Safety Architectures for Automotive Cross Domain Servers – Challenges and Potentials

Dirk Geyer, Christian Miedl, AVL Software and Functions GmbH Regensburg

Abstract
Next generation vehicle E/E architecture technology will leverage cross-domain servers (CDS) and benefit from. The underlying structures provide more flexibility in terms of upgradeability, system optimization, and more by separating sophisticated computing power from basic sensor/actuator routines. New architectures and the corresponding design of cross-domain servers require the application of Safety Elements out-of-Context (SEooC) to develop the basis for an application-ready overall functional system ranging from infotainment applications with low criticality to ADAS - High Performance Computers (HPC) and more with requirements up to ASIL D.

This presentation addresses relevant aspects of future cross-domain servers that require the integration of different applications with different levels of criticality in terms of Functional Safety and Cybersecurity.

The main measure to ensure these mixed-critical operations is the implementation of a hypervisor (ECU virtualization). This server-hypervisor architecture requires diverse and redundant structures including safety-related mechanisms to detect hardware- and basic software/middleware failures as well as intentional attacks on these layers. In addition to basic software, silicon hardware (SOC) can support safety- and virtualization mechanisms with high performant stable functions.

The presentation and talk will provide insights into implementation options for a "fail-safe" server system as well as implementations that meet ASIL D requirements. Examples of hardware protection of reaction paths and optimized partitioning for running different safety levels will be discussed.
**The paradigm shift for E/E - vehicle architectures**

The evolution of vehicle architectures over the past 50 years has been driven by the increasing number and complexity of electronic and programmable components in the car. Starting with the necessary electronics of internal combustion engines, such as ignition control, more and more electronic control units have been installed to fulfill innovative and new vehicle features, such as power steering, brakes and assistance functions. The number of interacting electronic control units quickly increased to about > 100 per vehicle, and the exchange of information between them exploded. Accordingly, the on-board network structures were gradually expanded, resulting in evolved distributed topologies.

Traditionally, most function development has been done at the component level and then integrated into the vehicle system from the bottom up.

The increasing complexity of functions as well as their interactions brought the distributed topology beyond its limit. The need for dedicated development steps at the E/E architecture levels (vehicle, domain, component) became more and more important, mainly driven by aspects as shown in Fig. 1

![Fig. 1: The drivers for changes in E/E Architectures](image)

As a general approach for numerous OEMs the function oriented structuring and the coming orientation towards high centralization, Fig. 2 shows the respective revolutionary steps from the distributed architecture to the domain-based- and the server-and-zone architectures of the
next generation. The main features of this trend are the centralization of high-level functionality in powerful computers and the localization of sensors and actuators close to their associated physical domains. This enables optimization of runtime requirements on the one hand, and real-time requirements near the sensors and actuators on the other. Signal interactions between functions inside of a domain and also between different domains become faster and more reliable.

Fig. 2: Present and future architecture types and characteristics

In addition to the aforementioned benefits of these architectural changes, we are currently facing new challenges in terms of functional safety engineering. The implementation of the functions at the various levels of vehicle, subsystems and control components, based on the vehicle-level safety goals, must also be realigned. The main implications for safety engineering concepts and processes are described in the next chapter.

From Safety Goal to distributed but consistent Functional and Technical Safety Mechanisms
ISO26262 [1] as the main standard for functional safety in the automotive industry describes firstly the path from the analysis of the criticality of features through Functional (FSC) and Technical Safety Concepts (TSC) to the hardware and software requirements. Then, after the implementation of appropriate mechanisms, safety-related integration verification and validation of the functions have to take place. Figure 3 shows a typical extension of the QM-
based development process with a safety lifecycle. The lifecycle is a holistic structure of sequential activities that are distributed and delegated to multiple stakeholders from the vehicle level to the software/hardware functional units.

Fig. 3: Functional Safety Lifecycle according to ISO26262 [1]

Starting with analyzing the criticality of vehicle functionality through Hazard and Risk Analysis (HARA) and deriving safety requirements from the top down, the "traditional" distributed architectures allow refinement and mapping of functional and technical safety requirements along a tight chain of effects. This was simply drawn from the inputs (sensor, HMI input, network request) through one or just a few connected Electronic Control Units (ECUs) to the safety-related actuators. In most cases, the safety goals of the respective vehicle characteristics were assigned to an ECU and implemented according to the associated TSC. In the past, this led to established ECU-centric safety concepts, such as the "EGAS three level monitoring concept". Based on a concept developed by Robert Bosch GmbH and further standardized and extended by a working group of German OEMs [2], it was originally developed for combustion engines, but has meanwhile been the saving part of various powertrain ECUs for more than 25 years as well as in other fields. Figure 4 shows a significant abstraction of this observer-based concept. The advantage: an asymmetric safety concept with a large part of non-safety-relevant main functions versus a small part of safety-relevant mechanisms for monitoring the main functions (function monitoring) and the processor (dual-processor monitoring with redundant HW monitoring unit). The TSC includes direct addressing of safety requirements to the relevant ECU - of course, signals from and to the vehicle network...
can also be safety relevant and must be taken into account, so this concept includes end-to-end communication monitoring for buses such as Ethernet, CAN and Flexray. In principle, the structure is suitable for fast and independent degraded fail-safe reactions to critical errors.

Fig. 4: 3-Level-Monitoring Concept (Basis: Robert Bosch GmbH / EGAS-AK [2])

What does this mean for the developments of next generation E/E architectures? The additional requirements of the next generations of feature sets are not covered by the state of the art above described. The main reasons for necessary extensions are:

- Splitting the chain of effects for a safety target among several components with different communication channels
- Operation of various safety-relevant functions with the demand for independence on individual high-performance controllers or servers under mixed criticality
- Growing share of fail-operational requirements
- Increasing portion of core functions cannot be efficiently monitored but must be "self-safe," e.g. ADAS functions
- TSCs must be consistently and budgeted via metrics across multiple functional units, including quantitative proof of each concept part
- Signal timing and fault tolerance times (FTT) require more cascaded concepts
- Distributed supply chains, especially for software, contribute to safety goals
• Greater virtualization of hardware and differentiation from software
• Consideration of "function-on-demand" requirements and "horizontal deployments" of SW
• Strong influence of cybersecurity on safety properties.

The following discussion uses the example of an ADAS controller to show that existing monitoring principles can be adapted and extended to meet the requirements of modern high-performance computers (HPC) mentioned above. Figure 5 describes the level of application software for ADAS functions. The software is implemented on two redundant "Systems on Chip" (SOC). The exemplary Automatic Highway Pilot (AHP) functionality is provided by the sensor configurations 1 and 2 including their redundant power supply (V_Batt1,2) to avoid common cause failures. A certain subset of configuration 2 provides the input to the safety SOC, where the plausibility checks for SOC 1 and SOC 2 are performed. Depending on the result of the comparison, a degraded mode, a reconfiguration of SOC 1 and SOC 2, or a substitute signal is applied. A comparable structure is used to monitor the program/data integrity of computing resources. Anomaly detection for both SOCs is combined with processor monitoring of the safety SOC itself. With this concept, safety requirements up to ASIL D and fail-operational reaction can be met.
System safety developers, working e.g. at the OEM together with the vehicle’s E/E architects, have the task to stringently refine and quantify the safety requirements for each of the involved electronic components and the associated software functions. Safety concepts are increasingly distributed due to the segmentation of the effect chain into zones and servers. Accordingly, the maturity and trustworthiness of safety requirements must be strongly emphasized in the FSC phase, including their verification, before they are addressed and passed to suppliers of I/O component, zone ECUs and CDS. On the right side of the V-cycle (Fig. 3), system safety integrators and system testers must be able to integrate all software and hardware components involved into the system in a "safety-compliant" manner, verify it and validate it for product release.
The Safety Design of Cross-Domain Servers (CDS)

With respect to the above criteria, the development of high-performance computers will differ in some aspects from state-of-the-art development with respect to functional safety. Considering the existence of mature safety requirements for a function to be provided by a cross-domain server, the architecture and TSC of this HPC must be defined. In contrast to the development of "traditional" control units, the supply chain for CDS will be more differentiated between hardware and software development. It can be assumed that the OEM will control both hardware and basic software development for these computers. According to the definition of the software-defined vehicle, the OEM will provide the "hardware on wheels", including the high-performance runtime hardware.

Corresponding services and functions, including aftersales and "features-on-demand," should then enable the integration and operation of application software (ASW) within the scope of these runtime resources. Accordingly, safety concepts, together with software interfaces between ASW and basic software (BSW), must define standardized mechanisms to provide the secure paths to the HW, especially for secure bus communication.

CDS enable the transition from domain-oriented architecture to vehicle-centric architecture (server/zone), see Figure 2. Software functions of different domains such as ADAS, body or infotainment are implemented in virtual ECUs as part of the server. The dedicated functions differ strongly in the safety objectives as well as in the appropriate operating systems (guest OS), depending on the runtime requirements of the software.

In general, the CDS is a suitable method for partial virtualization of real ECUs, e.g. domain controllers with potentially high hardware integration as SOC solutions (Fig. 6).

![Complex SoC (ECU)](image)
Figure 7 shows an example of the structure of a CDS. Three elements form the generic basis of this computer: the hardware layer, the hypervisor, and the control domain (domain 0), which has access to physical I/O resources and interacts with the virtual machines. These virtual machines, which are controlled and monitored by the hypervisor, provide the runtime resources for the application domains (domain 1...n). Depending on the type of hypervisor, a host OS (not shown here) is located between the hardware and the hypervisor.

In general, the CDS is a suitable tool for partial virtualization of real ECUs, e.g. domain controllers (Fig. 7) with potentially high hardware integration as SOC solutions. The given structure of CDS shows that the functions of the control domain as well as a hypervisor will care of most of the safety requirements in terms of freedom from interference and data integrity. The guest operating systems of the safety domains must ensure the effectiveness of the intra-domain safety mechanisms. Looking at the applied safety mechanisms for processor monitoring according to the 3-level concept (Fig.2), there are corresponding transitions into the mechanisms of OS and hypervisor:

Fig. 7: Schematic structure of CDS (example)
Table 1: ECU hardware – related safety mechanisms and application in ECU and CDS

<table>
<thead>
<tr>
<th>Safety mechanism</th>
<th>Classical ECUs</th>
<th>Cross-domain server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program flow monitoring</td>
<td>Task execution monitoring</td>
<td>Time partitioning, limit parameters, priority inheritance</td>
</tr>
<tr>
<td></td>
<td>(call/return)</td>
<td></td>
</tr>
<tr>
<td>Instruction set test</td>
<td>Reference calculation</td>
<td>Safety / Checker core</td>
</tr>
<tr>
<td>Safety RAM test</td>
<td>SW -function / ECC/ divers</td>
<td>HW-MMU, replication</td>
</tr>
<tr>
<td>Safety-ROM test</td>
<td>SW- function / checksum</td>
<td>SW- function /checksum</td>
</tr>
</tbody>
</table>

Future business models will consider CDS as a kind of element out of context. Comparable to personal computers, Tier-1 will supply hardware, OS, middle- and basic software, while the ASW of the application domains will be supplied by several sources, including cloud-based services. Thus - it is crucial to provide generalized interface structures towards the mixed-criticality application domains. Based on the standard for Adaptive Platform [3], servers can be organized in domain functions, that are running in virtual platforms. An application example under development in cooperation with a consortium, uses the generic CDS structure with the insertion of a virtual platform layer (Fig. 8)

Fig. 8: Virtual platform implementation by AVL SFR
In general, the required flexibility and scalability of CDS applications brings the reliability of deterministic safety mechanisms beyond their limits. Anomaly detection algorithms, designed to address these requirements will be very complex and must frequently be updated in order to cope with changed requirements. Here, the available runtime performance allows machine learning (ML) principles to be used to adapt safety mechanisms to changing operating conditions without losing the ability to meet safety safety goals. An example of a safety diagnosis utilizing ML approach can be seen in Figure 9. Machine learning algorithms can be implemented into specific safety domains and used to monitor safety objectives in different areas.

Fig. 9: Application of neuronal networks in safety functions of CDS

**Conclusion and Outlook**

Current and future architectures require a significant further evolution of functional and technical safety concepts. Encapsulated concepts on a single control unit, which represents the largest part of the safety-relevant chain of effects, must be transferred to distributed concepts with intelligent sensors, actuators and components on the zone and server levels. The outline of safety requirements, their implementation, integration and V&V in complex and distributed supply chains must be kept consistent and traceable, supported by stringent safety-
related contracts (development interface agreements) and reliably exchanged work products in the functional safety lifecycle.

Diversified-redundant implementation of safety functions with additional means to support fail-operational behavior follows the increased requirements of domains such as ADAS, Body and Chassis.

The requirements of functions with dynamic environment, such as object detection for automated driving and rapidly changing attacker models for cybersecurity mechanisms, cannot be secured by deterministic observers alone. The power of CDS allows to implement appropriate suitable machine learning algorithms.

As an outlook, the taken considerations for onboard components have to be extended to the principles of edge and cloud computing. Safety and security concepts will then be even more distributed in the vehicle and environment in the future, supported by fast and reliable V2X communication. Finally, automotive processes and methods will then consequently also be applied to this external part of the vehicle functions.