Due to the above-mentioned functions, the amount and complexity of software in the vehicle continues to increase rapidly. This trend is by no means new. But right now, an extraordinary number of sectors in the automotive industry are undergoing significant changes to meet new market requirements. This is particularly true for automotive electronics. Since the computing power required for the new tasks is high, microprocessors are increasingly being used in electronic control units (ECU) and complement the microcontrollers previously used in the automotive sector. One or more of these microprocessors, often in combination with a microcontroller, form so-called high-performance ECUs. The microprocessors in these ECUs are very similar to those used in smartphones or PCs and require new software architectures. One aspect of such an architecture is the POSIX-compliant operating systems typically used to efficiently utilize the computing resources. These operating systems allow a more dynamic handling of the executed software and abstract more thoroughly from the hardware than previously used real-time operating systems. To seamlessly integrate these microprocessors into the existing vehicle network, the middleware running on top of the operating system is based on the standard AUTOSAR Adaptive.

The E/E architecture of a vehicle is also going through a change. Domain and central server architectures integrate the mentioned high-performance computers into the vehicle. Supported by fast data networks and powerful processors, the focus is no longer on efficient data transmission, but on stronger decoupling of individual ECUs. Changing a single ECU should have as little impact as possible on the rest of the system. A typical approach is, for example, introducing service-oriented architectures.

**New E/E Architecture with Central Servers**
In addition to decoupling individual components, increasing the reusability of hardware and software is another goal. This means that components can be used across vehicles...
Bild 1a: Distributed architecture

Bild 1b: Domain architecture

Bild 1c: Central server architecture
and even manufacturers. With the classic E/E architectures of recent years, this requirement cannot be met. Figure 1.a shows such an ECU-oriented architecture. Here, a function is realized by exactly one ECU. It brings along an associated set of sensors and actuators and receives additional data from the vehicle network. The communication matrix of the vehicle describes these necessary communication channels between the ECUs. Such a design, however, restricts reusability. Sensors and actuators are directly connected to a functional ECU. If these values need to be used by other bus participants, the communication matrix needs to be changed. To overcome this problem, the domain controller architecture has been established (Figure 1.b). Typical domains are e.g. “Body”, “Drivetrain” and “Info-tainment”. The basic idea of this architecture is using one powerful controller per domain, to which a large part of the necessary sensors/actuators is connected. In the domain, it also coordinates the subsequent ECUs. This greatly increases the flexibility for extending functionality within a domain, since adaptations often result in domain-local changes only. But the use cases mentioned at the beginning cannot be assigned directly to a domain. Highly automated driving functions require information from all domains and also feed data back to them.

The next development step of this approach is the central server architecture (Figure 1.c). The domains are combined in a large high-performance computer or computer cluster. However, there are more differences to the domain architecture than this. It is, for example, no longer possible to connect the sensors/actuators directly to the central control unit, since so many I/O peripherals cannot be connected to processors available today. Sensors/actuators are now connected directly to the network as so-called “Smart Sensors” and “Smart Actuators” and perform mechatronic tasks. Hence, they become ECU and vehicle independent, enabling a modular system design with a high reuse potential. With low-cost sensors, though, this procedure would not have a good cost-benefit ratio. To use these sensors, they can also be connected directly to the integration nodes shown in Figure 1.c in blue. These node ECUs also have another important function: They act as a gateway between the bus systems of the sensors and actuators, i.e. CAN, LIN, FlexRay and Ethernet. In this network, Ethernet represents the main bus system in the direction of the central computer. A modular and functionally expandable architecture is created by a suitably abstracted interface to the sensor and actuator ECUs.

Complex Architectures of a Central Server
Central servers or integration nodes are complex ECUs. They usually consist of several microcontrollers and microprocessors. This heterogeneous structure offers some advantages, because controller and processor complement each other in their properties. One example is the faster start time of the microcontroller. After being switched on, it is quickly ready for operation and can thus participate in communication with other ECUs and perform its function. Furthermore, even the highest timing requirements in the microsecond range with low jitter can be met with a microcontroller. The microcontroller is also better suited if the implemented function requires frequent interrupts.

The microprocessor has its strengths in other areas. Most important is of course its performance. The used computing cores provide a higher clock rate and bring along many functions like high multi-scalarity or jump predictions, which improve the average performance. Larger caches also allow the connection of slower but larger external memory devices. In addition to more resource capacity, microprocessors offer better hardware virtualization support, making it easier to use hypervisor technology.

A further advantage of the heterogeneous equipment with microcontroller and microprocessor lies in the fulfillment of safety requirements. According to ISO 26262, current processors achieve integrity levels up to ASIL B. By using redundancy, the ASIL D level required for highly automated driving can nevertheless be achieved. In such a system, the additional microcontroller performs two tasks: On the one hand, it executes monitoring functions. Yet, it can also be used to provide degraded functionality in the event of a fault, so that the system can continue to perform its function with a high degree of reliability. This is an important feature required for fail-operational systems, i.e. systems that must continue to function in case of failure (Figure 2).

The fact that the ECU is equipped with several programable components results in another aspect: From the outside, it is still a single ECU. Internally, however, many independent software components implement the ECUs functionality. This leads to technical and organizational challenges. From a technical point of view, the components must be able to communicate with each other to provide a common function. The task of the ECU manufacturer is now to connect the components by using inter-processor communication (IPC) and to describe the exchanged data. This is a new task for the ECU manufacturer, as this step did not occur in the previous workflow. Only the data exchange between the ECUs had to be described so far. However, this responsibility lays exclusively in the hands of the vehicle manufacturer. The same applies to diagnostic functions, software update and network management of the system: what has been defined by one party for more simple ECUs now requires distributed and coordinated realization.

From an organizational point of view, integrating the various software components represents an increasing challenge. The modular design of the ECU and the POSIX-com-
With a service-oriented approach, however, information can be subscribed. But there are other, less obvious advantages: The hardware drivers and the high-level software are more strictly separated. The hardware-independent applications in the vehicle are therefore highly portable. This enables much greater resource optimization than with AUTOSAR Classic ECUs. For example, software in the development phase can easily be moved between different ECUs if the resource limit is exceeded to avoid changes in the hardware design. Another advantage: The reusability of software components for several vehicle types is increasing.

In AUTOSAR Adaptive projects, the separation of software from hardware also enables a completely new work distribution between vehicle manufacturers and suppliers. Whereas previously functionality was always ordered as a physical device in the vehicle, it can now be fully purchased in software. To make this work, each AUTOSAR Adaptive application is now a separate binary file. Application development is therefore independent of ECU development. The driver of the vehicle can thus become an integrator himself by installing additional applications from an app store.

But who is responsible if malfunctions occur in the system on the road? An untested combination of applications could be installed in the vehicle. This situation conflicts with the typical integration approaches in AUTOSAR Classic ECUs, where each configuration is thoroughly tested. To avoid testing all app combinations, the freedom from interference between the applications must be guaranteed. Commercial operating systems can guarantee that memory limits are not exceeded for safety-relevant applications.

**AUTOSAR Adaptive as Platform for Central ECUs**

As already described, the software components executed on the microprocessor are generally not based on the AUTOSAR Classic standard. Instead, AUTOSAR Adaptive is used on this hardware to meet the requirements for modularity, dynamics and update capability. AUTOSAR Adaptive is becoming the de facto standard for high-performance computer platforms in vehicles. The AUTOSAR Adaptive middleware uses POSIX-compliant operating systems such as Linux, PikeOS or QNX and completes them with all necessary automotive extensions. One of the main functions of AUTOSAR Adaptive is the communication layer ara::com. This enables communication with other AUTOSAR Adaptive applications as well as with other software components (SWC) in the vehicle (Figure 3).

Diagnostics as well as security and safety functions supplement the functional features. This may sound very much like AUTOSAR Classic basic software. However, there are several architectural and technological differences. For example, ara::com is a service-oriented middleware. This allows dynamic communication paths to be established at runtime. This dynamism is a prerequisite for application software that can be installed during runtime. A classic communication matrix would have to be adapted to send new content to an ECU.

With a service-oriented approach, however, information can be subscribed. But there are other, less obvious advantages: The hardware drivers and the high-level software are more strictly separated. The hardware-independent applications in the vehicle are therefore highly portable. This enables much greater resource optimization than with AUTOSAR Classic ECUs. For example, software in the development phase can easily be moved between different ECUs if the resource limit is exceeded to avoid changes in the hardware design. Another advantage: The reusability of software components for several vehicle types is increasing.

In AUTOSAR Adaptive projects, the separation of software from hardware also enables a completely new work distribution between vehicle manufacturers and suppliers. Whereas previously functionality was always ordered as a physical device in the vehicle, it can now be fully purchased in software. To make this work, each AUTOSAR Adaptive application is now a separate binary file. Application development is therefore independent of ECU development. The driver of the vehicle can thus become an integrator himself by installing additional applications from an app store.

But who is responsible if malfunctions occur in the system on the road? An untested combination of applications could be installed in the vehicle. This situation conflicts with the typical integration approaches in AUTOSAR Classic ECUs, where each configuration is thoroughly tested. To avoid testing all app combinations, the freedom from interference between the applications must be guaranteed. Commercial operating systems can guarantee that memory limits are not exceeded for safety-relevant applications.
They offer hard real-time scheduling methods for this purpose. This requires the definition of memory limits and a worst-case execution time (WCET) for the application. Since there is no direct interaction with the hardware, time-related side effects caused by changed interrupt loads are no longer significant.

Of course, this effort is only necessary for safety-related applications. The use of a hypervisor makes it possible to operate systems with different degrees of dynamics and safety in parallel. QM applications can be located in a more dynamic and IT-like partition of the system, which can also use open source software. In security-related partitions, however, caution is advised, as software errors and safety gaps cannot be eliminated at the necessary speed. The use of open source software also involves a risk in terms of product liability.

**Outlook**

The increased demands on performance and flexibility in software development with simultaneously increasing cost sensitivity require extensive changes in the entire supply chain. AUTOSAR Adaptive is an essential software component that will make a significant contribution to the development of high-performance ECUs in the future. Adaptation of the E/E architecture has already begun in current vehicle generations. However, the shift of functionality from sensor and actuator components to central ECUs must be implemented even more consistently.

**Authors**

Dr. Markus Oertel
joined Vector in 2015 and is as a team leader responsible for the product Adaptive MICROSAR, Vector’s AUTOSAR Adaptive solution.

Dr. Bastian Zimmer
joined Vector in 2015 and is team leader for “solution management - embedded software and systems”. Together with his team he is responsible for the establishment of innovative topics and technologies.

*Originally published in "ATZ elektronik" magazine, Issue 05 - May 2019*

Image rights: Vector Informatik GmbH