**Autosar**

Autosar is automotive open system architecture. It pursues the objective to create and establish an open and standardized software architecture for automotive electronic control units.

It is a common platform where applications are created.

Autosar is a basic software to communicate with particular hardware (MC) for different application.

Companies like BOSCH, BMW, GM ,FORD,TOYOTA ..etc are the core partners of autosar.

The main aim of AUTOSAR to reduce the electronic complexity in modern vehicles,

In modern vehicles there are many ECUs & each ECUs have number of functions which was performed by different hardwares so we want recode frequently once the hardware (MC) changed.

So its very important for Automobiles for software application independent with hardware , inorder to achieve this fundamentals func AUTOSAR basic software is created.

**ADVANTANGE OF AUTOSAR**

Software sharing can be possible b/w companies

Reusability of SWC

SW code can be reused

Cost & development time is reduced

Autosar consists of different layers:

1. Application
2. RTE (Run Time Environment)
3. Service layer
4. Complex drive
5. ECU abstraction layer
6. Microcontroller abstraction layer
7. Hardware(microcontroller)

BSW divided into three major layers service layer, ECU abstraction layer, MC abstraction layer

Application- In this layer software component is placed. It is hardware independent.

RTE- it is used for multiple communication. (CAN , I2C, UART , SPI, ETHERNET )

Service layer- Different functionalities like operating systems, memory services, Diagnostics services… are under this layer.

Complex drive-used for some special occasions. Directly access Microcontroller.

ECU abstraction layer-interface with mc abstraction layer. access the peripherals and devices

MC abstraction layer- contain internal drivers, internal peripherals. making MC independent of higher software.

Each layer includes different functions:

MC abstraction layer includes: MC drivers (WDT, GPT, MCU Driver, Core test)

Memory drivers (EEPROM driver, RAM test, internal flash driver, flash test)

Crypto drivers

Wireless communication drivers (Bluetooth, WIFI)

Communication drivers (CAN, flex Ray, ethernet, SPI, Lin)

I/O drivers (port driver, ADC, PWM driver, ICU driver, DIO)

ECU abstraction layer includes: onboard device abstraction, memory hardware abstraction, crypto hardware abstraction, wireless communication HW abstraction, communication hardware abstraction, i/o hardware abstraction.

Service layer includes: system services, memory services, crypto services, offboard communication services, communication services.

VFB : virtual function bus

In application layer there are so many software components are places as per the particular applications (swc1, swc2, sw3,sw4….etc) each of the software compoents communication is based on the hardwares it may use many (can , i2c, lin, ethernet , uart) , but each of the connected via a single bus called VFB.