

KPIT



Migration of Non-AUTOSAR Application to Adaptive AUTOSAR Application

Bharathi Vetsa

Ramesh CV

Praveen Pandey





Agenda

- 1.** Emerging Trends & Challenges Shaping Evolving Powertrain Requirement
- 2.** Industry wide focus on Adaptive AUTOSAR for Powertrain
- 3.** Mapping Powertrain Trends from Classic to Adaptive AUTOSAR
- 4.** Evolution in Software Architecture: Classic Vs. Adaptive
- 5.** Migration workflow and common issues

Emerging Challenges Shaping Evolving Powertrain Requirement

Balancing costs and complexities in electrification

Short development cycles for vehicle launch

Navigating the intricate landscape of global regulatory standards

Robust end to end safety & cybersecurity

Software Reusability and Scalability

Over 20%

Consumer expect over 20% cost reduction for new EV models

18-24 months

Typical timeline of launching new vehicles model by new entrants

Strict & dynamic

Compliance to multiple frequently updating emission standards across regions

225%

Increase in cyberattacks on vehicles globally over last 3 years

scalable software solutions

Reusability across multiple vehicle platforms

Industry wide focus on Adaptative AUTOSAR for Powertrain Requirements

High-Performance Computing (HPC)

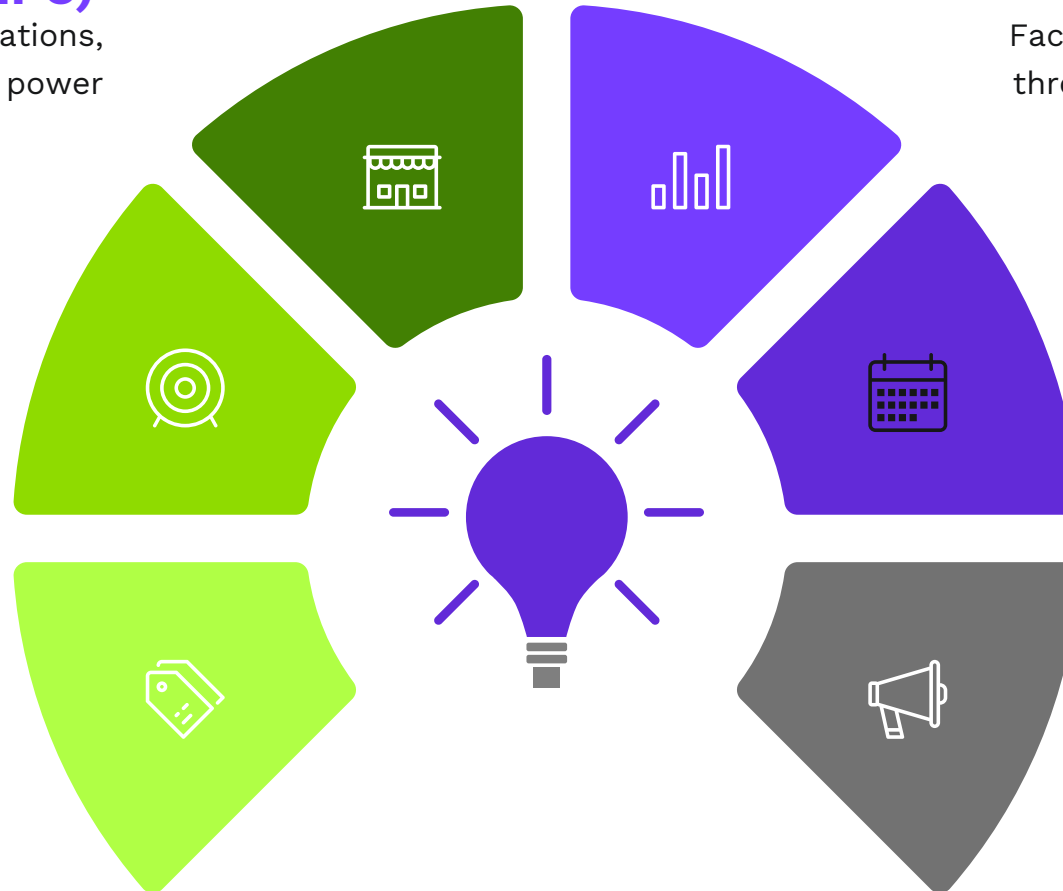
supports high-performance applications, providing the computational power

Over-the-Air (OTA) Updates

Built-in capabilities for seamless software updates

Scalable and Flexible Development

Enables significant scalability and flexibility, allowing for easy adaptation



Dynamic Data Exchange

Facilitates real-time data exchange through a publish-subscribe model

Enhanced Usability

Focuses on reusability, testability, and maintainability, simplifying development cycles

Event-Driven Execution Model

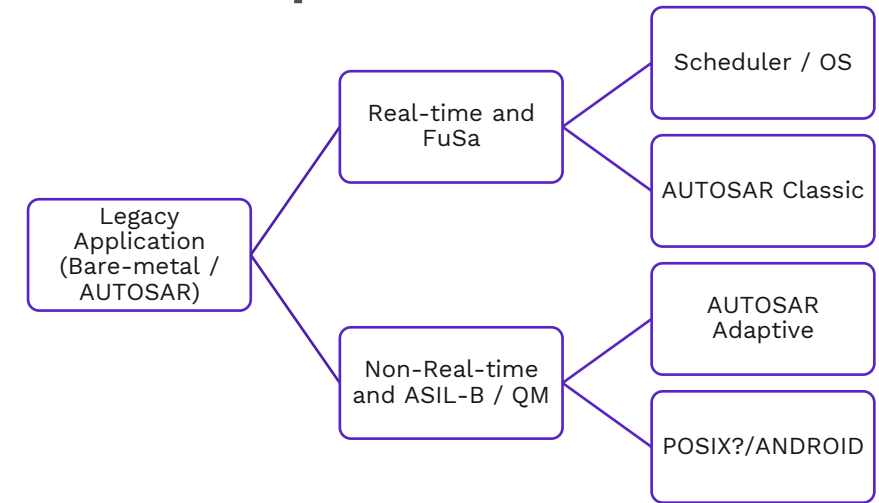
Respond dynamically to real-time conditions and inputs, improving overall system

Mapping Powertrain Trends from Classic to Adaptive AUTOSAR

Observed Industry Trend on Classic Vs Adaptive Distribution at domain level



- Classic AUTOSAR / RTOS based on multi-core CPU in safety partition
- Classic AUTOSAR / RTOS based on multi-core CPU in non-safety partition
- Adaptive AUTOSAR / POSIX OS in safety partition
- Adaptive AUTOSAR / POSIX OS in QM partition
- Linux / Android platform for e-Cockpit
- Linux platform for Edge with cloud connectivity



Migration to HPC – Features

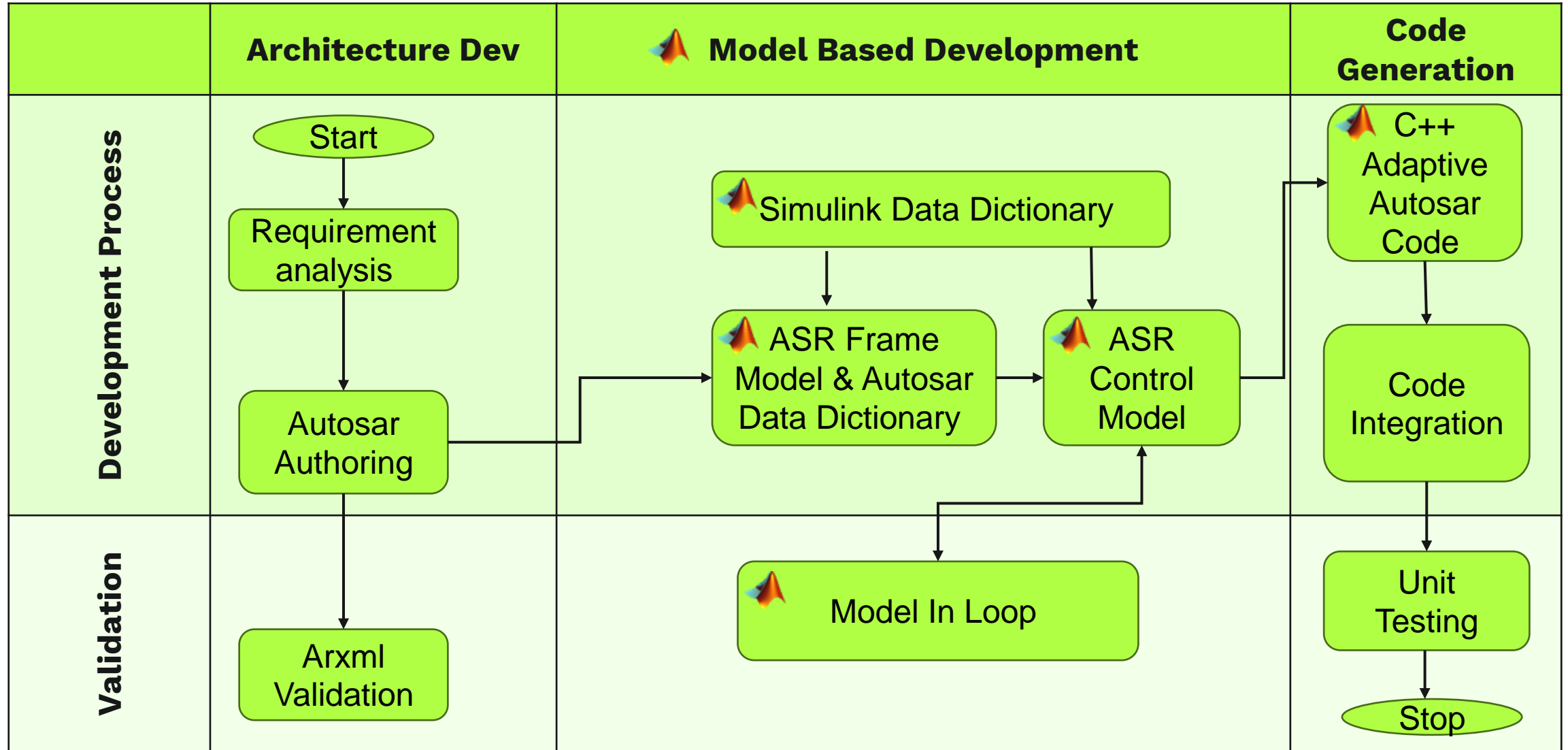
- e-PT**
- Torque arbitration
 - Battery performance and Metrics
 - Thermal management
 - Fleet Management
 - Travel guide for charging
 - Plug and Charge
 - Wireless charging
 - Data Logger

Carmakers are evaluating technical approaches to reuse and migrate relevant legacy application to Adaptive platform

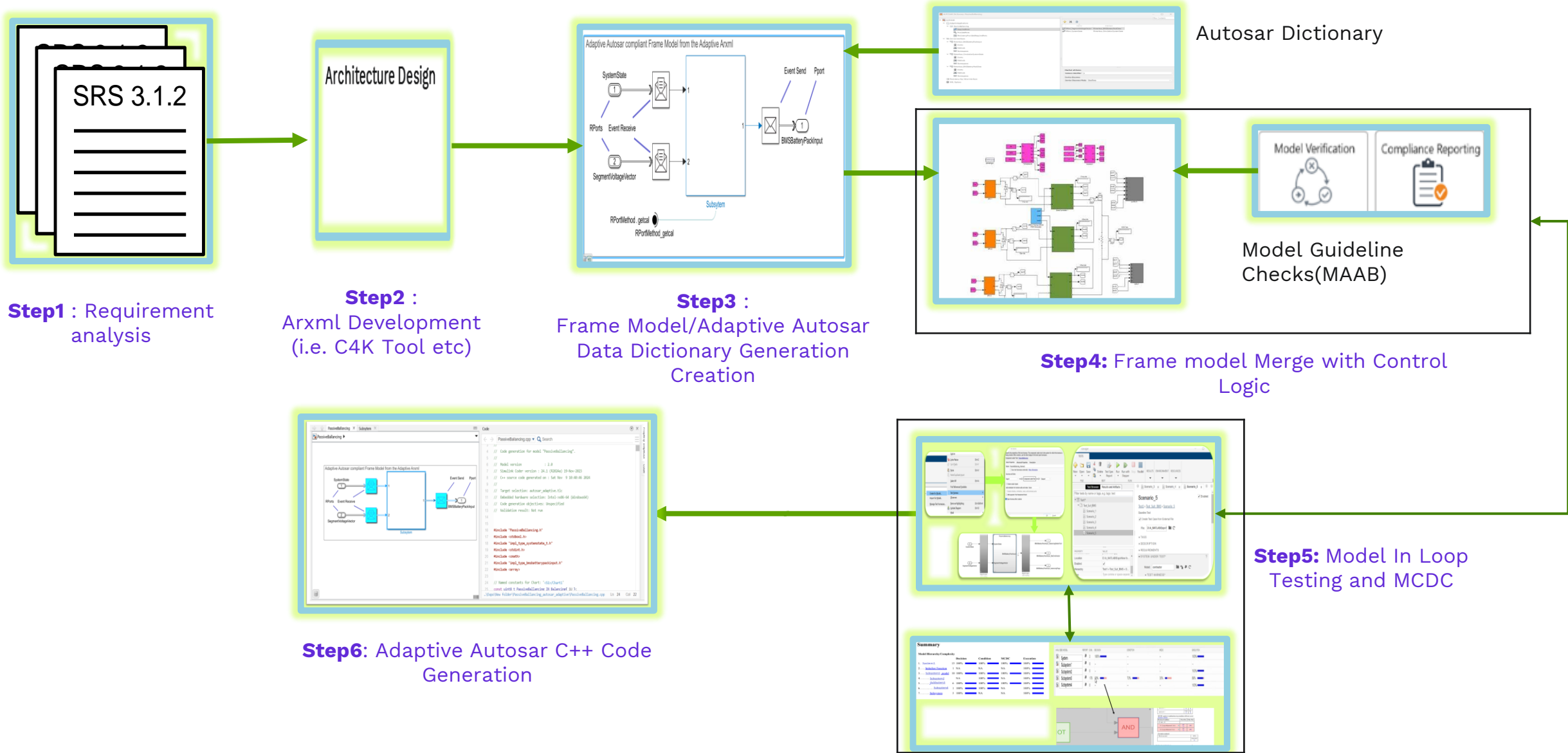
Evolution in Software Architecture: Classic Vs. Adaptive

No	Challenge	Classic Autosar	Adaptive Autosar
1	Performance	Time critical applications	High performance applications
2	Architecture	Fixed, pre-defined topology	Dynamic, service-oriented architecture
3	Execution model	Static, periodic and time triggered scheduling	Event-driven, tasks executed dynamically
4	Data Exchange	Deterministic data exchange	Publish-subscribe

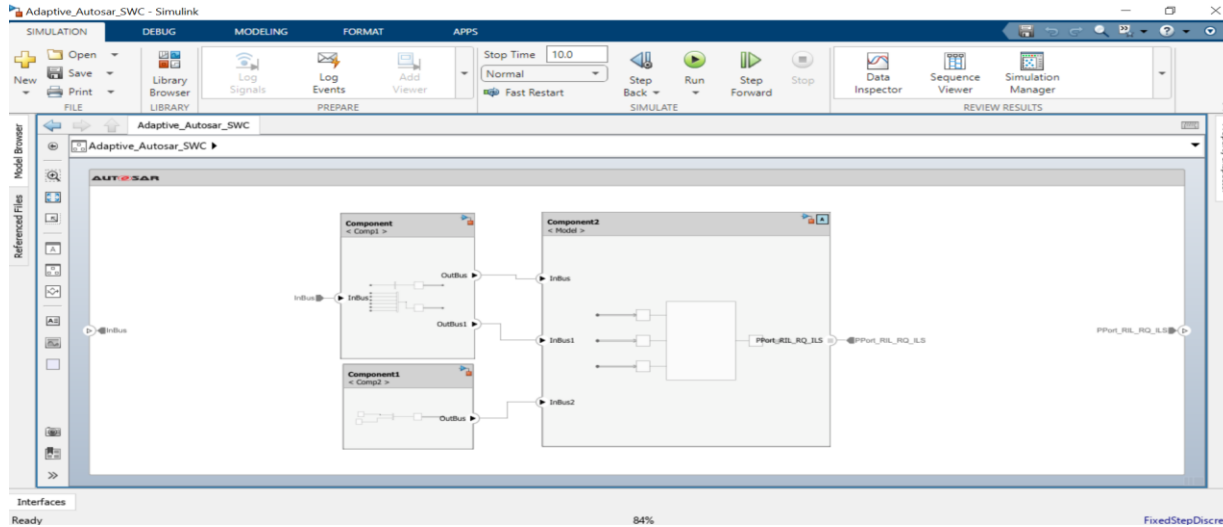
High-level Migration Workflow Using MathWorks Toolchain



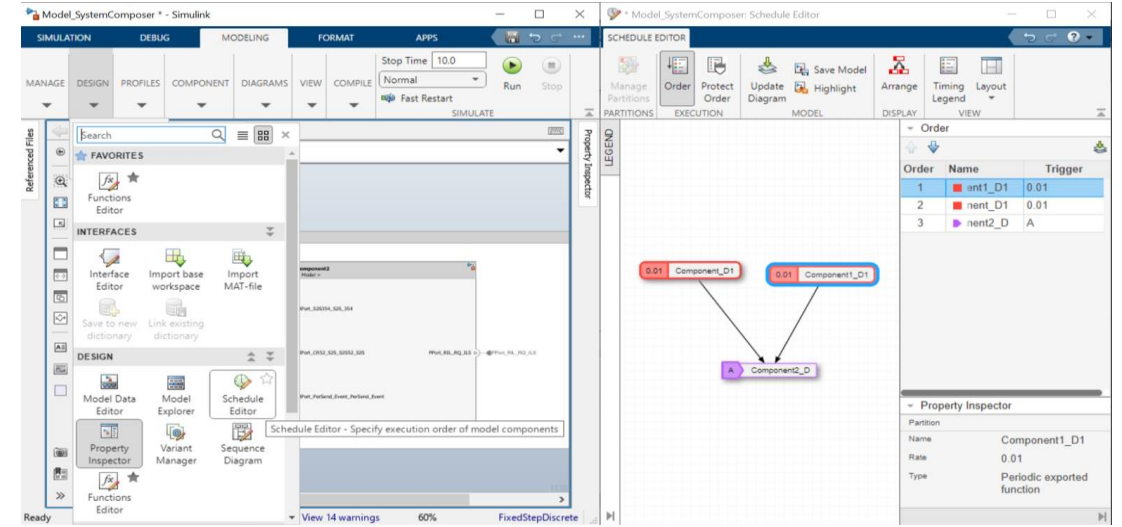
Comprehensive AUTOSAR Migration Strategy



Model In Loop Testing Using Multiple Components



System Composer



Schedule editor

- Test composition creation in system composer for connected adaptive Autosar components
- Schedule editor window used for controlling execution order of different components inside a composition

Addressing Common Issues during the Migration

Persistency path:

Challenge : Persistency interface path does not appear as per Autosar standards



Solution: Modify Autosar properties API to resolve this issue

ARA Log:

Challenge : Usage of ARA log library block does not show up in generated code



Solution: Modify ARA log tlc file in installation path for different log levels to show up in generated code

Extern Functions:

Challenge: Establishing communication handling between classic and adaptive applications



Solution: Extern function calls to be introduced in generated code using C function block or S function block

Future Scope

Shaping the future for Adaptive Autosar



SIL testing

Enabling SIL testing with Adaptive Autosar Layer



Static violations

Closer look into static violations in auto generated code



Vector type

Vector element inside a structure datatype

An aerial photograph of a multi-level highway interchange with several overpasses and ramps. The scene is overlaid with a digital network of glowing cyan circles and lines, representing a data or communication network. The circles are centered on various vehicles and points along the roadways, with lines connecting them to form a mesh. The overall color palette is dominated by the cyan of the digital overlay and the grey and asphalt of the highway.

Thank You