Whitepaper

The next step in E/E architectures

The automotive industry is facing one of the greatest transformations in its history. New business models promise additional revenue throughout the vehicle life cycle. Consumers increasingly expect their cars to blend into their digital, private, and professional lives. For manufacturers, there are many new opportunities to stand out from competition, however, especially for the established ones, leveraging this is a challenge.

Abstract

Current E/E architectures do not scale in terms of updateability, customizability, and connectivity. Where speed is becoming vital, their design, underlying development methods, and tools pose a significant limitation. This hinders the introduction of the envisioned softwarecentric technology platforms. To succeed in the transformation to the softwaredefined vehicle (SDV), multilevel strategies based on a common analysis of the different technology domains are required.

In this whitepaper, we analyze the SDV from the perspective of the E/E architecture. We describe the impact of the SDV within and between the individual functional domains and use this to derive technology needs and reference architectures.

The implementation of the SDV is closely linked to new vehicle-centric and zonal architectures, since these provide the technical foundation for simplified development of functions using high-performance central computers and zone controllers. To ensure that the design remains costeffective, it is essential to drive the technical transformation pragmatically, in particular by considering existing architectures. There is no one-size-fits-all solution, so we include in this whitepaper possible fusion, migration, and convergence scenarios, which can also be implemented in several steps.

As the world's largest tier 1 supplier with many years of experience in all vehicle domains, we provide a comprehensive view of the disruptive topic of new vehicle architectures. In doing so, we want to help our customers on their way to the next generation of vehicles.

Table of contents

1 Introduction

The demand for new E/E architectures has massively increased in the past years, and the first vehicle manufacturers are already moving to centralized and zonal architectures. However, there is still some criticism of the new designs. In E/E architecture circles, they are said to be too expensive, hard to scale, and that they lack the potential for a risk-mitigating step-by-step migration path from legacy systems. But at the same time, the new architectures help create new opportunities. They offer a relevant competitive advantage through simplification, enabling holistic updates, and making softwarebased configurability and extensibility the central focus. Even though there are currently some slowdowns, the race to fundamentally advance vehicle architectures is on.

With the new vehicle architectures, consumers' changing expectations of the driving experience can be better responded to. In today's market, engine performance, exterior, and safety are no longer the only criteria customers consider when choosing a vehicle. Especially younger people buy a vehicle because it promises them entertainment, convenience, and integration into the online world. They seek products that seamlessly connect the worlds of transport, home, and productivity. The vehicle is currently evolving into a technological stage where a personalized user experience (UX) and continuous evolution are key performers. This is an opportunity, in particular for established automotive manufacturers, to enter new domains and tap into new sources of revenue.

New E/E architectures, however, have to balance the conflicting priorities of ensuring costeffectiveness in the initial production of the vehicle and the newly expected support for reconfigurability and extensibility over the vehicle's lifetime. One-time "packages" that can be configured at purchase are increasingly being augmented by vehicle feature "subscriptions" that can be added later (even temporarily). Features can then be developed well after the initial start of production of a vehicle line.

The automotive industry's response to these challenges is the software-defined vehicle, which represents a shift of the technological solution space from hardware to software. This new focus brings the envisioned flexibility; however, it also requires changing the overall vehicle architecture. In addition to the need to replace many small, functionally oriented control units with a few powerful vehicle computers, the internal and external connectivity in the vehicle and the "vehicle OS," a construct aimed at reducing variance in the hardware and software development, are central topics of the next generation of E/E architectures. In these new architectures, static functions and add-on features are complemented by dynamic and distributed services, for example, those that use cloud data to optimize the driving strategy. As a result, the boundaries between classic domains are becoming more blurred. In terms of the technical elements, we often find this to be less pronounced, but there is a significant increase in functional interdependencies. It is clear that legacy architectures, markets, and a vehicle segment focus require a variety of solutions.

Many features and goals of the SDV, however, can already be implemented through less disruptive changes. This allows the optimizations of today's architectures to be preserved throughout the ongoing evolution.

In this whitepaper, we assess the SDV specifically from the perspective of the E/E architecture. We identify the influence of the SDV paradigm on the different functional domains and determine the relevant drivers that must be considered when changing existing solutions. Tapping into our experience as a comprehensive solution provider, we explore how existing architecture components such as control units, sensors, actuators, and the wiring harness need to be modernized or replaced and where new solutions need to be added.

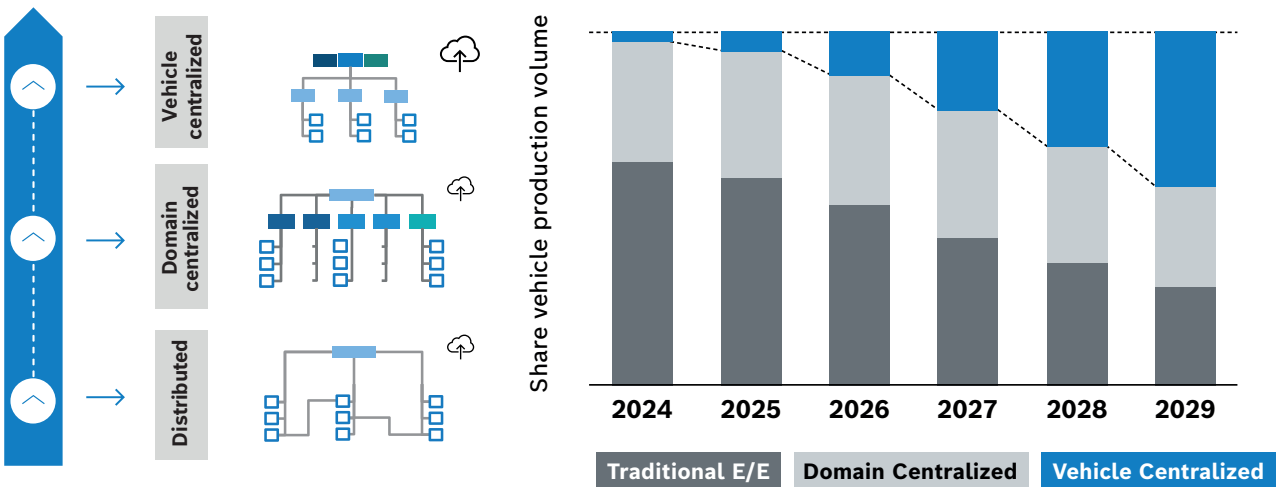


Figure 1: OEMs are ramping up new vehicle-centralized architectures and gradually replace prior architectural patterns.

2
The software-defined vehicle is changing the vehicle domains but will not eliminate them

There are high expectations that the software-defined vehicle will provide endless new opportunities. But one may wonder: since subscriptions and servicebased functionality (for example, monthly plans for traffic data) are already available in today's vehicles, why does the technological backbone of a software-defined vehicle need to be adapted at all?

The novelty is in the dynamics. In the SDV, features grow over time and across all functional domains, where they were previously defined only once. A vehicle becomes a living platform, with the production date only a secondary indicator of its capabilities. There are several preliminary steps necessary to enable this.

First, it needs capable centralized hardware that is designed with corresponding performance reserves or can be replaced throughout the vehicle's lifetime. This is the most obvious change. The SDV cannot be implemented in legacy distributed architectures.

A second fundamental requirement is a stable connection to a cloud platform, through which features are managed, and continuous interaction with the vehicle (for telemetry, data, and services). The cloud platform and the vehicle become a strongly interwoven technical ecosystem. Finally, the update path becomes the most vital element. Updates will be used for different objectives. These are:

01 Freshness: updating or expanding existing functions to improve the user experience

02 Bug fixing: fixing errors that arise after development before they affect the individual customers

03 Offerings: introducing new features to continue adding short-term and long-term value to the vehicle

The demand for technically enabling new offerings over the vehicle's lifetime requires new system layouts and "softwareification." But the solutions are not the same for every part of the vehicle. Even though all domains will eventually move to more software-driven development approaches, what form this takes varies. Domains will continue to have individual constraints and optimization points that must be considered. To understand the technical needs of the SDV, we have collected the new driving forces and expectations from each domain. This allows us to examine the architectural decisions and variance points for each of the new E/E architectures. We will start with the infotainment and body/comfort domains, move on to motion-related functions, and conclude this chapter with our analysis of the driver assistance and automated driving functions.

2.1
Connecting information, entertainment, and convenience creates added value

New SDV features obtain their automotive-specific character by connecting different domains of the vehicle. Such cross-domain functions already exist today. In the software-defined vehicle, however, the connection points between the domains will no longer be feature-specific; instead, they will be designed more broadly and generically to support reuse in a potentially large variety of future applications. We expect the infotainment domain and body domain to be initially affected the most by this shift, because they jointly provide the oftenenvisioned digital and physical interactions between passengers and the vehicle. Oftentimes, the infotainment domain is also the functional endpoint of the cloud uplink, extending the interactions to smartphones and remote services.

To quickly implement new, value-generating services that can be experienced immediately by the consumers, the new vehicle platforms provide easy access to the necessary sensor information, actuators, and vehicle information. Safety and security objectives, however, must not be jeopardized in this new setup. This particularly applies to vehicle functions that control the lights, doors, and seats, where implementation or execution errors may create a safety hazard. Additionally, personally identifiable information must always remain protected.

For reasons such as these, we expect that the physical functions and their low-level controls will continue to remain largely separated from the infotainment domain and be accessed by the infotainment apps only through logical interfaces, known as vehicle APIs or hardware abstraction layers. For practical reasons such as connector sizes, it even makes sense to keep body functions at least partially on dedicated control units (possibly as added functionality in a zone ECU) or concentrate them in a protected execution environment on another central controller. This provides the separation that is necessary from a security and safety perspective.

However, we have observed that, in practice, manufacturers plan for different allocations of the control logic of sensors and actuators. Logic is centralized, implemented locally on existing control units, or even distributed between zone ECUs. The allocation is often decided for each function individually, and there are few cases where it follows a generic strategy. Due to stricter requirements relating to self-diagnostics, startup timing, and fallback behavior, however, hardware sensor/actuator-specific implementations will never entirely disappear from the deeper system levels.

As stated above, generic application interfaces are a cornerstone of the simplified implementation of cross-domain functions. For further optimization, SDV platforms implement functional safety objectives through generic, high-level, software-based protection mechanisms so that critical actuators, for example, can be locked while driving. This significantly reduces the processing burden for SDV applications and can be combined with simple, locally implemented protective mechanisms in the actuator.

Vehicle APIs are a software construct that allows technical endpoints to be moved almost completely freely within the E/E architecture through corresponding abstractions. Nevertheless, the underlying communication buses need to be capable of supporting the emerging dynamics.

This is why such APIs are combined with transport protocols that allow connections to be established dynamically and support a guaranteed quality of service. Examples are SOME/IP and DDS, both of which are used on Ethernet-type buses, often in combination with Time-Sensitive Networking. Another change we see in the systems engineering for these architectures affects the variant and version management. Version management will be expanded to include not only software component version management but also API version management. The latter provides for easier dependency management. Nevertheless, any additional regulations, such as for releasing the overall functional chains, always need to be taken into account in addition to managing technical compatibility.

Figure 2: Vehicle-centralized architectures are a response to expectation from the C.A.S.E.* paradigm. It requires a shift of technological solutions into software, and a technology platform supporting fast iterations.

***(connected, autonomous, sharing, electrificed)**

2.2
Driving dynamics and comfort are based on the actuators installed in vehicle production

While assisted and automated driving functions are on the rise, the manual driving experience will remain a vital part of the design, since it is still the essence of the brand for many automotive manufacturers. Consequently, we expect that the driving software, in particular the parts responsible for regular driving situations, will not follow the same continuous evolution path, but will already have reached a very mature state with the start of vehicle production. Subsequent software-based updates are expected to happen primarily “under the hood.” Some extensions will come to the SDV in the form of maintenance and performance improvements that use cloud data or run non-critical additional functions offboard.

While end customers might not be able to experience the new SDV architectures in this domain right away, these architectures can still bring a significant benefit for manufacturers. They can use the introduction of central control systems to improve the performance of their vehicles in terms of safety, comfort, and efficiency for dynamic longitudinal and lateral motion coordination. With a further shift of steering, braking, acceleration, and suspension functions to that system, the motion functions can be realized in a more uniform development environment.

Despite this trend towards centralization, however, we still have not seen any major step towards dissolving the close connection between high-frequency control/regulation tasks and the sensors and actuators of the domain. The comparatively low benefits in terms of system costs, very strict safety requirements, and individual components that are highly optimized for performance and cost stand in the way of this.

For the subsystems mentioned above, the SDV has the advantage that, in addition to simplified joint development, updates can be rolled out without bringing the vehicle to a mechanic. Improvements to driver support features used in critical driving situations, such as traction control and vehicle dynamics, can also be introduced using this option. For reasons of safety and stricter vehicle homologation regulations, however, the relatively high update rates and variety of versions for the functions seen in other domains are not expected.

An online channel to an automotive backend, most likely implemented in an execution environment separate from the direct driving coordination functions, provides an opportunity to make the driving experience even safer through cloud-based functional extensions, such as predictive warnings in the event of poor street conditions due to snow and ice. However, since the vehicle must remain useable even without a remote connection, these are primarily non-essential additions to the core functionality.

The same is true of functions for optimizing the operating strategy, which (depending on the stage of development) are implemented in local or cloud environments that are optimized for dynamic loads. Depending on how relevant these functions are for vehicle homologation, however, there may be limitations in the freedom of allocation and updates.

For vehicle manufacturers, this presents opportunities for new, indirect monetization models. Data collected through standard interfaces can be used for value-creating services such as predictive maintenance and for creating battery certificates. Features can also be activated at a later time.

2.3
Data-driven development and new business opportunities shape assisted/automated driving experience

The development of assisted and automated driving functions is a great beneficiary of the constant connectivity envisioned for the SDV. With a constant influx of scene and telemetry data,the automated/assisted driving system (ADAS) can be continuously improved. Through iterative updates, these improvements are rolled out to all customers. Consequently, the expected update rates will likely be higher.

The mentioned elements of SDV technology are generally necessary to sustainably improve the quality of the product, but they can already be implemented today in existing domain-centric architectures. So in terms of the user experience, at a first glance it seems that there is no need for action. Does this mean that ADAS is a “non-SDV” domain? No.

There is a change in end customer business models for ADAS that affects the underlying E/E architecture. Previous architectures are built to optimize costs by allowing optimal selection of sensor sets, Compute performance, and storage per vehicle and purchased feature set. While cost-effective, this rules out any post-production feature upgrades.

Architectures that aim to enable general SDV capabilities in ADAS take a completely different approach. Possible increases in the costs of vehicle production (due to overprovisioning of sensors and Compute performance) are counteracted by revenue from subscription models and pay-per-use options. Furthermore, reducing the number of different variants lowers costs. This is done despite the fact that not all of this functionality may be needed in individual cases.

The SDV cloud connection is used for feature release management and billing. It is clear that this only represents a profitable business model when the business case covering vehicle production costs, additional revenue over lifetime, and the possible reduction in the number of variants comes out positive.

According to current observations, automotive manufacturers with a focus on the upper-midrange and premium vehicle segment in particular have used this approach so far. The transition from assisted (<=L2+) to automated (>=L3) driving functions significantly increases the need for sensors, performance, and redundant system structures. In order for the additional feature purchasing model to work outside of the premium segment, making a distinction between vehicles allowing or not allowing the later integration of >=L3 driving functions represents a valid criterion for creating variants in production. In new vehicle architectures, this can be done through purely additive expansion designs with additional centralized controllers, redundant power supplies, and additional sensors that do not disrupt the original <=L2+ architecture.

Through closer integration with the cloud, additional data sources can be included in the SDV that continue to improve the functional performance of the assistance features (for example, through predictive planning using online maps with detected road conditions such as black ice). Until more robust wireless communication technologies become available, these more volatile SDV features will not replace the robust, universally available core of the ADAS functional platform. Instead, they will complement it.

From the perspective of the overall architecture, centralization of the ADAS functions will continue to be a focus when it comes to implementing the SDV. This is required in particular to enable wider software-based scaling in homogeneous hardware platforms. We expect that ADAS-specific sensor systems that require a lot of bandwidth (cameras, lidar systems, and raw data radar systems) will initially remain connected with a star topology through broadband interfaces specific to the technology domain (e.g., LVDS for cameras) or multigigabit Ethernet, as per the current state of the art. Shifting within the ADAS domain towards a zonal architecture in the currently envisioned architectures appears to be advantageous for interfaces with lower bandwidth requirements. For ultrasonic sensors, a zonal coupling can also prove to be advantageous when it comes to overall geometries.

3
The primary challenge in the software architecture is managing complexity

In the previous chapter, we outlined the different needs of the functional domains and how they benefit from an SDV-enabled E/E architecture. Centralization and consolidation were key aspects here. In the following, we will conduct a small excursion into the software domain to look at technical separation needs, because this is an overarching driver that impacts the extent to which consolidation in the E/E architectures is possible and sensible.

In the SDV, software becomes the fulcrum of the technological development of the vehicle. In order to keep the growing amount of software manageable, complexity must be reduced and controlled. This means that both the software platforms of the SDV and the tooling used must be suitable, but also the hardware designs need to be assessed for their capability to act as “software carriers.” Here, the support for good separation mechanisms, rooted in hardware, is vital. Where previously complexity was reduced by placing functions into different control units, now the controller platform’s internals are becoming the primary contributor to the separation.

With the evolution of the SDV, technical separation mechanisms such as hypervisors and containers play a key role. For microprocessor platforms, technical solutions from other industries, such as from data centers, have already been transferred and enhanced for use in an automotive context. This allows functions from different domains to be merged, even onto a single system-on-chip (SoC). The operating systems available in the automotive domain provide many features that allow concerns to be separated as desired, so they can be managed, verified, and validated by small teams, as autonomously as possible and without regression. This also applies to safety-related freedom from interference requirements. The evolution of automotive microprocessor platforms, however, is not yet complete and continues to be driven by consortia such as SOAFEE. Nevertheless, a high maturity level has been reached already, enabling domain and functional consolidation onto single microprocessor platforms.

On the other hand, microcontroller platforms are still in the early stages of supporting modular deployment of functions. This partially stems from limitations in the hardware itself, but also applies to many microcontroller software frameworks that target singlepoint integrations and monolithic deployments. Their granularity is therefore, at most, at the level of separated virtualization-based partitions, but with shortcomings in their independence and hardware agnosticism. This imposes a hindrance with regard to finely granular deployability, which currently leads to the perception that dynamic SDV functions are only developed on microprocessor platforms.

Especially for the driving and assisted/automated driving domains with their great importance for safety and tight functional coupling, the fact remains that, despite decoupled development, the software as a functional implementation must still be managed as a whole. This is necessary, for example, to meet the requirements of UN ECE R156 regarding the release of software updates. In this context, the decoupling of homologation-related functions is far more relevant than the ability to install or update software components individually. This leads to beneficial partitions in the E/E architecture that strictly separate, for example, homologation-related functions from other functions.

4
The Pragmatic implementation of zonal E/E architectures

As is now obvious, centralization is essential in SDV E/E architectures, but may be constrained by the technical capabilities of the Compute hardware or software frameworks used. To complete our picture of these architectures, we will now move on to the underlying layers of the E/E architecture, here in particular the use of zone controllers for decoupling functions in the SDV.

One often-mentioned primary driver for zonalization is that geometric vehicle partitioning using zone ECUs allows the wiring harness to be simplified through segmentation and de-/multiplexing. From the perspective of the communication network, these zone controllers bundle lower-speed communication channels and connect them to a larger central backbone. The power network in a zonal E/E architecture benefits from a finely granular safety and control architecture using, for example, electronic fuses. This simplifies the overall power subdistribution, which is advantageously implemented in combination with central power management that takes care of the main distribution. Zonalization can provide benefits both for production costs and for automating wire harness production and installation, but the advantages vary depending on the amount of equipment in the vehicle. Especially in the cost-optimized budget vehicle segment with significantly smaller feature sets, additional ECUs for simplifying the wire harness may ultimately not pay off.

Zone ECUs can leverage further cost benefits if they simultaneously contribute to a noticeable reduction in the overall number of ECUs. This is particularly evident for body-related vehicle applications, where many small ECUs may be combined.

As already noted in the title of this section, however, the task of the zones is decided pragmatically – from the “extended arm” of the central control units to the largely independent geometrically oriented extended body domain controller. Service-oriented architectures allow for a greater degree of freedom in allocation and use the zone controllers as an infrastructure element for the implementation of abstracting vehicle APIs.

The scaling of zone ECUs is limited by pin count and heat dissipation limits (especially for power electronics), which means that the use of sensitive components such as high-performance SoCs or large memories is subjected to strict limitations. Higher-performance zone ECUs are possible with a more comprehensive and costly cooling solution, but centralization in the direction of actively cooled central vehicle computers generally appears more favorable.

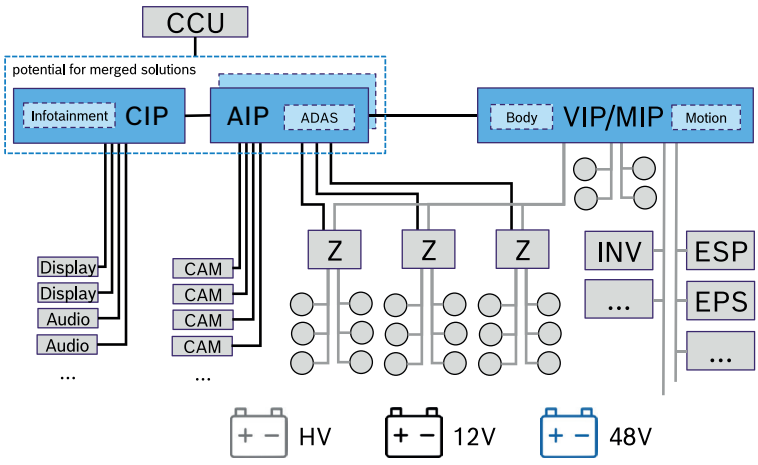
A further shift of functions into zone ECUs, especially sensor preprocessing for the ADAS domain, does not provide clear advantages. Either increased centralization with simpler sensors or preprocessing within the sensor make more sense in terms of domain independence and technology specificity. For example, the processing algorithms of radar systems are very sensor-specific. Moving this to a zone ECU requires common component development and puts additional burden on the zone ECU design. Functionally, the benefit of zone-based preprocessing is limited, as quickly becomes evident in the example of a 360-degree camera view with overlapping image areas that need to be merged. Note that this does not mean that the camera belt will never be zonalized, because technology standards and silicon availability are constantly improving. The zones would, however, act only as a multiplexer in this case.

Finally, zone ECUs specialized for distributing communication have the potential to perform in-vehicle diagnostic gateway functions. However, in terms of the SDV, this must be reconciled with the general need for a central data telemetry and update master function, which can be advantageously implemented at the central Compute layer.

5
The next centralization steps come in different variants

Considering SDV architectures from a domain perspective, as presented in the previous sections, shows a clear trend towards centralization, with consolidated ECUs for ADAS, infotainment, and driving and body functions. Between the “new” domains and the underlying zone ECUs, more standardized interfaces will allow cross-domain implementation of functions. The next integration steps will take place at the granularity level of this now-completed domain consolidation, resulting in different technological variants. Single-domain controllers will therefore successively disappear from the E/E architecture. Within the ECUs, however, they will remain – for reasons such as different requirements for approval and management processes but also due to varying technical constraints.

One integration step that has already been implemented by some vehicle manufacturers today is that of a “shared housing,” in which separate Compute units are accommodated in a common mechanical frame. The ADAS and infotainment domains in particular, both of which rely on high-performance SoCs and heat-sensitive memory components, benefit from a common cooling system and possibly additional shared infrastructure. The shared housing concept can be designed as a fixed multi-PCB solution or, with more flexibility in mind, using adapted module concepts. This makes them cheaper to maintain and allows them to be potentially replaced by better hardware versions over the lifetime of the vehicle.



AIP = ADAS Integration Platform
CAM = Camera Head
CCU = Connectivity Control Unit
CIP = Cockpit Integration Platform
ECU = Electronic Control Unit
ESP = Electronic Stability Program ECU
EPS = Electric Power Steering ECU
INV = Inverter ECU
MIP = Motion Integration Platform
VIP = Vehicle Integration Platform
Z = Zone Controller ECU

Figure 3: First vehicle-centralized architectures will introduce centralized vehicle computers and zone ECUs. As visible in this example, the degree of cross-integration and shift to a purely zonal communication/power network will differ between domains.

Multi-SoC solutions on a single PCB represent a more rigid level of integration while also increasing synergies and cost benefits. They can also be combined with a modular design concept for scaled variants.

Especially in the cost-sensitive segment, solutions based on fusion SoCs represent a valid integration solution. In this case, separation is implemented at the software level, for example, with hypervisors and container frameworks, always paired with strong hardware-supported separation and quality-of-service mechanisms. This degree of integration is also the most complex in terms of functional freedom from interference.

The question of which co-integration option is best suited cannot be answered from a technical perspective alone. However, using elements from existing architectures in particular makes a step-by-step approach from the shared housing concept to the fusion SoC appear advantageous as a method of minimizing risk.

In addition to the primarily microprocessor-centric integration platform for ADAS and infotainment, a comparable microcontroller-centric integration step can also be considered for motion, gateway, and body functions. These platforms are dominated by much stronger separation mechanisms, which are primarily designed in hardware. This is the only way to handle the higher determinism requirements and accompanying topics such as the decoupling of homologation-related functions. This is associated with more static software frameworks, which are, however, based on established standards such as AUTOSAR Classic. These platforms offer lower idle current consumption, fast startup, and easier implementation of functions up to maximum safety levels.

6 Conclusion

The next generation of vehicle architectures puts user experience and software at the forefront. Driven by changing consumer preferences, it allows automotive manufacturers to tap into new business models that build upon data-centric interconnection of the vehicle with cloud-based services and that lead to a further dissolution of functional domain boundaries in user-centric areas. Abstraction at both the hardware and vehicle level combined with greater freedom and decoupling in software development are necessary for coping with the increasing complexity of the vehicle system. This is helped by adapting the E/E architectures in such a way that they significantly simplify the development of the vehicle functions. In both geometric and vehicle-wide centralization, the convergence of control units has to be considered. Despite a recent deceleration, the trend towards new vehicle architectures, which manifests itself in high-performance computers and zone ECUs, continues unabated.

In order to meet the cost, time, and quality targets for this transformation, a holistic and targeted approach is required that aligns the specific technical, commercial, and functional considerations of the affected vehicle domains and provides for meaningful deviations from the often rather dogmatically conceived vehicle-centric architectures. This is particularly relevant to enable E/E architectures to be continued to be developed step by step from legacy architectures and with the aim of also covering the entry-level market segments.

The new software-centric approach, however, cannot scale sufficiently in the existing ecosystem. New initiatives such as SOAFEE, COVESA, and Eclipse.SDV provide access to software resources and developers on an unprecedented scale. However, fulfilling the quality requirements of the automotive world will take many more collaborative efforts, including with regard to hardware standards. New partnerships need to be established to generate joint synergies. In doing so, the specifics of the highly regulated and safety-focused automotive industry, which operates under high cost pressure, have to be taken into account. The lack of joint processes, methods, and tools required for collaborative project and product development presents an important challenge to be overcome.

The creation of a software-defined vehicle will succeed if the requirements for updating, integrating cloud-based services, and managing variants and configurations are clearly mapped to preferred technical solutions and functional views. As a cross-domain system provider with expertise in the classic automotive sector as well as in the field of the SDV, Bosch is hard at work addressing these issues. As a global player, we bring our extensive expertise to projects and studies to enable content-centric, viable architecture advancements, such as the ones outlined in this whitepaper.

Imprint

Why Bosch?

Bosch is your full-stack technology provider for innovative hardware, software, and systems products. With our extensive experience in systems design, our network of experts supports our customers to realize cost-efficient, scalable, and safe solutions across all automotive domains.

Authors:

Dr. Thorsten Huck
Dr. Andreas Achtzehn
Andreas Deberling
Erik Goerres
Dr. Karsten Wehefritz

Cross-Domain Computing Solutions
Engineering System Architecture

Contact us now

Get in touch with our expert team. We are looking forward to support you with our expertise to design your next steps in E/E architecture.

thorsten.huck@de.bosch.com

+49 173 3175436