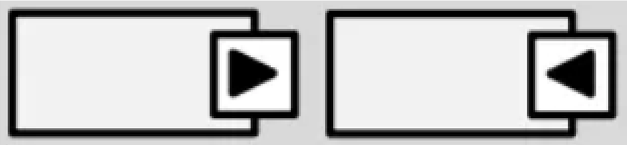
3. Standardized Interface

- predefined by the AUTOSAR standard as an API in the C language
- BSW modules in an ECU, between the RTE and the operating system or between the RTE and the BSW module Com



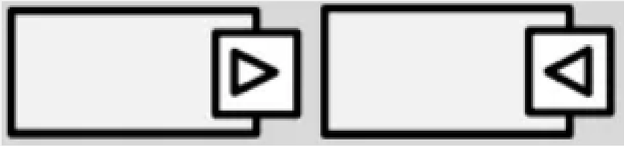
3. AUTOSAR MEHTODOLOGY

Q. Type of Ports?



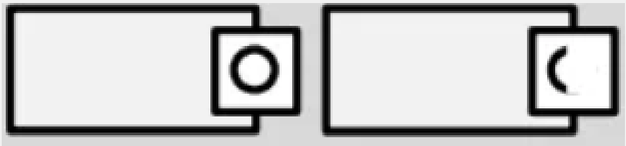
Sender (left) and receiver (right) ports symbol

- Sender Port: Sends data to other components.
- Receiver Port: Receives data from other components.



Parameter provider (left) and parameter requirer (right) ports symbol

- Used to set or read parameters of a component. Parameters typically provide initialization or configuration values.

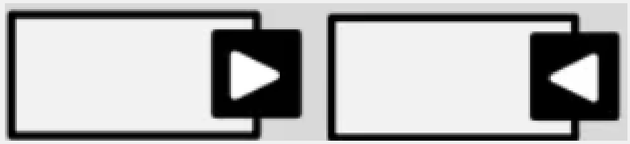


Server (left) and client (right) ports symbol

- Client Port: Requests services from other components.
- Server Port: Provides services to other components.

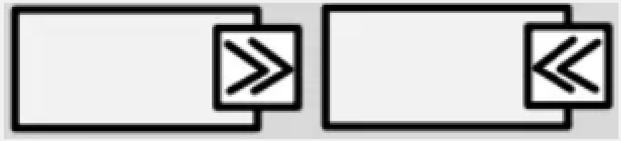
3. AUTOSAR MEHTODOLOGY

Q. Type of Ports?



Symbol of service ports

- Client Port: Initiates service requests (symbol: rectangle with inward arrow).
- Server Port: Provides services in response to requests (symbol: rectangle with outward arrow).

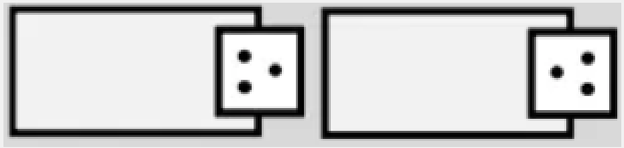


Trigger source (left) and trigger sink (right) ports symbol

- Trigger Source Port: Sends trigger signals to initiate actions (symbol: outward-pointing arrow).
- Trigger Sink Port: Receives trigger signals and initiates corresponding actions (symbol: inward-pointing arrow).

3. AUTOSAR MEHTODOLOGY

Q. Type of Ports?



Mode switch manager (left) and mode switch user (right) ports symbol

- Mode User Port: Uses system modes. It detects changes in the system mode and acts accordingly.
- Mode Manager Port: Manages system modes. It changes the system mode and informs other components about the current mode.

NvData sender (left) and NvData receiver (right) ports symbol

- Reads from and writes to non-volatile memory. It is used for data that needs to be preserved between power cycles.