## G-1 Group - AUTOSAR Architecture

|  |  |
| --- | --- |
| Group Name | Points |
| **Jyotirmay Chowdhury(coordinator) - 30639** | **1** |
| **Shikha  - 31136** | **2** |
| **Abhigyan- 30382** | **3** |
| **Nandhu -30400** | **4** |
| **Gourab Das - 30392** | **5** |

Major Layers of AUTOSAR Architecture

AUTOSAR (AUTomotive Open System ARchitecture) is a standardized automotive software architecture that separates software from hardware. It is designed to ensure reusability, scalability, interoperability, and maintainability. The architecture is layered to abstract and modularize the system effectively.

# Application Layer

- Contains Software Components (SWCs) that implement vehicle-specific functionalities.  
- SWCs are independent of hardware and communicate through well-defined interfaces (ports).  
- Examples: Engine Control, Brake Assist, Adaptive Cruise Control.  
- Supports Atomic SWCs (independent units) and Composite SWCs (combination of multiple SWCs).  
- Enables modularity and reusability of application logic.

# 2. Runtime Environment (RTE)

- Middleware between Application Layer and Basic Software (BSW).  
- Handles communication between SWCs and between SWCs and BSW.  
- Automatically generated glue code at build time.  
- Ensures location transparency and decoupling between software modules.  
- Example: Transfers sensor data from BSW to application logic without hard-coded logic.

# 3. Basic Software (BSW)

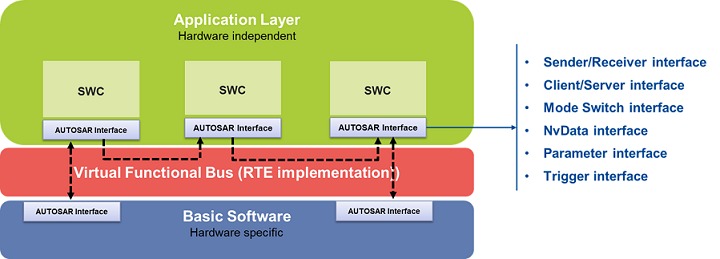
- Provides core system services and drivers for microcontroller and ECU hardware.  
- Divided into several sub-layers for modularization and abstraction:

a. Service Layer:  
 - Provides communication, diagnostic, memory, and system services.  
 - Modules: PDU Router, DCM, COM, NvM, etc.  
 - Supports over-the-air updates, diagnostics, and networking.  
  
 b. ECU Abstraction Layer:  
 - Abstracts hardware-specific details.  
 - Interfaces with on-chip peripherals (ADC, PWM, watchdogs).  
  
 c. Microcontroller Abstraction Layer (MCAL):  
 - Provides standardized drivers for microcontroller peripherals.  
 - Ensures hardware independence of higher software layers.  
  
 d. Complex Drivers:  
 - Optional, for non-standard or time-critical functionality.  
 - Example: High-speed sensor drivers, proprietary logic.

# 4. Microcontroller / Hardware Layer

- The physical hardware (e.g., NXP, Infineon microcontroller).  
- Interfaces with sensors, actuators, and performs instruction execution.  
- Base layer for AUTOSAR software execution.

# Diagram: AUTOSAR Layered Architecture



# Details

* **- Application Layer**
* Contains software components (SWCs) with runnables and ports that implement vehicle functions independently of hardware.
* **- Runtime Environment (RTE)**
* Generated middleware that routes data and method calls between SWCs and Basic Software, hiding ECU specifics.
* **- Basic Software**
* Abstracts and manages hardware via four sublayers:
* - **Microcontroller Abstraction Layer (MCAL)** – low-level drivers for on-chip peripherals
* - **ECU Abstraction Layer** – unified API for external devices and ECU interfaces
* - **Services Layer** – OS, communication stacks, diagnostics, memory management
* - **Complex Device Drivers (CDD)** – specialized drivers for nonstandard or timing-critical hardware

# Key Advantages of Layered AUTOSAR Architecture

- Hardware Abstraction: Software can run on different ECUs without change.  
- Reusability: Software components can be reused across vehicle platforms.  
- Scalability: Supports both simple and complex ECU systems.  
- Interoperability: Promotes collaboration between OEMs and suppliers.  
- Maintainability: Easier to update or isolate faults.

# Conclusion:

By organizing software into these layers, AUTOSAR achieves scalable, maintainable, and hardware-agnostic development, accelerating integration and reducing overall automotive development costs.