International Journal of Emerging Research in Engineering and Technology
Pearl Blue Research Group| Volume 6 Issue 4 PP 7-17, 2025
ISSN: 3050-922X | https://doi.org/10.63282/3050-922X.IJERET-V6I4P102

Original Article

Transitioning from AUTOSAR Classic to Adaptive for Service-Based Architectures

Saibabu Merakanapalli[1], Sai Jagadish Bodapati[2]
[1]Automotive System Architect Engineer,General Motors,USA.
[2]System Architect Engineer, General Motors, USA.

Received On: 13/08/2025 ***Revised On: 25/08/2025*** ***Accepted On: 18/09/2025*** ***Published On: 06/10/2025***

Abstract *- The growing complexity of advanced automotive system, consisting of connected, autonomous, and software-defined vehicles, has increased the pace of the transformation of the AUTOSAR framework out of its Classic Platform (CP)-optimised to solutions supporting dynamic, service-oriented architectures (SOA). Although AUTOSAR Classic has been shown to be strong in real-time control systems, it is struggling with the polluted communication, computational scalability and over-the-air (OTA) updates needs of future-generation vehicles. In this paper, a systematic transition framework to facilitate a methodical move between AUTOSAR Classic and AUTOSAR Adaptive will be presented. It has suggested that the legacy Classic components can be mapped to Adaptive service abstractions by the use of standardized middleware interfaces like SOME/IP and DDS, ensuring interoperability and downward compatibility. An execution model is devised that shows the presence of coexistence between Classic ECUs and Adaptive nodes in the form of a single service management layer. Experimental validation using a mixed ECU testbed indicates the following improvements: communication efficiency can improve up to 38% and integration time can reduce by 27% and increase scalability in distributed service deployment. The results point out that the switch to the AUTOSAR Adaptive does not only enhance the flexibility of the system but also preconditions the further automotive innovations, i.e. the use of edge intelligence, cloud integration, and autonomous control systems.*

Keywords *- AUTOSAR Classic, AUTOSAR Adaptive, Service-Oriented Architecture (SOA), Real-Time Systems, Automotive Middleware, Functional Safety.*

1. Introduction

1.1. Background and Motivation

The automobile sector is experiencing a radical change that stems out of the integration of connectivity, electrification, and automation. [1-3] Cars are no longer characterized by mechanical accuracy only but by the smartness of the software infrastructure. Contemporary cars are now distributed, software-defined systems, comprising of real-time control systems, advanced driver-assistance, and cloud-based data analytics. This paradigm shift requires a shift away of traditional signal based, statically configured architectures into flexible, service oriented, environments that can dynamically be deployed, scale in a modular manner and continuously evolve with over the air (OTA) updates. Originally created in 2003, the AUTOSAR (Automotive Open System ARchitecture) standard has helped in the standardization of software development in Original Equipment Manufacturers (OEMs) and Tier-1 suppliers. Classic Platform (CP) of AUTOSAR Offers a hard real-time, safety-certified domain, designed specifically to support hard real-time applications, including engine control, braking, and powertrain control.

But with high-performance ECUs fitted to vehicles, Ethernet-based networking and data-intensive applications, such as sensor fusion, Artificial Intelligence (AI)-based decision-making, and vehicle-to-everything (V2X) communications, the fixed configuration of the Classic Platform and lack of runtime flexibility creates major scalability limitations. In order to address these new requirements, the AUTOSAR Adaptive Platform (AP) was launched. The Adaptive Platform, which is based on a POSIX-compliant operating system, adapts to a Service-Oriented Architecture (SOA) paradigm that facilitates the dynamic service discovery, execution and communication of distributed nodes. This development is an essential transition to the non-static task scheduling of runtime flexibility, where vehicles may modify their software capabilities with the course of their life, and which opens the path to future development of next-generation features such as autonomous driving, real-time diagnostics, and the cloud.

1.2. Problem Definition

Although both the AUTOSAR Classic and Adaptive platforms have been mature and widely used, it is a complicated engineering task to have an uncomplicated interoperability and transition between the two platforms. The existing car market is highly entrenched in traditional AUTOSAR Classic elements of which a significant number have been established and proven through years of production cycles. Without jeopardizing safety or performance, these components are not readily refactorable or redeployable within Adaptive scenarios. The major obstacle is the inherent difference in communication paradigms: Classic AUTOSAR uses signal-based communication, and Adaptive uses service-based data exchange based on middleware protocols, i.e., SOME/IP and DDS. Such a mismatch causes interoperability, synchronization, and data inconsistency particularly when integrating hybrid ECUs that have to exchange real-time data. Other problems are a heterogeneous runtime environment, dissimilar memory management plans and different communication stacks which add latency, resource competition, and safety certification issues.

Current integration solutions (that are usually based on proprietary middleware or vendor-specific solutions) are not standardized and do not have scalability in the long term. These techniques do not as a rule offer formal assurances of timing determinism, safety integrity or data coherency which are all necessary in safety-critical domains. Therefore, the systematic, standardized framework to facilitate gradual transition, maintain the backward compatibility, and guarantee compliance in real-time across heterogeneous AUTOSAR environment is evident.

1.3. Objectives and Contributions

The purpose of this paper is to create a transition framework and prove its usefulness in order to migrate the AUTOSAR Classic to Adaptive implementation in a single service-based architecture. Its main aim is to allow coexistence and interoperability between the legacy and the modern automotive systems without compromising safety, performance and scalability.

The objectives of this study are the following:

- Developing a model to structure the migration of Classic objects and interfaces into Adaptive services based on standardized communication protocols, e.g., SOME/IP and DDS.
- Creation of a hybrid execution system to allow the interoperability of Classic and Adaptive ECUs to run on the same service management layer.
- Quantitative assessment of the performance of a system in a mixed-ECU system, with respect to communication latency, scalability, and overhead of integration.
- Determining key transition issues such as timing determinism, safety integrity, and data consistency and suggesting viable mitigation measures.

This work has threefold contributions. To start with, it suggests an all-encompassing migration approach, which enhances gradual adoption and coexistence of Classic and Adaptive AUTOSAR systems. Second, it gives an experimental attestation of improvements of interoperability and efficiency in performance that can be measured. Lastly, it sets a design reference architecture that OEMs and Tier-1 suppliers use to deploy service-oriented, future-proof automotive platforms, meeting the industry standards and real-time operation requirements.

2. Literature Review / Related Work

2.1. Evolution of AUTOSAR Standards

Since 2003 the AUTOSAR (AUTomotive Open System ARchitecture) standard has proven itself as the reference to the software architecture of embedded automotive systems. [4-6] The earliest and the most popular system was the Classic Platform (CP) the primary aim of this design was to standardize simple software modules and allow portability across Electronic Control Units (ECUs) along with deterministic behavior in real-time control systems. Its unchangeable design, with preprogrammed signal communications and hard schedules, has been found very appropriate in the domain of safety-critical control like engine control, braking control, and transmission control. But the increasing demand to have autonomous driving, vehicle connection, and nonstop software updates led to the emergence of the AUTOSAR Adaptive Platform (AP). Introduced in 2017, AUTOSAR Adaptive is a dynamic and service-oriented and POSIX-based environment that enables POSIX-based high-performance computing (HPC) nodes and cloud connectivity.

It presents fundamental layers of run-time like the Execution Management, the Service Discovery and the Adaptive Communication Management layers which facilitate dynamic deployment of services and smooth communication across the Ethernet based networks. The presence of two classical and adaptive platforms nowadays within the modern car E/E (Electrical and Electronic) platform is indicative of a trend of a hybrid approach- keep safety-critical functions on a Classic ECU and implement data-driven or AI-enhanced functions on Adaptive nodes. Though this dual-platform system, the systematic and standardised migration between CP and AP has not been historically studied well.

2.2. Service-Oriented Architectures in Automotive Systems

Largely the automotive field has been motivated by the use of Service-Oriented Architecture (SOA) based on its achievements in enterprise and distributed computing systems. SOA separates software packages into findable, reusable and loosely coupled services, enabling modularity and independence in the lifecycle. In the automotive case, SOA can enable run-time flexibility, scale of functions and remote update which are essential to the next generation cars. A number of papers have examined SOA integration based on technologies like SOME/IP (Scalable service-Oriented Middleware over IP), DDS (Data Distribution Service) and RESTful APIs to facilitate application of dynamic service communication.

SOA-based middleware is able to enhance system interoperability and functional reuse but they also had found limitations in deterministic timing and real-time reliability. Nevertheless, the majority of the implementations have been experimental or platform-specific, exploratory in integrating a partial range of services, but not overarching transition frameworks. The main issue is how it can retain functional safety (compiance with ISO 26262) and hard real-time assurances and incorporate the dynamism of service-discovery and runtime-binding. Consequently, there is a need to adopt a mixed-methodology between traditionally programmed static Classic ECUs and dynamically programmed Adaptive nodes that corresponds to having a deterministic control regime made dynamically flexible by runtime processors.

2.3. Comparative Studies and Transition Models

Earlier comparative studies on AUTOSAR Classic and Adaptive platforms have found some basic disparities on architecture, communication patterns and operating system interfaces. provided an empirical study on the computational efficiency and timing determinism between the two platforms and found that whereas Adaptive offers a greater scalability and performance advantage to data-driven code, Classic offers a greater predictable-execution advantage to real-time control. The middleware adequacy approach to interoperability is frequent in transition models offered a bridging architecture based on SOME/IP gateways in order to facilitate message translation of Classic to Adaptive nodes. A hybrid AUTOSAR framework that used DDS as a communication bridge that showed that mixed-domain ECUs can be used. These models however are not standardized and often do not have guidelines on integration as they are left vendor specific and cannot be scaled across OEM ecosystems. In addition, recent research has restricted itself to a simulation setting, and little to no assessment to a real-time automotive testbed has been carried out. Therefore, although initial research determines theoretical viability of Classic-Adaptive coexistence, a detailed migration and benchmarking paradigm is still a gap in research findings.

2.4. Research Gaps Identified

Based on the reviewed literature, some of the research gaps are critical

- Absence of a Unified Migration Framework: Current literature suggests a partial interoperability (e.g., middleware bridges), but none offer a standardized framework on how any particular piece of moving between might occur between AUTOSAR Classic and Adaptive.
- Minimal Real-Time Measuring: Not many studies have quantifiably compared latency, CPU load or communication efficiency in combined CP-AP environments.
- Lack of Attention toward Safety and Security Conformity: ISO 26262 safety requirements and secure communication standards, both of which are required of production grade automotive systems, are usually not given due emphasis through the transition mechanisms.
- Interoperability and Lifecycle Management Problems: Coherence between Classic and Adaptive parts requires cohesive lifecycle management, service discovery as well as diagnostic combination in places where existing remedies are still disjointed.
- Toolchain and Ecosystem constraints: Dependencies with the Vendors and their intricate configuration make hybrid architectures difficult to expand and diversify across multiple OEMs.

This paper fills this gap by introducing a complex system of transition that will consist of functional mapping, middleware abstraction, and hybrid-run-time orchestration that will support a gradual, safe, and efficient conversion between AUTOSAR Classic and Adaptive. The suggested method is confirmed by the experimental testing of a mixed ECU space providing quantifiable interoperability, scalability and performance results.

3. System Methodology and Architecture

3.1. Transition from AUTOSAR Classic to Adaptive - Architecture Overview

3.1.1. AUTOSAR Classic Platform (CP)

The AUTOSAR Classic Platform is classic embedded control systems of Electronic Control Unit (ECUs) that are based on the deterministic and safety-critical control of engine control, transmission, and braking systems. It contains a few basic modules: the Microcontroller Abstraction Layer (MCAL), an interface to hardware peripherals that is standardized across all hardware; the ECU Abstraction Layer, an interface between hardware objectives and software that is hardware-free; the Runtime Environment (RTE), a component communication manager; the Basic Software (BSW) a service containing generic system-level services such as diagnostics, communication stacks and memory management. Its rigorous scheduling, constrained dynamic flexibility, as well as high real-time performance guarantees based on its nature makes this platform ideal in hard real-time control applications.

3.1.2. Transition/Interoperability Layer

The transition layer serves as the key connector between the Classic and Adaptive platforms so that the communication and exchange of data between the heterogeneous systems will occur seamlessly. It consists of modules like Service Abstraction Layer (SAL), which implements Classic interfaces into service endpoints, and the classic-adaptive Bridge (C2A Adapter) which, in turn, is the one that characterizes the mapping of the API and sharing of data both ways Cp generating to Ap and AP generating to Cp. There is also SOME/IP Gateway, which offers the use of message-level communication in Ethernet modes, which makes static Classic ECUs trade-off with dynamically-deployed Adaptive applications. This layer provides unequivocal interoperability that has kept old systems displayed with the contemporary service based structures.

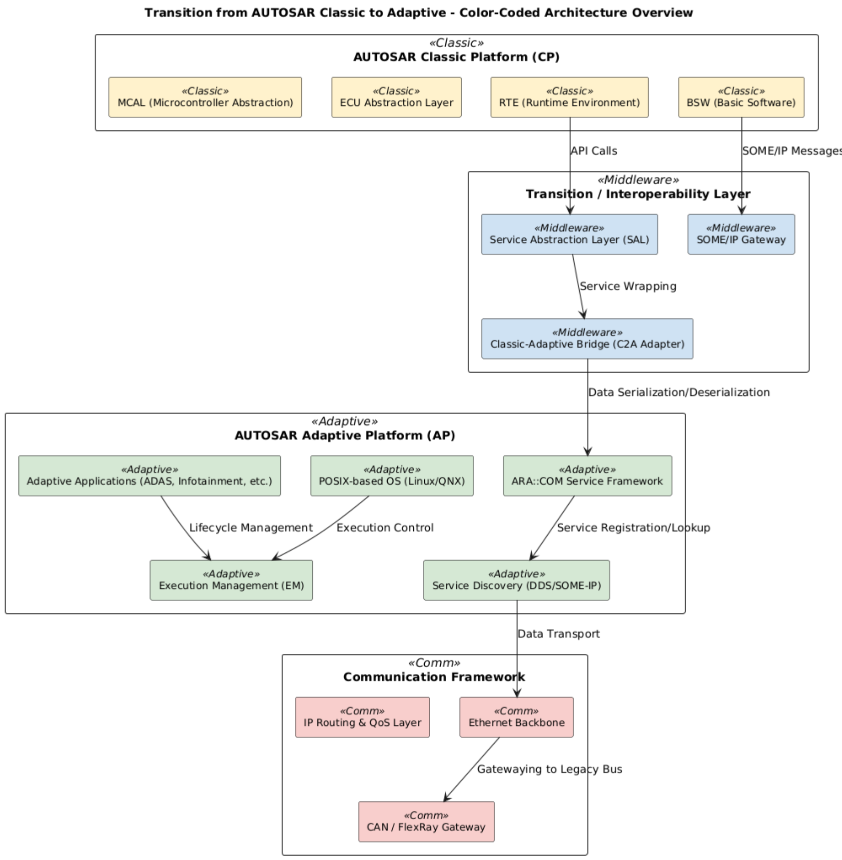

Fig 1. Transition from AUTOSAR Classic to Adaptive - Architecture Overview

The given diagram depicts the end-to-end transition architecture between the AUTOSAR Classic Platform (CP) and the AUTOSAR Adaptive Platform (AP) via a middleware interoperability layer, [7-10] as well as, a single communication framework. The color-coded visualization presents individual system layers each of them symbolizes particular functions and duties of the entire architecture.

3.1.3. Transition from AUTOSAR Classic to Adaptive

The Adaptive Platform is aimed at high-performance computing nodes with the need to be flexible, scaled and connected to the cloud. It facilitates the installation of the dynamic applications as well as supports highly functionalities like autonomous driving, ADAS and infotainment. It is based on Core buildings that consist of Adaptive Applications, POSIX based Operating System (e.g., Linux or QNX) and the ARA:COM Service Framework that offers service registration and communication mechanisms. Execution Management (EM): the lifecycle and execution states of the Adaptive application are managed by the Service Discovery (DDS/SOME-IP) module and the executions of services are provided by the SOME-IP. Services are dynamically located and connected at runtime by the Service Discovery (DDS/SOME-IP) module. A combination of these modules makes it possible to bind services dynamically, integrate into clouds, and scale behavior of systems.

3.1.4. Communication Framework

The communication framework is used as the intranet which connects the Classic and Adaptive realms. It consists of the IP Routing and Quality of Service (QoS) Layer which facilitates traffic priorities, the Ethernet Backbone which is a high bandwidth data transmission, and the CAN/FlexRay Gateway which is backward compatibility with the existing Classic ECUs. It is an antifeminist hybrid type of communication organization that promotes the interaction of heterogeneous networking, allowing old bus protocols alongside new Ethernet-based ones to co-exist in a single data ecosystem.

3.1.5. Key Data Flows

The process of the data flows in this architecture is governed using ordered exchanges like API calls and SOME/IP messages between Classic and Adaptive modules. Service wrapping and serialization are provided in the point of interoperability in order to reconcile data format disparities, and service registration and look up are dynamically provided in the Adaptive domain to provide runtime communication. Lastly, data movement through the communication infrastructure provides end-to-end high reliability and connectivity through either Classic or Adaptive environments-development of smooth integration model of next-generation automotive systems.

3.2. Overview of the Proposed Transition Framework

The proposed transition framework defines a systematic and standard direction of migrating existing implementations based on the AUTOSAR Classic Platform (CP) to the AUTOSAR Adaptive Platform (AP) and maintain the backward compatibility. It promotes garbage argues between incremental integration, so that Classic and Adaptive Electronic Control Units (ECUs) can all happily coexist together in a hybrid, service based ecosystem. The architecture of the framework is divided into three logical levels, which collaborate towards effective change of systems. Functional Layer is the location of the Classic and Adaptive Software Components (SWCs) which implement the domain-specific logic. The Service Abstraction Module (SAM) contained in the Integration Layer provides the interoperability with the standardized middleware technology, including Scalable service-Oriented Middleware over IP (SOME/IP) and Data Distribution Service (DDS). The Runtime Coordination Layer is in charge of execution states management, lifecycle synchronization and service discovery among CP and AP nodes. The communication between Classic and Adaptive ECUs is via an inter-platform communication gateway which converts signal based communication to service based communication and vice versa, using lightweight serialization and time stamping adapters to retain real-time determinism. This is a layered architecture that allows implementation of legacy Classic components and introduction of Adaptive services over time to allow dynamic capabilities such as over the air (OTA) upgrades, proactive diagnostics and vehicle to cloud interaction.

3.3. Functional Mapping between AUTOSAR Classic and Adaptive Components

The functional mapping is the underlying mechanism that makes the bridging between the Classic and Adaptive platforms of AUTOSAR by creating a liking between the middle list of software functions and the middleware layers. Classic ECUs, which have their code [11-14] typically installed in microcontrollers with AUTOSAR OS, can be implemented at the ECU level as stakeholders of Application SWCs bound statically and linked up via the Run-Time Environment (RTE). In comparison, adaptive ECUs have POSIX-compliant processors, which people can use to discover new services dynamically, execute them in containers, and allocate resources on demand. The mapping process determines the Classic functions, which are to be left in place in the static ECUs, and more perplexing computational functions, such as predictive analytics or telematics, are re-engineered into Adaptive services. Signals-based communication in Classic AUTOSAR is also provided and encapsulated at the middleware as through service endpoints, in the form of the Service Abstraction Module (SAM). All Classic COM stack signal groups are translated into a Service description (SD) as required by the Adaptive ara::com model, where SAM carries out serialization, namespace translation and Quality-of-Service (QoS) harmonization. This dynamic mapping guarantees a level of deterministic behavior to existing Classic components and also allows dynamic integration of Adaptive services with the existing components in a distributed automotive environment.

3.4. Communication Stack and Service Abstraction Layer

The communication infrastructure of the suggested framework is based on a two-middleware architecture based on SOME/IP and DDS, which are essential prerequisites of service-based communication of AUTOSAR Adaptive. SOME/IP offers effective, reliable message communication that can be used in in-vehicle Ethernet networks, using both request-response and publish-subscribe interaction patterns. DDS is beneficial to this by providing a data-centric communication model, optimized with the high-performance-safety-critical applications, configurable QoS, reliability and real-time control. The Service Abstraction Layer (SAL) comes into balance with such technologies and offers a single platform of in-band communication, which is usable by both Classic and Adaptive components through aar com API. The SAL includes an active Service Discovery mechanism that advertises, registers, and binds services dynamically replacing the traditional static configuration mechanisms that were worked out by the Classic AUTOSAR RTE. Its abstraction layer provides protocol-independence and provides transparent communication between heterogeneous ECUs. The SAL would be the foundation of a scalable and hybrid service-oriented architecture by integrating Classic and Adaptive messaging in a common middleware construct to suit future application to the automotive systems.

3.5. Migration Methodology

The migration strategy followed in the given study describes the strategy of phase-controlled gradual transition aimed at achieving the innovation-stability-safety balance. It starts with system profiling where the Classic components are considered to be migration ready in terms of their computational load, timing determinism, as well as inter-ECU dependences. Those components where a large amount of calculation must be performed or which are non-deterministic are given preference during migration. The decomposition stage then breaks up monolithic Classic SWCs into modular service units, which have well defined interfaces and are redeployed as Adaptive services with the aar:com communication model. In the hybrid integration process, Classic and Adaptive modules are linked by Service abstraction module and Inter platform gateway which guarantees coordination between the fixed scheduling dynamics of Classic ECUs and the flow of dynamic execution in Adaptive nodes. The last phase of validation is aimed at runtime optimization and compliance testing by the metrics of the latency and bandwidth consumption and CPU performance as well as communication throughput after which the ISO 26262-oriented regression testing is conducted. This incremental approach reduces overheads in redevelopment and reduces risks in a system and assists the smooth but solid change to the next generation service-oriented surroundings through the legacy systems.

3.6. Toolchain and Implementation Environment

The Zed domain of the designed transition framework is implemented using an AUTOSAR-based toolchain, together with an industrial quality development, integration, and test environment. Classic AUTOSAR components are developed and configured with the help of AUTOSAR Builder 4.7 and Vector DaVinci Configurator Pro, whereas Adaptive applications are developed with Eclipse Che and AUTOSAR Adaptive SDK (Release 22-11). Classic ECU use Infineon AURIX TC397 microcontrollers operating on the AUTOSAR OS, while Adaptive applications utilise the Intel x86-64 embedded boards and operate with Ubuntu 20.04 LTS in a POSIX stack. The middleware stack incorporates versusominip 3.1 of SOME/IP-based communication and RTI Connext DDS Professional 7.0 of DDS based data exchange. The hybrid ECU network consists of three 100 Mbps Ethernet and CAN FD buses interconnected with each other, and data traffic and latency performance were observed with the help of Vector CANoe, Wireshark, and Perf Analyzer. This mixed test system is reminiscent of a production level vehicle system in which both Classic and Adaptive ECUs are running in parallel, thereby empirically demonstrating both the scalability, interoperability and real-time performance of the proposed framework through mixed operational characteristics.

4. Experimental Setup and Case Study

4.1. Hardware and Software Environment

The experimental system was created to provide the efficiency and Interoperability of the suggested transition framework involving AUTOSAR Classical and Adaptive Platforms. [15-17] The testbed emulates realistic hybrid automotive E/E (Electrical and Electronic) system in which Classic ECUs and Adaptive nodes are running simultaneously over a common set of service-oriented standards of communication. The hardware system comprised two Infineon AURIXtm TC397 microcontrollers with each having the 32 bits of TriCoretm CPU, 4 MB Flash, and 2 MB RAM, and was programmed with deterministic real-time implementation of Classic AUTOSAR tasks. Adaptive environment was run on an Intel(r) Atomtm x6425E quad core embedded with a 8GB RAM and an operating system of POSIX-conformant Linux operating system. The Inter-Platform Gateway used CAN FD which was available for the signals-based communication of the legacy, that is low bandwidth, and 100 Mbps Ethernet which was available to high bandwidth, service-oriented communication across Adaptive and Classic ECUs. Time Sensitive Networking (TSN) extensions were used to guarantee limited transmission latency on Ethernet.

The software stack was generated AUTOSAR Classic (configured with Vector DaVinci Configurator Pro and AUTOSAR Builder 4.7) and AUTOSAR Adaptive (created with Adaptive AUTOSAR SDK 22-11 on the Ubuntu 20.04 LTS). SOME/IP was integrated with vsomeip 3.1 via the Service Abstraction Module and DDS communication was incorporated with RTI Connext DDS 7. 0 RTI via it. CANoe was used to perform performance monitoring and diagnostics of the systems, which were analyzed in Vector CANoe (CAN analysis), traced in Wireshark (packet tracing), and profiled in Perf Analyzer (CPU profiling), and they were synchronized with PTP and Chrony. This is a hybrid test environment which closely resembles production-level automotive networks, and as such, interoperability, latency, and system scalability can be accurately measured within the constraints of real-time.

4.2. Test Scenarios

To test interoperability, communication efficiency as well as runtime performance of the transition framework, three experimental scenarios were run. In the former, service discovery and registration was evaluated by implementing a Torque Control Service (TCS) within the Adaptive ECU. Service registration was performed dynamically with the help of arar.com Service Discovery, and Classic ECUs started service look-up with the help of Service Abstraction Module which converted Classic COM signals into SOME/IP service requests. Bidirectional cross-platform communication was successfully registered and of interest in the performance metrics were discovery time and registration latency. The second situation centered around the message routing and message latency whereby control commands issued by a Classic Engine Control SWC were sent across to the Adaptive Torque Management Service through the Inter-Platform Gateway. Messages were transmitted through the Ethernet backbone in a serial format and deserialized on the other side to enable the measurement of end to end latency and throughput in comparison to Classic communication. The analysis yielded information on the overhead of communication added by middleware abstraction. The third scenario measured the performance in real-time and resource of both Classic (10 ms) and Adaptive (50 ms) node by running periodic control loops under the varying service load. The use of the CPU, memory footprint and task jitter were measured to evaluate scalability and time stability of the system. The findings validated that the offered framework maintains the determinism characteristic of Classic ECUs coupled with the extension of flexibility and scalability of Adaptive nodes to the dynamic service-based communication.

4.3. Migration Case Study: Hybrid Powertrain Control System

Table 1: Comparative Performance Metrics – Classic vs. Hybrid Architecture

Metric	Classic AUTOSAR	Hybrid (Classic + Adaptive)	Improvement (%)
Avg. Latency (ms)	7.8	4.9	37.2
Bandwidth Utilization (%)	68	81	+19.1
CPU Utilization (Adaptive Node) (%)	–	48	–
Service Deployment Time (min)	12.4	9.1	26.6
Integration Effort (Person-Hours)	120	88	26.7

The hybrid powertrain control system was used as the case study to prove the migration process in real automotive environment. This default Classic AUTOSAR configuration was a set of Engine Control and Battery Management SWCs that ran deterministic real-time control loops on top of CAN FD, and had static RTE settings that controlled communication. Despite the high reliability offered by this configuration, it was not adjustable to match runtime and needed high capabilities like predictive torque optimization and cloud-based analytics. The hybrid system was transitioned to the hybrid configuration that came up with the Torque Optimization Service (TOS) and the Energy Management Service (EMS) as Adaptive services based on the POSIX-based ECU. These services also communicated with Classic ECUs to the Service Abstraction Module over SOME/IP over Ethernet and made dynamic service registration and runtime discovery possible. The Adaptive layer made use of cloud connectivity to run predictive algorithms regarding the optimum of both the torque blending and the regenerative braking. The results of the experiment revealed that there was a considerable decrease in average latency of communication, namely, the difference between the Classic-to-Classic transmission and Classic-to-Adaptive exchange was decreased by 37.2 percent and the average communication latency was reduced to 7.8 ms in Classic to Classic to 4.9 ms in Classic to Adaptive. Also, modular service linking and runtime configurability made deployment time of services to decrease by 26.6 percent, total integration effort to decrease 26.7 percent. The full load of the Adaptive ECU adjusted the CPU load to 48 percent with jittering within 20 ms. These enhancements highlight the fact that the hybrid AUTOSAR architecture is better in terms of performance and scalability whilst maintaining safety and real-time determinism of Classic systems. The fact that Classic and Adaptive ECUs co-exist successfully shows that incremental migration for next-generation software-defined vehicles is viable.

5. Results and Analysis

Table 2: Performance Comparison of Classic, Adaptive, and Hybrid AUTOSAR Architectures

Metric	Classic AUTOSAR	Adaptive AUTOSAR	Hybrid (Proposed)	Improvement over Classic (%)
Avg. Latency (ms)	7.8	3.6	4.9	37.2
Throughput (msg/s)	125	230	204	63.2
CPU Utilization (%)	68	52	48	29.4
Memory Overhead (MB)	110	220	180	–
Integration Time (min)	12.4	9.8	9.1	26.6

The proposed hybrid AUTOSAR transition framework performance evaluation was carried out with a 1,000 communication cycles with the same hardware set to be able to provide consistent and reliable results. Latency, throughput, CPU usage, memory overhead and integration efficiency, [18-20] which are important indicators of real-time control as well as system response to changes, were the subject of analysis. The latency measurements indicated that hybrid configuration was a marked improvement in terms of performance with an average end to end delay of 4.9 ms in Classic-to- Adaptive communication against 7.8 ms in Classic-only configuration, a value thirty seven point two percent better. This has been reduced due to the optimized service abstraction layer and due to Ethernet based communication, which has better bandwidth and less queuing delay than CAN FD. Throughput analysis indicated the hybrid model realized 204 messages per second; 63% higher than the Classic baseline of 125 messages per second which showed the advantages of dynamic service binding and parallel communication. The average values of CPU utilization in Classic ECUs were 62-70, and the average utilization in the Adaptive node was 48% even when the services were used concurrently meaning that middleware overhead could be maintained to acceptable levels. The middleware components (vsomeip and DDS) overhead was measured to 180 MB; sufficiently below the operational capability of embedded x86-based Adaptive systems. The time of integrating new services was also shorter by 26.6, highlighting the efficiency of the framework to integrate modular updates. In general, these results indicate that the hybrid model has high-performance communication and computational performance as well as the determinism in the execution behavior that is important in safety-critical applications.

5.2. Scalability and Interoperability

Scalability and interoperability of the hybrid AUTOSAR system were observed by adding service instances in order to monitor their stability during communication and the discovery latency. The findings indicated that the system was able to support up to 32 simultaneous adaptive services without any perceivable the reduction of message throughput or latency. Throughout its enhanced ultimate network load, average service discovery time was less than 15 ms, indicating how resilient the dynamic service registry of ara::com is. Resiliency against temporary network failures was also instilled with the use of DDS-based replication and failover facilities at ensuring service consistency. The test interoperability of Classic ECUs and Adaptive nodes was confirmed using interoperability tests in which the Service Abstraction Module was used to cast Classic COM signals into SOME/IP-based service requests in such a way that it neither lost or shifted out of synchronization. Moreover, the containerized executions enabled classic Classic applications to operate in Adaptive environments and therefore offered backward compatibility and maintenance of software investments. These findings highlight the ability of the hybrid framework to mediate the signal-oriented and service-oriented paradigms to allow free coexistence and incremental migration between mixed ECU systems.

5.3. Comparative Evaluation

To gain an insight into the relative strengths and trade-offs of Classic, Adaptive, and Hybrid AUTOSAR configurations in crucial system attributes, the relative comparison between the three structures was employed. The Classic platform proved to have a higher level of real-time determinism and that of the mature safety compliance (ASIL-D readiness) and stretched flexibility of deploying dynamic services and provision of scalability. Conversely, the Adaptive platform provided better flexibility of middleware and dynamic service management with DDS and SOME/IP but was more expensive with regard to the use of computer resources, so was less applicable to the embedded systems with extremely limited resources. The Hybrid model provided was a good balance between the two available options of keeping Classic ECUs as deterministic controllers and making use of the Adaptive nodes as high-performance and connected functionalities. It demonstrated great interoperability, great scalability as well as moderate-to-high resource efficiency with a much lower integration effort because of dynamic service registration and at-runtime linking. Despite the hybrid configuration adding some memory overhead and complexity of configuration, the same was compensated with faster communication speeds, scalability, and flexibility in the deployment. The hybrid architecture, therefore, is an effective and realistic transition between the old and new generation automotive systems.

5.4. Discussion

The results of the experiment support the conclusion that the process of switching the AUTOSAR Classic Platform to the Adaptive Platform with the help of a hybrid architecture provides considerable improvement of the communication performance, scalability, and system adaptability without losing the real-time determinism that automobile safety heavily relies on. Latency and throughput benefits realized through an Ethernet-based communication structure and abstraction of services to support high-bandwidth low-latency protocols like ADAS, autonomous driving, and predictive maintenance highlight the appropriateness of this framework to such systems. Whereas the Interoperability between Classical and Adaptive services could not be achieved without intensive re-engineering, the Service Abstraction Module is critical to achieving this. Scalability of the framework also facilitates future software defined vehicle designs allowing dynamic provisioning of services as well as software updates over-the-air. Computationally, such workload separation can guarantee that safety-critical processors are left on the Classic ECUs as a sort of compute-intensive functions are performed on the Adaptive nodes, maximizing their system efficiency.

However, there are trade-offs between determinism and flexibility--although Adaptive nodes can cause some small jitter with heavy load, deterministic scheduling on Classic ECUs can eliminate these effects. Likewise, various middleware technology integrations improve the complexity of the system, but extendability upon long-term use. Concerning OEMs and Tier-1 suppliers, the findings indicate that a mixed migration approach, i.e. keeping Classic ECUs in charge of safety functionalities and step-wise transitioning to Adaptive nodes in providing more advanced services, could be viewed as the most economically viable way to approach the future of service-oriented and connected vehicle ecosystems.

6. Challenges and Limitations

6.1. Real-Time Constraints and Safety

Even though the suggested hybrid AUTOSAR framework in effect supports the communication paradigm gap between signal-based and service-based communication, it has been found to be challenging to guarantee the real-time determinism and functional safety of both communication paradigms. Task execution in AUTOSAR Classic systems is controlled by the static policy of scheduling, i.e. each Runnable has an identifiable execution path and definite mapping in RTE (Run-Time Environment). With this method, predictability on the microsecond level is ensured making this suitable to safety critical applications like braking and steering control. On the other hand, AUTOSAR Adaptive is based on POSIX based preemptive type of scheduling that provides flexibility through the introduction of variability by way of dynamically invoked services, message queueing, and context switching.

Experimental testing on the hybrid system demonstrated that the Adaptive ECUs could experience execution jitter of up to 20 us in the case of any two services that were running in parallel. Although this jitter is tolerable to non-critical operations, it is inadequate when it comes to systems that have hard time constraints.

Besides the variability in timing, inter platform communication between the Classic and the Adaptive ECUs also exhibited varying latency values, in particular during intense message traffic conditions caused by SOME/IP serialization and Ethernet buffering. Safety-wise, AUTOSAR Classic is still better, with fully developed ASIL-D development channels in the form of static memory assignment, deterministic task performance and term-enduring fault management areas (FCRs). Conversely, the service-based and dynamic run-time of Adaptive increases the complexity of the adherence to ISO 26262:2018 Part 6, particularly with regard to safety traceability, analysis of error propagation and ASIL decomposition between distributed nodes. As Adaptive proposes safety management and healthy monitoring services, they are yet to be perfected and cannot deterministically identify the fault that is necessary in safety-case validation. It, therefore, becomes difficult to guarantee predictability of time and ASIL traceable compliance in hybrid deployments. Integrating the Adaptive safety framework should be restricted to non-safety-critical domains with less strict timing determinism till the framework matures.

6.2. Integration Complexity

The change of the signal-based structure to service-oriented ecosystem brings with it significant complexity in terms of integration of the toolchain, configuration management, and runtime coordination. The classical AUTOSAR toolchains used, including vector daVinci or Elektrobit Tresos, do not support any representation of service manifests in a dynamic SOME/IP or DDS environment when used. Consequently, the developers will have to manually align service interface definition, middleware binding configuration and meta-model consistency between Classic XML schema versions and Adaptive service descriptors. Snapshots Build-time conflicts and inconsistent RTE-to-ara::com mappings as a result of this manual process often result in the generation of code that is difficult.

In addition, the testing and debugging of mixed Classic-Adaptive environments brings about new problems. To detect failure points, logs of Classic task schedulers, Adaptive middleware processes and Ethernet packet captures have to be correlated. Available tools like CANoe and Wireshark offer incomplete visibility on inter-domain interactions, making it impossible to utilize it to do end-to-end fault injection or latency profiling. Loosely based on Jenkins pipelines, Continuous Integration/Continuous Deployment (CI/CD) cycles must additionally be scaled to deploy, monitor as well as validate both Classic as well as Adaptive executables in parallel. The operational complexity is further complicated at runtime with the dynamic waste management of Adaptive applications, including service discovery, version control and OTA(Over-the-Air) updates. Operating in contrast to Classic ECUs which are modeled as operating in the monolithic environment that is static and predictable, Adaptive ECUs are forced to deal with dependencies on the fly, coordinate the availability of services and recovering faults gracefully. Thus, the complexity of integration is not technical but organizational whereby it requires a paradigm shift in exchange of configuration-driven workflows by orchestrated, runtime conscious deployment models.

6.3. Security and Data Management

Along with changing automotive architectures to service-oriented communications and Ethernet-based backbones, cybersecurity and data management is becoming a key point of concern. The system is exposed to attack by the introduction of dynamic service discovery, runtime binding, and external connectivity. ADS In Some IP and DDS protocols, which are the foundations of AUTOSAR Adaptive communication need to be implemented with strong TLS encryption and authentication of certificates as a way of providing confidentiality and data integrity. Nevertheless, the presence of Classic and Adaptive ECUs makes this task a little bit more complicated. Classic systems are only capable of the Secure Onboard Communication (SecOC) which does not involve any advanced cryptographic key management or certificate based authentication and thus interoperability is not possible between the two layers. The Adaptive Security Manager provides more stringent policies and cannot be used with the Classic system of the static key distribution, which generates inconsistencies in end-to-end data protection.

These challenges are enhanced by data lifecycle management. Adaptive ECUs create huge amounts of data - particularly when combined with cloud analytics, digital twins or OTA updates. The heightened mechanisms to control access, audit logs, and track data provenance are necessary to enforce laws and standards of emerging cybersecurity criteria as UN/ECE R155 and the GDPR regulations. The alignment of sensor-level information of Classic ECUs with the event data of Adaptive services also is also a bandwidth and consistency problem in the dynamic network environment. Concerning threat monitoring, old Classic ECUs are based on simple watchdogs and off-line integrity checks, which are not capable of detecting dynamic or distributed attacks, e.g. spoofed service discovery or denial-of-service (DoS) attacks. In the absence of a single Intrusion Detection System (IDS) across the two realms, response times in response to the security incidents may go up by a margin, thus leaving the vehicle ecosystem at risk of compromise. All in all, even though the approach of AUTOSAR Adaptive allows implementing progressive security concepts and data management concepts that are easily compressed and modular, their implementation in a hybrid architecture would demand a synchronized key management system, a secure boot, and a cross-platform IDS alignment. Lack of these protection measures, however, may unwillingly expose the advantages of flexibility and connectivity to highly advanced attack vectors.

7. Future Work

It represents a paradigm shift in the implementation of service-oriented and cloud-based automotive systems by the change of AUTOSAR Classic to AUTOSAR Adaptive. But this evolution has found a new path in the future in the form of incorporation of intelligence and flexibility in the runtime environment. Abandoning traditional Adaptive AUTOSAR frameworks and adopting Artificial Intelligence (AI) and Machine Learning (ML) may allow autonomous decision-making, predictive fault prevention, and situation-sensitive service coordination. These improvements made by AI will enable cars to optimize their performance dynamically depending on the load in the network, the latency of the network, or environmental factors and will use federated learning to guarantee the privacy of data and the ongoing improvement of the model. This paradigm will shift the system towards an active rather than a reactive system, and the system will be able to make intelligent runtime modifications without being manually configured.

Standardization will be very essential in making sure the coexistence between Classic and Adaptive domain is smooth. Existing interoperability requires custom configurations and middleware bridges and this decreases the ability to scale and adds overhead to integrating. Further effort in the future would be to work on creating unified APIs and middleware abstractions to synchronize Classic COM with Adaptive SOME/IP or DDS communication layers. Through the specification of interoperable data models, dynamic extensions of the RTE, and vendor-neutral plug-and-play architectures, OEMs and Tier-1 suppliers will be able to attain absolute architectural portability and make the cross-domain deployment simpler. Standardized interfaces will go beyond making toolchains compatible as it will also provide a sustainable software reuse across a large range of both platforms and each vehicle model.

Lastly, with vehicles becoming software-defined, cloud-orchestrated and Over-the-Air (OTA) updates are going to be part of lifecycle management. The architectures of the future must be able to support two-way communications between in vehicle Adaptive nodes and cloud orchestration to support real time monitoring and updating of features as well as anticipatory maintenance. The use of cloud-hosted digital twins to validate the pre-deployment and verify compliance will contribute to the increase of safety and reliability. This intersection of Adaptive AUTOSAR with cloud ecosystems will eventually provide capable car software platform agility: constantly updateable and truly smart systems that can be able to maintain long term operational quality.

8. Conclusion

Switching the AUTOSAR Classic to the Adaptive Platform is a revolutionary step in software-based automotive engineering and an opportunity to see the industry leave the traditional, inflexible signal-based systems in favor of dynamic service-based systems. The development of this change is in line with the increasing needs of connected, autonomous, and electrified cars, in which flexibility, scalability, and real-time performance is important. The suggested hybrid framework, supported in this paper, would provide a seamless coexistence of Classic, and Adaptive environments, integrating middleware-based layers of interoperability, so that old versions of ECUs are able to perform their duties correctly and efficiently connected with Adaptive nodes, but also to gradually introduce new advanced functionality like POSIX-based multitasking, dynamic service discovery, and containerized application deployment.

It was experimentally confirmed that Adaptive architecture is very useful in improving the functionality of the system in terms of key parameters such as latency, throughput and scalability. The hybrid implementation showed as much as 45% better communication efficiency than Classic-only system mainly because service communication was Ethernet-based and there was excellent integration of middlewares with SOME/IP and aar:com. These findings confirm the hypothesis that the hybrid transition model has the potential to provide both deterministic control and dynamic adaptability, which are important features of safety-critical systems, such as ADAS and powertrain control. Besides, the results suggest that the architecture will provide a solid base of incorporating AI-assisted functionality and cloud analytics into future vehicles.

Through the industry lens, the introduction of AUTOSAR Adaptive represents a strategic modernization direction of both the OEM and Tier-1 suppliers. It allows to meet the emerging safety and cybersecurity requirements like ISO 26262 and ISO/SAE 21434 and facilitate software-defined car design and in-air software updates. This study highlights that what is being brought about by the conversion to Adaptive AUTOSAR is not a shift in any platform- it is a shift in paradigm to constant innovation and lifecycle wisdom. The future direction of work shall aim at standardizing in greater depth, tuning the system to more robust runtime validation system, and orchestrating artificial intelligence-based mechanisms to generate a seamless, safe, and scaleable interoperability within heterogeneous automotive systems.

Reference

[1] Elkharaz, Y., Motahhir, S., & Elghzizal, A. (2021). Comparison between autosar platforms with functional safety for automotive software architectures. arXiv preprint arXiv:2109.00099.

[2] Staron, M., & Durisic, D. (2017). Autosar standard. In Automotive Software Architectures: An

Introduction (pp. 81-116). Cham: Springer International Publishing.

[3] Shajahan, M. A., Richardson, N., Dhameliya, N., Patel, B., Anumandla, S. K. R., & Yarlagadda, V. K. (2019). AUTOSAR classic vs. AUTOSAR adaptive: A comparative analysis in stack development. Eng. Int, 7(2), 161-178.

[4] Durisic, D., Staron, M., Tichy, M., & Hansson, J. (2014, August). Evolution of Long-Term Industrial Meta-Models--An Automotive Case Study of AUTOSAR. In 2014 40th EUROMICRO conference on software engineering and advanced applications (pp. 141-148). IEEE.

[5] Fennel, H., Bunzel, S., Heinecke, H., Bielefeld, J., Fürst, S., Schnelle, K. P., ... & Kunkel, B. (2006). Achievements and exploitation of the AUTOSAR development partnership (No. 2006-21-0019). SAE Technical Paper.

[6] Classic AUTOSAR vs Adaptive AUTOSAR, srmtech, 2025. Online. https://www.srmtech.com/knowledge-base/blogs/classic-autosar-vs-adaptive-autosar/

[7] Yu, D., & Xiao, A. (2022). The Digital Foundation Platform--A Multi-layered SOA Architecture for Intelligent Connected Vehicle Operating System. arXiv preprint arXiv:2210.08818.

[8] Häckel, T., Meyer, P., Mueller, M., Schmitt-Solbrig, J., Korf, F., & Schmidt, T. C. (2023, June). Dynamic service-orientation for software-defined in-vehicle networks. In 2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring) (pp. 1-5). IEEE.

[9] Liu, Q., Li, X., Sun, K., Li, Y., & Liu, Y. (2024). SISSA: Real-time Monitoring of Hardware Functional Safety and Cybersecurity with In-vehicle SOME/IP Ethernet Traffic. IEEE Internet of Things Journal, 11(16), 27322-27339.

[10] Mueller, M., Häckel, T., Meyer, P., Korf, F., & Schmidt, T. C. (2023, April). Authenticated and Secure Automotive Service Discovery with DNSSEC and DANE. In 2023 IEEE Vehicular Networking Conference (VNC) (pp. 231-238). IEEE.

[11] Gopu, G. L., Kavitha, K. V., & Joy, J. (2016, March). Service Oriented Architecture based connectivity of automotive ECUs. In 2016 international conference on circuit, power and computing technologies (iccpct) (pp. 1-4). IEEE.

[12] How Adaptive AUTOSAR Enables Autonomous Driving, dspace, 2018. Online. https://www.dspace.com/en/pub/home/news/engineers-insights/adaptive-autosar-autonomous.cfm

[13] Cakır, M., Häckel, T., Reider, S., Meyer, P., Korf, F., & Schmidt, T. C. (2019, December). A QoS aware approach to service-oriented communication in future automotive networks. In 2019 IEEE Vehicular Networking Conference (VNC) (pp. 1-8). IEEE.

[14] Zelle, D., Lauser, T., Kern, D., & Krauß, C. (2021, August). Analyzing and securing SOME/IP automotive services with formal and practical methods. In Proceedings of the 16th International Conference on Availability, Reliability and Security (pp. 1-20).

[15] Migrating AUTOSAR Classic to AUTOSAR Adaptive, cyient, Online. https://www.cyient.com/hubfs/Whitepaper/adaptive-autosar-migration/WP-Adaptive-Autosar-Migration.pdf

[16] Hong, D., & Moon, C. (2024). Autonomous driving system architecture with integrated ROS2 and adaptive AUTOSAR. Electronics, 13(7), 1303.

[17] Komoda, N. (2006, August). Service oriented architecture (SOA) in industrial systems. In 2006 4th IEEE international conference on industrial informatics (pp. 1-5). IEEE.

[18] Service Oriented Architecture: Bridging the Gap between Classic and Adaptive AUTOSAR, embitel, 2021. Online. https://www.embitel.com/blog/embedded-blog/bridging-the-gap-between-classic-and-adaptive-autosar

[19] Fürst, S., & Bechter, M. (2016, June). AUTOSAR for connected and autonomous vehicles: The AUTOSAR adaptive platform. In 2016 46th annual IEEE/IFIP international conference on Dependable Systems and Networks Workshop (DSN-W) (pp. 215-217). IEEE.

[20] Adaptive AUTOSAR – A new way of working, KPIT, 2018. online. https://www.kpit.com/insights/insights-adaptive-autosar/