Automotive Software Engineering in an increasingly Data-Driven Automotive Sector

Oliver Denninger [1], Joachim K. Axmann [2], Severin Kacianka [3] und Bernd Westphal [4]

Abstract:

Automotive trends such as power-train electrification, personalization, connectivity, and automated driving are not well supported by the classical approach to hardware/software architectures that centre around numerous, dedicated electronic control units (ECUs) where software is delivered as part of the ECU and it and its environment does not change much after vehicle assembly. Similarly, current electronic architectures and vehicles do not exploit data-driven software development practices and do not have the capability to make use of unprecedented amounts of data on the vehicle, but also its environment and the Internet. These trends ask for a data-driven approach where the development, production, and operation data of automotive software feed back into continuous correction, improvement, and personalization. In this paper, we report findings from the Transformation Hub Automotive Software Engineering (TASTE) with two years of intensive discussions and workshops with a wide range of companies regarding the challenges facing the German based automotive industry in general, as well as individual companies from Original Equipment Manufacturers (OEM) to different suppliers (TIER-n). We discuss how previously different approaches need to be integrated into new software-centred development, update and maintenance processes throughout the automotive software value chains.

Keywords: Automotive Software Engineering, Data-Driven, Software-Centric, Automotive trends

1 Introduction

The automotive industry currently undergoes a major transformation that is driven by trends such as powertrain electrification (e.g., [WZD15]), personalisation (e.g., [En24]), connectivity (e.g., [Ra21]), and automated driving (e.g., [Yu20]). Tomorrow's cars and trucks will be fully integrated into digital environments and will produce large amounts of data. Vehicles are evolving into IoT devices on wheels, where new functions will be implemented primarily through software based on newly organised electronic platforms. The traditional boundaries between automotive electronics and consumer electronics will fade and disappear, and the autonomy of the individual vehicle will increase constantly.

[1] FZI Forschungszentrum Informatik, Karlsruhe, Germany, denninger@fzi.de, https://orcid.org/0000-0002-8649-7374

[2] Niedersächsisches Forschungstechnik für Fahrzeugtechnik (NFF), Hermann-Blenk-Straße 42, 38108 Braunschweig, Germany, j.axmann@tu-braunschweig.de, https://orcid.org/0000-0000-0000-0000

[3] fortiss, Center for Code Excellence, Guerickestraße 25, 80805 München, Germany, kacianka@fortiss.org, https://orcid.org/0000-0002-2546-3031

[4] Deutsches Zentrum für Luft und Raumfahrt e.V. (DLR), Systems Engineering for Future Mobility Institute, Escherweg 2, 26121 Oldenburg, Germany, bernd.westphal@dlr.de, https://orcid.org/0000-0002-6824-0567

Connected vehicles give OEMs real-time insight into how their customers drive, allowing them to make faster technical adjustments to improve the experience and optimise their manufacturing processes and services. In this way, they will be able to respond to customer needs and wishes as they arise, based on data and insights collected in real time. All these transformative aspects are often summarized under the term software-defined vehicle (SDV) [ZSL22].

On the other hand, many former industry certainties about business models, supply chains and competitive principles are changing in the area of automotive software and electronics. Traditional international OEM with globally recognised groups and brands, as well as established big automotive suppliers with engineering and manufacturing capacities all over the world, now face hard competitions with non-automotive industrial technology players who are very familiar with data-driven product development of software and electronics platforms from other technology sectors. To adapt to these changes, the automotive industry needs new business and collaboration models.

Our contribution is a discussion of possible future technological developments that will allow the industry to follow the above-mentioned trends to exploit available data more intensively in several ways.

The respective challenges and opportunities of OEMs, suppliers and other players are considered individually, and from the perspectives of processes and organisation, methods (e.g., systems engineering, DevOps), open source software, software modules and architectures, and the post-deployment phase. We see this discussion as an important starting point for future academic research on automotive software engineering.

2 Automotive Software

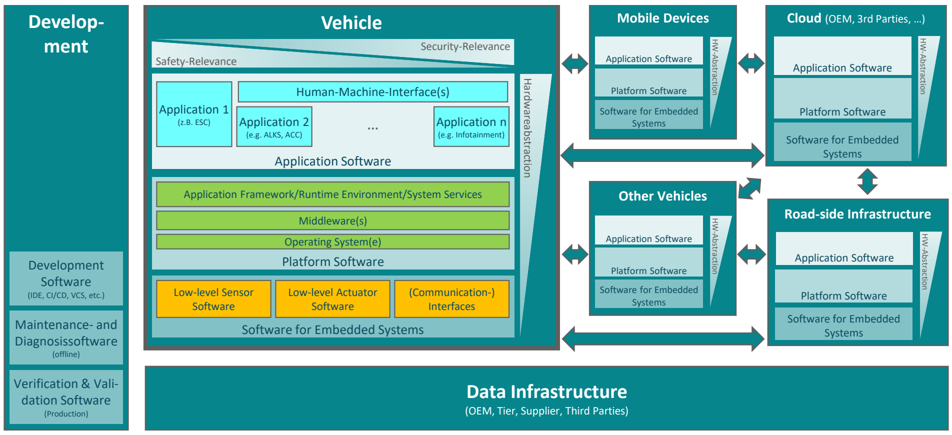

Abb. 1: Automotive Software

Automotive software as a term is defined differently depending on the focus of consideration. In addition to the narrow view, which refers to the software in the electronic architecture within the vehicles, there is a much broader view that includes the backend and also the software usage lifecycle. The respective data space must also be considered.

Directly recognisable for vehicle users is software that we call application software, which ranges from the software that realises safety functions like ABS, ESP, or ACC with or without graphical user interface to navigation and entertainment system [Ha17]. This software is usually built on some kind of platform software that abstracts from the underlying, distributed hardware in the vehicle cf. Box “Vehicle” in Figure 1. Elaborated libraries may already provide a notion of system services (like the indicator sub-system) which are used by applications such as remote lock/unlock.

Communication between ECUs (Electronic Control Units) is usually supported by middleware(s) for specific bus systems that are realised on top of the employed operating system. Further down in these levels of abstraction, we see classical embedded systems software for sensors and actuators of different complexity, and for communication infrastructure. Figure 1 represents all automotive software of the vehicle according to level of abstraction together, not the structure of a single ECU or centralised architecture (for that, see Section 2). In the traditional approach [SZ10] with distributed, dedicated ECUs developed by different TIERs, each ECU will use a software stack that covers multiple levels of the ones depicted in Figure 1. In a more centralised approach, there may, e.g., be less instances of middleware(s) running: one per zone SoC (System-on-Chip) as opposed to one per ECU.

With the trend to consider vehicles as devices in digital environments, the scope of automotive software in general becomes larger. There is a demand from customers to interact with vehicles through mobile devices, vehicles may communicate with other vehicles or roadside infrastructure (topic V2X), and vehicles may receive and contribute data from/to cloud services, which provide the connection to the data infrastructure of OEMs and suppliers as well as third parties, in particular within open data ecosystems.

Regarding this software, we mainly consider the counterparts running inside the vehicle where the boundaries between functions that are provided directly in the interaction between vehicles or with the integration of cloud back-ends are becoming blurred.

Software that is used in the development of automotive software in the above sense is considered to be tools in Automotive Software Engineering as opposed to the development of such software tools or the software that is used for vehicle production or supply chain management etc.

3 Emerging Trends

The following trends emerged from discussions in workshops, webinars and personal meetings over two years as part of the TASTE Transformations-Hub [TA24] that has centres spread throughout Germany's main automotive development areas. These trends fundamentally are the result from wider changes: connectivity, power-train electrification, personalization and automated driving mentioned in the introduction [Bä24; Bl24; Bu24]. Topics on which there is less consensus, such as the question of changes in individual vehicle ownership or changes towards public transport, were not considered.

3.1 More Centralized E/E Architectures

Without a significant increase in complexity and intricacy on the level on parts and connections, the increasing number of ECUs and the need for more computing power due to driver assistance functions can only be met by increasing centralization of the E/E architecture. In the future, functions previously implemented by different control units will be realized together on a more powerful System-on-a-Chip (SoC) [Ji24].

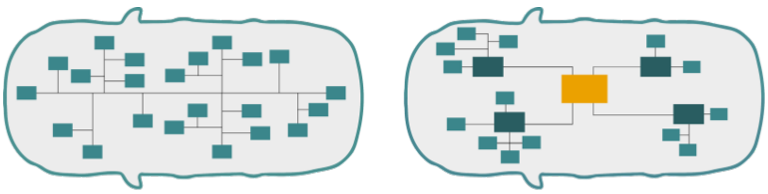

Abb. 2: E/E-Architecture: traditional ECU vs. centralised SoC

Zone- and domain-oriented concepts are being discussed, in which either functions in a specific area of the vehicle or similar functions are consolidated. Mixed forms are also possible. From the suppliers' point of view, there is a risk that different OEMs will take very different approaches here, resulting in high porting costs for the TIER-1 to (n) if functions have to be implemented as a dedicated ECU on the one hand and as a software component for an SoC with a complex software stack on the other.

3.2 More Complex Software Stacks

Due to the increasing centralization of the E/E architecture and the associated introduction of more powerful SoCs, the software stack is also changing. While ECUs are usually operated with a manageable stack consisting of e.g. OSEK OS and AUTOSAR Classic platform, SoCs are usually partitioned and/or virtualized in order to separate different functions and guarantee functional safety requirements. This is supplemented by middleware that enables connectivity both within the vehicle and with the cloud. In the future, even multi-level virtualization is possible, e.g. close to the operating system to separate different functions according to their ASIL level and at middleware level using containers for easy deployment of applications.

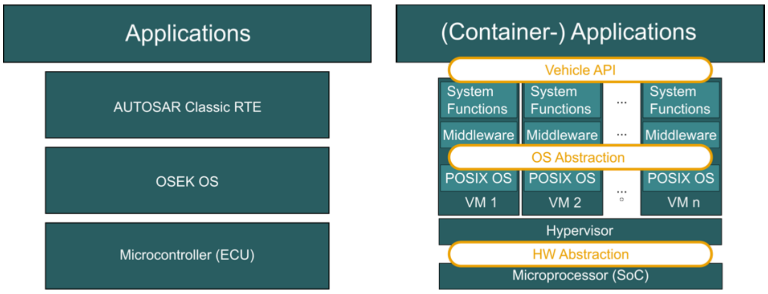

Abb. 3: Software stack: OSEK based ECU vs. complex virtualised SoC

As a result, a large number of new technologies are moving into the vehicle and the design space for how OEMs organize this is increasing considerably. For many suppliers, there is likely to be a need to master several layers of this complex software stack.

3.3 Data Access to the Vehicle

Automotive software will also be increasingly data-driven - AI-based assistance systems process large amounts of data, digital services are closely linked to cloud backends, driving data must be recorded for regulatory reasons and diagnostic data is analysed in real time [No23]. There is currently considerable political controversy over how access to vehicle data should be organized. Numerous variants are being discussed between completely free access and the OEMs as gatekeeper of vehicle data. This issue has a significant impact on suppliers. Whether diagnostic data can be retrieved directly from vehicle components or whether data transfers have to run via the OEM or data spaces such as Catena-X [CA24] or other cloud data storage vendors has a considerable influence on the type and software architecture of applications as well as quality of service that can be implemented. Another major change is that data allows manufacturer to understand the use of functionality (e.g. how often is seat-heating really used) and can help diagnose problems. Furthermore, with the rise of AI systems, data will be necessary to continuously improve the whole fleet and bring the need to ensure the quality of this data at runtime into focus.

3.4 User Expectations of a Lifestyle IT Product

Customers increasingly expect to be able to use familiar services such as social media, music or video streaming etc. in the vehicle. However, the simple solution of integrating customers' smartphones into the vehicle has disadvantages. On the one hand, OEMs are ceding control to smartphone manufacturers and, on the other, smartphone apps cannot exploit the potential of using vehicle data without additional interfaces and specific adaptations. Many players therefore prefer to run dedicated apps in the vehicles. OEMs would like to establish manufacturers' own app stores. For suppliers and service providers in particular, a landscape of highly heterogeneous app ecosystems threatens to emerge.

Regardless of the question of technical and organizational implementation, apps follow a completely different release cycle than vehicles. The automotive sector will have to meet customer expectations here and also enable fast release cycles. Apps as the realization of digital services cannot be considered separately from the rest of the vehicle software, as they access vehicle data via corresponding interfaces and these must also be adaptable in a timely manner. OEMs and TIER-1 to (n) in the development chains or networks need to reorganize their cooperation's under the new customer requests.

The issue of cyber security will also require rapid updates. As part of the Internet, the vehicle of the future will be exposed to all the attacks that are already a reality for other IT systems. European and international regulations create an impact on the software supply chain to comply with new laws, norms and standards

The ability to update quickly requires a high degree of automation of build processes and the seamless integration of all actors contributing software. Currently, however, a strong heterogeneity in the tools used and the procedures is a reality.

Basically, these trends are not new. They are more or less the same as they were 20 years ago [Br06]. In fact, people have gained much more experience with ubiquitous computing by carrying around smartphones and wearing smart watches. At the same time, we have a much deeper understanding of the amount of data produced by an intelligent system like a car. But this does not change the important issues, which are still complexity, development processes, software quality and reliability.

4 Consequences of these Trends

In recent decades, a highly differentiated structure of the automotive value chain has been established, which aggregates parts into components across several OEMs, TIERs and engineering suppliers. For software functions, the TIERs are perceived less strongly, as individual third-party software libraries that become part of a software function are often not considered individually. The detailed consideration of the entire software bill of material has only become established in recent years. Software functions are often created in a small group of companies in close cooperation with the OEM and not in a classic supply chain.

This approach, which has become established over time, is opposed by younger OEMs in particular, who aim to create a significant proportion of the software functions in-house. In terms of cooperation, they are usually only dependent on one SoC supplier and, if necessary, support in the realisation of components of the E/E architecture.

These two approaches are currently also reflected in the political objectives - more or less protectionism in China and the USA, while Europe, Japan and South Korea continue to orientate themselves towards open markets and international trade agreements.

With regard to the needs of the automotive sector to work more data-driven and to bring software functions into the vehicle more quickly, the two different approaches have a significant impact. The collection, processing and exchange of data are much easier to implement within an organisation or a smaller group of cooperating companies, both from a technical, organisational and regulatory perspective. For the flexible exchange of data in more complex cooperation structures, however, at least standardised interfaces are necessary. The concepts of open source software and open data spaces are presumably also necessary for efficient implementation with regard to organisational and regulatory aspects. Standardized and highly automated build chains are necessary for the rapid deployment of software functions and updates in the vehicle, for which the same arguments apply as for the previously discussed data exchange.

A change from the cooperative model to the old, more OEM-oriented model would only be possible very slowly for established software developers, as both models would have to be maintained in parallel over a very long period of time for the large electronics architectures in the field.

5 Ways Forward

Considering these trends and their consequences, how should the automotive sector adjust? And while the answers, as usual, depend on the context, the move to data and the software utilizing it suggests a few actions that are true for every company, especially small-to-medium enterprises (SMEs) who often do not yet have extensive experience with data-driven, software-centric products and business models. We describe these changes along three axes: the business model, the product life-cycle and the software development process. All three need to put data in their centre and ask what data are available, what data should be available, how is this data being validated, how can it be used and, finally, what value can be derived from this data in combination with data from other sources. While German-based OEMs and Tier-1s are likely to find ways to improve internally and adapt their supply chains, smaller suppliers – unless they are already software development experts –have a great need for better training in software engineering, modern software development methods and a continuous flow of information regarding new regulations, standards, software techniques and data usage.

5.1 Changes to Business Models

OEMs must embrace agile business models to remain competitive. The traditional revenue model centred around unit sales is evolving into one that includes software services, data monetization, and subscription-based offerings. Real-time data from connected vehicles offers OEMs the chance to craft personalized user experiences, creating new revenue streams. This shift will allow OEMs to form closer relationships with their customers, fostering brand loyalty and unlocking innovative service opportunities.

As the debate around access to vehicle data intensifies, TIER suppliers face the uncertainty of whether OEMs will act as gatekeepers or if an open-data ecosystem will prevail. This ambiguity affects not only software architecture decisions but also the feasibility of services offered by suppliers. Yet suppliers have the opportunity to position themselves as leaders in data analytics and utilization. By offering advanced analytics they can deliver value-added insights and services that enhance OEMs' and consumers' experiences.

5.2 Impact on the Product Life-Cycle

Vehicle life-cycles are becoming more iterative and dynamic as software updates and digital services extend product lifespans. This shift necessitates ongoing enhancement and lifecycle management strategies. By embracing software as a continual service, OEMs can engage customers with regular updates, creating sustained value beyond the point of sale. The opportunity here lies in turning vehicles into ever-evolving products that adapt to customer needs. As connected cars start integrating into a broader mobility environment, the collaboration with tech companies in the field of data ecosystems also opens new opportunities for OEMs and suppliers to provide greater value to end-users by additional and new business fields, especially by using collected data. For example, Microsoft offers automotive OEMs the framework and infrastructure to develop their own custom autonomous development tools. Providing non-differentiated tools and technology that can give OEMs and TIER-1 greater efficiency enables a continuous feedback loop to create continuously improving products.

5.3 Adaption of the Automotive Software Development Process

As research institutions with a focus on processes, methods and tools in the field of automotive software engineering, the methodical approach to software business processes is a key aspect of our research that will also be addressed here.

As vehicles become more software-centric, the traditional V-model may need adaptation. Integrating agile methodologies can help address the dynamic requirements, rapid iterations and more complex integration of automotive software development. This hybrid approach can facilitate faster feedback loops, enabling more responsive and efficient development processes. Consequently, the V-model's evolution offers the opportunity to enhance collaborative development practices across the industry. Especially the deep integration of agile methods such as iterative development cycles, continuous testing and cross-functional teams, will allow companies to foster innovation and quickly react to customer demands.

To overcome the limitations of external and internal collaboration and to avoid insular or missing processes and data streams, which can't be easily shared between development partners and operation phases in the product lifecycles, well-known and established product development methods from other sectors are taken over to the automotive domain. Especially process parameters, features and systems – so-called special characteristics according to VDA (German Association of the Automotive Industry) – which have significant impacts on safety, compliance with regulations, fit, function, performance or further processing of the product have to be identified and managed during the lifecycle. Systems Engineering (SE) and Model-based Systems Engineering (MBSE) according to ISO 15288 and ISO 12207 built methodical collaboration frameworks, based on technical, organisational and procedural recommendations and templates offering operational excellence in the development. SPICE (ISO 15504) – with its automotive derivate ASPICE – helps to improve relevant processes for the development and operation of technical hard- and software systems.

OEMs and TIER-1 suppliers are in the process of implementing further developed product development processes (PDPs) that better adapt the different development processes of electronic hard- and software as well as to develop, offer, and deploy customer-exciting features, both from a technical, temporal and organizational perspective. Current vehicle-development cycles show mainly hardware related facelift or model cycles. Innovations in features and software need more flexible and faster options to retrofit or upgrade solutions for existing fleets (DevOps). Over-the-air (OTA) updates are a promising method for connected car monetization, as Tesla in the German automotive landscape has successively demonstrated. New autonomous vehicle regulations like UNECE R156 being enforced in mid-2024 will help accelerate commercialization of this OTA technology.

Former organizational setups around electronic architectures are mainly structured according to the development organisations of hardware subsystems like chassis, body, and interior components. The effect of Conway's law is particularly evident in the electronic architectures of recent years [Co68]. OEMs and suppliers intend or are changing to different organisational setups for vehicle-related electronic topics. Companies show a growing adoption of agile and iterative principles, particularly in software development (SCRUM, DevOps), to increase flexibility and speed as well as increasingly established co-located, cross-domain teams to accelerate development times and improve internal collaboration. These changes require not only proficiency in traditional automotive systems but also advanced skills in virtualization, middleware, and multi-level software abstraction. Balancing safety and performance in highly integrated systems requires collaboration across software, hardware, and IT domains. This means that developers need to be able to work in interdisciplinary teams.

Finally, automating the testing of complex automotive software systems is paramount. The adoption of continuous integration/continuous deployment (CI/CD) pipelines, alongside advanced simulation tools, will enhance the efficiency of development cycles. These automation technologies, coupled with robust validation frameworks, present opportunities to improve product quality and accelerate time-to-market while ensuring safety and reliability. It is important to include data into the testing. Modern machine learning systems are no longer “finished” and then deployed once and forever. These systems constantly ingest data and adjust their behaviour. This means that such systems need to be tested at runtime to detect deviations from their expected behaviour.

6 Outlook

All these simultaneous changes mean that OEMs and TIER suppliers need to react quickly. An industry that used to think in decades, needs to adjust to a world in which OpenAI changed the expectation of what AI can look like and do almost overnight. This requires persistence and a shared view of the future. Industry, academia, and state funding for research need to work together and work towards such a shared data-driven and software-centric automobile. Not only on a German level, but now more so than ever with a broader view on manufactures that share an open view on collaboration of OEMs and geographically distributed suppliers. Projects such as TASTE, with the aim of helping SME suppliers to reach the level of capability to write software that can handle these new requirements are a vital step in ensuing the perseverance of the vibrant and diverse set of companies that made car accessible to a wide public. The hard engineering skills are all still there, but they need to be augmented with the skills to handle software integration and data management. Here Europe can lead the way by enforcing open standards and open source software solutions.

These recommendations are quite close to what was discussed 20 years ago [Br06]. At that time, with the increasing adoption of Automotive ASPICE for process maturity and AUTOSAR for a more unified and comprehensive architecture, people were excited about the future of automotive software. Today we feel again that our methods and processes do not meet the expectations to compete in the global market. It is time to create a new vision for automotive software.

Acknowledgement

We would like to thank the many industry experts who have shared their views and opinions on the current changes in the way automotive software is created with the TASTE team, making this publication possible.

The Transformation-Hub Automotive Software Engineering (TASTE) is funded by the German Federal Ministry for Economic Affairs and Climate Action under grant number 16THB0009.

References

[Bä24] Bähr, C.; Kempermann, H.; Kuttler, F.; Brauckhoff, E.: Begleitforschung des Expertenkreises Transformation der Automobilwirtschaft zum Thema Resilienz der automobilen Wertschöpfungs- und Liefernetzwerke Thema „Innovationen“, https://expertenkreis-automobilwirtschaft.de/home/veroeffentlichungen, [Online; accessed 24-Oct-2024], 2024.

[Bl24] Block, L.; Herrmann, F.; Wizl, J.; Borrmann, D.: Deutschland zum Innovationsstandort für das automatisierte und vernetzte Fahren machen./, 2024.

[Br06] Broy, M.: Challenges in automotive software engineering. In: Proceedings of the 28th International Conference on Software Engineering. ICSE ’06, Association for Computing Machinery, New York, NY, USA, S. 33–42, 2006, ISBN: 1595933751.

[Bu24] Burstedde, A.; Risius, P.; Tiedemann, J.; Werner, D.: Weiterbildungsbedarfe der Automobilbranche in der Transformation, https://expertenkreis-automobilwirtschaft.de/home/veroeffentlichungen, [Online; accessed 24-Oct-2024], 2024.

[CA24] CATENA-X: CATENA-X: Das Catena-X-Datenökosystem, https://catena-x.net/de/, [Online; accessed 24-Oct-2024], 2024.

[Co68] Conway, M. E.: How do committees invent. Datamation 14/4, S. 28–31, 1968.

[En24] Entrapeer: Personalization in Automotive Industry, https://innovate.entrapeer.com/theme/personalization-in-automotive-industry, [Online; accessed 24-Oct-2024], 2024.

[Ha17] Haghighatkhah, A.; Banijamali, A.; Pakanen, O.-P.; Oivo, M.; Kuvaja, P.: Automotive software engineering: A systematic mapping study. Journal of Systems and Software 128/, S. 25–55, 2017.

[Ji24] Jiang, Shugang: Vehicle E/E Architecture and Key Technologies Enabling Software-Defined Vehicle. In: WCX SAE World Congress Experience. SAE International, 2024.

[No23] Norouzi, A.; Heidarifar, H.; Borhan, H.; Shahbakhti, M.; Koch, C. R.: Integrating machine learning and model predictive control for automotive applications: A review and future directions. Engineering Applications of Artificial Intelligence 120/, S. 105878, 2023.

[Ra21] Rahim, M. A.; Rahman, M. A.; Rahman, M. M.; Asyhari, A. T.; Bhuiyan, M. Z. A.; Ramasamy, D.: Evolution of IoT-enabled connectivity and applications in automotive industry: A review. Vehicular Communications 27/, S. 100285, 2021.

[SZ10] Schäuffele, J.; Zurawka, T.: Automotive software engineering. Springer, 2010.

[TA24] TASTE: TASTE Transformations-Hub, https://transformations-hub-taste.de/, [Online; accessed 24-Oct-2024], 2024.

[WZD15] Wu, G.; Zhang, X.; Dong, Z.: Powertrain architectures of electrified vehicles: Review, classification and comparison. Journal of the Franklin Institute 352/2, S. 425–448, 2015.

[Yu20] Yurtsever, E.; Lambert, J.; Carballo, A.; Takeda, K.: A survey of autonomous driving: Common practices and emerging technologies. IEEE access 8/, S. 58443–58469, 2020.

[ZSL22] Zhao, Fuquan; Song, Haokun; Liu, Zongwei: Identification and Analysis of Key Technical Elements and Prospects for Software-Defined Vehicles. In: SAE 2021 Intelligent and Connected Vehicles Symposium Part II. SAE International, 2022.