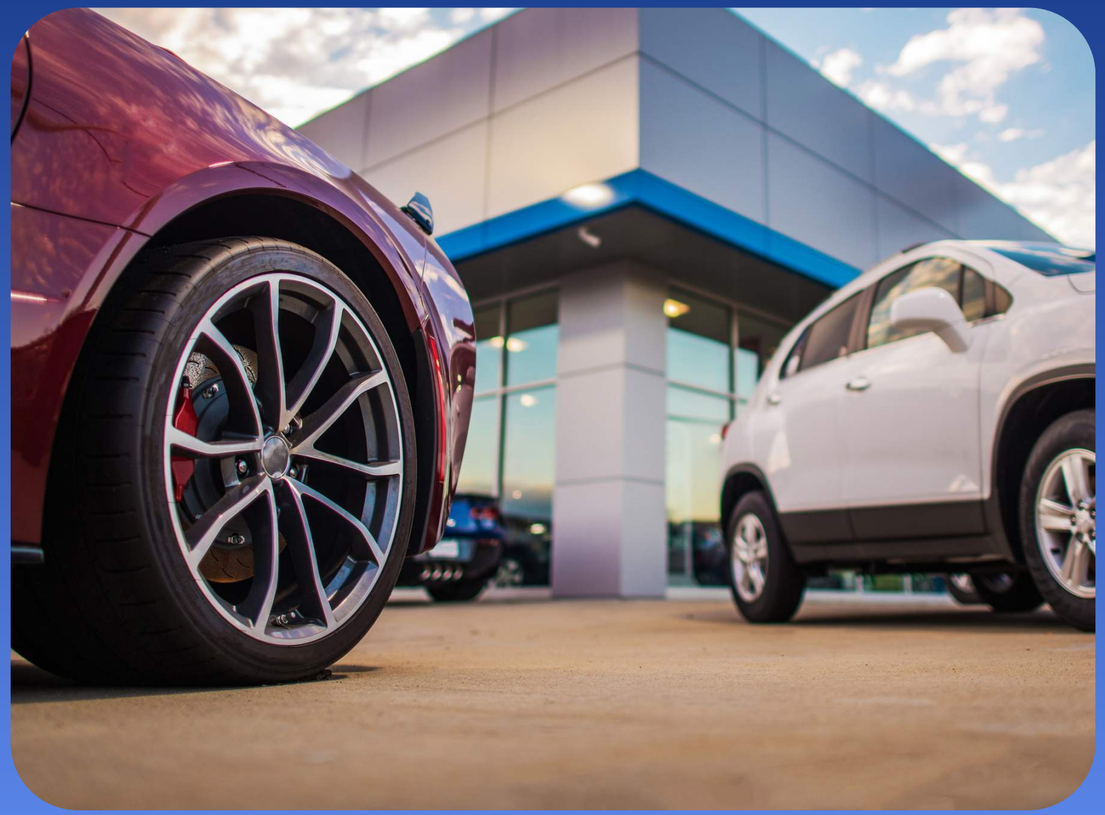
AUTOSAR TRAINING
COURSE CONTENT

Comprehensive AUTOSAR
Fundamentals and Implementation

AUTOSAR

OVERVIEW AND FUNDAMENTALS



Introduction to AUTOSAR

What is AUTOSAR?
(Automotive Open System Architecture)
Historical evolution and partnership structure Benefits and challenges of AUTOSAR adoption AUTOSAR Classic Platform vs. Adaptive Platform

AUTOSAR Architecture Concepts

Layered architecture overview Virtual Functional Bus (VFB) concept
Service-Oriented Architecture (SOA) in Adaptive AUTOSAR
Methodology and workflow

Application Layer

Software Components (SWCs) Ports and Interfaces (Sender-Receiver, Client-Server) Runnable Entities and Events Component Internal Behavior

AUTOSAR LAYERED ARCHITECTURE

AUTOSAR LAYERED ARCHITECTURE

Runtime Environment (RTE)

Today, it's driven by automation, digital design, and globalized operations, offering higher quality, consistency, and speed across all production phases.

Basic Software (BSW) Layer

Service Layer : Diagnostics, memory, communication services ECU.
Abstraction Layer : Hardware-independent interfaces Microcontroller.
Abstraction Layer (MCAL) : Direct hardware access Complex Drivers: Non-standardized functionality

AUTOSAR METHODOLOGY AND TOOLS

AUTOSAR Methodology

System configuration (System Description) ECU configuration (ECU Extract) Software component configuration Integration process and workflow

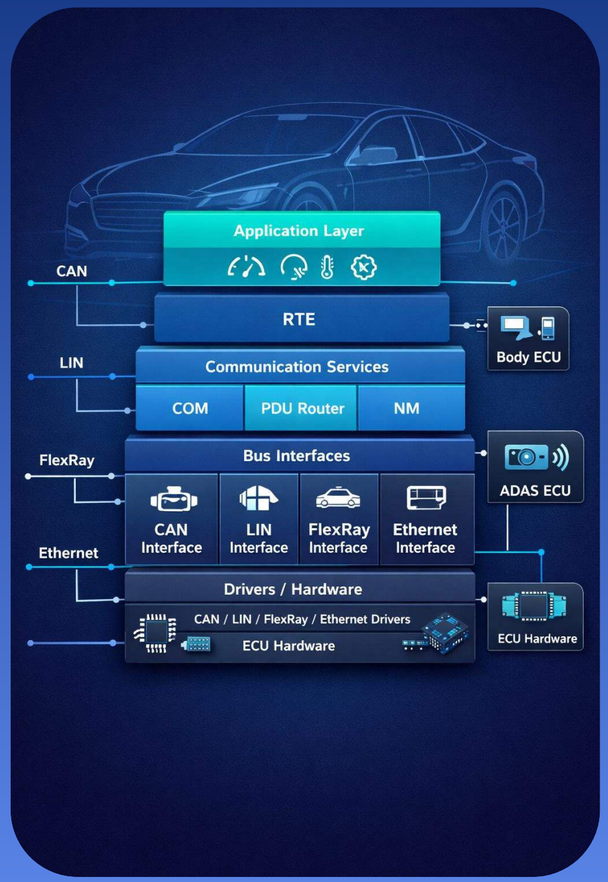
Tooling Ecosystem

Authoring tools (Vector Davinci, ETAS ISOLAR, Elektrobit Tresos) Configuration and code generation tools Integration and build tools Simulation and testing tools

ARXML (AUTOSAR XML)

ARXML file structure and schema Modeling software components System and ECU descriptions Best practices and common patterns

AUTOSAR METHODOLOGY AND TOOLS



COMMUNICATION STACK (COM STACK)

Communication Architecture

PDUs, SDUs, and signals CAN, LIN, FlexRay, Ethernet communication CAN Stack: CAN Interface, CAN Transport, CAN Network Management CAN NM, PDU Router, COM module

Diagnostics

Diagnostic Communication Manager (DCM)
Diagnostic Event Manager (DEM)
Diagnostic over CAN (DoIP) and UDS Error handling and fault memory

Memory Stack & Microcontroller Abstraction Layer (MCAL)

NVRAM Manager (NvM) Memory mapping and allocation Flash drivers and EEPROM emulation Data integrity (CRC, ECC)

GPIO configuration ADC/DAC drivers PWM and capture modules Watchdog and clock management SPI, I2C, UART drivers

MEMORY AND MICROCONTROLLER ABSTRACTION

AUTOSAR OS

Task management and scheduling Interrupt handling (ISR1, ISR2) Resource management (RES_SCHEDULER) Protection mechanisms (OS Application, Memory Protection)

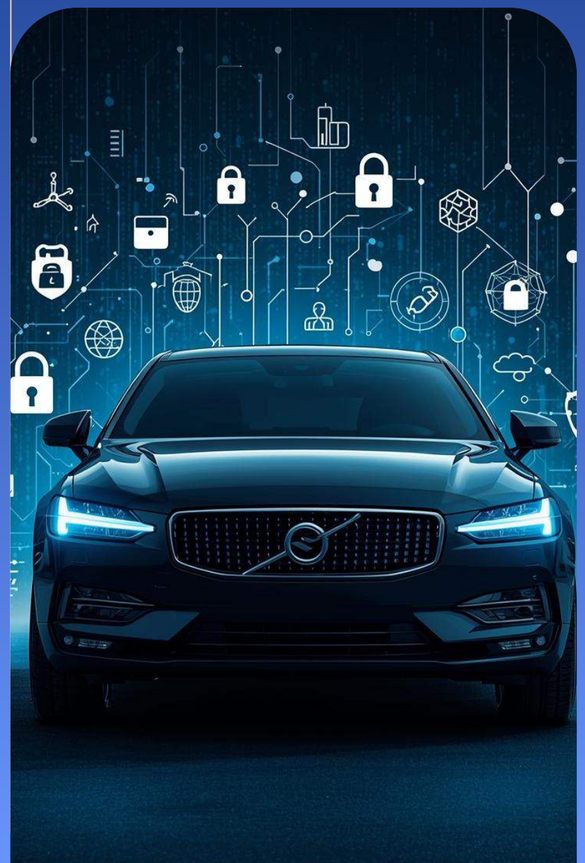
Timing and Scheduling

Timing constraints and requirements Scheduling tables (OSEKtime) Mode management Execution time measurement

OPERATING SYSTEM AND SCHEDULING

SAFETY AND SECURITY

SAFETY AND SECURITY

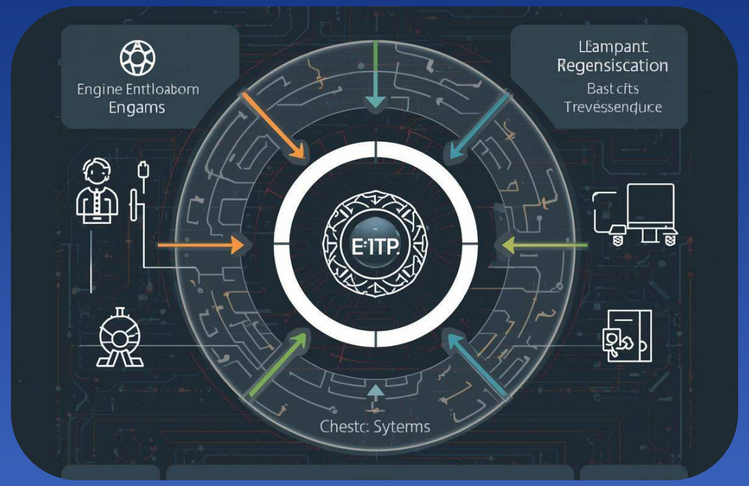


Functional Safety (ISO 26262)

Safety concepts in AUTOSAR BSW Modules for Safety: WdgM, FIM, E2E Protection Safety mechanisms and diagnostics ASIL decomposition and requirements

Automotive Security

Cryptography stack (Crypto Service Manager, Crypto Driver) Secure communication (SecOC) Intrusion detection and prevention Secure boot and firmware update



INTEGRATION AND TESTING

INTEGRATION AND TESTING

Integration Process

Integration of BSW modules RTE generation and integration Multi-core ECU considerations Integration with legacy code

Testing Strategies

Unit testing for SWCs Integration testing with RTE Communication stack testing Timing and performance analysis

ADVANCED TOPICS AND TRENDS

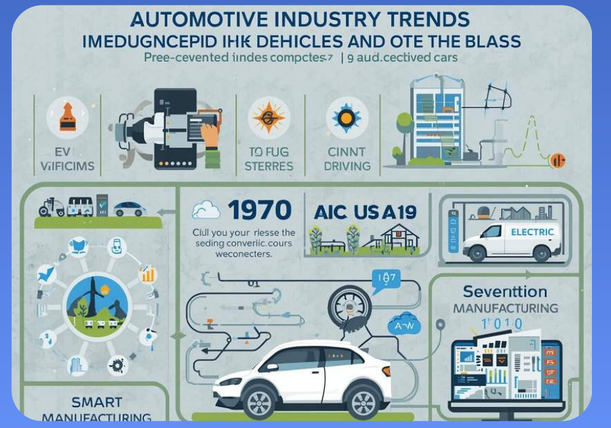
Current AUTOSAR Standards

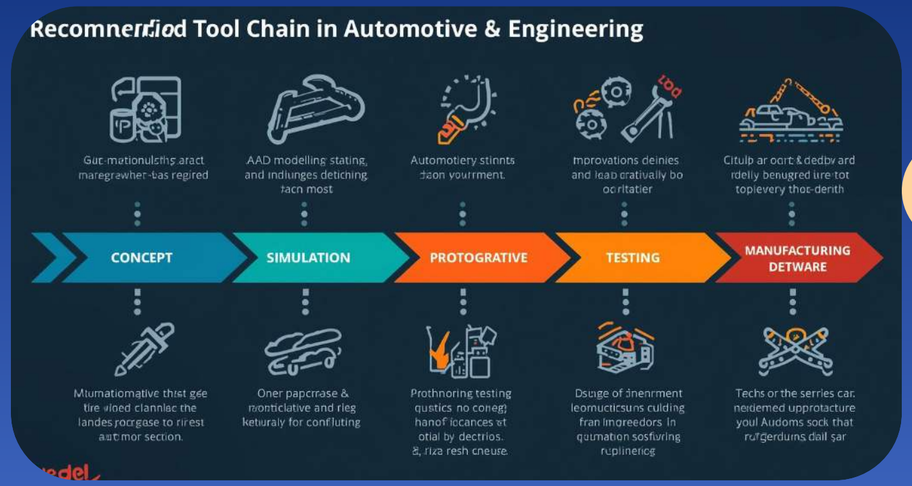
AUTOSAR Release 21-11 and newer features Electrification-specific modules SOA in automotive and cloud integration

Industry Trends

AUTOSAR and autonomous driving Mixed-criticality systems Cloud-native automotive software Continuous Integration/Deployment in automotive

ADVANCED TOPICS AND TRENDS





TOOLS AND ENVIRONMENT

TOOLS AND ENVIRONMENT

Recommended Tool Chain:

Modeling: Vector Davinci Developer/Configurator Code.
Generation: EB Tresos, ETAS.
ISOLAR Simulation: CANoe, Vehicle SPY, vVIRTUALtarget.
Hardware: Infineon Aurix TC3xx, NXP S32K.

Development Environment:

MATLAB/Simulink with AUTOSAR support Eclipse–based AUTOSAR tools Git for configuration management Jenkins for CI/CD pipelines

LEARNING OUTCOMES

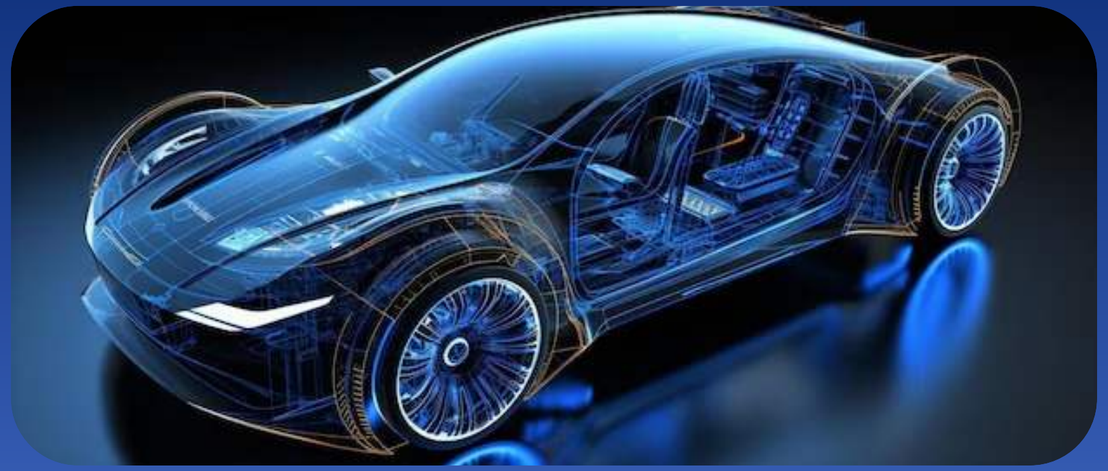
UPON COMPLETION, PARTICIPANTS WILL BE ABLE TO:

1. Understand AUTOSAR architecture and methodology
2. Design and configure Software Components
3. Configure and integrate BSW modules
4. Implement communication stacks (CAN, Ethernet)
5. Work with AUTOSAR tools and ARXML files
6. Apply safety and security concepts
7. Differentiate between Classic and Adaptive platforms
8. Integrate and test AUTOSAR-based ECUs

CERTIFICATION PREPARATION

AUTOSAR CERTIFIED PROFESSIONAL
SYLLABUS COVERAGE SAMPLE QUESTIONS AND PRACTICE TESTS INDUSTRY BEST PRACTICES AND PATTERNS COMMON PITFALLS AND TROUBLESHOOTING

THANK YOU



Get In Touch

Automotive manufacturing continues to evolve, blending innovation and sustainability. Thank you for joining us. For more insights, contact our team. Together, let's shape the future of mobility.

+91 8851002535

www.autosarindia.com

alignteam@aligntechnology.in

Sector-51, Noida-201301