AUTOSAR Vehicle Operating System (AVOS) – The Safe and Secure Framework for Automotive System Architectures

Standards or proprietary solutions – a conflict?

Dr. Günter Reichart, AUTOSAR Spokesperson, Munich; Rinat Asmus, BMW Group, Chairperson of AUTOSAR, Munich; Bernd Mattner, AUTOSAR IAO, Munich

Abstract
The AUTOSAR partnership with more than 300 partners provides a globally recognized standardized software framework for intelligent mobility. The AUTOSAR stand includes the Classic Platform used all over the world and the Adaptive Platform for modern E/E systems architectures.

Some companies, including AUTOSAR partner companies, have announced their vehicle software development strategies to address the challenges in the mobility sector. Some strategies, for example, deal with the implementation of an own OEM "Vehicle OS".

This paper presents the working hypothesis on the AUTOSAR "Vehicle Operating System", showing what can be understood by this and how even such radical-seeming approaches can be effectively supported with existing offerings from the AUTOSAR standard already today. At the same time, we can identify fields of action for the further evolution of the standard.

Challenges in the Mobility Sector
The stakeholders in the automotive industry are still addressing the challenges of future mobility, with the guiding principles of connectivity, autonomy, sharing and electrification (CASE), as well as increased user expectations, in various ways.

A modern car is sometimes compared to a "mobile device," referring to our smartphones. That may be true when seen from the perspective of vehicle use. Here, key factors for the user experience play an essential role, for example end-to-end connectivity, applications for accessing vehicle functions, or even regular software updates in the background.

In vehicle development, there are differences that have a significant influence on the complexity of the requirements. On the one hand, there are the requirements for real-time behavior or the scalability of memory and computing power, considering automated driving, for example. In addition, there are strict regulatory requirements for functional safety and
cybersecurity, which must be applied to reduce or eliminate the risk to humans posed by a vehicle.

The major part of the necessary innovations in these areas takes place in E/E systems and software, with the result that one can also speak of a software-defined vehicle. Compared to a smartphone, the requirements here increase further when you consider the entire life cycle. In the case of smartphones or enterprise software, we expect it to last five to ten years, but in the case of a vehicle, ten or even more than 15 years. And maintainability and updates must be ensured over the entire life cycle.

And it becomes even more complex when we consider the entire value chain and, in addition, aspects such as integration and certification. And then of course the question is how this will fit together.

**Strategies for Automotive Software Development**

In response to this increased complexity, we are seeing numerous changes in the automotive industry with regard to automotive software and E/E system architecture development. Vehicle manufacturers are announcing different strategies in this area, ranging from improvements of existing concepts, the scale-up of in-house software development, the cooperation with big IT players, the cooperation to create industry standards, up to the development of individual "Vehicle OS" or "Automotive.OS" platforms. As a result of some publications in this context, the impression has arisen that the OEMs want to rely on proprietary solutions by bringing SW development in-house or by introducing their own "Vehicle OS".

The motivations of vehicle manufacturers to introduce their own "Vehicle OS" may be manifold, but what they have in common is that the term "Vehicle OS" is initially undefined and probably includes much more than what we usually understand by an operating system. In addition to a definition of "Vehicle OS", it is therefore necessary to analyze which software components are affected, whether and where proprietary approaches can be suitable, and what is to be achieved in detail.

**Motivation for a “Vehicle OS”**

To understand the motivation for the introduction of a “Vehicle OS”, a comparative look once again at the development approaches and potentials for smartphones and today's cars may help. The obvious feature of the smartphone development is the separation of hardware and software by means of an operating system platform that allows applications to be developed independently of this platform with the support of software development kits. In addition, a
separation of applications and backend systems allows far-reaching scalability and updates of functions and business models over the entire lifecycle. In comparison, we still have a huge number of hardware units in today's vehicles, often fully integrated with embedded software with a software release at the time of ECU production. These embedded Electronic Control Units (ECUs) are supplied via a large value chain and integrated into the vehicle. This means that millions of lines of code come together in a vehicle, and it is a challenge to standardize essential functional elements and update or upgrade them over the entire lifecycle.

The requirements described at the beginning are mainly functionally realized by software. This results in different life cycles for hardware and software, since, for example, the software has to be updated when functional safety or cybersecurity requirements change. This requires OEMs to look at hardware and software separately, so that their procurement or development as well as integration can be reassessed.

The central goal from the perspective of an OEM thus would have to be the decoupling of the development cycles of vehicle functions and hardware. In this context, consideration of the in-house shift of software development or a dedicated "Vehicle OS" would be understandable. With a "Vehicle OS", an attempt could be made, similar to the smartphone, to have an operating system platform for the entire vehicle with a standardized Vehicle Application Programming Interface (Vehicle API) and SDKs in order to completely decouple the development of applications and functions from the hardware and to achieve the desired scalability and update capability over the entire life cycle.

Additional impact could be seen through the ability to accelerate application deployment, optimize the use of software development capacity, increase confidence in system and software integration by manufacturers, and more effective responsiveness to ever faster changing regulatory requirements. Through the platform approach, a larger community of software developers could be interested in developing vehicle applications, thus being able to cope with the increasing amount and complexity of software in terms of integration into mobility ecosystems. On the other hand, a dedicated "Vehicle OS" could also be seen as an important competitive differentiator to demonstrate the software capabilities.

The AUTOSAR “Vehicle Operating System” (AVOS) Hypothesis

To say in a nutshell: AUTOSAR will be the global established standard for software and methodology enabling scalable E/E system architectures for future intelligent mobility supporting high levels of dependability, especially safety and security, but not provide a standardized "Vehicle OS".
The term AUTOSAR “Vehicle Operating Software” (AVOS) refers to the ambition within AUTOSAR to describe a generic model of how a “Vehicle OS” for future intelligent mobility from a perspective of an OEM could be composed, developed, deployed and operated. This working hypothesis is based on publicly available statements and publications and is the basis for maintaining the AUTOSAR standard's trusted role as a safe and secure framework for automotive E/E system architectures in the future.

Architecture of a “Vehicle OS”
A “Vehicle OS” in the AUTOSAR working hypothesis is both an “operating system” and a “software platform” for vehicles as elements of future intelligent mobility. It is the “operating system” for a vehicle from the perspective of a mobility ecosystem and provides all the in-vehicle and off-board capabilities that are required to operate a vehicle in an IoT/Cloud/Backend ecosystem. It comprises software functionality to enable and support user applications for vehicle operation and in-vehicle user experience, for diagnosis, monitoring and management of the vehicle E/E systems and software. The “Vehicle OS” provides the mechanisms to integrate vehicles into mobility ecosystems, ensures dependability with a special focus on safety and security, and is distributed over different hardware like micro-controller (µC), micro-processors (µP), Systems-On-Chip (SoC), Electronic Control Units (ECU) and the on-premise and cloud server.

The in-vehicle software of a "Vehicle OS" remains in the layered architecture, where the real-time operating systems (e.g., OSEK-OS, POSIX), non-real-time operating systems, hypervisor- or container-based virtualization, and hardware abstraction (BSP, MCAL) are arranged on the lower OS layer. The Middleware layer is characterized by a typical middleware providing basic software, common services and functions for applications (e.g., communication or crypto algorithms in AUTOSAR Adaptive and Classic). An automotive grade middleware enhances the runtime capabilities in the context of dependability.

On top of the Middleware Layer an additional Platform Layer will include the “Vehicle API”, SDKs for development of user applications and all services required to decouple application development from the underlying layers. Additionally, the Platform Layer provides OEM applications relevant for the car and system deployment. On the top layer you will find the user application and functions that extend the basic functionality and include the essential market differentiating software.
The “Vehicle OS” as a software platform includes state-of-the-art tools, vehicle APIs, safety and security building blocks, simulation and methodology to allow DevOps a rapid iteration for shortening the time from concept to deployment. This approach can trigger similar effects as in the development of mobile devices and enabled the vehicle manufacturer to continuously improve user experience and provide additional value to customers through continuous enhancements. The “Vehicle OS” software platform provides capability to the manufacturer to implement new vehicle software life cycles with automated continuous development and automated integration, test and deployment. Beside the aspect of user experience such an approach supports the obligation of vehicle manufacturers to continuously maintain safety and cyber security for more and more connected and automated vehicles.

The “Vehicle OS” to Manage Market Differentiation

At the moment, no unified way is published how vehicle manufacturers who decide on an individual ”Vehicle OS" will realize it. The limited information available indicates that approaches could range from "green field approach" to integration of "the existing supply chain and standards as before". From the perspective of time to market and efficient use of development resources, this becomes more relevant when it is estimated that in the future nearly half of the software could no longer be market differentiating.
Differentiating in this context is everything that an end user in the vehicle associates with the respective brand, i.e., the user experience, which ties people to the brand and differentiates it from the competition. From the OEM's point of view, it should initially be irrelevant here whether the market differentiating part is 50% or less. In any case, they have to get this share sustainably under control, because the respective business models are likely to depend on it. But even the non-differentiating portion will include a huge number of innovations in software, which will be driven in particular by the principles of CASE. Among these, there will also be a lot of important and critical software components that need to be globally or at least regionally harmonized. As these also increase complexity, vehicle manufacturers will also use their “Vehicle OS” or in-house development considerations to pay particular attention on scalability and reuse here in order to balance efficiency over the entire lifecycle.

**AUTOSAR Partnership - Trusted Collaboration**

AUTOSAR as a standardization organization is obliged to its partners in the development cooperation and will optimize the standardization activities to meet the new challenges. The OEMs that are already part of the AUTOSAR partnership generated about 80% of total automotive revenues in 2019. Currently more than 300 partners and this enormous market share demonstrate the global acceptance and trust that has been placed in the AUTOSAR standard for many years. The openness of the AUTOSAR standard and the new topics for standardization make the standard attractive in the relevant IT sector and expand the AUTOSAR partnership into this area. This allows the AUTOSAR partnership to bring together the leading experts in the automotive industry and the IT sector to realize a comprehensive framework for vehicle software. The partners share their intellectual property and can rely on confidentiality and compliance with the code of conduct agreed by the partners. Therefore, it is only natural for a standardization organization like AUTOSAR to show how the global standardization work can also provide a valuable contribution in the context of a "Vehicle OS". That the AUTOSAR partnership is also prepared for these challenges as a result of its many years of trusted collaboration can be seen from a brief historical review and the current standardization activities.

**A Historical Review to AUTOSAR**

A brief look back into history shows that essential reasons for the emergence of AUTOSAR almost 20 years ago are also applicable to the current changes. The concept for an open system architecture emerged in early 2002 at BMW electronic division. After several experiences with late announcements of processor changes for ECUs
under development Dr. Günter Reichart, at his time vice president for system architecture, decided with his team to take a rather radical step forward. The idea was to get rid of the late and risky adaptations of basic SW for ECU's and of the high manpower effort involved in these adaptations. HW changes led due to the strong dependencies between SW and HW to failure prone SW developments frequently on last minute risking the start of production. The proposal was to move towards open system architectures (OSAR) and to create an HW abstraction layer.

The first document defined an open system architecture as “an open system architecture is characterized by standardized, unambiguously defined and non-proprietary interfaces and a hardware-abstracting software layer that allows functions (software modules with defined interfaces) to be assigned to computer nodes within the system architecture in a largely free and unrestricted way”.

It was clear from the very beginning that such a major change cannot be realized by one OEM alone and a collaboration of various OEMs and suppliers to achieve the goals is a must. The goals have been from the very beginning to define and standardize central elements of an open system architecture and its related development processes and tools between globally leading OEM’s, Tier1 and supplier.

An initial goal was to effectively integrate the functional modules of different suppliers and OEMs within a domain or cross-domain into one system configuration, meeting the functional and non-functional requirements. The standard should ensure the maintainability of the systems over the entire life cycle, taking into account the available technical capabilities for software updates and upgrades.

Standardization should allow the use of commercial off-the-shelf (COTS) hardware and ensure scalability and extensibility of the system architecture to support vehicle and configuration variations, as well as allow compilation of certified and verified SW modules with newly developed SW modules for system design.

Essential targets were the standardization for the implementation of basic software functionalities (e.g., memory and network management, diagnostics) and the compliance with the high requirements for functional safety, security, maintainability and availability of the systems.

As a first step, the evolution of AUTOSAR was already started in 2015 by introducing the Adaptive Platform. This initiated support for new use cases such as highly automated driving, vehicle-to-everything (V2X), and greater interaction with off-vehicle infrastructure services. The support of the latest E/E system architectures with domain controllers and central computers with multi-core processors, the support of Ethernet networks and the introduction
of service-oriented software architectures were defined as supplementary goals of the standardization. To benefit from the developments in other industries, the support of operating systems with Portable Operating System Interface (POSIX) was included, which support extensive software, allow dynamic communication and a partial update of software.

Looking back, we can see that standardization is keeping pace with the changing framework conditions. With regard to the AUTOSAR "Vehicle OS" hypothesis, it can be seen that the previous objectives for the AUTOSAR standard continue to be valid. The AUTOSAR partnership will therefore continue to provide the appropriate support for the emerging evolution with regard to a "Vehicle OS".

**AUTOSAR – the Safe and Secure Software Framework**

The AUTOSAR standardization follows the same layered approach intended as a fundamental assumption in the "Vehicle OS". AUTOSAR provides a standardized automotive middleware based on the proven AUTOSAR Standards for the Classic and Adaptive Platform supporting existing and future vehicle E/E architectures. These middleware gives the vehicle manufacturer the choice to integrate their preferred OSEK or POSIX compliant operating system on the lower layer. It provides basic automotive functions (e.g., communication, diagnostic, monitoring, safety, security) on the middleware layer and the required interfaces to the user applications on the higher layer. The AUTOSAR methodology supports effective software development and operation life cycles.

In the automotive Industry there has been a common development for a standardized operating system which is called the OSEK-OS and since 2005 is published as international standard ISO 17356-3. This OSEK-OS standard has been the basis for the AUTOSAR OS Specification which was published with the AUTOSAR Release 4.2.2.

The AUTOASR OS can be hosted on low-end controllers and without external resources, and provides a priority-based scheduling policy and runtime protective features (memory, timing, etc.). It is statically configured and scaled and suitable for hard real-time performance.

This feature set defines the type of OS commonly used in the current or recent generation of automotive ECUs based on the so-called AUTOSAR Classic Platform, with the exception of Telematic/Infotainment systems which use proprietary OSs (e.g., Windows CE, Android, VxWorks, QNX).

Since 2003 the AUTOSAR standard has influenced very much the kind of basic software and middleware layers which are built for the ECUs of a particular OEM. A lot of functionalities are similar if not equal between OEMs, specific OEM requests are considered in the so-called complex device driver. The whole concept is summarized in the AUTOSAR Classic Platform
and is supported by a standardized development methodology and standardized templates. This has been and still is a very successful approach to reduce the complexity of system design, to allow an easier reuse of software solutions and to get rid of late notice adaptions to hardware changes.

With the new use cases and their increasing complexity and safety relevance as well as their potential vulnerability to external hacking attacks a need for the additional AUTOSAR Adaptive Platform emerged.

New applications like highly automated driving (HAD), Vehicle-to-X, software updates over the air (SOTA), or vehicles as part of the Internet-of-Things (IoT) raised completely new requirements to a software platform for the next generation of ECUs. The AUTOSAR Adaptive Platform supports dynamic deployment of customer applications, provides an environment for applications that require high-end computing power and allows to connect deeply embedded and non-AUTOSAR systems in a smooth way while preserving typical features originated in deeply embedded systems like safety, determinism and real-time capabilities. Built around existing standards such as POSIX, the AUTOSAR Adaptive Platform complements automotive specific functionalities enabling the platform to run in an automotive network. It is quite obvious that these new requirements cannot be addressed by a classic OSEK based operating system, therefore a POSIX-based OS supporting multi-core processors are used for these specific requirements.

The Adaptive Platform has a much more restricted configuration space as compared to the Classic Platform but provides a lot of important features for supporting high levels of safety, security and over the air updates and upgrades. They key characteristic is its compatibility with the Classic Platform and elements which are common to both platforms are addressed in the so-called AUTOSAR Foundation.

AUTOSAR has standardized the software architecture within the middleware layer based on a layered architecture concept for the Classic Platform and on a software cluster concept for the Adaptive Platform. The latter offers more flexibility and is suited for Service-Oriented Architectures (SOA).

In the coming development stages the Adaptive Platform is planned to include ICC3 Architecture specification and configuration for safety and security relevant functional clusters, the standardization for hypervisor and System-on-Chip (SoC) support, the standardization and demonstration of Graphical Processing Unit support, and standardization and demonstration Over-The-Air software update as well as a Vehicle API. Additionally, the exploitation license for such a Vehicle API publication is planned to be compliant to licenses models common for collaborations on backend solutions.
At the beginning of the standardization of AUTOSAR, it was defined that this standard would not cover the application software, i.e., it would not specify an infotainment system, for example, but is enabling the integration of such software packages by providing standardized interfaces.

**Strengthening the Value Chain**
Traditionally much of the software of today’s vehicles comes from Tier1s. They provide software which is designed to be flexibly adapted to different OEMs. They deliver application software for many not differentiating vehicle functions and also the basic software for the interplay of all systems components in an E/E-architecture. It can be assumed already today that around 60-70% of the functions are available in all vehicles in the top segment and thus no longer constitute a fundamental differentiating feature. They may differ in minor implementation details or user experience, but the basic functionality is similar or identical. Software for market-differentiating functions is usually developed by the OEMs themselves. These are typically new innovative functions or user interface software functions that come from many different sources and can be a major challenge for system integration. With an increasing software scope and requirements for certification especially due to the necessary innovations in the area of connected, automated, shared and electrified use cases, the effort for software development and operation at the OEMs still increases significantly.

Since many of the innovations concern basic technology solutions, and will be required globally for the realization of the new use cases, they will not be suitable for market differentiation even in the short term. This is where the main opportunities for a strong mobility value chain exist, by addressing the basic technological solutions in joint collaboration, for example via joint standardization in AUTOSAR. In this way, development resources can be used effectively and comparable solutions do not have to be developed again and again for each OEM. Furthermore, the joint coordination of standardized interfaces creates a reliable framework that reduces the risk of misguided developments and accelerates the implementation of solutions. As a result, the increasing development capacities for vehicle integration and approval could be organized more effectively at the OEMs, and the burden of software development could be distributed more evenly across the entire value chain.

Such effective integration of the supply chain could also be supported by joint collaboration on the methodology by incorporating the considerations for software development, integration and verification in the context of the "Vehicle OS" into the standardization.

The consortium will further work on the improvement of methodology and systems engineering to allow a seamless integration into the general E/E system architecture development.
processes. There will be a strong focus on a user-friendly configuration and a limitation of the configuration space. Demonstrators for the Adaptive Platform as well as the Classic Platform will allow software engineers to validate AUTOSAR specifications and to test implementations. In addition to this, the provision of platform agnostic Software Development Kits as well as AUTOSAR conformance test support is planned. In this context, AUTOSAR will also further develop the connections with other relevant standardization organizations (e.g., ASAM, MISRA, CVII, Khronos).

**Conclusion**

From the AUTOSAR perspective presented above, standards and proprietary solutions are not a conflict. Quite the contrary. Attractive proprietary offerings or functions are initially the essence of market differentiation and may allow a company to exist in the market. In the automotive environment, the creation of proprietary solutions is also subject to high safety and security requirements and is often governed by strict approval regulations. The effort required to develop and operate proprietary solution parts in the vehicle can only be provided economically if the non-differentiating shares in a vehicle can be supplied and integrated as efficiently as possible. The key to this is the use of offerings from the open-source community or on the basis of standards (e.g., AUTOSAR) combined with a high degree of reuse. Consequently, the efforts of individual vehicle manufacturers to implement their own “Vehicle OS” should probably not be understood as an approach to implementing the entire software for a vehicle as a proprietary solution. The approach is more understandable because the vehicle manufacturer has overall responsibility for integration and approval. With increasing automation of the vehicles and integration into overarching mobility systems, the risks in the context of certification and safe operation during the entire life cycle could increase, so that the ability to control significant software components could be crucial. The discussion about a "Vehicle OS" makes it clear that the boundaries of vehicle functionality are shifting significantly and will consist of an in-vehicle and an off-board component in the future. The situation is complicated by the fact that the solution components to be used have historically been developed under completely different framework conditions, in the vehicles under extremely regulated conditions and high safety and real-time requirements in contrast to cloud or backend solutions that are often based on open source and often have indeterminate response times. This is where a standardization organization with a development agreement like AUTOSAR shows its strengths. For nearly 20 years AUTOSAR has demonstrated to be the best
established and well-suited organization to coordinate and drive the standardization for software and platforms meeting the requirements of automotive E/E system architectures. It enables the partner companies to mutually contribute intellectual property via the expert discussions and at the same time includes free licensing as part of the membership. In this way, global acceptance for the standardized solution approaches can be created very quickly. AUTOSAR, together with its partners from the automotive industry and the IT sector, will also make its contribution to “Vehicle OS” or similar approaches and provide the necessary standardization to connect these worlds of future intelligent mobility.