World Electric Vehicle Journal

MDPI

Article

Cloud-Enabled Reconfiguration of Electrical/Electronic Architectures for Modular Electric Vehicles

David Kraus *, Daniel Baumann, Veljko Vučinić and Eric Sax

Institut fuer Technik der Informationsverarbeitung (ITIV), Karlsruhe Institute of Technology (KIT), Engesserstr. 5, 76131 Karlsruhe, Germany; daniel.baumann@kit.edu (D.B.); veljko.vucinic@kit.edu (V.V.); eric.sax@kit.edu (E.S.)
* Correspondence: d.kraus@kit.edu

Abstract: Modern mobility faces increasing challenges, like carbon-free transportation and the need for flexible transportation solutions. The U-Shift II project addresses these problems through a modular electric vehicle architecture, a drive unit (Driveboard) and a vehicle body (Capsule). This separation offers high flexibility in different use cases. Current architecture paradigms, like AUTOSAR, face limitations in cost and development speed. To address these issues, this paper introduces a hybrid software architecture that integrates signal-oriented architecture (e.g., CAN bus) with service-oriented architecture for enhanced flexibility. A integral component of the hybrid architecture is the dynamic link system, which bridges these architectures by dynamically integrating Capsule-specific components into the Driveboard software stack during runtime. The performance of the developed systen and its functionality were evaluated using a hardware setup integrated into a Driveboard prototype. The dynamic link aystem was evaluated including latency measurements, as well as functionality tests. Additionally, a cloud-based reconfiguration process enhances the versatility of the Driveboard by allowing for over-the-air software updates and resource allocation. The results show a promising hybrid, reconfigurable E/E architecture that aims to enable a robust transition towards a pure service-oriented architecture required in future electric autonomous vehicles.

Keywords: electric connected vehicles; e/e architecture; software architecture; service-oriented architecture; cloud computing

Academic Editor: Michael Fowler

Received: 15 January 2025
Revised: 9 February 2025
Accepted: 17 February 2025
Published: 18 February 2025

Citation: Kraus, D.; Baumann, D.; Vučinić, V.; Sax, E. Cloud-Enabled Reconfiguration of Electrical/Electronic Architectures for Modular Electric Vehicles. *World Electr. Veh. J.* **2025**, *16*, 111. https://doi.org/10.3390/wevj16020111

1. Introduction

In an era characterized by rapid urbanization and technological innovation, the global transportation landscape is undergoing a profound transformation. Urban mobility is being reshaped by three key trends, namely, the sharing economy, electrification, and the advent of autonomous vehicles, fundamentally altering how people and goods move within and between cities [1]. Traditional paradigms of personal vehicle ownership and conventional public transportation face significant challenges as urban centers grapple with congestion, environmental sustainability, and equitable accessibility. Amid these challenges, the concept of shared modular mobility emerges as a viable and innovative solution, offering a flexible and adaptive framework to address the dynamic needs of modern societies. The interplay between shared mobility models and advanced electrical/electronic (E/E) architectures redefines vehicles as intelligent and interconnected systems rather than mere modes of transportation. Through the prioritization of signal-based communication, vehicles achieve seamless interaction with their components and dynamically respond to their surroundings, creating a safe, efficient, and responsive network. Furthermore,

the integration of service-based software enhances modularity, enabling vehicles to deliver tailored functionalities based on their current configuration. As this trend gains momentum, established automotive manufacturers are adapting to meet the demands of modularity and scalability. For instance, Daimler has announced plans to mass-produce a modular and scalable "Van Electric Architecture" [2], which decomposes the vehicle into three interchangeable units during manufacturing. Similarly, Kia introduced its "Platform Beyond Vehicle" at CES 2024 [3], adopting a comparable approach. These innovations enable the creation of modular vehicles with interchangeable components, allowing for a diverse range of hardware configurations.

However, traditional software architectures, typically designed for fixed hardware setups, face significant challenges in accommodating the variability introduced by modular components. Signal-oriented software architectures are by design limited in their ability to efficiently adapt to the dynamic requirements of modular vehicles, particularly when unique software adaptations are necessary for different configurations.

Service-oriented architecture offers a promising solution to address these challenges. By decoupling software functionality from specific hardware configurations, Service-Oriented Architecture (SOA) enables greater flexibility, scalability, and adaptability in managing the complexity of modular and scalable vehicle architectures, paving the way for more efficient and innovative automotive solutions. An SOA approach is already well established in consumer web and cloud services where APIs based on HyperText Transfer Protocol (HTTP) are typically used for exchanging information. However, due to the fundamental differences of the consumer cloud and automotive applications, HTTP cannot be used in the automotive industry [4]. As a solution, current standard architecture solutions like AUTOSAR Adaptive are already focusing on an automotive, service-based architecture with Scalable Service-Oriented Middleware over IP (SOME/IP). Combined with AUTOSAR Classic via a gateway Electrical Control Unit (ECU), it is already possible to combine bus systems with SOA. However, this system lacks two major points. First, as AUTOSAR is economically driven, the annual fee for using a AUTOSAR Basic Software stack by a provider is well above EUR 10.000, depending on the company size and volume. Secondly, of significant importance is to mention that the AUTOSAR documentation is in need of improvement [5]. Therefore, frameworks like ROS2 are a promising alternative to the current popular software architecture solutions like AUTOSAR, as there are already some efforts towards automotive applicable ROS2 versions [6]. Furthermore, the field of connectivity has emerged in the automotive sector. Most modern vehicles are connected to a cloud, and some vehicles, like the Volkswagen ID.3, are leveraging Vehicle-to-Vehicle (V2V) to enhance the overall awareness through messages like Cooperative Awareness Message (CAM) [7] and Decentralized Environmental Notification Message (DENM) [8] defined by the European Telecommunications Standards Institute (ETSI). This article is an extended version of the authors' contribution to the International Electric Vehicle Symposium and Exhibition (EVS37), held from 23 to 26 April 2024, in Seoul, Republic of Korea [9]. Building on that work, Section 3 has been revised and expanded with Section 3.2, which provides a more detailed description of the cloud reconfiguration process. Additionally, the evaluation has been extended to incorporate the new content in Section 5.

1.1. Problem

The U-Shift II platform represents a solution for modular electric autonomous vehicles tackling challenges of modern mobility [10], similar to the vehicle concept from the first U-Shift project (see Figure 1). Within U-Shift II, an on-the-road modular vehicle, which has to seamlessly integrate different capabilities and dependencies, and which comes equipped with the current Capsule topology, is developed. With that flexibility, the conception of

a suitable E/E architecture becomes a major challenge. Besides the power supply needs, the Capsule could also provide a prior unknown set of sensors and actors, which the Driveboard has to accommodate within the current system configurations. With that in mind, typical automotive E/E architectures of modular vehicles quickly reach their limits, endangering their potential (benefits) to the overall mobility.

(**a**) Driveboard and Capsule separated (**b**) Driveboard with attached Capsule

Figure 1. U-Shift vehicle concept [11].

1.2. Aim

The transition towards a native automotive SOA needs to be initiated with hybrid software architectures. Hybrid software architectures are an emerging system bridging the gap between current technical standards and future software-defined cars by exploiting the flexibility and fast development cycles enabled through SOA systems while still being able to have a failure-proof and cost-efficient way to distribute information within set time constraints. Considering the fully autonomous functionality combined with high modularity achieved with different Capsules, U-Shift presents the unique demonstrator for implementing and evaluating a hybrid and modular software architecture.

1.3. Contribution

We provide a detailed understanding of our developed E/E architecture for automated modular vehicles. This is achieved through the development of a prototype Driveboard and its construction within the U-Shift II project. Within that prototype, the proposed concepts for a hybrid software architecture, combining service- and signal-oriented architectures, are implemented and tested. Through a dynamic software coupling mechanism, namely, the dynamic link, the signal- and service-oriented architectures enable a seamless transition between Capsule topologies and therefore its use cases, leading to new requirements to the overall E/E architecture. Through 5G connectivity, the Driveboard itself can leverage cloud computing capabilities and over-the-air updates to enhance the overall performance, reliability, and adaptability towards new Capsule technologies. In Section 5, measurements are conducted to show the capability of the proposed dynamic link concept. We provide performance benchmarks to evaluate the necessity and suitability of the cloud-based reconfiguration mechanisms.

2. State of the Art

The automotive industry is undergoing a significant transformation as it shifts from traditional signal-oriented architectures to more flexible service-oriented architectures. Signal-oriented architectures have historically been the backbone of automotive E/E systems and they rely on predefined communication protocols, such as Controller Area Net-

work (CAN). This approach has limitations in the sense of modular vehicles, particularly in terms of scalability and flexibility, as it requires static configurations that can slow down the rapid development and adaptation of new functionalities [12].

Recent research explores various aspects of SOA implementation in automotive systems, focusing on resource allocation, network architecture, timing predictability, and communication protocols. A method for optimization-based resource allocation in SOA was proposed in [13]; it utilizes mathematical optimization to schedule service parameters while considering dependencies between data flows and computations within services. The research focused on optimizing resources locally inside vehicle ECUs and across multiple services and ECUs. The integration of Software-Defined Networks (SDNs) with service-oriented architectures was investigated in [14] through a simulation environment as a means to support dynamic network configurations in vehicles. This research demonstrated the use of SDNs to support a dynamic, service-oriented network architecture using the SOME/IP protocol. Furthermore, a formal timing analysis of the discovery phase in SOME/IP was developed in [15]. This work highlighted the importance of the SOA discovery phase in determining the readiness of vehicle sub-systems, as the time required to complete this phase is a critical parameter for overall system performance. A case study [16] discussed the implementation of an SOA distributed feedback control loop automotive application using SOME/IP and programmable data planes, and demonstrated the potential benefits of incorporating SOME/IP awareness of overall automotive SOA applications and network devices. Efforts to improve the performance and platform independence of SOA implementations have also been made. Bitroute SOME/IP, a new implementation of the SOME/IP communication protocol, focuses on execution performance and deployment platform independence, addressing the need for scalable and efficient communication middleware in modern automotive systems in [17]. These advancements in SOA applications for automotive systems represent significant progress towards more flexible, scalable, and efficient vehicle software architectures. The UNICARagil research project is based on a scalable drive platform and various usage units [18], similar to the U-Shift project. Compared to U-Shift, the authors used a native SOA approach for their autonomous vehicle solutions [19,20].

Currently, hybrid signal- and service-oriented architectures that include a combination of Ethernet, CAN, and FlexRay, are more realistic in practice, as stated in [13]. One of the first efforts of hybrid service-oriented architectures with a practical case study was shown in [21]. Applications of hybrid software architectures were also discussed in the context of the classical signal-based, CAN and ECU-driven AUTOSAR Classic and the Adaptive platforms that form a SOA using Ethernet [22]. The state of the art indicates that in order to fully leverage the advantages of a pure SOA required for modular autonomous electrical vehicles, a controlled transition of hybrid architecture presents itself as an intermedial solution for the current state of industry based on signal-oriented approaches.

In the cross-section area of SOA and cloud technologies for automotive applications, there is a significant shift towards more dynamic, flexible, and interconnected systems. The authors of [23] introduced an Orchestrator for dynamically extending automotive E/E architectures to the cloud, enabling Control Over the Air (COTA) capabilities for non-safety-critical functions. This approach aligns with the growing trend of service orchestration across edge and cloud environments, as exemplified by the study on object detection in industrial vehicles [24]. Further work carried out in [25] shows a multi-layered SOA which provides abstractions for heterogeneous computing hardware and facilitates the integration of cloud computing and big data analysis for advanced autonomous driving technologies. To ensure high availability and enhance Quality of Service (QoS) in cloud computing, various replication strategies have been developed, categorized into data-oriented and

service-oriented approaches in [26]. Furthermore, a demonstration of the vehicle update management systems connecting in-car communication networks with out-car services, including cloud vehicle services, was provided in [27].

The utilization of cloud technologies in the automotive sector spreads towards the online reconfiguration of automotive software architectures. The expansion of on-board computing and storage capacity through cloud computing has given rise to the concept of cloud-based vehicle functions, which use cloud capabilities to enhance vehicle control strategies and enable new functionalities that would otherwise not be feasible with on-board resources [28]. The authors of [29] emphasized the potential of cloud connectivity in vehicles, enabling computationally intensive machine learning models to be executed in the cloud, with applications such as reducing the energy impact of air conditioning in city buses. By leveraging the scalability and computing power of the cloud, a mixture-of-experts approach can be used [30], where a gating mechanism dynamically selects specialized models based on context-specific data to optimize vehicle functions and adapt to different scenarios. The authors of [31] proposed a reconfiguration protocol for Ethernet TSN in automotive systems, which enables dynamic over-the-air network operation using the cloud with features like traffic isolation, fault recovery, and performance adjustments, enhancing both safety and user experience with reconfiguration. Furthermore, Ref. [32] introduced a cost-aware multifaceted reconfiguration approach for dynamic routing applications, leveraging infrastructure-as-code modules for efficient resource allocation and the rescheduling of idle components. Finally, Ref. [33] presents a smart cloud management system using a knowledge base to model cloud resources, manage service-level agreements, and optimize resource allocation through RESTful services.

3. Hybrid E/E Architecture Concept

3.1. Dynamic Link System Architecture

For the use case of the Driveboard, a dynamic link system architecture emerges as a viable solution to address the challenges associated with the evolving landscape of Capsule technologies. Due to the inherently static nature of signal-based systems, such as CAN bus, a well-defined information flow interface is required to bridge the service-oriented and signal-based components of the hybrid software architecture. In addition, updates to the signal-oriented ECU occur infrequently and are typically performed during major maintenance in a workshop. To ensure a universally applicable Driveboard, a universal dynamic link system message becomes essential.

The Dynamic Link System (DLS) message comprises two essential parts: the Universal Information Field (UIF) and a corresponding Rule Identifier (RID). The UIF acts as a unique identifier that dictates how to interpret the data within the DLS message. Essentially, the RID serves as an interface definition for the Orchestrator's Dynamic Link X service. It creates a standardized communication protocol between the Driveboard and Capsule. This design ensures both backward and forward compatibility. This means that the Driveboard can understand messages from Capsules built with different technologies and functionalities. An RID-based design enables the DLS architecture to support modular and extensible interactions. As new Capsule features or functionalities are developed, corresponding RIDs can be introduced to define these advancements without requiring modifications to the Zone Controller ECU's software. This modularity ensures that the Driveboard remains versatile and capable of supporting a wide array of Capsules without necessitating fundamental redesigns or software updates. The UIF within the DLS message acts as a container for transmitting various types of data, including sensor readings, control commands, and status updates. Capsule services leverage the DLS message to interface with the signal-oriented components of the Driveboard, such as CAN bus and

other communication buses. The encapsulated information within the UIF is translated and bidirectionally routed, enabling efficient and standardized communication between the Driveboard and the Capsule. This architecture ensures a robust and adaptable platform for seamless interaction across diverse Capsule technologies.

A key factor of the DLS is the dynamic integration of Capsules during runtime. Therefore, a Plug-and-Play compatibility is needed to process of connecting and disconnecting Capsules. The overall DLS architecture is displayed in Figure 2. As can be observed on the left side, one Capsule X of a variety of Capsules connects the DLS to the Driveboard, where it initially shares the Capsule type with the Driveboard Orchestrator, as can be seen in Figure 3. The Orchestrator then contacts the Capsule Service Backend to download the corresponding Dynamic Link X service and integrates it into the already operating SOA. As soon as the new Capsule connects with the Driveboard, it announces itself with its Capsule type. After a handshake process, the Driveboard first downloads the new service and starts the Capsule-specific Dynamic Link X service. Once this Dynamic Link X is ready to operate, it also sends an announcement message. After receiving a final acknowledgement flag from the Capsule X, the Dynamic Link X service starts to operate. This Dynamic Link X can then be used for a wide variety of applications. For example, the Capsule can have, depending on the Capsule type, a battery storage of its own, which can be used to either support the battery storage of the Driveboard or, vice versa, to balance out the state of charge (SoC). By exchanging information on the SoC, encoded with a specific RID, the Driveboard can dynamically adjust power distribution to balance the SoC between the Capsule and the Driveboard. This ensures the optimal utilization of energy resources, prolongs the range, and can prevent deep discharges of the batteries, therefore extending the battery life. Another application would be a Capsule weight-based motor controller adjustment. With the ability to communicate weight information from sensors within the Capsule to the Driveboard, the DLS enables adaptive motor controller adjustments based on the Capsule's weight. For example, in a Capsule carrying varying payloads, the motor controller can dynamically adjust torque output or regenerative braking settings to optimize performance and efficiency based on the current load. Taking safety into account, the DLS enables the coupling of safety-critical elements of the Capsule and the Driveboard (e.g., the door interlocks of the people mover Capsule with the motor enable signal of the Driveboard). Other applications synchronize similar actuators of the Capsule and Driveboard, such as light controls, enabling the real-time monitoring of the overall system, among others. To maintain robust operation within the CAN bus by ensuring timely message transmission while preserving the flexibility of SOA, the system retains the most recent successful transmission between the two domains.

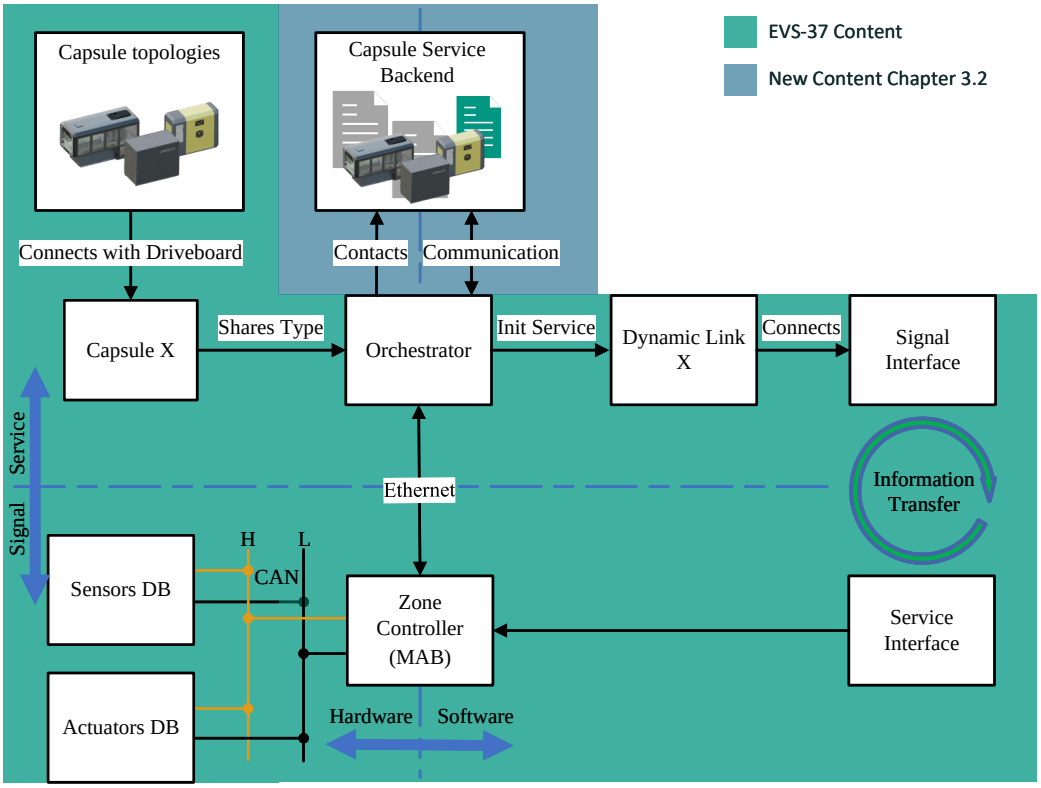

Figure 2. System architecture of dynamic link system based on [9].

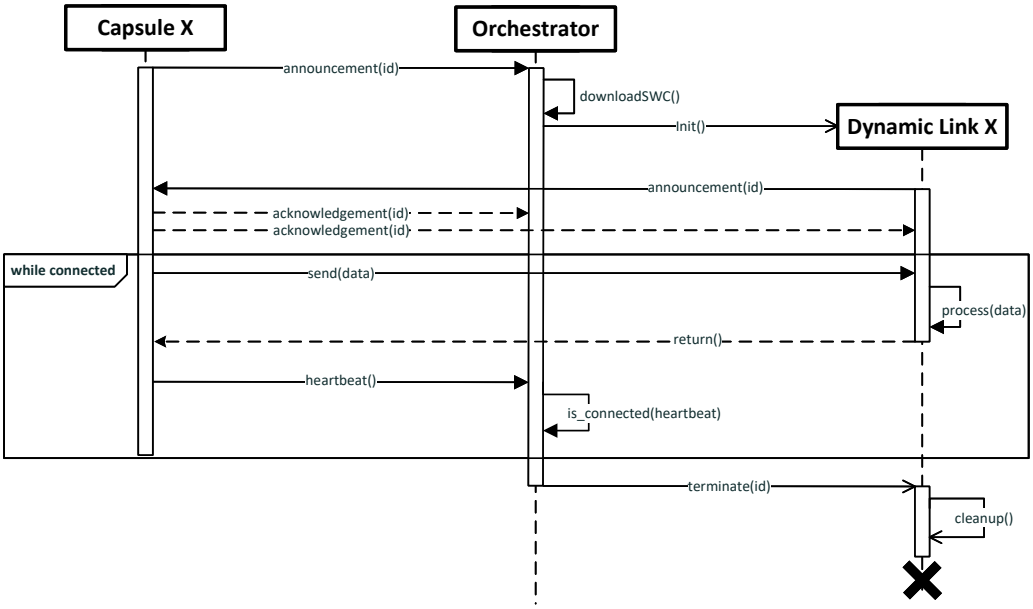

Figure 3. Startup sequence of new DLS X-Software Component (SWC) and exemplary interaction between Capsule and Driveboard.

3.2. Cloud-Enabled Software Reconfiguration

Running an SWC in the cloud offers a number of advantages. A key advantage of integrating the cloud is the ability to dynamically include real-time data and external information in the SWC. Relevant data include, for example, current weather data or traffic forecasts, which can be processed in real time and adapted to the individual requirements of individual vehicles. Furthermore, the cloud enables the creation of a comprehensive fleet database that centrally stores and aggregates information from all vehicles in a fleet.

Another advantage of outsourcing to the cloud is that it overcomes the resource limitations imposed by the hardware within a vehicle. Cloud systems offer virtually unlimited computing and storage capacities, so that computing and data-intensive applications can be executed without using up the capacities of the on-board systems. Finally, the cloud architecture ensures the global availability of outsourced functions and data. Relevant information can be provided, updated, and used regardless of the vehicle's location, which is particularly advantageous for globally operating fleets. Before an SWC can be executed in the cloud, it must first be evaluated for its feasibility to be offloaded (see Figure 4). Preliminary work on this subject has already been conducted in [34] and can be incorporated into this process. The evaluation process consists of two steps. The first stage determines the operational relevance of the function based on criteria such as real-time requirements or Automotive Safety Integrity Level (ASIL). If a function is deemed not operationally relevant, its suitability for offloading to the cloud can then be assessed. This suitability is determined by a combination of quantitative and qualitative criteria. Quantitative criteria include potential energy savings at the hardware compute level and potentially faster response times due to faster processing in the cloud. The potential added value of outsourcing a function, such as the benefits of fleet learning, is assessed qualitatively and is also included in the evaluation process. At the end of this process, each function is assigned a suitability score that is normalized to a range between 0 and 1, with higher scores indicating greater suitability for cloud offloading. The main advantage of running SWC in the cloud is the ability to have not just one general model for a functionality, but different experts. An expert model is a specialized machine learning model that has been trained on a specific context or dataset and thus achieves particularly high performance in its area of application. However, a general base model is still required to serve as a fallback solution if no suitable expert is available. The use of a general base model as a fallback solution is essential to ensure that the system remains operational even when no suitable expert model is available or applicable. This base model can serve as a versatile solution capable of providing acceptable results across a wide range of contexts. However, it may offer lower performance compared to the specialized expert models. The combination of expert models and a base model allows the system to achieve both flexibility and efficiency, dynamically adapting to various requirements and data contexts.

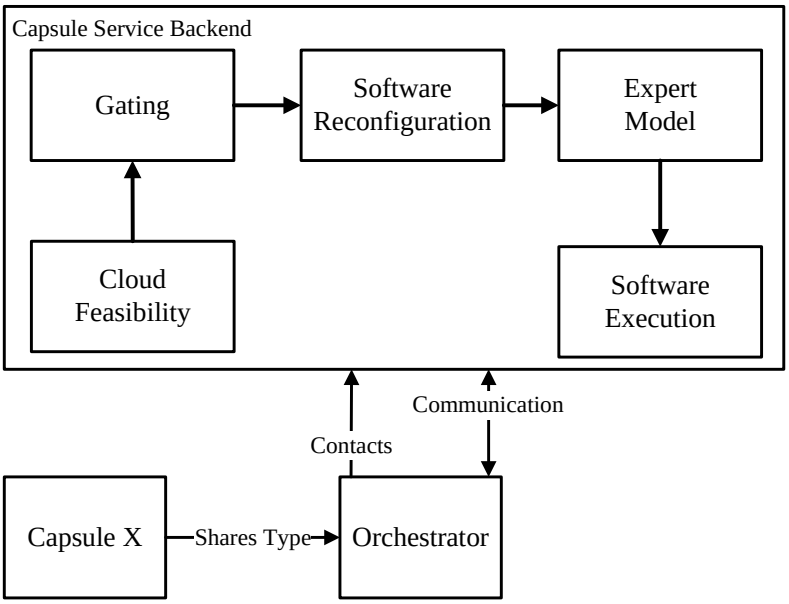

Figure 4. Reconfiguration of a software component in the cloud with a gating mechanism and the context for selecting expert networks.

A general machine learning model that offers a broad coverage of all possible context areas serves as a starting point. In addition, a number of expert models are required, each of which has been trained for specific contexts. These expert models should have clear areas of application that can be adapted depending on the context. Furthermore, these expert models can be gradually learned and incorporated during the system's operation. As new contexts emerge or existing ones evolve, the system can continuously refine its expertise by training and integrating new specialized models. This dynamic approach allows the system to improve its performance over time.When a request is made to the cloud, the appropriate expert must be selected via a gating (see Figure 4). The gating can be a simple lookup table or a trainable mechanism such as a neural network or a decision tree. Based on the input data, a decision is made as to which expert is relevant for the given task. If no expert model is suitable or available, the general model or the model with the greatest coverage is selected. Depending on the expert selected, the software is reconfigured accordingly and executed in the cloud (see Figure 4). The reconfiguration process involves dynamically loading the required model, adapting the processing pipeline, and ensuring compatibility with the incoming data format. Software execution is then orchestrated by allocating the necessary computational resources. The result is sent back to the vehicle via the communication channel.

4. Prototypical Setup

The goal of sustainable mobility can be achieved through modular electric vehicle architectures, in which fewer vehicles with a higher daily usage time complete the necessary journeys. In order to achieve the highest possible modularity, the U-Shift II vehicle concept breaks up existing structures and divides the vehicle into a drive unit (Driveboard) and a vehicle body (Capsule), as can be seen in Figure 1a,b, where a prototype Driveboard with a people mover Capsule from a predecessor project can be seen.

The Driveboard integrates all the components required for fully automated electric vehicle operation, including electric drives, a battery system, control units, and environmental sensors. Automation is achieved using the novel hybrid A* path planning algorithm proposed by Schumann et al. [35], which integrates the unique driving characteristics of the Driveboard. The specific application of the vehicle is determined by selecting an appropriate Capsule. Depending on the chosen Capsule, the vehicle can support diverse use cases such as passenger transport, food delivery, bicycle collection, or cargo logistics. A lifting mechanism embedded within the chassis enables the vehicle to autonomously pick up and dock with Capsules without external assistance.

A key design requirement for the Driveboard is its adaptability to the varying architectures of different Capsules, addressing the challenges of modern mobility. This adaptability ensures compatibility with ongoing advancements and dynamic developments in Capsule technologies. Since new Capsule designs, functionalities, and configurations are likely to emerge after the Driveboard's production, it must accommodate these changes to remain effective and versatile across diverse vehicle applications. This adaptability is essential for sustaining the Driveboard's relevance in an evolving technological landscape. The hardware setup implemented in the U-Shift II system for dynamic linking is illustrated in Figure 5. This setup is divided into two primary sections. The main section is located at the front of the Driveboard and houses components facilitating the integration of service-oriented and signal-oriented architectures, highlighted in blue. This section also includes the Capsule coupling mechanism, which interfaces with key Driveboard components, such as the Orchestrator. The Orchestrator manages communication and operates the service-oriented architecture. The dSpace MicroAutoBox (MAB) serves as the Zone Controller, functioning as a real-time-capable rapid prototyping system that manages the control and

communication of most sensors and actuators on the Driveboard. Communication in this system occurs via signal-oriented methods, utilizing various CAN bus components to meet hard real-time requirements.

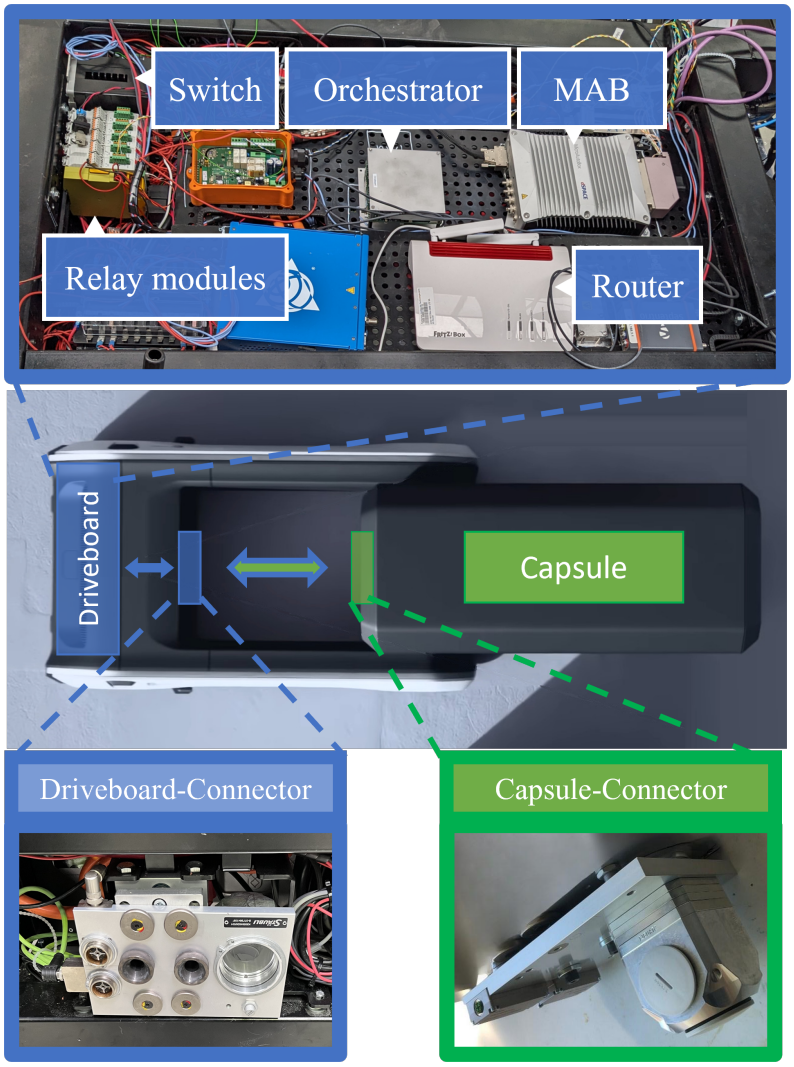

Figure 5. U-Shift II prototype hardware setup.

The MAB and the Orchestrator execute software components designed to facilitate dynamic integration between signal-oriented and service-oriented architectures. To meet the real-time performance requirements of the Driveboard and ensure the execution of safety-critical functions within specified temporal constraints, the MAB operates at a clock frequency of 1 kHz. This configuration ensures that all essential tasks for controlling the Driveboard are executed reliably. Communication between the MAB and other control units within the network is established through the Ethernet interface using the UDP protocol. This approach enables the seamless integration of service-oriented architecture elements into the signal-oriented framework of the Driveboard. The switch is used as a physical interface between Orchestrator, MAB and possibly connected Capsules. The time synchronization in normal operation is established between the MAB, Orchestrator, and Capsules through a high-precision Precision Time Protocol (PTP) Grandmaster. The second part of the hardware setup is indicated in green and represents the Capsule and its coupling mechanisms between the Driveboard and Capsules, respectively. They offer mechanical, power, and networking coupling that ensures the full modularity of U-Shift II. Within the Capsule, there is at least one small ECU that acts as an interface between the Orchestrator

and the Capsule itself. As a cloud provider, we use our in-house server infrastructure to create a dedicated Virtual Machine (VM) for U-Shift II as the Capsule Service Backend (CSB). The VM is vertically and horizontally scalable as needed for the deployment of updates to the Driveboard and the execution of other SWCs. The Driveboard initiates an IPSec-secured VPN connection with Advanced Encryption Standard (AES) to the CSB if needed. This ensures the authenticity and security of the communication between the Driveboard and CSB.

5. Measurements and Evaluation

For the experiment, an implementation of the DLS architecture was evaluated by taking two of the aforementioned scenarios into account. Firstly, there was the shared weight information of the Capsule attached, which could then be used to optimize control parameters within the MAB, and, secondly, to measure the transmission delay from synchronizing signals from the Dynamic Link X service to the signal-oriented CAN bus, we controlled the light actuators of the Driveboard which are components of the CAN bus system. For the measurements including latency information, the Orchestrator PC and the MAB were first synchronized directly with PTP, where the MAB serve as master clock.

The performance of the PTP synchronization was observed with approximately 9000 measurement points and can be seen in Figure 6, with a median time deviation of the MAB and Orchestrator of 56 µs, where the maximum outlier peaked at 246 µs. Therefore, the worst case approximation of the synchronization in our hardware setup in the measurements can be assumed at 250 µs. The results shown in the Figure 6 reflect the specific hardware setup used in the case of the U-Shift II demonstrator, but the reasoning and performance will be scaled to this solution. The average transmission times from the Orchestrator to the MAB were 1.23 ms, which is below 2.5 ms, which therefore can be assumed as the worst case transmission time between two control units in the vehicle. This opens up the possibility of controlling even, as aforementioned, safety-critical functions like battery shutdown or emergency stopping, via the dynamic link interface, although these were not explicitly considered in the current work. The variance in the transmission times depending on the direction resulted from the constant clock of the MAB, which was firmly defined to maintain real-time capability and was not further reduced due to the computing time required for other tasks. For further processing and sending the corresponding message on the CAN bus, an additional 1.25 ms was required on average until the message arrived on the CAN bus and the lights were activated. Once the weight had been successfully transmitted from the Orchestrator to the MAB, the MAB set various model variables, such as control parameters for trajectory tracking, depending on the weight.

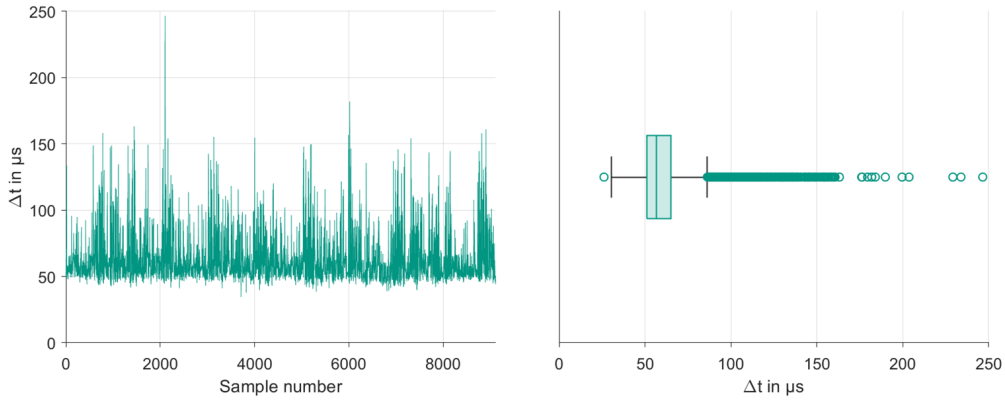

Figure 6. PTP synchronization of MAB and Orchestrator.

Within Figure 7, a sequence of message frames and the corresponding information extraction can be seen, which is executed by the MAB and the Orchestrator. In the upper measurement, the raw UIF message frame, not yet separated (but for clarity already colored properly), can be observed. Note that there are two axes, as the scale deviates too much otherwise. In the second measurement is the corresponding RID, which is needed to interpret the information correctly. The MAB receiving the DLS message can use the RIDs as context markers to determine how to interpret the information contained within the UIF. This contextual interpretation ensures that each component processes the data according to its specific requirements and functionalities. This process can be seen in the lower two measurements, where the separation is performed and processed accordingly. While the RID value is 11, the Driveboard shares the information to synchronize the left turn signal from the CAN bus to the Orchestrator (and therefore the Capsule's Dynamic Link X service), whereas within the time RID equals 10, the Capsule shares its current weight. This weight can then be used further to adjust controller parameters to gain a higher performance.

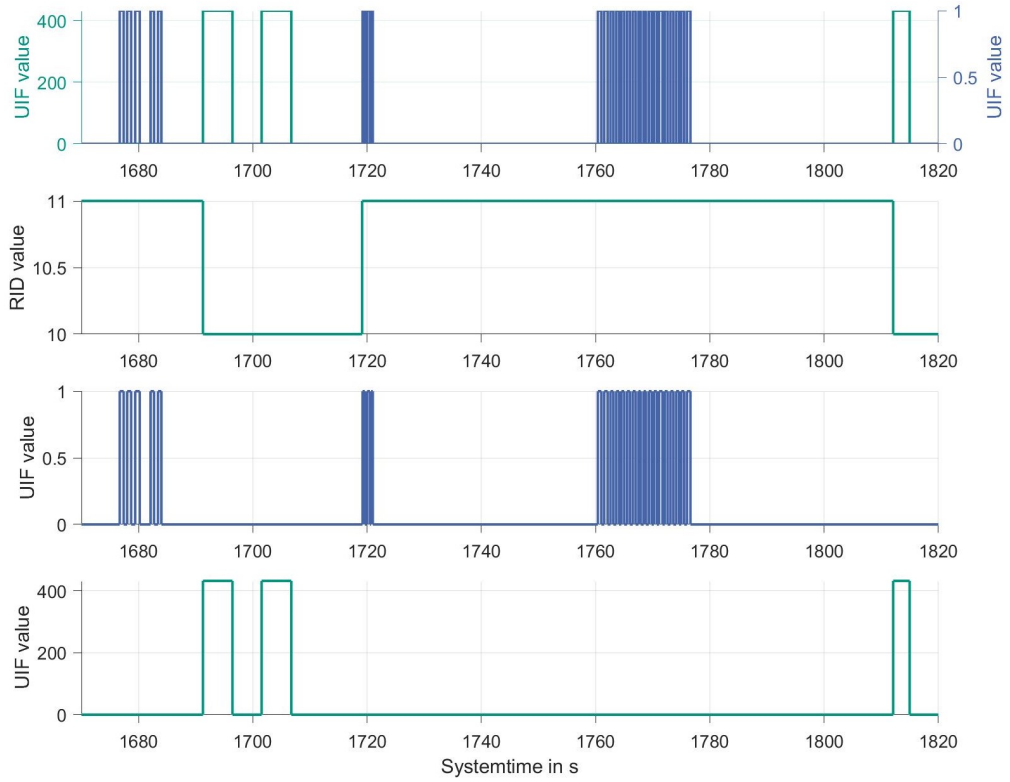

Figure 7. Received raw UIF message and separated information based on RID.

Figure 8 presents the relationship between the CPU utilization and the message frequency measured over time. Within the measurement, a benchmark is conducted to better understand the effects of a high CPU load. The red line indicates the start of the benchmark, where the CPU load is slowly raised from basic load utilization at roughly 5%, linearly increasing within 30 s up to 100%. During the increasing utilization, a steep drop in the service frequency from well above 90 Hz to 40 Hz occurs. The reduced frequency stays well below the target frequency of 100 Hz until the CPU utilization decreases again to the base load. Both the drop and rise in the service frequency happen at roughly 60% CPU utilization, and therefore a correlation between CPU utilization and service frequency becomes evident. More precisely, the correlation between CPU utilization and service frequency is −0.39, which therefore hints towards a moderate negative correlation. Consequently, a high CPU load should be avoided during operation. The growing computational demands of

additional software components, such as Capsule-specific SWCs, can negatively impact the performance of sensor data processing as shown in Figure 8, potentially compromising the effectiveness of automated driving functions. In the worst case, this could result in a failure to execute these functions successfully. To prevent this, resource-intensive, non-safety-critical tasks should be offloaded to the cloud whenever possible. By reallocating only non-time-critical SWCs, this limitation can be effectively mitigated.

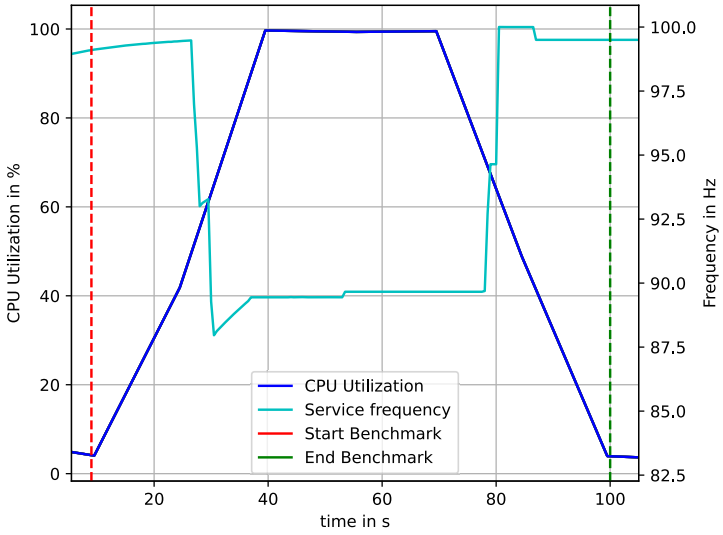

Figure 8. Impact of CPU utilization on the frequency of an SWC.

6. Conclusions

In summary, the novel DLS architecture demonstrates its potential to address key challenges in developing a modular, on-the-road vehicle concept. By enabling seamless communication, adaptability, and integration among diverse Capsule and Driveboard components, the DLS facilitates advanced functionalities, including safety-critical features. These include coupling motor control with safety mechanisms in the Capsule or dynamically balancing the state of charge between the Driveboard and, if available, the Capsule's battery storage. In collaboration with our partners, we developed a full-sized Driveboard prototype featuring the proposed hybrid E/E architecture. Our initial testing confirmed that the DLS enables a low-latency connection between the signal-oriented architecture and SOA. The measurements included various bidirectional messages, demonstrating promising results—most notably, a dynamic link message that meets hard real-time requirements. The DLS offers the flexibility to adapt to new Capsule technologies, even after the Driveboard is already on the road. Through the Orchestrator and MAB, interactions can be dynamically adjusted using the proposed RIDs that define universal information fields. Additionally, the cloud-enabled software reconfiguration allows for offloading SWCs to the cloud, offering advantages such as enhanced computational power, real-time data processing, and global availability. This reconfiguration process is facilitated by expert models and a gating mechanism that dynamically selects the most suitable model for a given context, optimizing performance and resource usage. In the context of hybrid software architectures, our focus on a single bus protocol in the signal-oriented architecture aligns with the prototype-driven nature of this research. While the prototype is built around a single bus protocol, the methodology remains flexible. This enables future research to investigate alternative protocols for broader feasibility in automotive applications. In order to transfer our cloud-based reconfiguration methodology to other E/E architectures, an individual study must be carried out to determine the system capabilities. This is due to the fact that

different bus protocols, physical transmission layers, and processing units can result in different behavior, for example, in terms of latency.

Further research is needed to address connectivity failures, security challenges, the integration of new Capsule topologies, and a comparison of different E/E architectures. To improve robustness against connectivity failures, our current system keeps the last successful transmission between the service- and signal-oriented architectures. The proposed system could be enhanced through internal redundancies or fallback functionalities in the event of transmission errors. Regarding cybersecurity, while IPSec with AES encryption provides a solid foundation, additional measures for cyberattack detection and mitigation can improve secure operation. Furthermore, an evaluation of our system through different Capsule topologies is necessary to assess the feasibility and scalability of the DLS. To support this, we are actively collecting extensive data, enabling further analysis and direct comparisons with alternative technologies.

For the comparison of different E/E architectures, we aim to focus our research on examining how multiple CAN domains and other in-vehicle networks interact with the DLS architecture. In parallel, we are actively collecting data from our prototype to evaluate system performance and will explore strategies for integrating safety-critical communication within the Capsule.

Author Contributions: Conceptualization, D.K. and D.B.; Methodology, D.K. and D.B.; software, D.K. and D.B.; validation, D.K., D.B. and V.V.; formal analysis, D.K., D.B. and V.V.; investigation, V.V.; resources, D.K. and V.V.; data curation, D.K. and V.V.; writing—original draft preparation, D.K.; writing—review and editing, D.K.; visualization, D.K.; supervision, E.S.; project administration, E.S.; funding acquisition, E.S. All authors have read and agreed to the published version of the manuscript.

Funding: We acknowledge the support from the German State Ministry of Economic Affairs Baden-Wuerttemberg within the project U-Shift II (AK 3-433.62-DLR/60).

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Acknowledgments: This study is based on an oral poster presentation delivered at EVS37 in Seoul, Republic of Korea, on 24 April 2024. We extend our gratitude to the EVS37 program for the opportunity to contribute this research paper to the *WEVJ* Special Issue.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Dlugosch, O.; Brandt, T.; Neumann, D. Combining analytics and simulation methods to assess the impact of shared, autonomous electric vehicles on sustainable urban mobility. *Inf. Manag.* **2022**, *59*, 103285. [CrossRef]
2. Mercedes-Benz Group. Mercedes-Benz Vans Sets the Course for a Fully Electric Future. 2023. Available online: https://group.mercedes-benz.com/innovation/drive-systems/electric/van-ea.html (accessed on 18 February 2025).
3. Kia. CES 2024 | Kia Global Brand Site | Movement That Inspires. Available online: https://worldwide.kia.com/int/ces2024 (accessed on 18 February 2025).
4. Trifunović, N.; Vujanić, M.; Kenjić, D.; Antić, M. Data Exchange Interfaces in Automotive SOA. In Proceedings of the 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia , 23–27 May 2022; pp. 1416–1419. [CrossRef]
5. Henle, J.; Stoffel, M.; Schindewolf, M.; Nägele, A.T.; Sax, E. Architecture platforms for future vehicles: A comparison of ROS2 and Adaptive AUTOSAR. In Proceedings of the 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), Macau, China, 8–12 October 2022; pp. 3095–3102. [CrossRef]
6. Pöhnl, M.; Tamisier, A.; Blass, T. A Middleware Journey from Microcontrollers to Microprocessors. In Proceedings of the 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE), Antwerp, Belgium, 14–23 March 2022; pp. 282–286. CrossRef]
7. *EN 302 637-2-V1.3.1*; Intelligent Transport Systems (ITS), Vehicular Communications, Basic Set of Applications, Part 2: Specification of Cooperative Awareness Basic Service. European Telecommunications Standards Institute (ETSI): Sophia Antipolis, France, 2014.

8. *EN 302 637-3-V1.3.1*; Intelligent Transport Systems (ITS), Vehicular Communications, Basic Set of Applications, Part 3: Specifications of Decentralized Environmental Notification Basic Service. European Telecommunications Standards Institute (ETSI): Sophia Antipolis, France, 2019.
9. Kraus, D.; Fuchs, S.; Vucinic, V.; Kiebler, J.; Wagner, A.; Sax, E. Introducing a Dynamic Link for Hybrid E/E-Architectures of Modular Electric Autonomous Vehicles. In Proceedings of the EVS37 Symposium: COEX, Seoul, Republic of Korea, 23–26 April 2024.
10. Münster, M.; Brost, M.; Siefkes, T.; Kopp, G.; Beeh, E.; Rinderknecht, F.; Schmid, S.; Osebek, M.; Scheibe, S.; Hahn, R.; et al. U-Shift II Vision and Project Goals. In *22. Internationales Stuttgarter Symposium—Automobil- und Motorentechnik*; Bargende, M., Ed.; Springer: Berlin/Heidelberg, Germany, 2022; p. 18. [CrossRef]
11. DLR, U-Shift On-The-Road Modular Vehicle, DLR Transport. Available online: https://verkehrsforschung.dlr.de/en/projects/u-shift (accessed on 18 February 2025)
12. Bengtsson, H.H.; Hiller, M.; Migge, J.; Navet, N. Signal-Oriented ECUs in a Centralized Service-Oriented Architecture: Scalability of the Layered Software Architecture. In Proceedings of the Automotive Ethernet Congress 2022, Munich, Germany, 1–2 June 2022.
13. Kampmann, A.; Lüer, M.; Kowalewski, S.; Alrifaee, B. Optimization-based Resource Allocation for an Automotive Service-oriented Software Architecture. In Proceedings of the 2022 IEEE Intelligent Vehicles Symposium (IV), Aachen, Germany, 4–9 June 2022; pp. 678–687. [CrossRef]
14. Häckel, T.; Meyer, P.; Mueller, M.; Schmitt-Solbrig, J.; Korf, F.; Schmidt, T.C. Dynamic Service-Orientation for Software-Defined In-Vehicle Networks. In Proceedings of the 2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring), Florence, Italy, 20–23 June 2023; pp. 1–5. [CrossRef]
15. Fraccaroli, E.; Joshi, P.; Xu, S.; Shazzad, K.; Jochim, M.; Chakraborty, S. Timing predictability for SOME/IP-based service-oriented automotive in-vehicle networks. In Proceedings of the 2023 Design, Automation & Test in Europe Conference & Exhibition (DATE), Antwerp, Belgium, 17–19 April 2023; pp. 1–6. [CrossRef]
16. Nayak, N.; Ambalavanan, U.; Thampan, J.M.; Grewe, D.; Wagner, M.; Schildt, S.; Ott, J. Reimagining automotive service-oriented communication: A case study on programmable data planes. *IEEE Veh. Technol. Mag.* **2023**, *18*, 69–79. [CrossRef]
17. Vujanić, M.; Trifunović, N.; Kaštelan, I.; Kovačević, B. Bitroute SOME/IP: Implementation of a Scalable and Service Oriented Communication Middleware. In Proceedings of the 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia, 23–27 May 2022; pp. 1426–1429. [CrossRef]
18. Keilhoff, D.; Niedballa, D.; Reuss, H.C.; Buchholz, M.; Gies, F.; Dietmayer, K.; Lauer, M.; Stiller, C.; Ackermann, S.; Winner, H.; et al. UNICARagil–New architectures for disruptive vehicle concepts. In *19. Internationales Stuttgarter Symposium: Automobil-und Motorentechnik*; Springer: Berlin/Heidelberg, Germany, 2019, pp. 830–842. [CrossRef]
19. Kampmann, A.; Alrifaee, B.; Kohout, M.; Wüstenberg, A.; Woopen, T.; Nolte, M.; Eckstein, L.; Kowalewski, S. A dynamic service-oriented software architecture for highly automated vehicles. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 2101–2108. [CrossRef]
20. Mokhtarian, A.; Alrifaee, B.; Kampmann, A. The dynamic service-oriented software architecture for the UNICARagil Project. In Proceedings of the 29th Aachen Colloquium Sustainable Mobility, Aachen, Germany, 5–7 October 2020. [CrossRef]
21. Baresi, L.; Ghezzi, C.; Miele, A.; Miraz, M.; Naggi, A.; Pacifici, F. Hybrid service-oriented architectures: A case-study in the automotive domain. In Proceedings of the 5th International Workshop on Software Engineering and Middleware, Lisbon, Portugal, 5–6 September 2005; pp. 62–68. [CrossRef]
22. Tischer, M. The computing center in the vehicle: Autosar adaptive. Elektronik automotive, special issue "Bordnetz", September 2018. Available online: https://cdn.vector.com/cms/content/know-how/_technical-articles/AUTOSAR/AUTOSAR_Adaptive_ElektronikAutomotive_201809_PressArticle_EN.pdf (accessed on 18 February 2025)
23. Sommer, M.; Guissouma, H.; Schindewolf, M.; Sax, E. An Orchestrator for the Dynamic Extension of Automotive E/E Architectures to the Cloud. In *Service-Oriented Computing*; Aiello, M., Barzen, J., Dustdar, S., Leymann, F., Eds.; Spring: Cham, Switzerland, 2025; pp. 24–41. [CrossRef]
24. Pettinen, H.; Hästbacka, D. Service orchestration for object detection on edge and cloud in dependable industrial vehicles. *J. Mob. Multimed.* **2022**, *18*, 1–26. [CrossRef]
25. Yu, D.; Xiao, A. *The Digital Foundation Platform—A Multi-layered SOA Architecture for Intelligent Connected Vehicle Operating System*; SAE Technical Papers; SAE International: Warrendale, PA, USA, 2022. [CrossRef]
26. Slimani, S.; Hamrouni, T.; Ben Charrada, F. Service-oriented replication strategies for improving quality-of-service in cloud computing: A survey. *Clust. Comput.* **2021**, *24*, 361–392. [CrossRef]
27. Aust, S. Vehicle Update Management in Software Defined Vehicles. In Proceedings of the 2022 IEEE 47th Conference on Local Computer Networks (LCN), Edmonton, AB, Canada, 26–29 September 2022; pp. 261–263. CrossRef]
28. Milani, F.; Beidl, C. Cloud-based Vehicle Functions: Motivation, Use-cases and Classification. In Proceedings of the 2018 IEEE Vehicular Networking Conference (VNC), Taipei, Taiwan, 5–7 December 2018; pp. 1–4, ISSN: 2157-9865. [CrossRef]

29. Baumann, D.; Sommer, M.; Dettinger, F.; Rösch, T.; Weyrich, M.; Sax, E. Connected Vehicle: Ontology, Taxonomy and Use Cases. In Proceedings of the 2024 IEEE International Systems Conference (SysCon), Montreal, QC, Canada, 15–18 April 2024; pp. 1–6, ISSN: 2472-9647. [CrossRef]
30. Masoudnia, S.; Ebrahimpour, R. Mixture of experts: A literature survey. *Artif. Intell. Rev.* **2014**, *42*, 275–293. [CrossRef]
31. Kostrzewa, A.; Ernst, R. Achieving safety and performance with reconfiguration protocol for ethernet TSN in automotive systems. *J. Syst. Archit.* **2021**, *118*, 102208. [CrossRef]
32. Amiri, A.; Zdun, U. Cost-Aware Multifaceted Reconfiguration of Service-and Cloud-Based Dynamic Routing Applications. In Proceedings of the 2023 IEEE 16th International Conference on Cloud Computing (CLOUD), Chicago, IL, USA, 2–8 July 2023; pp. 428–438. [CrossRef]
33. Chyad, H.S.; Mustafa, R.A.; George, D.N. Cloud resources modelling using smart cloud management. *Bull. Electr. Eng. Inform.* **2022**, *11*, 1134–1142. [CrossRef]
34. Sommer, M.; Baumann, D.; Rösch, T.; Dettinger, F.; Sax, E.; Weyrich, M. Process for the Identification of Vehicle Functions for Cloud Offloading. In *Science and Information Conference*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 596–608. [CrossRef]
35. Schumann, O.; Buchholz, M.; Dietmayer, K. Efficient Path Planning in Large Unknown Environments with Switchable System Models for Automated Vehicles. In Proceedings of the 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC), Bilbao, Spain, 24–28 September 2023; pp. 2466–2472. [CrossRef]