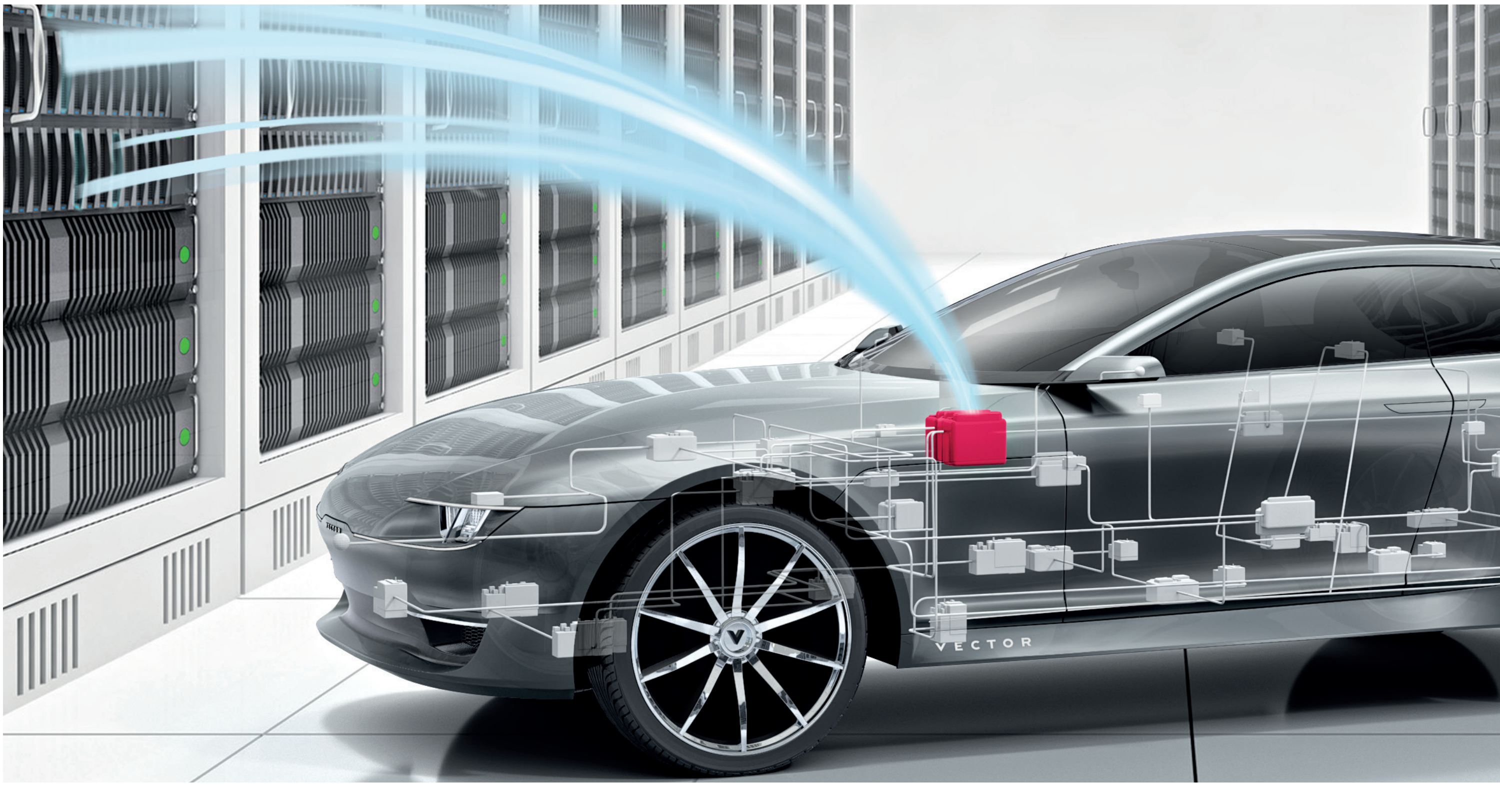

The Computing Center in the Vehicle

AUTOSAR Adaptive

The new AUTOSAR Adaptive software standard enables more powerful and more flexible E/E architectures in the vehicle. Ethernet-based ECUs can now be used as central application servers in an AUTOSAR Adaptive architecture. The big advantage: Adaptive ECUs make it possible to update applications over a vehicle's entire life cycle and add new software functions at a later time.

In future vehicle generations, highly developed driving functions will have a tremendous influence on the networking architecture and the interplay between individual ECUs. Continued growth of driver assistance systems, up to the level of highly automated driving, requires advanced sensors consisting of radar, lidar and various cameras. These sensors supply the vehicle with a coherent environment model. This means that the vehicle network must transport and process large amounts of data in a short time. The data transport alone requires networking architectures that can achieve high data throughput with low latency. In addition, the fusion of sensor data requires high-performance, in-vehicle computing systems which can process complex algorithms with the help of special hardware. The entire process chain, including the sensors and actuators, needs to be integrated in the vehicle safely and with little delay.

Effects on the Vehicle Architecture

Digital services, which are available over an IT backend, have a large influence on the vehicle architecture. Mobile radio interfaces with high bandwidth and vehicle-to-x communication (V2X) with low latency supply information on environmental conditions such as the weather, traffic flow and construction zones. The ECUs aggregate this data together with route information and data on the surroundings acquired by the vehicle sensors. Computing systems in the vehicle execute the necessary graphic-intensive operations. The results are shown on user-configurable and high-resolution display clusters or head-up displays.

It is not just these wireless interfaces to the outside world that require secure, and therefore complex, data transmission. Even a relatively simple electric vehicle must be able to establish a secure connection with a charging station for electrical billing purposes. This secure charging communication also demands heightened computing power in the vehicle.

All of these new functions will boost the share of added value by electronics and software in the vehicle to as much as 65 percent by 2025. The software share will grow the most, and it will be approximately 25 percent by then **[1]**. The increased complexity results in a conspicuous trend towards a stricter partitioning of tasks in ECU development. Although the hardware development of an ECU is usually performed by a Tier-1 supplier, the automotive OEM typically has a much larger variety of sources for the software, which is a very relevant competitive factor. The OEM either develops the algorithms in-house or purchases the software components from specialized third-party providers.

New Systems for High Requirements

The automotive OEM or the end user should be able to add software functions or modify them at a later time – including after the ECU has gone into production – e.g. to extend vehicle functions. However, a new market situation is created by the potential sourcing of software from third-party providers. The dynamic software integration environment needed for this is, in any case, a required property of upcoming vehicle platforms. In the future, even more powerful ECUs will be needed in the vehicle to execute computing-intensive algorithms and functions. They will be equipped with 64-bit multicore processors and MMU (Memory Management Unit) support.

These processors bring additional hardware support for virtualization and fast interfacing to external memory. Graphics, image and algorithm processing is handled by systems-on-chip (SoCs), which, as hardware accelerators, achieve the necessary processing rates with low power loss. Also used in an ECU are additional microcontrollers or SoCs for implementing safety-relevant functions as well as components for protecting against unauthorized access such as the Hardware Security Module (HSM) and Trusted Platform Module (TPM).

Interfacing to internal vehicle networks and to the world outside the vehicle continues to play an important role. Inside the vehicle, Ethernet – implemented as IEEE 100BASE-T1 (100 Mbit/s) or 1000BASE-T1 (1 Gbit/s) – will assume the role of a backbone, persistently changing the networking structure and multiplying data throughput. The external communication interfaces – such as WiFi, Bluetooth, 5G and V2X – also enable high data throughput. It also will not be uncommon to see interfaces to sensors and actuators via Low Voltage Differential Signaling (LVDS) or Automotive Pixel Link (APIX) which have transfer rates in the gigabit range. Previous conventional bus systems will still be used, but they will no longer play a central role in overall vehicle integration.

Efficient use of this hardware with high data throughput requires flexible software architecture. Generally, a POSIX-based operating system is a prerequisite for the use of complex processors and their peripherals. Such operating systems include Linux and, for safety-relevant systems, PikeOS. Many types of libraries and development frameworks are available for efficiently developing the software functions. Libraries for graphics and artificial intelligence applications are especially noteworthy.

AUTOSAR Adaptive: The New Standard

To fulfill these new requirements, the AUTOSAR Consortium defined another standard in addition to the established Classic platform: the AUTOSAR Adaptive platform. The Classic platform supports cost-optimized microcontrollers. Nonetheless, even these less powerful processors can be used to implement challenging applications in the areas of Ethernet, security and functional safety. The traditional platform is primarily used to implement ECUs which access sensors and actuators directly and need to fulfill stringent real-time requirements. The Adaptive platform, on the other hand, is designed to support applications with exceptional performance requirements such as highly auto-

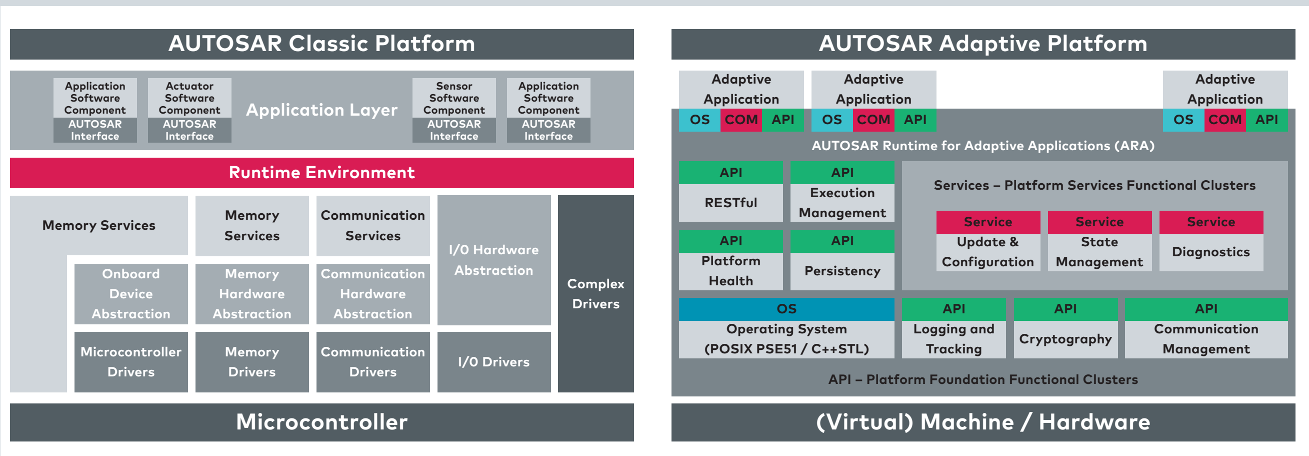

Figure 1: Differences between the AUTOSAR Classic and AUTOSAR Adaptive platforms.

mated driving. It provides a flexible integration environment whose distinguishing features include its ability to be updated and extended. Clearly, Adaptive platforms must also fulfill stringent time requirements. AUTOSAR Classic continues to be better suited for processing events that occur with high frequency. From today's perspective, it is only possible to fulfill the most stringent safety requirements by combining Classic ECUs with AUTOSAR Adaptive ECUs. That is because the AUTOSAR Adaptive standard enables more freedom and flexibility in the components and operating systems used – the keyword here is POSIX-OS. Comprehensive studies are currently being conducted to determine whether the Adaptive platform will, in the future, be used uncompromisingly as the sole system for supporting functional safety in every use case.

In its interplay with AUTOSAR Classic, the Adaptive standard now offers support for a large cross-section of ECUs and gives system designers the necessary building blocks for developing high-performance vehicle platforms. Previously, such ECUs were implemented by proprietary approaches. But which components make up AUTOSAR Adaptive? And what distinguishes the two platforms from one another technically **(Figure 1)**?

In the Adaptive platform, applications utilize the "AUTOSAR Runtime for Adaptive Applications," also known as ARA. This runtime environment gives users standardized interfaces for efficiently integrating different applications into the system. ARA offers mechanisms for ECU-internal and inter-network communications as well as access to basic services such as diagnostics and network management. In addition, the application programmer can directly access a subset of operating system functions known as the "Minimum Real-Time System Profile" (PSE51).

An Adaptive application is implemented as at least one process in a POSIX-based operating system. At runtime, application processes are loaded in their associated virtual address spaces and are executed there. Coordinated startup of these processes is handled by a module for application control (Execution Management).

The Adaptive environment continues to offer basic services such as diagnostics and network management, which are needed to integrate an ECU into an E/E architecture. Other services that provide basic functions are persistent saving of data (Persistency), functional monitoring of the platform (Platform Health Management), access to cryptographic operations (Crypto) and logging of measurement values (Logging and Tracing).

In addition, Update and Configuration Management (UCM) is a central capability of the Adaptive platform. While the Classic platform can typically replace the entire ECU code during an update, the Adaptive platform now offers the option of removing, updating or adding individual applications. Each Adaptive application is defined by a package

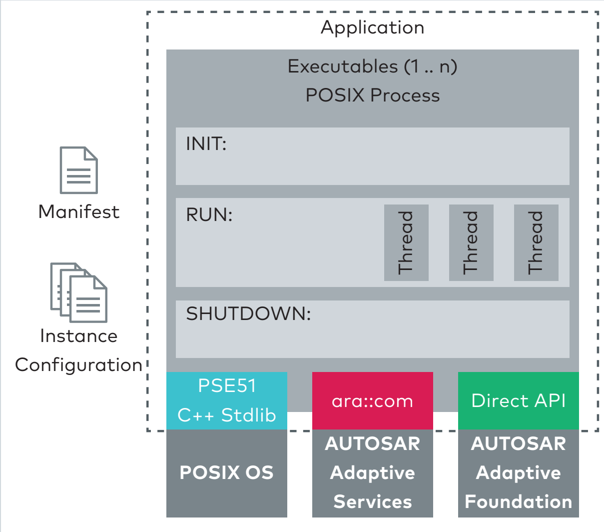

Figure 2: Interfaces of an AUTOSAR Adaptive application.

consisting of an executable program and a manifest file. This enables flexible handling of ECU functionality. The manifest file models the application's interfaces such as the ports and IP addresses for service-oriented communication. In addition, it establishes the preconditions for executing an application start with call parameters.

Unlike AUTOSAR Classic, C++ is used as the programming language in AUTOSAR Adaptive. The application programmer can efficiently implement challenging, reusable software thanks to the combination of object-oriented programming language, dynamic memory management and existing standard libraries.

Another property of the Adaptive platform is its transition to an exclusively service-oriented architecture paradigm, which offers greater flexibility in system design. Applications provide their functionality as a service via the Adaptive platform, and they can use services that are offered **(Figure 2)**. ECUs on which an Adaptive platform is executed are interconnected over an Ethernet network. A client requests a service that was offered by the server beforehand. This information is transmitted over Ethernet with the network protocol Scalable Service Oriented Middleware over IP (SOME/IP). The server replies to the request with the relevant data. Serialization of this data is also defined by SOME/IP. By comparison, the focus of the Classic platform is primarily on signal-oriented communication. Nonetheless, it is also possible to use AUTOSAR Classic in a service-based way for communication between multiple ECUs. As with the Adaptive platform, an Ethernet connection and SOME/IP are used for this.

In practice, the main properties of the AUTOSAR Adaptive and Classic platforms complement one another. It may therefore be assumed that ECUs based on both standards will be used in future vehicle generations – resulting in a

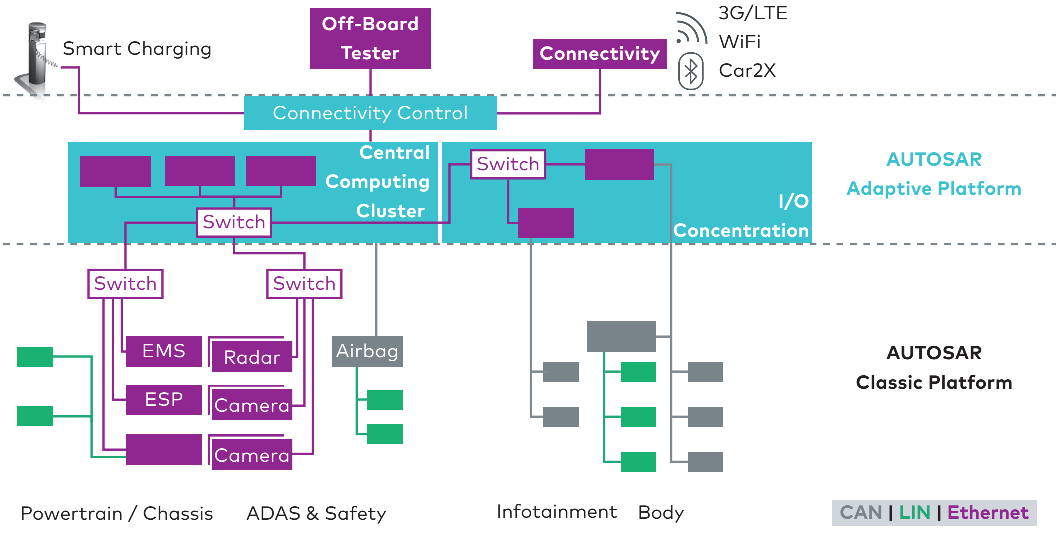

Figure 3: Vehicle architecture with AUTOSAR Adaptive and AUTOSAR Classic.

heterogeneous architecture. Besides familiar Classic implementations, Adaptive platform solutions are already available on the market such as Adaptive MICROSAR from Vector.

Domain Controllers and Gateways

Some modern vehicles have a domain-centered E/E architecture which groups vehicle functions according to their logical assignments into domains such as Infotainment, Body Controller and Drivetrain. Each domain has its own domain controller. These domain controllers are interconnected over Ethernet, and they each coordinate other ECUs which are assigned to them and which, in turn, enable access to sensors and actuators. Vehicle buses such as CAN or LIN produce the connection to the specific domain controller. Another controller, known as a Connectivity Unit, provides the connection to the outside world via off-board diagnostics, Bluetooth and mobile radio.

The use of AUTOSAR Adaptive does not change the fundamental vehicle architecture abruptly **(Figure 3)**. The most noticeable changes are the universal use of Ethernet in the vehicle and the related use of switches, each of which produces a collision-free point-to-point connection between ECUs. To optimize the wiring harness, these switches are integrated and are not implemented as separate ECUs. Furthermore, there will be several high-performance computers which connect to radar/lidar sensors, cameras, displays and other ECUs over many different high-performance interfaces. Above all, these include ECUs which assure access on the sensor level and actuator level as well as gateways between conventional bus systems which transfer data that occurs with high frequency and is time criti-

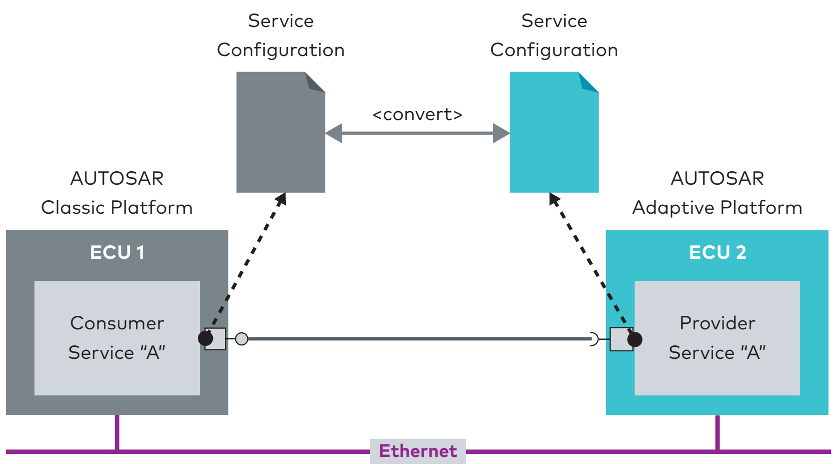

Figure 4: Inter-platform communication over Ethernet and SOME/IP.

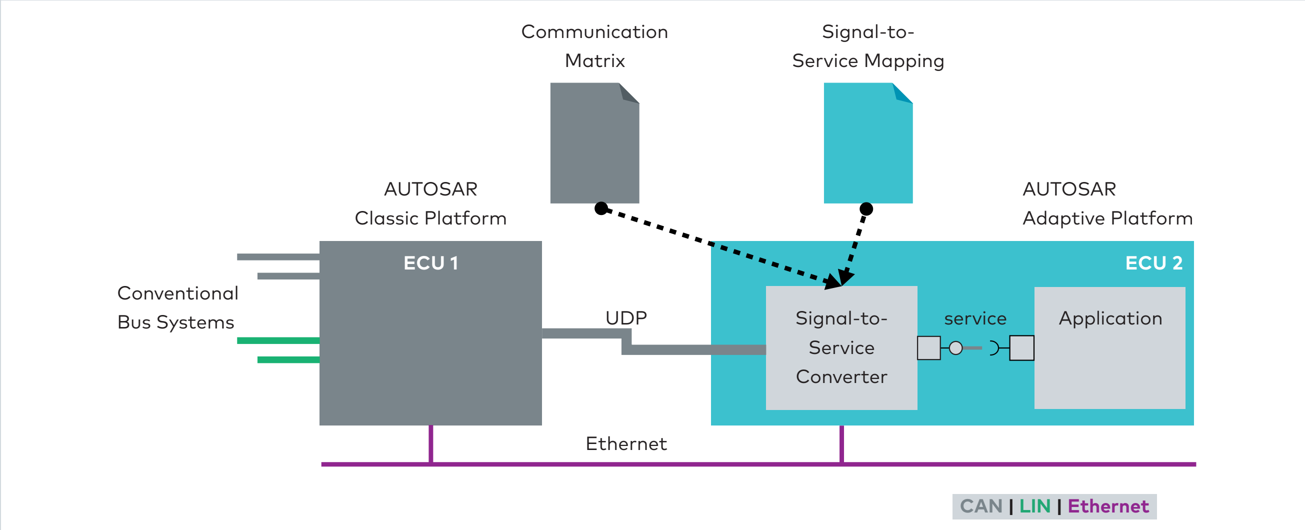

Figure 5: Connection to an AUTOSAR Classic signal gateway.

cal. The high-performance computers which are equipped with AUTOSAR Adaptive software, on the other hand, handle communication tasks and computing-intensive tasks in the vehicle network.
The question arises as to how Adaptive and Classic ECUs communicate with one another in such a scenario. In the simplest case, the ECUs – which are interconnected over Ethernet – use service-oriented communication over SOME/IP **(Figure 4)**. In this example, the AUTOSAR Classic ECU1 is connected to multiple bus systems to which other ECUs are connected. ECU1 operates as a gateway in this configuration and is responsible for "packing" the message signals from the bus side into a service so that they can be accessed directly by the AUTOSAR Adaptive platform. The communications layout is a fixed component of the design of AUTOSAR ECUs, whether it is a Classic or Adaptive platform. Because the configuration format is different for the two platforms, it is necessary to map the service configuration in the form of a conversion.
The situation is somewhat more multifaceted for communicating with an AUTOSAR Classic ECU whose operation is exclusively signal-based **(Figure 5)**. In this scenario, ECU1 is designed as a signal gateway, and it converts message signals directly into UDP frames on Ethernet. The AUTOSAR Adaptive ECU now converts signals from the UDP frame to a service that is available within ECU2 by signal-to-service mapping. In addition to the signal description via a communication matrix, an implementation based on this schema requires the mapping rules to a service as a component of an AUTOSAR Classic configuration. These mapping rules are already standardized as part of the AUTOSAR Adaptive specification. The service-to-signal mapping is coded based on these rules and is executed as an independent service in the ECU.

Mastering Software Complexity

The paradigm shift towards automation and extended communication of vehicles requires the processing and transport of large amounts of data and mastery over the resulting software complexity. Today's systems hardly satisfy these stringent requirements. That is why the AUTOSAR Adaptive platform – with its powerful central computers and Ethernet-based and service-oriented communication – provides an optimal foundation for future vehicle architectures. AUTOSAR Adaptive and Classic ECUs can be used in parallel here. Flexible gateway scenarios make it easy to interconnect these two worlds.

Dipl. Ing. Mirko Tischer
has worked in product development and product management at Vector Informatik since 1997. Since 2018 he is responsible as product manager for the AUTOSAR basic software Adaptive MICROSAR.

Translation of a German publication in Elektronik automotive, special issue "Bordnetz", September 2018

Literature References:

[1] Hansen Report No. 6, 2017, More supplier disruption coming.