35th International Electric Vehicle Symposium and Exhibition (EVS35)
Oslo, Norway, June 11-15, 2022

An Approach to Implement Service-Oriented Software Architecture on Energy Management Systems of Autonomous Electric Vehicles

M.Sc. Dorsa Mohammad Zaheri[1], M.Sc. Jonas Maier[1], M.Eng. Matthias Hisung[1], Prof. Dr.-Ing. Hans-Christian Reuss[1]

[1]*University of Stuttgart, Institute of Automotive Engineering, Pfaffenwaldring 12, 70569 Stuttgart, Germany*
dorsa.mohammad-zaheri@ifs.uni-stuttgart.de

Summary

The design complexity of automotive Electrical/Electronic (E/E) architectures is constantly aggravating by the increasing number of functions and electronic systems. The automotive industry is adapting Service-Oriented Architecture (SOA) concepts within the automotive domain to meet the challenges posed by this growth. Implementing service-oriented architecture on energy management systems of battery electric vehicles is a major challenge due to the fact, that commercially available batteries do not possess an SOA interface. In this paper, we intend to tackle this challenge by proposing an architecture that enables adapting an electrical energy management system as part of a SOA system in an automated vehicle. The proposed approach has been implemented and approved in the UNICARagil project.

Keywords: autonomous vehicle, BMS (Battery Management System), BEV (battery electric vehicle), communication, infrastructure

1 Introduction

In upcoming years, the automotive E/E architectures will face extreme complexity due to advanced features and innovations in future vehicles. This has been largely driven by the development of technologies such as Advanced Driver Assistance Systems (ADAS), connectivity, electrification, and smart mobility. Modern architectures are expected to evolve towards enabling autonomous driving. The complexity of these modern architectures makes them highly challenging to design and validate using traditional approaches that are being applied by many manufacturers [1]. Moreover, current function-oriented design-time integrated architectures cannot fully support new technologies because of their lack of flexibility and reusability [2]. Hence, a real paradigm shift in the area of software and E/E architecture is expected [3]. As a result, many original equipment manufacturers (OEMs) and suppliers have been advocating for centralization as the E/E architectural trend of the future to reduce complexity and costs in the long run [4]. On the other hand, with the ever-increasing amount of data being transferred between various Electronic Control Units (ECUs) and sensors, the demand for networking solutions

with higher bandwidth and transmission speed is rising rapidly. In this regard, vehicle manufacturers are relying on Ethernet technology for communication within the vehicle and its environment [5].

Consequently, the automotive E/E architecture is evolving from distributed architectures with function-specific ECUs towards a centralized software-oriented design to provide more flexibility in adding new features, facilitate reuse in hardware and software, and meet the demand for higher bandwidth.

Currently, Service-Oriented Architecture (SOA) is drawing attention within the automotive domain to tackle these challenges. SOA is an architectural pattern, which provides flexible integration of components at runtime. SOA encapsulates different functions into services and allows any service providing data to be connected to any other service requiring this data at runtime [6]. This approach provides updateability, modularity, and extensibility to the system. Although bringing Ethernet into automotive applications paved the way for the use of SOA in the automotive industry, there are still some challenges such as security and reliability that need to be addressed [6, 7]. Additionally, in order to integrate all automotive systems and units into SOA, an adaptive environment that provides SOA functionality while being compatible with the existing automotive embedded systems is required.

The latest research shows that autonomous vehicles will have to be electrified due to the environmental benefits and low operating costs [8]. The battery system is considered the fundamental part of any electric vehicle, which provides the required electricity to power the vehicle motor and accessories. A battery of an electric vehicle is often composed of battery cells and battery management systems (BMS) to manage the task of monitoring and controlling all aspects of batteries [9].

The BMS also calculates data such as state of health (SoH) and state of charge (SoC) and provides them to the rest of the vehicle using the available communication interface—often over the CAN bus. The lack of an accepted SOA interface in current batteries poses a major challenge for developing autonomous vehicles. This work seeks to address this issue by introducing a practical approach that is already used in the UNICARagil project. The proposed architecture consists of a gateway and a Raspberry Pi and can be adapted to different projects. The remainder of this paper is structured as follows: Section II provides information about the automotive SOA concept and the Electric Power Supply (EPS) used in the UNICARagil project. Our proposed approach is presented in section III. Finally, section IV summarizes our work and outlines possible research topics for future work.

2 SOA concept and EPS in UNICARagil

The E/E architecture of a vehicle consists of ECUs, software applications, and network communication. Today's domain-centric E/E architectures may have up to 150 individual ECUs which are integrated at design time to perform the vehicle functions [10]. Since the software and hardware in such architectures are closely coupled, the software modification and updating the functionality of a system during its lifetime is unlikely to be reached. Therefore, these E/E architectures cannot cover the requirements of the mentioned technology trends in an efficient manner. As a result, the long-established signal-oriented paradigm will be replaced by service-oriented architectures to provide upgradability and updateability [3]. In the next sections, the SOA concept used in the UNICARagil project as well as the EPS system of UNICARagil are introduced.

2.1 SOA in UNICARagil

Over the last few years, a lot of research has been conducted on bringing SOA into the automotive domain, describing its benefits and challenges. However, few studies have focused on SOA implementation approaches [7, 11]. In [12], the authors present αSOA approach and investigate the capability of applying their approach to automotive software. A dynamic service-oriented architecture for automated vehicles is introduced by Kampmann et al. in [2]. The proposed approach was later used in the UNICARagil project [13].

The UNICARagil project aimed to investigate a disruptive modular architecture for agile, automated electrified vehicles [14]. Within this project, four prototypes of SAE level 4 driverless vehicles are developed based on a so-called "Automotive Service-Oriented Architecture (ASOA)". Each software function is defined as a service

in ASOA. The interfaces between services in ASOA are established at design time and are realized in the forms of requirements and guarantees [13]. Service's guarantees are the output functions that are provided by a certain service for other services. On the other hand, service's requirements are defined as input functions that a service requires to operate. Thus, for each service's requirement, an appropriate guarantee must be available so that the service can operate.

Each requirement or guarantee of a service contains three types of data, which describe the data being sent or received [13]:

- Data: the data structure contains the user data that is sent by a guarantee of this type or received by an associated requirement.
- Quality data: it defines the quality (e.g. accuracy) of the sent data.
- Parameter data: This structure should allow a requirement to decide whether a guarantee provides usable information.

Services in ASOA are integrated at runtime and can be executed and updated independently. A central controller, a so-called orchestrator, is responsible for initiating and controlling the interaction between services at runtime [13]. This addresses some of the issues of previous software architectures. For one thing, updating service in ASOA does not always result in software adoption for all consumers of that service [5].

The lifecycle of ASOA services has three states. In the "running" state, services perform their functions and interact with each other. The "prepared" state is used to make a service ready for its operation while in the "stopped" state, a service enters idle state.

The ASOA interface is implemented in C++. Each interface definition file contains a structure that defines the data type as well as a function describing data transfer in the physical bus system. This function determines how the data is serialized and transmitted using Real-Time Publish Subscribe (RTPS) protocol and how it is deserialized on the receiving side [13].

2.2 EPS in UNICARagil

The Energy supply of UNICARagil comprises four 48V Lithium-ion batteries. To ensure reliability and availability of the EPS system, the batteries are arranged in a redundant ring topology. Each battery is equipped with a battery module and an independent BMS that continuously sends SoC, SoH, state of function (SoF), the temperature of traction batteries, voltage, current, and status of the vehicle (charging or driving) over CAN bus to the vehicle [15]. According to the ASOA concept, this information should be provided to other services by using a guarantee. In Fig.1, the layout of the power supply as well as the battery management service in the UNICARagil are depicted.

As shown in Figure 1, the information of the batteries will be available as a service's guarantee for the rest of the vehicle. Other services use this information to perform their task. For example, a service that is responsible for trajectory control in the vehicle reads the SOC value and checks if it is not too low, before sending manipulated variables to the actuators.

The battery management service also provides information about the clamp 15 or KL15 status. It checks whether the ignition switch is on or off and sets a binary value to be either 0 or 1. The central controller in the vehicle (i.e. orchestrator) receives this service and checks KL 15 status. If the status changes from 0 to 1, the orchestrator shows "KL15 rising edge detected" message and start the wakeup process by sending Wake-on-LAN (WoL) messages to some of the ECUs in the vehicle.

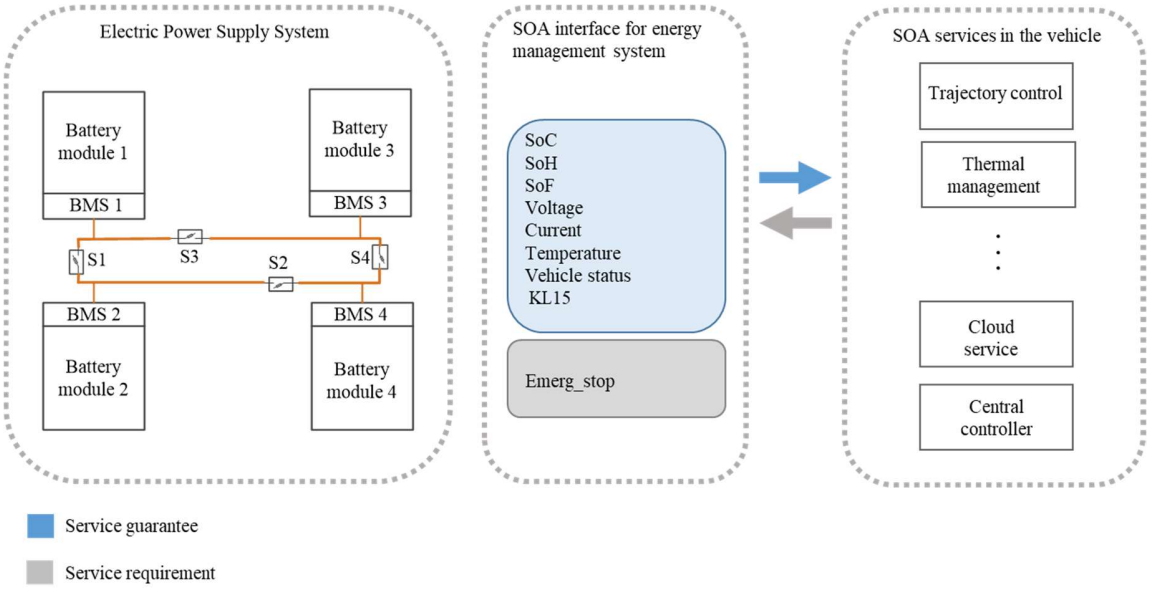

Figure 1: Layout of power supply inspired by [15] and battery management services in the UNICARagil vehicles

Battery management service consists of a service's requirement, which enables an emergency stop function in the vehicle to avoid a hazardous situation. In case of malfunction or danger, the orchestrator activates the emergency stop service. After receiving this service, a message is sent to the Energy Management System (EMS), which causes the batteries to be disconnected and all systems to shut down.

2.3 EMS in UNICARagil

The energy management system of UNICARagil manages and controls the operation of the batteries and monitors information and status of the batteries' BMS, by means of an industrial Programmable Logic Controller (PLC; SIMATIC S7-1200) and a CAN communication module. The EMS communicates directly with the BMS of the batteries through a CAN network.

Each BMS individually calculates electrical quantities such as cell voltage, cell current, SoC, and SoH as well as the temperature of its battery pack. Since only one value is defined for each of these parameters in ASOA, and because in UNICARagil four separate batteries are used, the EMS should perform a data calculation to provide a single value for each parameter. As a result, in order to satisfy safety concerns, EMS selects the lowest values of SoC and voltage, as well as the maximum values of current and temperature to be used in ASOA. Furthermore, as SoH is not a safety-critical parameter, the average value of all four SoHs is chosen by EMS [15].

The EMS is illustrated in Fig. 2. The CAN Bus not only transmits the battery information to the EMS, but also receives the control messages (e.g. emergency stop) from the EMS. As seen in Fig. 2, we use a Raspberry Pi along with a PLC to provide an interface to ASOA. This approach will be discussed in detail in the next section.

3 Proposed architecture

In this section, an approach to implementing SOA on the energy management system of a battery electric vehicle is discussed in depth. As already mentioned, the energy management system cannot be adapted as part of the SOA system because commercially available batteries do not have an SOA interface. Hence, we used a CAN-Ethernet gateway together with a Raspberry Pi to act as an interface between the data provided by BMS and the SOA in the vehicle network. The proposed architecture is shown in Fig. 3.

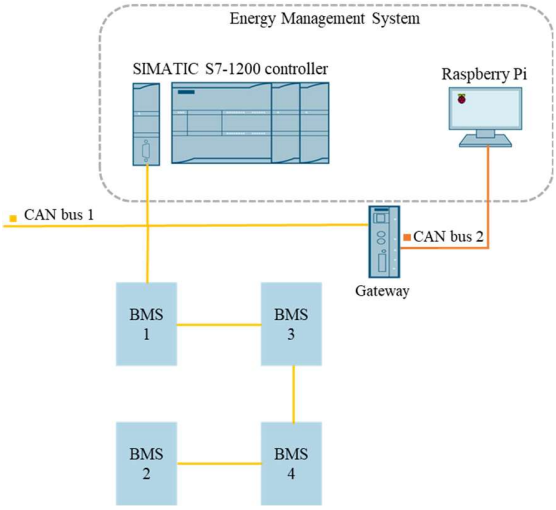

Figure 2: EMS communication network in the UNICARagil

In the first step of the communication process, the battery information is sent from BMS via CAN bus to the EMS. The PLC receives all the BMS messages and after applying data processing steps generates new messages containing values of SoC, SoH, SoF, temperature, total voltage, total current, and the status of the vehicle. It also sends a control message to the batteries through the CAN bus in order to turn the batteries' relays, depending on the situation, on or off. Then, the gateway receives messages from PLC and maps them to Ethernet and/or other available CAN buses in the network. The gateway is also used for monitoring and diagnostic purposes. We have created a CAN database file and uploaded it to the gateway in order to decode raw CAN bus data to physical values. Furthermore, we have configured a data logger on the gateway to record and save the entire CAN bus data. The recorded data can be used for debugging and verification in test vehicles. Table 1 indicates an example of the messages that are sent from PLC to the gateway as well as the extracted values that are monitored in the gateway.

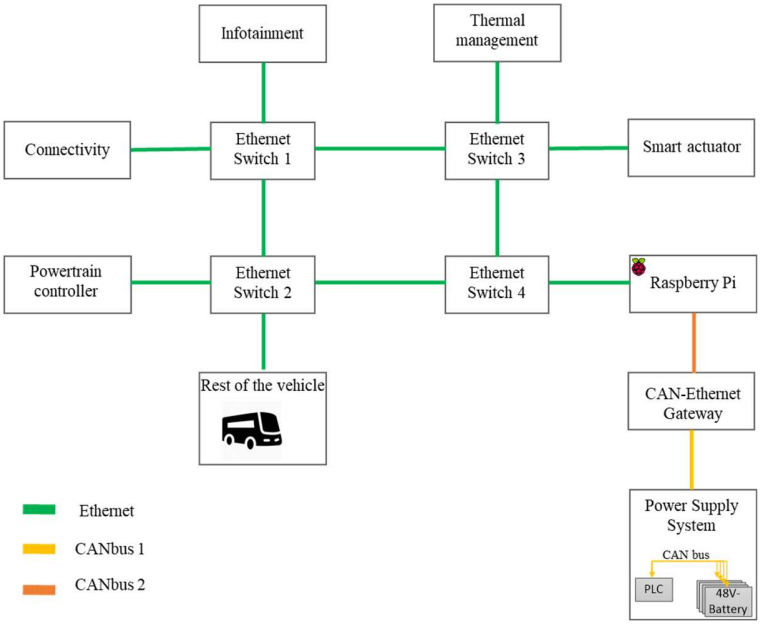

Figure 3: Proposed E/E architecture to facilitate SOA implementation on EMS

The battery information can be received in Raspberry Pi through either CAN bus or Ethernet. Since in UNICARagil vehicles the Raspberry Pi is connected to a different CAN network, we used the gateway to map messages between both available CAN networks. As Raspberry Pi SoC (System on a Chip) does not include a CAN controller, a CAN bus shield should be mounted into Raspberry Pi. In this project, we used Raspberry Pi 4 with Ubuntu OS and PiCAN 3 to provide CAN bus capability. In addition, we set up a SocketCAN interface to enable sending and receiving CAN packets through the Linux IP stack. The SocketCAN is an implementation of the CAN protocol that provides a socket interface to user applications and builds on the Linux network layer [16]. According to the predefined services in ASOA, the battery information is decoded in the Raspberry Pi and integrated into the corresponding service. We used the *can_read()* function to read the battery information and transfer it to the corresponding service's guarantee. All software functions and ASOA integration are implemented in Raspberry Pi using C++ programming language.

Table1: Extract battery information from CAN messages sent by PLC

Message 1								
CAN ID	Value							
0x85	02	64	00	17	00	00	00	00
Battery information								
	Status	SoF	SoH	SoC				
	Driving	100 %	0 %	23 %	-	-	-	-

Message 2							
CAN ID	Value						
0x86	00	00	00	41	0A	F0	01	CD
Battery information							
				Temperature	Current	Voltage	
	-	-	-	25°C	0 A	46,1 V	

As discussed in section 2.2, the battery management service should also provide information about clamp 15. Thereby, we connected a GPIO pin of the Raspberry Pi to the clamp 15 PDU (Power Distribution Unit) using a relay. In the KL 15 service, we wrote a function that continuously checks the GPIO pin and sets the KL 15 value to 1 when the ignition switch is turned on.

To add the emergency stop information to the battery management service, we used the *can_send()* function to send a CAN packet to the PLC, which leads to disconnecting battery relays and shutting down the vehicle. This information is defined as a service's requirement and is activated when the orchestrator enables its corresponding service's guarantee.

After creating service parameters and integrating them into ASOA, the Raspberry Pi publishes the developed service via Ethernet interface to switches and the vehicle network. As a result, the service that requires this data can receive it, which means the services can interact with one another.

It is important to mention that the battery management service should be available as soon as the vehicle is powered on. Therefore, we have brought up the CAN interface as well as the ASOA service in the boot process. Once the vehicle is turned on, the orchestrator establishes and controls the connection between services based on a predefined rule.

The proposed approach has been implemented and successfully tested in all four prototypes of the UNICARagil project. Our findings proved that the SOA concept may be successfully applied to the energy management systems of electric vehicles. This solution can assist E/E architecture developers and engineers to reach a secure and cost-effective solution to address the problem of implementing SOA on energy management systems of automated vehicles.

4 Conclusion

In recent years, new advanced functions and the increasing number of ECUs have led to ever greater complexity of E/E architectures. This dramatic growth in complexity severely challenges the existing architectures and necessitates an E/E architectural transformation. As a result, future E/E architectures have to become more centralized with the goal of satisfying the need for flexibility, energy efficiency, and updatability during the E/E systems lifecycle. Achieving the requirements of future E/E architectures calls for new dynamic concepts. That means vehicle software architecture must also alter from current function-oriented architecture to service-oriented design. Service-oriented architectures allow replacing a component or upgrading a vehicle function without having an effect on other components while providing the required network bandwidth. Implementing SOA on the energy management systems of battery electric vehicles is a major challenge, due to the fact that available batteries do not support SOA interface.

In this paper, we proposed a cost-effective approach to deal with this problem. An Automotive Service-Oriented Architecture (ASOA) is already developed and utilized in the UNICARagil project [13]. Based on ASOA, a service for energy management system is designed that provides battery information to other software functions in the vehicle. The suggested architecture consists of a Raspberry Pi and an embedded platform to transfer CAN packets to Ethernet and other CAN networks. The developed service is implemented in C++ and provides the information about SoC, SoH, SoF, temperature, current, voltage as well as vehicle and clamp 15 status. This approach is used in the UNICARagil project. Our experimental results show the successful integration of EMS into SOA while maintaining update and upgrade capability. In this paper, we did not consider the quality of the data being sent. Therefore, the next step would be to calculate the accuracy of the data and integrate it into ASOA.

Acknowledgments

This research is accomplished within the project “UNICARagil” (FKZ16EMO0289). We acknowledge financial support for the project by the Federal Ministry of Education and Research of Germany (BMBF).

References

[1] V. Cebotari and S. Kugele, *On the Nature of Automotive Service Architectures*, IEEE International Conference on Software Architecture Companion (ICSA-C), 2019, pp. 53-60, doi: 10.1109/ICSA-C.2019.00017

[2] A. Kampmann, B. Alrifaee, M. Kohout, A. Wüstenberg, T. Woopen, M. Nolte, L. Eckstein and S. Kowalewski, *A dynamic service-oriented software architecture for highly automated vehicles*, IEEE Intelligent Transportation Systems Conference (ITSC), 2019, pp. 2101–2108, doi: 10.1109/ITSC.2019.8916841

[3] M. Rumez, D. Grimm, R. Kriesten and E. Sax, *An Overview of Automotive Service-Oriented Architectures and Implications for Security Countermeasures*, IEEE Access, vol. 8, pp. 221852-221870, 2020

[4] V. Bandur, G. Selim, V. Pantelic and M. Lawford, *Making the Case for Centralized Automotive E/E Architectures*, IEEE Transactions on Vehicular Technology, pp. 1230-1245, 2021, doi: 10.1109/TVT.2021.3054934.

[5] M. Helmling, *Service-oriented Architectures and Ethernet in Vehicles: Towards Data Centers on Wheel with Model-based Methods*, Electronic automotive magazine, Special Issue "Automotive Ethernet", 2017

[6] Gopu, G. L., K. V. Kavitha, and J. Joy, *Service Oriented Architecture based connectivity of automotive ECUs*, International Conference on Circuit, Power and Computing Technologies (ICCPCT), 2016, doi: 10.1109/ICCPCT.2016.7530358

[7] N. Kukulicic, D. Samardzic, *Automotive Service-Oriented Architectures: A Systematic Mapping Study,* Master Thesis, Mälardalen University, Sweden, 2022

[8] K. Han, T. Nguyen, K. Nam, *Battery Energy Management of Autonomous Electric Vehicles Using Computationally Inexpensive Model Predictive Control*, Electronics, 2020, https://doi.org/10.3390/electronics9081277

[9] P. Aruna and P. V. Vasan, *Review on Energy Management System of Electric Vehicles*, 2nd International Conference on Power and Embedded Drive Control (ICPEDC), 2019, doi: 10.1109/ICPEDC47771.2019.9036689

[10] A. Vetter, P. Obergfell, H. Guissouma, D. Grimm, M. Rumez and E. Sax, *Development Processes in Automotive Service-oriented Architectures*, 9th Mediterranean Conference on Embedded Computing (MECO), 2020, doi: 10.1109/MECO49872.2020.9134175

[11] N. Niknejad, I. Ghani, A. Hussin, S. Jeong, *A Systematic Literature Review Towards Service Oriented Architecture Adoption from 2009 to 2015*, International Journal of Advanced Smart Convergence, 2018

[12] S. Kugele, P. Obergfell, M. Broy, O. Creighton, M. Traub and W. Hopfensitz, *On Service-Orientation for Automotive Software,* 2017 IEEE International Conference on Software Architecture (ICSA), 2017, pp. 193-202, doi: 10.1109/ICSA.2017.20

[13] A. Mokhtarian, A. Kampmann, B. Alrifaee and S. Kowalewski, *The Dynamic Service-oriented Software Architecture for the UNICARagil project,* 29th Aachen Colloquium Sustainable Mobility, 2020

[14] T. Woopen, B. Lampe et al., *UNICARagil - Disruptive Modular Architectures for Agile, Automated Vehicle Concepts*, 27th Aachen Colloquium, Aachen, Germany, Oct. 2018

[15] M. Goth, M. Hisung, D. Keilhoff, H-C. Reuss, *A Novel Electrical Energy Management System in Automated Battery Electric Vehicles*, 33rd Electric Vehicle Symposium (EVS33) Portland, Oregon, 2020

[16] *SocketCAN - Controller Area Network*, https://www.kernel.org/doc/html/latest/networking/can.html

Authors

Dorsa Mohammad Zaheri received the B.Sc. and M.Sc. degree in electrical engineering from the Shahid Rajaee University, Tehran, Iran, in 2014 and 2017, respectively. She is currently working toward the Ph.D. degree at the University of Stuttgart, Stuttgart, Germany. Her research interests lie in the area of E/E architecture design, optimization, functional safety, and autonomous vehicle.

Jonas Maier received his B. Sc. in technology management and M. Sc. in mechanical engineering from the University of Stuttgart, Stuttgart, Germany in 2017 and 2020, respectively. He is currently working towards a Ph. D. degree at the University of Stuttgart, Stuttgart, Germany. His research interests include vehicle power supply systems with a focus on design, energy management, functional safety, and packaging aspects.

Matthias Hisung is a research assistant at the chair in Automotive Mechatronics at the University of Stuttgart. He has a Master's degree in electrical and information technology from University of Applied Sciences Aschaffenburg. His research interests include inductive charging and automated driving.

Prof. Hans - Christian Reuss studied electrical engineering at the Technical University of Berlin and received his doctorate at the Institute of Electronics. Since April 2004 he has headed the Chair of Automotive Mechatronics at the University of Stuttgart and is a member of the Executive Board of the Research Institute for Automotive Engineering and Vehicle Engines Stuttgart (FKFS).