Citation Information:
32nd Aachen Colloquium Sustainable Mobility 2023. Aachen, October 09th to 11th, 2023.
Aachen: Aachener Kolloquium Fahrzeug- und Motorentechnik GbR. Paper No. 047.
ISBN: 978-3-00-076758-6.

AUTOtech.*agil:* Architecture and Technologies for Orchestrating Automotive Agility

Raphael **van Kempen**; Bastian **Lampe**; Marc **Leuffen**; Lena **Wirtz**; Fabian **Thomsen**; Gergely **Bilkei-Gorzo**; Jean-Pierre **Busch**; Ida **Feger**; Christian **Geller**; Christian **Kehl**; Olaf **Uszynski**; Lotte **Wagner-Douglas**; Lukas **Zanger**; Univ.-Prof. Dr.-Ing. Lutz **Eckstein**
Institute for Automotive Engineering (ika), RWTH Aachen University, Aachen, Germany

David **Klüner**; Julius **Beerwerth**; Dr.-Ing. Bassam **Alrifaee**, Prof. Dr.-Ing. Stefan **Kowalewski**
Chair of Computer Science 11 (Embedded Software), RWTH Aachen University, Aachen, Germany

Dr. Marco **Konersmann**; Felix **Steinfurth**; Prof. Dr. Bernhard **Rumpe**
Chair of Computer Science 3 (Software Engineering), RWTH Aachen University, Aachen, Germany

Max **Hartmann**; Dr.-Ing. Norbert **Siepenkötter** ; Univ.- Prof. Dr.-Ing. Dieter **Moormann**
Institute of Flight System Dynamics, RWTH Aachen University, Aachen, Germany

Dr. Boris **Böhlen**; Claudia **Hannig**
Daten- und Systemtechnik GmbH, Aachen, Germany

Esther **Hekele**
Vires Simulationstechnologie GmbH, Bad Aibling, Germany

Dr. Heinrich **Gotzig**; Paul-David **Rostocki**; Deepak-Kumar **Gautam**
Valeo Schalter und Sensoren GmbH, Bietigheim-Bissingen, Germany

Richard **Schubert**; Niklas **Braun**; Prof. Dr.-Ing. Markus **Maurer**
Institute for Control Engineering, TU Braunschweig, Braunschweig, Germany

Kai-Björn **Gemlau**; Sebastian **Abel**; Prof. Dr.-Ing. Rolf **Ernst**
Institute of Computer and Networking Engineering, TU Braunschweig, Braunschweig, Germany

Dr.-Ing. Melina **Lutwitzi**; Lorenz **Bayerlein**; Moritz **Berghöfer**; Alexander **Blödel**; Björn **Klamann**; Anton **Kuznietsov**; Prof. Dr.-Ing. Steven **Peters**
Institute of Automotive Engineering (FZD), Technical University of Darmstadt, Darmstadt, Germany

Dr.-Ing. Stefan **Leinen**; Jakob **Bahle**
Physical and Satellite Geodesy (PSGD), Technical University of Darmstadt, Darmstadt, Germany

Lars **Ullrich**; Prof. Dr.-Ing. Knut **Graichen**
Chair of Automatic Control, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany

Timo **Woopen**
Thinking Cars GmbH, Ettlingen, Germany

Dominik **Spychalski**; Prof. Dr. Christoph **Krauß**
INCYDE industrial cyber defense GmbH, Frankfurt am Main, Germany

Mohamad **Alayan**; Dr. Jens **Giesler**; Dr. Martin **Lilienthal**; Thomas **Schulik**
ZF Friedrichshafen AG, Friedrichshafen, Germany

Dr. Martin **Lauer**; Dr. Carlos **Fernandez**; Dr. Eduardo **Molinos**; Nick **Le Large**; Nils **Rack**; Marlon **Steiner**; Kaiwen **Wang**; Prof. Dr.-Ing. Christoph **Stiller**
Institute of Measurement and Control Systems (mrt), Karlsruhe Institute of Technology, Karlsruhe, Germany

Gideon **Arndt**; Benedikt **Schulz**; Prof. Dr.-Ing. Kai **Furmans**
Institute for Material Handling and Logistics (IFL), Karlsruhe Institute of Technology, Karlsruhe, Germany

Stephan **Rauber**
IPG Automotive GmbH, Karlsruhe, Germany

Dr. Frank **Diermeyer**; David **Brecht**; Nils **Gehrke**; Prof. Dr. Markus **Lienkamp**
Institute of Automotive Technology, Technical University of Munich, Munich, Germany

Walter **Zimmer**; Christian **Creß;** Xingcheng **Zhou;** Prof. Dr.-Ing. habil. Alois **Knoll**
Chair of Robotics, Artificial Intelligence and Real-time Systems, Technical University of Munich, Munich, Germany

Dominik **Püllen**; Prof. Dr. Stefan **Katzenbeisser**
Chair of Computer Engineering, University of Passau, Passau, Germany

Arlinda **Elmazi**; Andreas **Sailer**
Vector Informatik GmbH, Regensburg, Germany

Dr. Martin **Alfranseder**; Ralph **Mader**
Vitesco Technologies GmbH, Regensburg, Germany

Dr.-Ing. Felix **Berkel**; Dr.-Ing. Thomas **Specker**
Corporate Research, Robert Bosch GmbH, Renningen, Germany

Philip **Mayer**; Dr. Hermann **v. Hasseln**
Mercedes Benz AG, Sindelfingen, Germany

Lukas **Jung**; Marcel **Grandinetti**; Dominik **Neidhart**
iMAR Navigation GmbH, St. Ingbert, Germany

Dr. Dan **Greiner**; Dennis **Niedballa**; Dorsa **Zaheri**; Jonas **Maier**; Prof. Dr.-Ing. Hans-Christian **Reuss**
Institute of Automotive Engineering (IFS), University of Stuttgart, Stuttgart, Germany

Valentyna **Afanasenko**; Dr. Oleksandr **Solomakha**; Swapnil Sunil **Roge**; Prof. Dr.-Ing. Ingmar **Kallfass**
Institute of Robust Power Semiconductor Systems, University of Stuttgart, Stuttgart, Germany

Priv.-Doz. Dr.-Ing. Michael **Buchholz**; Robin **Dehler**; Matti **Henning**; Charlotte **Hermann**; Markus **Schön**; Prof. Dr.-Ing. Klaus **Dietmayer**
Institute of Measurement, Control and Microtechnology, Ulm University, Ulm, Germany

Contact: raphael.vankempen@ika.rwth-aachen.de

[VLL+23] R. van Kempen, B. Lampe, M. Leuffen, et al.
AUTOtech.agil : Architecture and Technologies for Orchestrating Automotive Agility.
In: 32nd Aachen Colloquium Sustainable Mobility 2023. Aachen, October 09th to 11th, 2023, DOI 10.18154/RWTH-2023-09783,
Aachener Kolloquium Fahrzeug- und Motorentechnik GbR, Aachen, Germany, Oct. 2023.

Abstract

Future mobility will be electrified, connected and automated. This opens completely new possibilities for mobility concepts that have the chance to improve not only the quality of life but also road safety for everyone. To achieve this, a transformation of the transportation system as we know it today is necessary. The UNICAR*agil* project, which ran from 2018 to 2023, has produced architectures for driverless vehicles that were demonstrated in four full-scale automated vehicle prototypes for different applications. The AUTOtech.*agil* project builds upon these results and extends the system boundaries from the vehicles to include the whole intelligent transport system (ITS) comprising, e.g., roadside units, coordinating instances and cloud backends. The consortium was extended mainly by industry partners, including OEMs and tier 1 suppliers with the goal to synchronize the concepts developed in the university-driven UNICAR*agil* project with the automotive industry. Three significant use cases of future mobility motivate the consortium to develop a vision for a Cooperative Intelligent Transport System (C-ITS), in which entities are highly connected and continually learning. The proposed software ecosystem is the foundation for the complex software engineering task that is required to realize such a system. Embedded in this ecosystem, a modular kit of robust service-oriented modules along the effect chain of vehicle automation as well as cooperative and collective functions are developed. The modules shall be deployed in a service-oriented E/E platform. In AUTOtech.*agil*, standardized interfaces and development tools for such platforms are developed. Additionally, the project focuses on continuous uncertainty consideration expressed as quality vectors. A consistent safety and security concept shall pave the way for the homologation of the researched ITS.

1. Introduction

Four full-scale driverless vehicle prototypes for different applications have successfully demonstrated the disruptive new architectures for driverless vehicle concepts developed by the UNICAR*agil* consortium at the final project demonstration in May 2023. Based on this advanced starting point, the AUTOtech.*agil* project develops open software and E/E architectures and accompanying tools and methods that shall enable the development of future intelligent transport systems (ITSs).

1.1. The UNICAR*agil* project: A multi-layer architecture for driverless vehicles

The UNICAR*agil* project was funded by the German Federal Ministry of Education and Research (BMBF) and was first introduced in [1] at the 2018 Aachen Colloquium. In the project, eight leading Universities in the field of automated driving partnered with nine industrial partners to develop disruptive architectures in software and hardware to empower automated and driverless vehicle prototypes. With the research on a multi-layer architecture for driverless vehicles being the focus of the project, four vehicle prototypes were built up from scratch to integrate and showcase the architectural developments (see Fig. 1). These prototypes show the potential of the architectures and illustrate different use-cases for future development.

Fig. 1 The UNICAR*agil* research vehicles. From left to right: autoSHUTTLE for public transportation, autoTAXI for individual business rides, autoELF as a private vehicle and autoCARGO for the delivery of goods. © ika

With a disruptive modular approach, UNICAR*agil* paves new ways for the development of agile automated and electrified urban vehicles and rethinks the evolutionary development processes. For this purpose, the project was divided into five research domains, which are integrated in the prototypes that have been specially developed from scratch. The research domains further can be transferred into architectural perspectives namely geometry, mechatronics, software, automation and safety.

Besides the modular bionic E/E architecture, a key contribution of the project is the major development in the field of a service-oriented software architecture (ASOA) for automotive applications summarized in [40]. This architecture lays the foundation of future automotive software systems by enabling the underlying orchestrator to integrate services at runtime. Furthermore, these services not only transmit their (payload) data but also information on the service and data quality. This allows services to be updatable and upgradable. Since the developed electrics and electronics architecture is based on Automotive Ethernet, each service may be connected to the central cloud intelligence as well.

1.2. The AUTOtech.*agil* project: Larger consortium and extended system boundaries

The UNICAR*agil* project did not involve any OEM or tier 1 supplier due to the goal to develop a neutral blueprint for driverless vehicle architectures from scratch and without the need to adopt any legacy that is established in the automotive industry. AUTOtech.*agil* has the goal to align this rather academic approach with research and development in industry and to combine the knowledge to develop a solid basis for future ITS and therefore extended the consortium to 12 industry partners and 17 university chairs (see Fig. 2).

Fig. 2 The AUTOtech.*agil* consortium consists of 12 industry partners and 17 chairs at nine universities. The project is funded by the Federal Ministry of Education and Research (BMBF) with approx. 25 million euros and runs three years.

The consortium's vision for a future ITS including fleets of automated vehicles, non-automated vehicles and vulnerable road users (VRUs) as well as intelligent infrastructure, coordinating instances and cloud backends is outlined in Section 2.

The potential of the described ITS is motivated by three exemplary use cases that are enabled by electrification, automation and connectivity. Section 3 addresses inclusive mobility, efficient transport of goods and a Guardian Angel function for VRUs and light vehicles.

The presented use cases can be realized by different mobility providers having their own control rooms to supervise fleets of automated vehicles. Several such control rooms may operate in the ITS of one municipality and require coordination by a control center. Mobility providers can use fleet data for continuous monitoring, maintenance, and improvement of their algorithms as part of a DevOps process. Control centers may combine fleet data with data from intelligent infrastructure to create a digital twin of traffic. Some potentials of such learning and connected traffic intelligence are outlined in Section 4.

The feasibility of such a complex system substantially depends on enabling software. In Section 5, we propose a software ecosystem including a toolchain for development and testing as well as software architectures for Cooperative Intelligent Transport Systems (C-ITS) and driverless vehicles. We develop a methodology for software orchestration in the whole C-ITS and extend the Automotive Service-oriented Architecture (ASOA) originating from the UNICAR*agil* project to optimally orchestrate software on the control units of vehicles.

The architecture is realized as an open modular kit of software modules as described in Section 6. We develop robust modules for the whole effect chain of vehicle automation including localization, environment perception, planning and control as well as a highly integrated low speed function. Each software module can quantify its performance in the form of a “quality vector” that is part of each output and can be considered in the orchestration of the software.

The trend towards service-oriented software goes hand in hand with a centralization of the E/E architecture, in which the software is embedded. This centralization has the potential to reduce the complexity of the on-board networks but comes with new challenges like standardized interfaces, generic computing capabilities, and optimal resource allocation, which are researched in the AUTOtech.*agil* project as described in Section 7.

Eventually, Section 8 outlines a consistent safety and security concept, and algorithms that will enable safety assurance and approval of automated vehicles in the targeted ITS based on services and their dynamic reconfiguration.

2. Our vision for future Intelligent Transport Systems

Future ITS will have to cope with mixed traffic of agents endowed with varying degrees of automation and connectivity, and their coordinating instances. In addition to conventional traffic participants without automation or connection, this includes the following entities:

- **Connected Vehicles (CVs)** are driven by a human driver and exchange data with other vehicles and infrastructure, e.g., through ETSI ITS messages [47].
- **Automated Vehicles (AVs)** come with an automated driving function of SAE level 3 or above, i.e., the driver can temporarily be out of loop.
- **Connected Automated Vehicles (CAVs)** are AVs that exchange data with other vehicles and infrastructure. This is required for automated driving functions at least from SAE level 4 in Germany according to the AFGBV (in English: autonomous vehicle approval and operation ordinance), which stipulates a technical supervision of such vehicles.
- **Control Rooms (CRs)** supervise fleets of CAVs. They are operated by mobility providers and aid to fulfil specific purposes like the transport of people or goods. CRs may provide the technical supervision, as required by AFGBV, and customer service. They may use fleet data in their cloud backends, e.g., for continuous improvement of their service.
- A **Control Center (CC)** coordinates all CRs operating in its designated traffic area, e.g., one city. It is the responsibility of the municipality and pursues sovereign interests such as road closures and traffic flow control. The CC operates Intelligent Transport System Stations (ITS-Ss) and may combine data from CAVs, CVs and ITS-Ss into a live digital twin of traffic.
- **Connected Vulnerable Road Users (C-VRUs)** may use the live digital twin of traffic to improve their road safety, e.g., through warnings on mobile devices or HMI in the event of detected hazards.

Fig. 3 visualizes the coordination responsibilities between the described entities in a C-ITS.

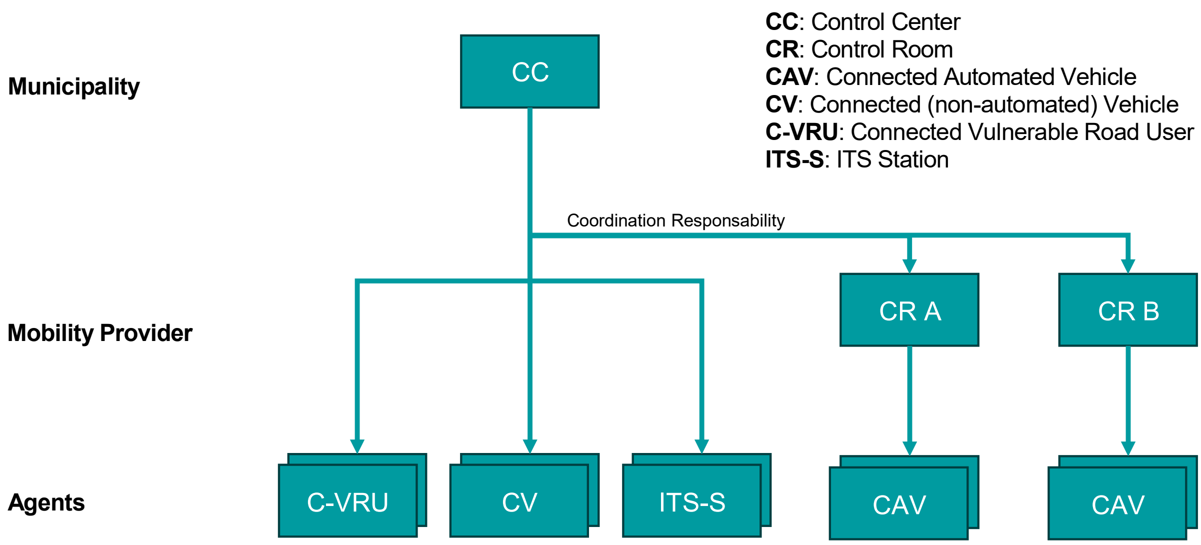

Fig. 3 Multiple mobility providers coordinate fleets of connected automated vehicles through their control rooms. A control center, that is responsibility of the municipality, coordinates all control rooms, other connected vehicles, and vulnerable road users as well as Intelligent Transport System Stations.

The AUTOtech.*agil* project addresses three relevant use cases as shown in Fig. 4 and described in Section 3:

- **Barrier-free mobility** allows all people to participate in future mobility. A digital cabin manager and connection to a control room ensure safe and comfortable passenger transport.
- **Efficient package delivery** combining automated delivery vehicles and mobile delivery robots allows fast door-to-door deliveries at off-peak hours and relieves traffic.
- **Vulnerable road users and light vehicles** are connected to the life digital twin of traffic by a Guardian Angel function that may warn them of hazards and allows them to benefit from the fleet knowledge to ensure their road safety.

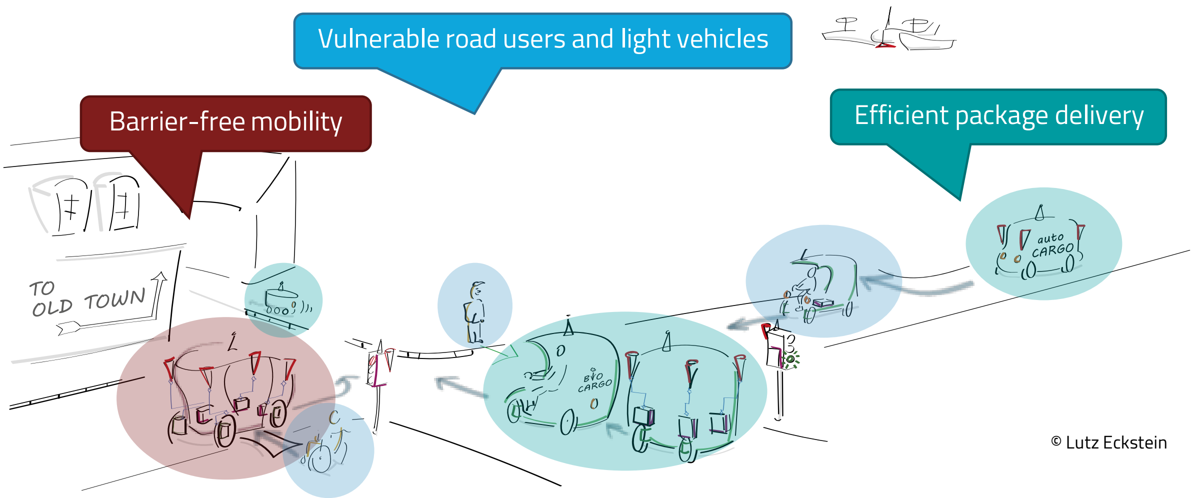

Fig. 4 Increasing automation and connectivity has the potential to enable barrier-free mobility, efficient package delivery and the inclusion of vulnerable road users in future mobility systems.

Barrier-free mobility and package delivery may be realized by mobility providers pursuing specific (e.g. economic) interests. Each mobility provider operating a fleet of connected automated vehicles will have to run a control room that provides technical supervision as well as customer support for the vehicles. As visualized in Fig. 5, AUTOtech.*agil* demonstrates an exemplary C-ITS with two mobility providers – one public transport service provider and one cargo service provider. Both operate in the same city in the area of responsibility of one control center that coordinates them and provides a digital twin of traffic based on data from ITS Stations that can be stationary, e.g., mounted at traffic lights, or instationary as fully autonomous unmanned aerial vehicles (UAVs). These UAVs automatically gather additional environmental data as requested.

To turn this vision into reality, AUTOtech.*agil* addresses the following core project goals:

- Distribution of intelligence through orchestration of services inside and outside the vehicles.
- Development platform for service-oriented E/E architectures.
- Intelligence & efficiency through digital twin of traffic and components.
- Agile updates & upgrades of services due to their explicit self-reported quality.
- Tools and methods for development and safeguarding.
- Interoperability with other ecosystems such as ROS 2 and AUTOSAR adaptive.

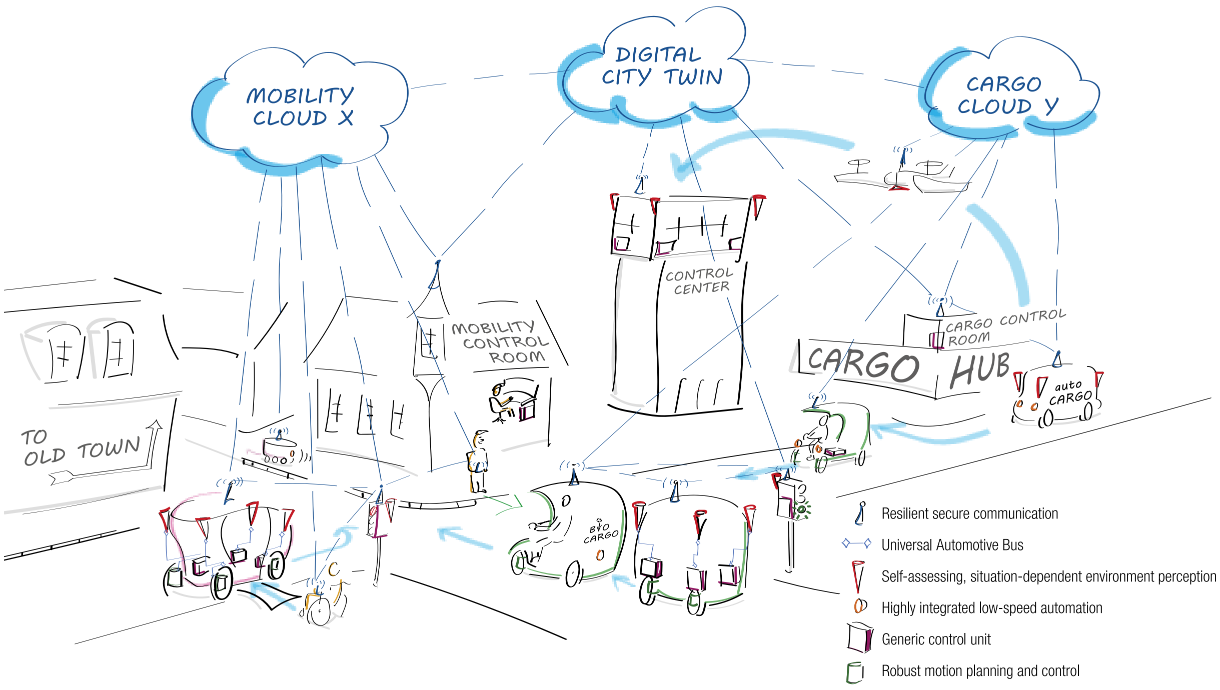

Fig. 5 Modular intelligence and technologies for future ITS

3. Significant use cases of future mobility

Future mobility systems should be designed so that all stakeholders can benefit from the transformation towards electrification, automation and connectivity. Three relevant use cases of future ITS motivate the work in the AUTOtech.*agil* project and are realized using the automated vehicle prototypes developed in the UNICAR*agil* project.

3.1. Inclusive mobility for all people

The advancing automation in the field of public transport has opened a promising future for mobility. In the predecessor project UNICAR*agil*, barrier-free access has successfully been demonstrated with the family vehicle autoELF [65]. In AUTOtech.*agil*, our focus is the creation of inclusive personal mobility that not only aims for efficiency and sustainability but is equally accessible to all people regardless of their individual abilities and needs. As an innovative form of passenger transportation, the autoSHUTTLE from the UNICAR*agil* project offers the opportunity to promote inclusion in the transportation sector, taking into account also passengers with different impairments such as visual, hearing, cognitive or physical limitations. By integrating new sensor technologies and adaptivity to the user, the automated shuttle can promote a barrier-free and safe mobility experience.

The AUTOtech.*agil* project will address three essential topics. The **adaptive cabin manager** focuses on providing an optimal experience for all autoSHUTTLE passengers. It helps users to overcome potential barriers when interacting with the autonomous shuttle by accompanying the user during the journey and by personally selecting and presenting information tailored to each user. To design the adaptive cabin manager, the User Experience (UX) principles [12] for adaptive systems are taken into account, as well as individual mobility needs of different user groups, including users with impairments. The first step is to conduct workshops to gather user needs using the Holistic Experience Centered (HEC) method [13]. The resulting needs will be transferred into requirements for the cabin manager and serve as a basis for the design of the human-machine interface (HMI), which will be developed in the project.

In order to increase and optimize passengers' comfort, the **Well-Being function** is implemented in the autoSHUTTLE. With a multimodal HMI system consisting of acoustic and visual interfaces, passengers could anticipate the shuttle's behavior by interacting with the visual and acoustic signals representing the upcoming driving dynamics and therefore prepare themselves for the upcoming driving maneuver. This anticipation should reduce the effect of motion sickness, and positively improve the driving experience and the overall acceptance to the AD technology.

Furthermore, the autoSHUTTLE will be expanded by the sense of hearing, i.e., **machine listening**. This enables a wide range of applications, the *outer ear* in the exterior of the vehicle for the detection of relevant acoustic events, such as siren detection [14][15], or for communication and function expansion in the context of inclusion. Likewise, the *inner ear* in the vehicle cabin is designed for interaction with passengers and extended inclusion, as well as for acoustic interior monitoring to detect critical situations in order to be able to act accordingly.

3.2. Efficient transport of goods

In the future, the transportation of goods is undergoing a significant transformation, as more and more automated solutions are being developed for efficient delivery of goods. The aim is to establish the most effective method for transporting goods. In the previous UNICAR*agil* project, the groundwork was laid with the autoCARGO system, enabling automated delivery of parcels to fixed receiving points. This functionality will be expanded to include a two-way material transfer to autonomous mobile robots (AMRs). This expansion involves the development of a universal decentralized communication system, which will eliminate manufacturer-specific dependencies. Consequently, this will enable the use of a heterogeneous fleet for last-mile delivery.

In conjunction with the communication system, a simulation model is developed that allows to model and simulate heterogeneous fleets for last-mile delivery. Using the simulation model, the design and control of heterogeneous fleets will be investigated and operation strategies will be developed and evaluated.

3.3. A Guardian Angel for VRUs and light vehicles

The group of Vulnerable Road Users (VRUs) comprises pedestrians, bicycles, motorbikes, and other traffic participants that are not secured by a stiff vehicle body. Consequently, their safety is of particular importance. Current accident statistics underscore the need for further improvement: A number of 347,030 people were injured in German road accidents in 2022, of which more than 45% were VRUs [2]. With 51%, their percentage among the persons killed is even higher [3], one reason for that being the lack of active safety mechanisms.

The active safety of current vehicles highly relies on various sensors perceiving their surroundings in the near and far field as well as powerful hardware to process the raw data, infer information about the environment, and interpret them to react to challenging situations as fast as possible [1]. This entire process is not feasible with VRUs and light vehicles because neither currently have a convenient possibility to carry any sensors nor provide the required computing power. Nevertheless, the majority of people carrying smartphones that are permanently connected to the internet offers the opportunity to make use of data collected by fleets of CAVs or Intelligent Transport System Stations (ITS-Ss) and processed to a live digital twin of traffic in the cloud.

The use case of the "Guardian Angel" targets exactly this gap in today's traffic systems. A mobile application will be developed with the goal to support VRUs as well as light vehicles and substantially increase their safety. For this purpose, the app will transmit its user's current location to the digital twin that returns a view of the environment model in the immediate surroundings of the requested position. In hazardous situations, the app shall warn the user appropriately. Relevant scenarios include fast vehicles driving along the road that a pedestrian is about to cross, emergency vehicles wanting light vehicles to yield right of way, or cars that have just stopped in a parking spot next to the bike lane are likely to have their door opened.

4. Learning and connected traffic intelligence

Future Cooperative Intelligent Transport Systems (C-ITS) are comprised of various connected and automated entities. A lot of effort in research and development is put into extending and improving the intelligence of individual entities. It is the cooperative character of the system though, which will possibly lead to the long-term success of automated vehicles compared to human operated vehicles. Although artificial intelligence has made substantial progress in recent years, it is the ability of computers to transmit and receive large amounts of data simultaneously to and from other entities at low latencies, which, at the moment, sets them apart from humans. This advantage enables them to form C-ITS, in which automated agents cooperate through communication both directly, during the driving task, and indirectly, by continually collecting data for learning to improve existing joint capabilities.

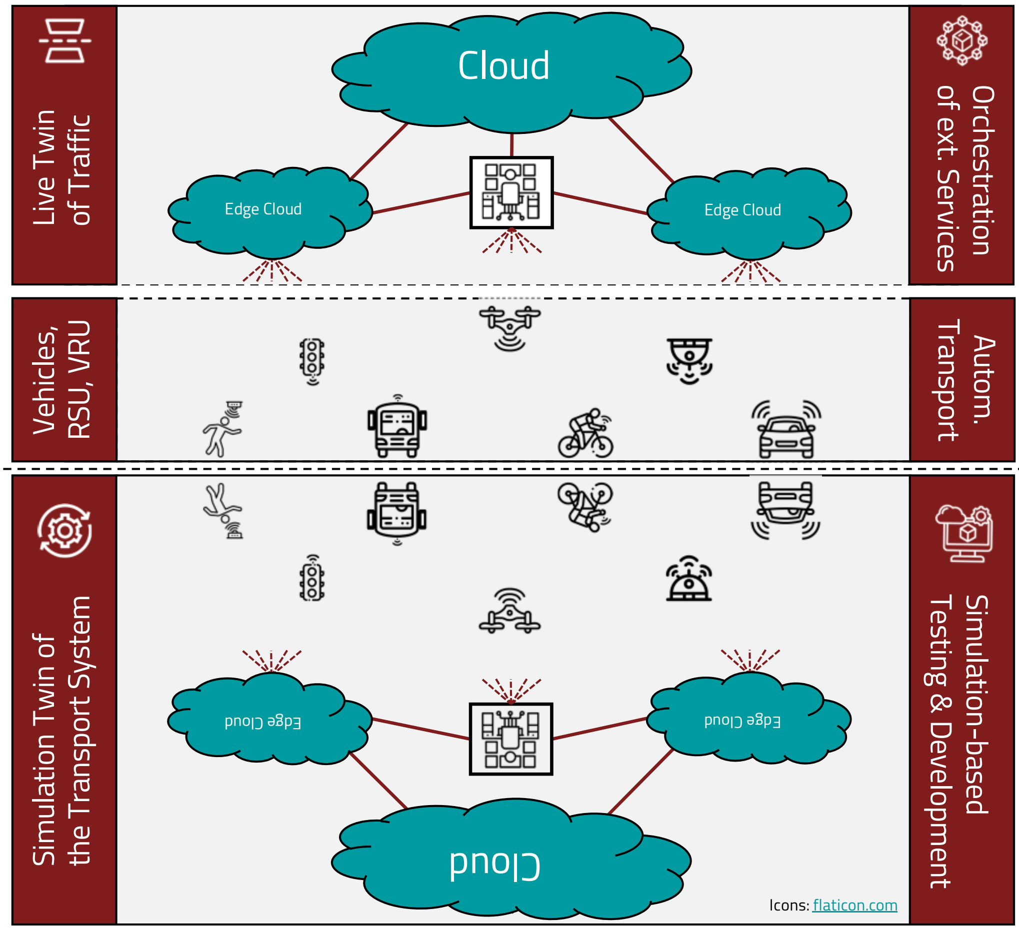

Fig. 6 The transport system is comprised of automated or semi-automated agents, and external elements such as roadside units, cloud servers, control rooms and control centers. The external elements compute, use, and provide a live digital twin which is a digital representation of traffic. A simulation twin is a digital mirror of the whole transport system including the external elements.

In our project, we develop concepts, software, and tools for software development which help pave the way towards a C-ITS as described above. Fig. 6 summarizes essential components of a C-ITS that are considered in AUTOtech.*agil*. A control center is conceptualized which coordinates individual fleet control rooms. Tools to automatically orchestrate software help manage the various connected entities. A live digital twin provides an accurate description of the current state of the entities forming the C-ITS. Connecting traffic participants to each other and with external entities such as sensor-equipped roadside units (RSUs) and the cloud allows cooperation through an exchange of data, negotiated behavior, and the sharing of processing power. A simulation twin of a C-ITS acts as an important development and safety assurance tool by enabling large-scale tests in a safe environment. Developed collective and cooperative functions are embedded into the overall service-oriented architecture and into a DevOps process aiming to automatically and continually improve the C-ITS.

4.1. The role of Control Rooms and Control Centers

The recent deployment of autonomous vehicles in San Francisco shows that despite their legitimate role in future mobility, further coordination of the vehicles must take place to successfully introduce them in the future transport system [49]. The coordination of autonomous vehicle fleets can be split into operative tasks and a municipal interest that both need to be realized by the fleet operator. Operative coordination specifies all coordination actions that aim at successfully completing the fleet's economic mission. Municipal interests include the superordinate goal of efficiency and safety in the transport system.

In UNICAR*agil* [42], a **control room** concept for automated vehicles including teleoperation as a fallback layer in case of automation failures was investigated and implemented [50]. With this concept, the operative coordination of a fleet was demonstrated at the final event in May 2023. The control room concept consisted of general coordination tasks on map-based fleet visualization and designated teleoperation workplaces that allow for a mission completion in the case of an autonomous function failing.

In this project, the concept of a **control center** is investigated to address municipal interests such as increasing road safety and efficiency. Operated by the municipality, the control center receives, among others, the current autonomous vehicle positions, their statuses and information from the RSUs via the live digital twin and the control rooms. In order to achieve the goals of safety and efficiency, souvereign instructions, such as redirection of the traffic in the event of an accident, can be sent from the control center to control rooms for coordination. Further tasks of the control center include traffic monitoring and control, the management of RSUs, the implementation of political guidelines and the implementation of smart city concepts.

4.2. Collective and cooperative functions

Among the C-ITS software developed in the project, there are both collective and cooperative functions, building upon results of UNICAR*agil* [42]. They are collective in the sense that all connected entities can benefit from a centralized collection, fusion, processing, and sharing of processed data. They are cooperative in the sense that entities may share data and negotiate behavior in a decentralized fashion.

An essential collective function is the **live digital twin** [42][41]. It is a digital representation of the C-ITS entities built from a combination of data these entities share. This representation may consist of multiple layers such as

- map data (e.g., lane topology and geometry, regulatory elements, static objects),

- sensor data (e.g., lidar point clouds, camera images),
- environment representations (e.g., object lists, grid maps),
- planned and predicted behavior (e.g., routes, maneuvers, trajectories),
- self-reported states (e.g., pose, speed, acceleration),

and more. In the project, a focus will lie on the integration of data from sensor-equipped infrastructure into the digital twin. The infrastructure can either be stationary, i.e., sensor-equipped RSUs, or mobile, i.e., UAVs which automatically gather additional data regarding specified areas upon request. Different instances of a live digital twin may be provided by different connected entities depending on, e.g., the requirements posed at the service quality. A multi-access edge computing (MEC) application server connected to nearby sensored RSUs may compute and provide selected layers of a local live digital twin to nearby automated vehicles at low latencies. A control center on the other hand may have less strict requirements regarding latency compared to automated vehicles and may access a cloud-based live digital twin holding the self-reported states of connected vehicles.

One focus of the project is the integration of sensor-equipped RSUs into the overall service-oriented architecture, and an event-driven **orchestration framework** for managing C-ITS applications running on connected entities such as the sensor-equipped RSUs. It will allow, e.g., computation offloading and data collection for collective learning [45]. Both are further means through which connectivity can help improve a C-ITS. Since computing and energy resources in traffic participants are rather limited, it can be beneficial to offload computative intensive tasks to more capable entities such as MEC application servers or cloud servers.

Data collection for collective learning is one aspect of a centralized C-ITS **DevOps process** aiming to improve the development life cycle of the C-ITS and its components. The overall DevOps process developed in the project comprises six essential steps.

1. Deployment of services
2. Monitoring and assessment
3. Collection of relevant data
4. Dataset creation
5. Data-driven development of services
6. Testing of services

As described in Fig. 7, these steps form a cycle aiming to mostly automate the continual improvement of individual C-ITS components and the C-ITS as a whole. One focus in the project is the development of a simulation framework which allows to develop and test functions in a simulation twin. Due to the vast amount of possible and relevant scenarios in which automated vehicles need to perform safely, and due to the limitations of real-world testing, simulations play a vital role in this step of the DevOps cycle. For this purpose, a **simulation twin** is developed. While a live digital twin describes the current state of a C-ITS, a simulation twin can also describe potential states based on models of both hardware and software of the C-ITS. These models allow risk assessments before new or updated software is deployed. A major challenge when developing a simulation twin lies in the validity of the simulation and its components, since different aspects of reality can only be modeled with limited fidelity. Nowadays, increasing fidelity often comes with a lot of human effort, which is why methods shall be developed that help automatically build high-fidelity simulation twins.

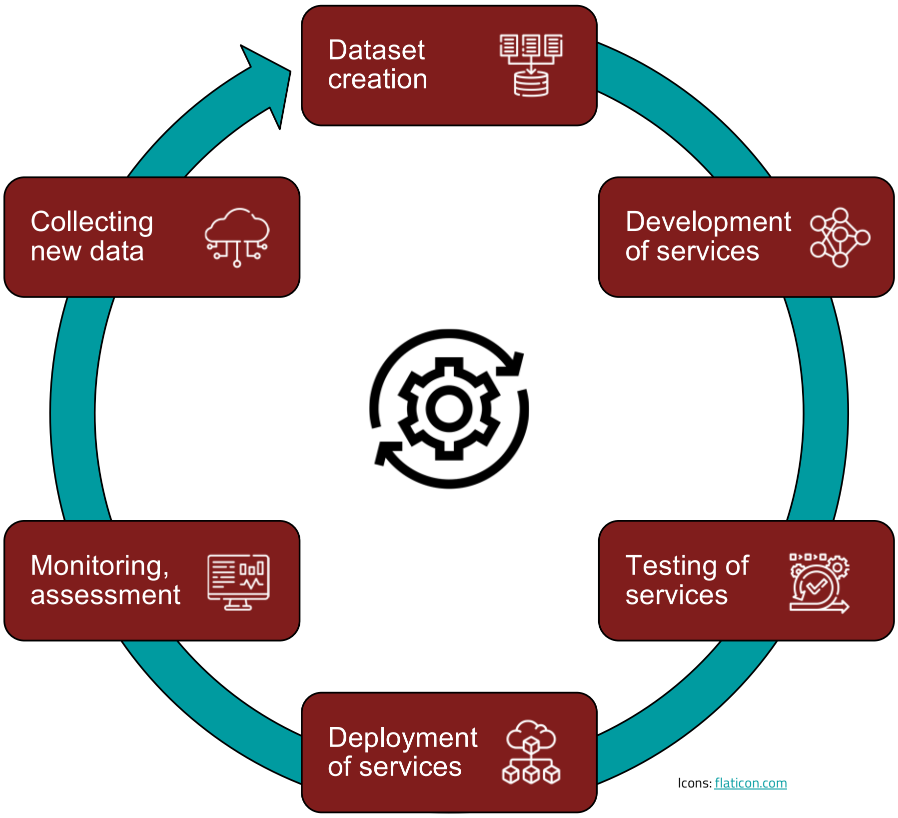

Fig. 7 The DevOps cycle describes how services are continually monitored, improved, and updated over their lifetime. The process is data-driven, meaning data gathered from the services is at the center of assessing their performance, and also essential for improving them, either through machine learning or other means.

The data exchanged in a connected ITS cannot only be used for “offline” collective learning but also for “live” cooperative functions such as **cooperative behavior** decisions. Although autonomous vehicles are called autonomous this doesn't mean that they are making decisions independently from everyone else. In some situations, they might need to consider pieces of information that weren't sensed by themselves but that were collected and communicated by others, e.g., by other vehicles, by sensors on the infrastructure, or even by drones that supervise traffic from bird's-eye view. Furthermore, autonomous vehicles might want to negotiate their behavior with other traffic participants to optimize traffic flow and to achieve more comfortable driving behavior. They might be influenced by commands and hints of a control room, and they want to benefit from data and experiences that other vehicles previously collected. These kinds of cooperative behavior are considered in AUTOtech.*agil*. We develop an architecture that enables us to integrate pieces of information from many different sources and that allows the vehicles to make optimal decisions in the context of other traffic participants. It is important that cooperative behavior remains safe, and that decision making can be retraced.

5. A software ecosystem for future Cooperative Intelligent Transport Systems

The complexity of future Cooperative Intelligent Transport Systems requires new approaches regarding the tools for building and testing software for these C-ITS, regarding the software architectures that can be represented in the C-ITS, and regarding the orchestration tools that manage the transition between different architecture instances. Our project therefore develops concepts, software and tools enabling these new approaches. In the following, we will describe a toolchain for prototype development and testing, a C-ITS architecture and an approach for C-ITS orchestration, and the Automotive Service-Oriented Architecture (ASOA) for agent-based orchestration.

5.1. Toolchain for prototype development and testing

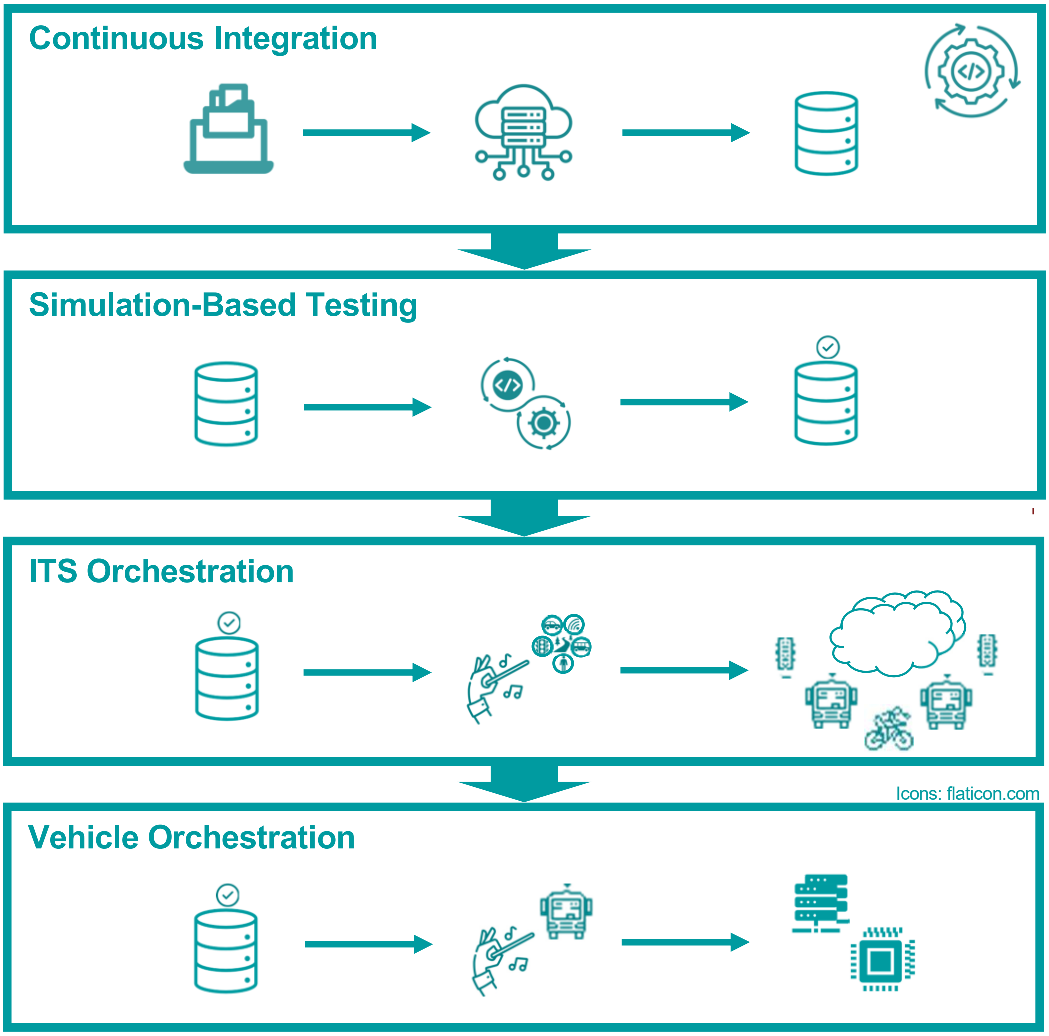

Fig. 8 The AUTOtech.*agil* software ecosystem comprises tools and processes for continuous integration, simulation-based testing and orchestration of services in a connected ITS.

The development of software for C-ITS typically involves many separate entities, e.g., teams in an organization or even across organizations. This poses a particular challenge when integrating the software of these different development teams into a complex system such as a C-ITS. The development of a first prototype for real-world testing is already immensely difficult, which is why, in the following, we focus on the toolchain to reach this first prototype and and not on a real-world validation strategy.

The toolchain, as shown in Fig. 8, builds on three important principles:

1. Documentation as code for common definitions
2. Continuous integration, testing, and containerization
3. Continuous simulation-based testing

Good documentation can help harmonize, e.g., interfaces, but risks being outdated when following agile software development processes. **Documentation as Code** is an approach to tackle this challenge. Here, documentation is written with the same tools as code. This can go as far as "Documentation is Software" meaning the documentation is what is actually compiled into the application software during the build process. In the microservice architecture employed in our project, this approach is especially useful when it comes to the interfaces between services. The communication between services typically involves the exchange of messages, which are defined in message files containing code readable by both humans and machines. Message files constitute one type of **common definitions** that are stored in definition repositories just like application code. They are therefore also subject to, e.g., version control, code reviews, and automated tests. Additional common definitions are also stored as code in repositories, e.g., coordinate frames and their relation to each other.

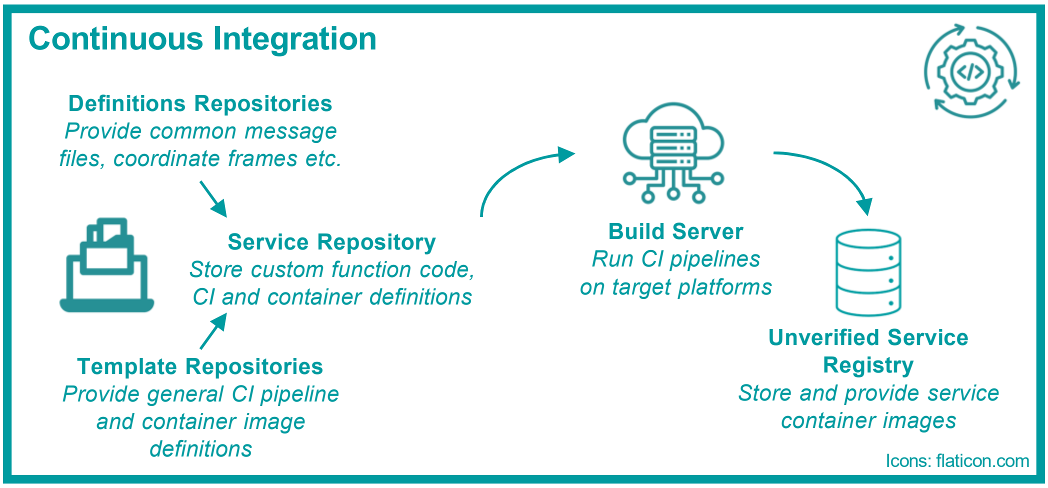

Fig. 9 Continuous Integration allows to run automated build tests on a dedicated build server any time a developer pushes code changes to a service repository. Service repositories include common definitions repositories that document message definitions as code, ensuring that different services can communicate with each other. Developers are provided templates for CI pipeline definitions and for container images, allowing them to focus on service development. Services which passed the automated CI tests are packaged into container images and stored in a registry of unverified services.

Continuous integration is an essential part of any successful DevOps process. Containers offer consistent, isolated environments, which are integral to Continuous Integration's automated testing, to ensure reliable code integration and deployment across diverse settings. Fig. 9 shows how we separate service development from the configuration of continuous integration and containerization as much as possible. All services may use common templates for continuous integration and containerization. Developers integrate the code of each service into a dedicated repository where they need to provide the names of required dependencies and choose the target processor architecture. Apart from these steps, service developers do not need to configure the CI or containerization. Instead, they make use of automated containerization [48]. A service is a piece of software that communicates via messages with other services and shall in principle be updatable and orchestrated independently of other services in the target system. All services include the applicable common definitions repositories ensuring a successful communication between all developed services. After a successful automated build during which services are containerized and subject to optional additional tests, the containers are pushed to an **unverified service registry**. The services are unverified in the sense that it is not ensured yet that their provided output data corresponds to the desired output data when running jointly with other services in a system of services.

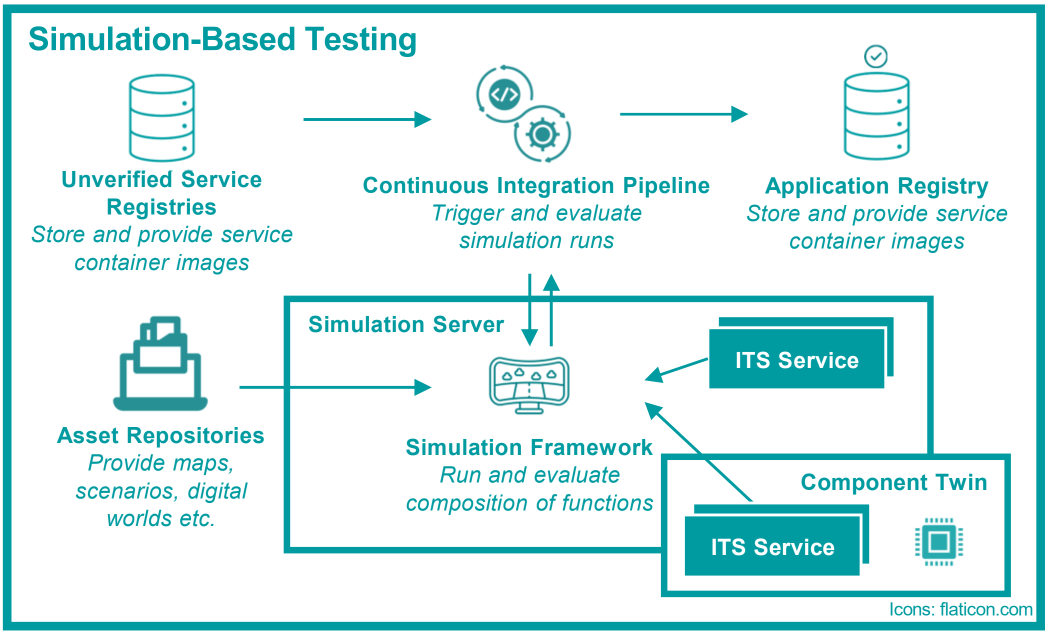

Fig. 10 A continuous integration pipeline for simulation-based testing evaluates whether unverified services produce desired behavior when run jointly, according to predefined metrics and scenarios. Services may run either on a simulation server, or on a component twin for hardware-in-the-loop tests. Sets of services that pass the tests are moved to application registries where they can be accessed for further, e.g., real-world testing.

Before deploying services in the C-ITS, they need to be verified jointly, because non-trivial services can produce undesirable and unpredictable emergent behavior when run jointly. One approach to joint testing services is **simulation-based testing** as shown in Fig. 10. Due to the vast amount of relevant scenarios, it is infeasible for developers to manually test their services, let alone compositions of services in a C-ITS, in a simulation. Therefore, a CI pipeline triggers simulation runs to test all relevant scenarios. The overall goal of the simulation CI pipeline is to test whether services are capable to jointly produce desired behavior. Those compositions of services that are capable to do so, are transferred to an **application registry**. An application in this context is a set of services that have been jointly tested and approved for, e.g., real-world testing. Just like the services, the simulation framework includes the common definitions repositories ensuring that the simulation framework and the services can communicate with each other. The simulation can run either only on simulation servers, or it can also partly incorporate hardware component twins, and run a set of services on these. Common assets repositories provide, e.g., maps, scenarios, and digital worlds necessary to build the simulation twin. Of course, simulation-based testing can only be one step in a chain of validation steps before software is actually deployed "to production". The aforementioned steps shall therefore describe an approach to create a relatively mature set of services suitable for, e.g., real-world testing, and not an all-encompassing safety assurance framework.

5.2. C-ITS software architecture and system-wide orchestration

Large-scale systems like C-ITS require software architectures and tools to manage them that work at scale. Cloud native computing approaches have already proven to be suitable for these large-scale systems. The Cloud Native Computing Foundation defines cloud-native as follows: "Cloud-native technologies empower organizations to build and run scalable applications in modern, dynamic environments […]. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach. These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil." [44] Due to its advantages, we adopt and adapt this approach for C-ITS.

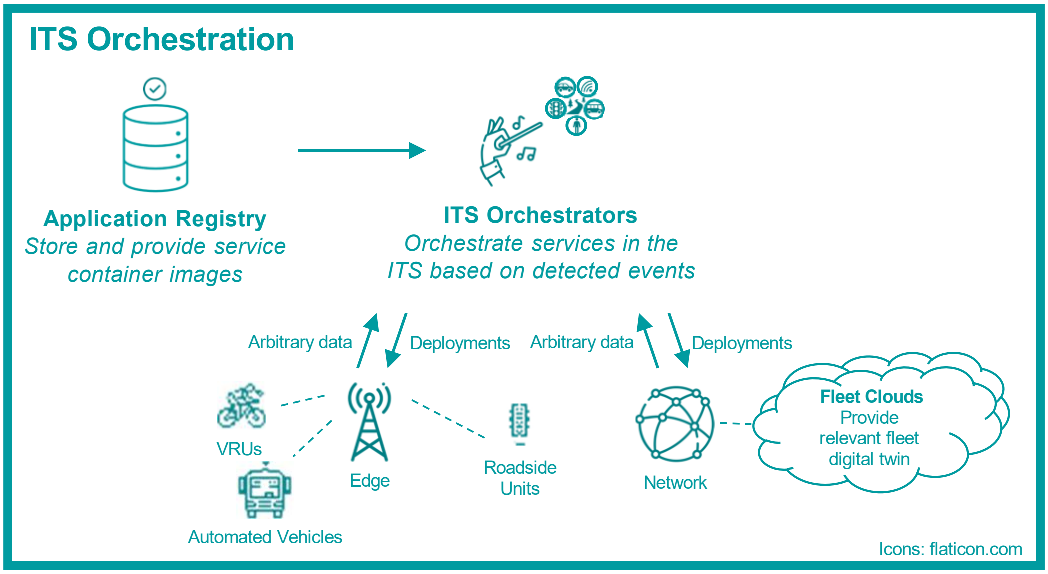

Fig. 11 Applications in a C-ITS can be managed by a set of application-specific ITS orchestrators that can access services stored in application registries. Orchestration is triggered by events detected by dedicated software components analyzing data exchanged by entities in the C-ITS. Events can for example trigger the deployment of a new application to certain ITS entities, or the reconfiguration or update of an already running application.

On C-ITS level, applications running in the various connected ITS entities such as sensor-equipped RSUs, MEC servers, and clouds are managed by **ITS orchestrators**.

Entities that perform safety critical tasks can additionally employ, e.g., a vehicle orchestrator that comes with specialized capabilities and privileges. Applications are provided by application registries, which are maintained and updated by DevOps processes. Provided applications have to go through multiple stages of safety assurance. ITS entities come with configurations for core services that they run as soon as they are powered. Additional applications on ITS entities can be orchestrated. This primarily happens through an event-based orchestration mechanism [45].

Software components for event detection can be employed together with a corresponding application manager, forming an ITS orchestrator that can trigger deployments of additional applications or the reconfiguration of already running services. These orchestrators are application-specific and run in a decentralized manner. One example event is the detection of a potentially safety-critical scenario triggering reconfigurations and deployments of services for recording, storage and analysis of corresponding data in the cloud. Another example event is the demand-driven deployment of supportive functions to sensor-equipped RSUs when suitable receiving automated and connected vehicles are nearby.

5.3. Automotive Service-oriented Software Architecture (ASOA)

In the automotive domain, software is subject to distinct constraints and requirements, including real-time responsiveness, limited computational capacity, and energy efficiency. These specialized conditions demand domain-specific architectural solutions. While traditional Service-Oriented Architectures (SOAs), common in the Cloud Computing domain, provide modularity and updatability, they do not address the specific requirements for automotive settings. Likewise, Microservices Architectures (MAs), similarly prevalent in cloud-based systems and also employed in our Intelligent Transport System (ITS) architecture, do currently not meet the real-time and resource-constrained demands of automotive systems. The highly distributed nature of MAs can lead to latency and resource overhead, making them imperfect for application in vehicles. As a result, specialized frameworks like the Automotive Service-Oriented Architecture (ASOA) bridge these gaps and address the unique challenges inherent to the automotive domain.

The ASOA represents an advancement in the automotive software domain, building upon the achievements made in the UNICAR*agil* project. Traditionally, automotive software architectures have been function-oriented. In keeping with new trends, ASOA facilitates a transition from function orientation emphasizing a service-oriented approach. This service orientation allows for the deployment of services on general-purpose hardware, enhancing flexibility and adaptability.

In the **ASOA framework** [40], services function as distinct software components, with each service fulfilling a different task in the system. To achieve the complete system functionalities, the services cooperate using ASOA's two communication paradigms. ASOA supports publish-subscribe and request-response based communication between services.

The execution model of ASOA is characterized by Tasks. The Task model transforms the service from a black box entity to a gray box system. This transparency enables the observation of computations within each service and enables both diagnoses, as well as optimization of information flow through the system.

The communication in ASOA makes use of requirements and guarantees. These are designed as generic typed interfaces which are unconnected at system start, ensuring that no system level information is encoded in the services. This design principle ensures that services remain modular and adaptable.

ASOA enables runtime integration, which integrates services during operation rather than being predefined during the design phase. Central to this dynamic integration process is the orchestrator as an architecture controller. The orchestrator ensures that the system conforms with the operational mode. To achieve this, it connects requirements and guarantees of services, pauses services to release their resources, and starts required services.

Building upon the foundational principles of ASOA, the AUTOtech.*agil* project seeks to further enhance and extend the capabilities of the architecture. This project poses a range of research questions aimed at improving the system's use of resources, determinism, and adaptability. Among the key extensions are strategies for the optimal allocation of computing resources, to ensuring determinism in system operations, and the introduction of new orchestration mechanisms. Additionally, AUTOtech.*agil* emphasizes the development of tools tailored for architecture description, simplifying interaction with other systems and the cooperation between partners. Furthermore, a notable direction of this research is the integration of service-oriented approaches for diagnosis and maintenance, ensuring the system's longevity and reliability.

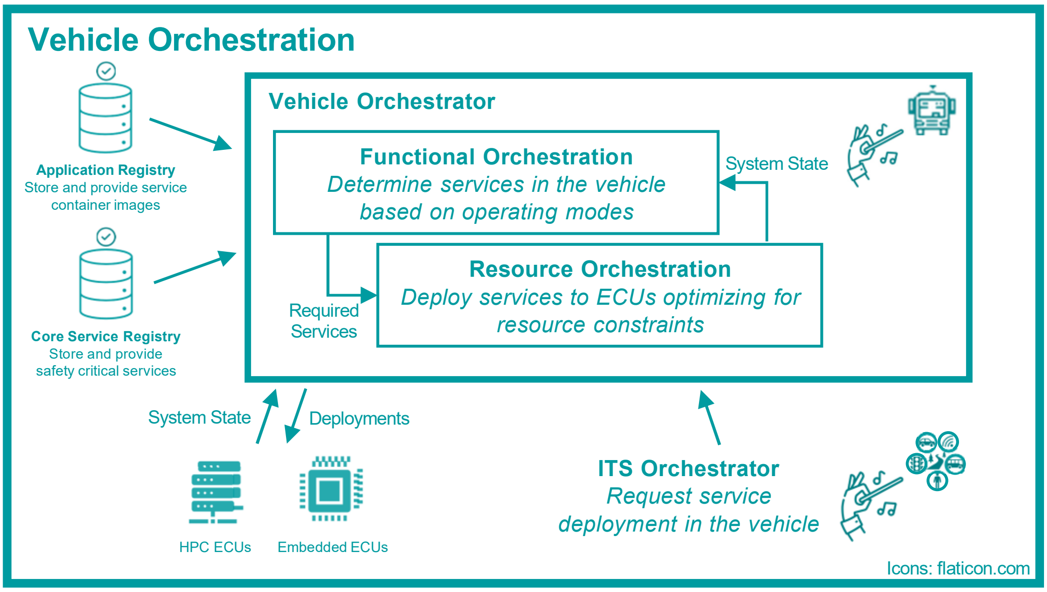

Fig. 12 Applications in the vehicle are managed by a central orchestrator. In functional orchestration, it considers the services to support the current operating mode of the vehicle as well as requests from the ITS. In resource

orchestration, the orchestrator computes the resource and efficiency optimal deployment of the services to the vehicle's ECUs.

A significant enhancement in AUTOtech.*agil* revolves around the **optimal allocation of computing resources** to vehicle compute units. As more complex functions need to be performed in vehicles, efficient use of resources is crucial. Our research aims to expand resource orchestration [47], emphasizing both online adaptability and more encompassing system models. While the system model dimension addresses challenges such as incomplete models for GPUs, Time-Sensitive Networking (TSN), and core sharing, the online aspect focuses on dynamic task scheduling during runtime. This ensures that the system can adapt in real time, optimizing its operations to improve performance.

In addition, the aim of our research is also to expand the **functional orchestration**. While the current state of functional orchestration offers a highly structured approach, the ambition is to transition towards a more flexible and adaptive model, exceeding the limitations of the existing state machine-based approach. A promising idea under consideration is the capability to automatically determine the necessary transitions between various system states. Additionally, this refined approach could be more tightly integrated with the resource orchestration, further optimizing the system's performance.

As part of the trend toward intelligent infrastructure, we seek to **expand orchestration beyond the individual vehicle**. The idea is to seamlessly integrate Roadside Units (RSUs) or cloud systems in ASOA to enhance its capabilities or temporarily reduce computational load. By shifting the load of certain services to RSUs or the Cloud, the vehicle's computational load could be reduced. Moreover, mechanisms are being developed to access sensors or data housed within these RSUs, further broadening the scope of information available to the system. The overarching plan is to expand the concept of orchestration beyond the limits of the vehicle, creating an interconnected system.

To ensure predictable computations in all these components, another aspect under active research in AUTOtech.*agil* is **determinism** in ASOA, especially given the inherent challenges it presents in distributed systems. Recent works have shown that non-deterministic behavior in distributed systems can yield unforeseen effects and system behaviors. The overarching aim is to achieve deterministic behavior in all ASOA components. However, the current concept presents only semi-deterministic outcomes. Factors such as operating system scheduling, influences from the Data Distribution Service (DDS), and networking effects contribute to this unpredictability. Research within AUTOtech.*agil* is directed towards mitigating these influences to enhance the system's determinism.

To support the development of systems based on ASOA, AUTOtech.*agil* aims to continue the **development of tools** for architecture design, especially in collaborative settings involving multiple partners. The objective is to enhance the current architecture development tool for ASOA-based systems design. These tools encompass detailed specifications, including the system architecture, and the types and definitions of interfaces. Furthermore, to streamline integration and communication, partners will be provided with machine-readable formats for service definitions and messages, ensuring seamless interoperability and consistency across the system.

To detect errors if they occur, we lay a strong emphasis on **service-oriented diagnosis and predictive maintenance**. The goal is to diagnose complex system failures during the vehicle's operation, ensuring timely interventions and minimizing disruptions. This diagnostic approach will be underpinned by meta-data and information from the orchestrator, providing a comprehensive view of the system's health and performance. Such a proactive approach not only enhances the reliability of the vehicle but also extends its operational lifespan.

Diagnosing an automotive service-oriented architecture requires a different approach than diagnosing classic vehicle architectures which focuses largely on electrical and mechanical problems. Problems of the vehicle’s software have been out-of-scope. For a largely software-driven vehicle, issues in the software can no longer be ignored. Software issues in this context can be differentiated in different categories:

- **Service composition**: As services depend on each other, problems in one service may propagate to other services. Thus, the service composition must be diagnosable to identify for a service whether all required services are available, whether messages are correctly routed, the service communication is error-free, or whether the dynamic service composition at run time operates as expected.
- **Service execution & execution environment**: Services have a relation to their execution environment, which can be subject to software-related issues. E.g., for each service an assumption regarding resource consumption has been made at development time. These assumptions must be validated at execution time to prevent errors in other services due to over-commitment of the used resources. In addition, the execution path of a request, e.g., originating from a control room, must be traceable down to the behavior of the vehicle to identify the root cause of an unexpected behavior and the service that caused this.

For diagnosing these problems, the widely common UDS protocol [5] is not sufficient anymore. Thus, the ASAM SOVD standard [6] has been developed. SOVD provides a unified REST API for diagnosing high performance computing units as well as classic ECUs. In addition, vehicles can be diagnosed remotely via the cloud. The current research already uncovered a couple of shortcomings of the ASAM SOVD standard, which will be addressed by upcoming versions of the standard. Based on the SOVD API, the **digital diagnostic twin** for a vehicle allows a control room to (a) get an overview on the fleet health status and (b) perform predictive maintenance to identify issues with a vehicle before they are out-of-service.

We develop digital twins for the purpose of diagnosis and predictive maintenance. This allows a) to outsource diagnosis operations from the vehicle to external, powerful computing resources, and b) to diagnose beyond single vehicles, with a view upon fleets of vehicles. More details on the diagnostics of the ASOA platform can be found in [4].

5.4. Integration of disruptive SOAs in industrial automotive platforms

Today's automotive systems are the result of a long-term co-evolution of hardware and software architectures, and comprise a well-established ecosystem of vendor specific tools. Our goal is to enable a fast transfer of disruptive research results into the industrial practice, by enabling interoperability of existing automotive systems with upcoming software architectures. The stringent safety requirements that have to be fulfilled for automated driving thereby imply a focus on a design process, which incorporates timing as a central aspect.

The state of technology for modern automotive systems is reflected by the AUTOSAR standard. It comprises two types of platforms, namely the AUTOSAR Classic and the AUTOSAR Adaptive platform. The traditional approach of mostly statically configured control systems with the highest safety requirements is represented by the Classic platform. It allows for a high degree of predictability regarding timing behavior by the specification means of the AUTOSAR Timing Extensions [60]. The Adaptive platform on the other hand puts focus on a flexible and SOA based software architecture, which is comparable to ASOA, at the cost of reduced time predictability. While the AUTOSAR standard only defines the specification means, different vendors compete in their implementation of software as well as development tools for configuration, synthesis and test. Enabling compatibility with the existing AUTOSAR standard is therefore important to introduce novel approaches in the software ecosystem, and future automated driving applications require a close interaction between all the involved platforms.

For future automotive systems, an efficient handling of timing behavior and data flow (the way how different data samples in the system are propagated and combined), is of special importance. This has to be done in a unified and systematic approach, which covers all development phases and all platforms. As a first step, the concept of Logical Execution Time (LET) [61][62] has been introduced for the AUTOSAR Classic platform, since it allows a time deterministic design as well as an efficient lock-free implementation of communication [63]. Nevertheless, LET is limited to a local scope and not applicable for SOAs. The System-Level Logical Execution Time (SL-LET) [64] paradigm therefore provides a platform agnostic abstraction of runtime behavior and thereby enables an efficient design of timing for distributed software, where robustness towards platform changes and frequent software updates plays a key role. SL-LET has already been introduced in the AUTOSAR standard for both, the Classic and the Adaptive platform. Since SL-LET allows to cover (distributed) communication, distributed clock bases as well as pipelined execution, it provides the developer with a powerful specification mean for complex systems with dependent cause-effect chains. Additionally, SL-LET is already applicable on the functional level, thus it enables to close the gap between safety requirements on a vehicle function and their technical realization.

Our goal is to demonstrate the benefits of SL-LET in all stages of the development process, while putting a focus on a throughout tool support. This includes consistent decomposition rules across design stages (and across the involved models), the automatic code generation during integration, as well as the generation of trace and diagnostics information for verification. To demonstrate the applicability of SL-LET in a real-world automotive process, we start with the identification of architectural characteristics that enable a mixed mapping of functions on AUTOSAR Classic and Adaptive. Particularly a cross-platform interface for safety-relevant communication is of high interest for future automated driving scenarios. In addition, concrete use cases are formulated at the functional level in order to examine the applicability and effectiveness of the solution approach. Finally, the use cases are implemented on a hardware demonstrator in order to validate the results through practical implementation.

6. Robust vehicle automation with continuous self-assessment

Driverless automation of vehicles in mixed traffic faces multiple challenges, e.g., the complexity of especially urban traffic scenarios or adverse weather conditions. As the overall concept (see Section 2), the automation concept of AUTOtech.*agil* builds upon our automation approach [7] from the UNICAR*agil* project (see also Section 1). It makes use of the ASOA concept (see Section 5.3) and bases also on a modular mechatronic architecture [8].

In AUTOtech.*agil*, we extend our approach by systematically integrating quality vectors to the output of each service or module, which will contain information from self-assessment that reflects the current capabilities. Furthermore, we extend the probabilistic framework, which is commonly used for perception tasks, also to the on-board planning and control task, to strengthen the robustness of the complete automation chain from sensors to actors. Additionally, the automation system will be adapted to interface with the new external service components of the intelligent and connected infrastructure (see Section 4). Furthermore, we will exemplarily show the realization of different low-speed AD use cases (e.g., Automated Valet Parking (AVP) and Shuttles) in one generalized function, which is based on a dedicated energy-efficient and upgradable short-range sensor module.

6.1. Robust and self-assessing localization in urban environments

Automated driving requires a robust and self-assessing localization service which provides ubiquitous accurate and reliable localization also under unfavorable conditions, e.g., in urban environments with difficult global navigation satellite system (GNSS) signal reception or under bad light and sight conditions. Accuracy requirements are, depending on the automated driving function to be serviced, in the few centimeters to decimeters range. Today however, localization quality is still vulnerable to GNSS outages in the urban area due to signal obstruction and multipath effects. Similarly, visual localization suffers problems under low light (dusk, dawn), bad weather conditions (fog, rain, snow), and dust.

These problems will be tackled in AUTOtech.*agil* with the aim of overcoming outages due to insufficient localization accuracy. The strategy followed in our project can be summarized by three interrelated activities:

- Firstly, the incorporation of new supplementary **localization technologies** to increase the redundancy in available measurement data from a set of complementary sensors to be used in the sensor data fusion process. Preference will be given to radio positioning technologies and perception sensors like lidar, which do not require additional infrastructure on ground.
- Secondly, improved **evaluation methods** in sensor data fusion and visual localization shall be developed aiming at an increased robustness of accurate localization. An architecture of diverse localization functions will be developed: They will run parallel, be tested against each other, and their results will finally be merged in a subsequent fusion layer.
- Thirdly, the redundant information is used to develop a **self-assessment** method which provides a clear description of the current localization quality and usability with respect to the requirements for automated driving. Self-assessment information will consist of performance metrics like accuracy and integrity as well as information about sensor failures or suspicious sensor performances. It will be compiled in a stringent way using information from the individual sensors and the results of sensor data fusion and localization information estimation.

A thorough benchmarking will accompany all development steps and innovations to quantify and assess the improvement and progress towards the above-described aim of robust and accurate localization everywhere in urban environments.

6.2. Self-assessing and situation-aware environmental perception

Sensor setups, like the modular setup from UNICAR*agil* [1][7][8], redundantly surveil the automated vehicle's environment with multiple sensor principles (camera, lidar, radar) to ensure safety in complex traffic scenarios even in the case of single failures (fail-safe operation). However, the processing of all sensors' data all the time leads to a significantly high energy consumption of the automation system. Depending on the current situation, e.g., traffic scenario and/or vehicle state, the requirements on the environment model can vary heavily, meaning that not all information from all sensors is required in all situations. Thus, focussing on processing only relevant data can significantly reduce the required energy [9][10][11]. AUTOtech.*agil* aims to integrate this situation-aware environment perception into the ASOA concept and additionally develop appropriate adaptive detectors exemplarily for lidar sensors.

Besides, AUTOtech.*agil* consequently enhances the idea of self-assessment from UNICAR*agil* by introducing quality vectors for each service or module. Thus, the respective methods to calculate appropriate self-assessment scores in the onboard perception will be investigated within the project. Here, exemplarily, a special focus will be laid on camera detectors and trackers under adverse weather conditions, for which the detection and tracking performance should also be enhanced, as well as the subsequent fusion of individual quality vectors of different sources into one perception output quality vector.

6.3. Robust planning and control through probabilistic methods

Another focus of AUTOtech.*agil* is the consistent probabilistic handling of uncertainty, from sensors and trajectory planning to controllers and actuators. While probabilistic approaches are already state-of-the-art in sensor technology and perception, planning and control are mostly based on deterministic physical system models, which do not allow any statement about the robustness and reliability of the system under the influence of uncertainties [7]. Especially for highly automated driving, the consideration of uncertainties is essential. For example, the states estimated from noisy sensor measurements must be taken into account during planning. In addition, the vehicle model is also subject to unknown changes, like varying weather conditions, that pose a particular challenge. However, the largest uncertainties arise from the inherent complexity of the task. In particular, the emerging behavior of other road users and the prediction of their behavior is challenging. The transition from deterministic to probabilistic methods also requires a focus on hybrid and data-based methods to address the non-trivial aspects of predicting other road users. In terms of robustness and reliability, therefore, data-based methods may need to be reconsidered and enhanced. Ultimately, instead of a single deterministic trajectory, there is the possibility of optimizing a distribution of trajectories and passing the additional information to the lower-level control system.

To achieve a holistic uncertainty treatment, the additional probabilistic descriptions of the trajectory planning are considered in the lower-level motion control system. Furthermore, the motion control system integrates abstract models of the actuators and handles disturbances and uncertainties acting on the vehicle and actuator system.

6.4. Highly integrated low speed automation

Low-speed AD applications offer many potentials for a wide range of stakeholders. First, they represent a potential path for manufacturers and regulators to introduce highly automated driving functions (SAE Level 3 and 4) into the existing traffic, as they pose limited risk due to the low kinetic energy. In addition, the technical complexity for perception, prediction, and planning is reduced. The most popular and technically advanced use case is Automated Valet Parking (AVP), which is already found in industrial series and regulatory approved since 2022 [28]. In addition, the use of automated shuttles in delimited areas enables many application areas, especially for industrial sites (e.g., cargo shuttle on company ground or group shuttle at an airport). The application of low-speed AD functions in public road traffic expands these possibilities. It is to assume that dedicated low-speed functions can take over the driving task in specific low-speed situations and thereby supplement capabilities of the full-range function or replace them with a more energy-efficient operation.

In summary, there are many opportunities and stakeholders for low-speed AD functions. In current research and state-of-the-art technology, these use cases are mostly realized by independent driving functions, tailored to each specific application. To efficiently combine them in one AV architecture, a holistic approach of a generalized low-speed function is of interest. However, first initiatives in this area, such as ISO 22737 [29], are still at an early stage. To close this gap, one goal in the AUTOtech.*agil* project is to develop a generalized AD function for low speeds (**"Generalized Low-Speed Function"**, GLSF). This GLSF comprises its own sensor module and software stack and shall cover as many different use cases in the low-speed range as possible. Thereby, the project builds on the experience gained from the function “Safe Halt” [30] of the project UNICAR*agil*, which is also a low-speed function with proprietary hard- and software. In order to meet the project’s idea of a modular, resource-efficient transport system, the GLSF is realized with an energy-efficient, low-latency, and upgradeable sensor module for short-range applications. The GLSF will further rely on its own planning module, which will send trajectories to the central vehicle controller and will also be designed towards low complexity and efficient responses. Low latencies shall enable applications in public road traffic, as for example traffic-calmed areas (in German: “verkehrsberuhigter Bereich”), which place high demands on response time, when, e.g., reacting to a sudden ball rolling on the street.

The **short-range sensor module** of the GLSF comprises several environment perception sensors, the perception software pipelines, and the computational hardware to execute them. The sensor module includes a mix of established and novel sensor technology, like currently emerging event-based cameras. One advantage of event cameras is low latency in data acquisition, as they sense changes in brightness instead of recording frames. Thereby event-based cameras are independent of frame rates and allow to adapt the data processing rate depending on the current situation of the vehicle.

Another novelty lies within the application of spiking neural networks in addition to classical perception algorithms, which is a further emerging technology that is not yet advanced to the level of classical artificial neural networks. The literature on spiking neural networks is largely dominated by theoretical research, while practical applications only take place on benchmarks like MNIST [31] (classification) or CIFAR [32] (object recognition).

The use of spiking neural networks goes hand in hand with the use of neuromorphic hardware. The combination of both technologies promises a highly energy-efficient form of perceptual data processing that has the potential to consume orders of magnitude less energy than today's perceptual architectures on modern hardware [33].

One of the project goals is to assess the potential of these novel technologies towards a highly integrated, efficient and reliable GLSF. A further interest is understanding the minimum necessary sensor information with different physical principles for L4 GLSF for ODD extension to meet the safety of the intended functionality (SOTIF) requirements, total system power consumption and latency time.

7. Service-oriented intelligent E/E platform

To effectively cope with the escalating complexity of hardware and software in automotive systems, the electrical/electronic (E/E) architecture has undergone a process of centralization [16]. The subsequent phase in this centralization progression is known as the "zonal architecture". Zonal architectures entail a consolidation of data processing, communication, and power transmission within specific zone controllers to minimize cable length and to reduce the ever-growing number of ECUs in the vehicle [17]. However, they introduce challenges like safety concerns, optimal resource allocation, and generic computing capabilities that require dedicated research and innovative solutions. In the AUTOtech.*agil* project, we investigate and resolve these challenges based on the architecture implemented in the UNICAR*agil* project [18].

7.1. The Universal Automotive Bus: A generic data and energy interface

Todays historically evolved, signal-oriented vehicle communication is insufficient to meet modern requirements regarding automated driving functions [51], software updates throughout the entire product life cycle, and the resulting disruptions in E/E and software architectures. As a consequence, three central development trends can be observed: The increasing establishment of Automotive Ethernet as a replacement for established vehicle bus systems such as CAN, LIN, FlexRay and MOST [51], the trend towards zone-orientated architectures to reduce the total number of ECUs [52] as well as the breakdown of the monolithic connection of hardware and application software on ECUs towards a service-orientated architecture [53]. These developments motivate the research project for a Universal Automotive Bus (UAB). The UAB uses the conceptual similarities between future centralized vehicle architectures and established solutions for connecting conventional computer peripherals. The research approach is to combine the beneficial characteristics of Automotive Ethernet [54] and the Universal Serial Bus (USB) [55][56][57] to simplify and universally support vehicle integration of hardware and software across the entire development process. The UAB should contain device definitions for a simple Plug&Play connection of hardware components and standardized tests for system integration. Energy supply and energy management of the connected devices as well as a potential cost reduction due to saved cabling are facilitated by the transmission of energy and data in the same cable. A support of meshed networks to increase redundancy and safety as well as standardized interfaces for telemetry and software updates takes automotive-specific requirements into account.

7.2. Management of software components and computing resources in service-oriented vehicle architectures

In a service-oriented system architecture, it is also imperative that the software components can be managed across all interconnected control units. Existing approaches permitting software updates lack uniformity across all networked control units [58]. For the service-oriented architecture, it is opportune to implement software component management across all networked devices as a service with standardized interfaces. For this purpose, each control unit can execute a local runtime, facilitating the administration of precompiled software components on the respective device. The runtime must enable fundamental management functions, such as creation, loading, initiation, termination, and deletion of software components. Furthermore, it must also allow the isolation of software components from the rest of the system to enhance security and fault tolerance. This also enables the updating of software components throughout the entire vehicle and the dynamic exchange of software components via uniform interface.

In modern vehicles, driving functions are increasingly automated, leading to enhanced computing capabilities of vehicle control units [59]. However, this computational power is not consistently in demand. The surplus computing power can be allocated to external services by equipping the runtime services with externally accessible interfaces. This enables the loading of precompiled software modules from external sources into the vehicle control units and harnessing their computational capabilities. Ensuring safety mandates the isolation of these external software modules from the rest of the system. For the management of external software modules executed on vehicle control units, a centralized administration is necessary. This administration's role is to optimally deploy the software modules into the vehicles. This process necessitates vehicle data, such as latency, available computing power, and memory capacity. Based on these parameters, an optimal distribution is derived, and in the face of changing conditions, currently running software modules are redistributed. The amalgamation of runtime and central administration has the capability to aggregate the vehicle fleet and integrated control units into a computational cluster, whose performance can be harnessed for computationally intensive parallelizable applications.

7.3. Tools and processes for E/E architecture design

The prevailing approach to E/E architecture design usually relies on the experience of engineers. Given the increased complexity of modern vehicular systems and the importance of functional safety, the utilization of appropriate methods and tools becomes imperative. The focus of AUTOtech.*agil* lies on the development of methods to solve two key issues: Firstly, the identification of the optimal number of zones. This involves the strategic positioning of zone controllers within the vehicle based on the efficient allocation of electrical loads [19]. Consequently, the selection of zones is guided by spatial criteria, which leads to a comprehensive consideration of packaging aspects during the design of zonal E/E architectures. Secondly, the optimization of the resource allocation and the architecture topology based on safety-related constraints. Both methods assist E/E architecture developers during the design phase in achieving a reliable and cost-effective solution while reducing development time. To take full advantage of the capabilities offered by this optimized E/E architecture, the error handling needs to be rethought. Most of the fault tolerant E/E architectures currently discussed in industry and research rely on static or reactive fault tolerance approaches. To cope with future requirements regarding flexibility and reliability of cost-intensive hardware, a dynamic approach for proactive fault tolerance in zonal architectures is being developed. The combination of novel fault prediction techniques with systemwide agent-based self-reallocation of services has the potential to tackle the overhead of future zonal architectures.

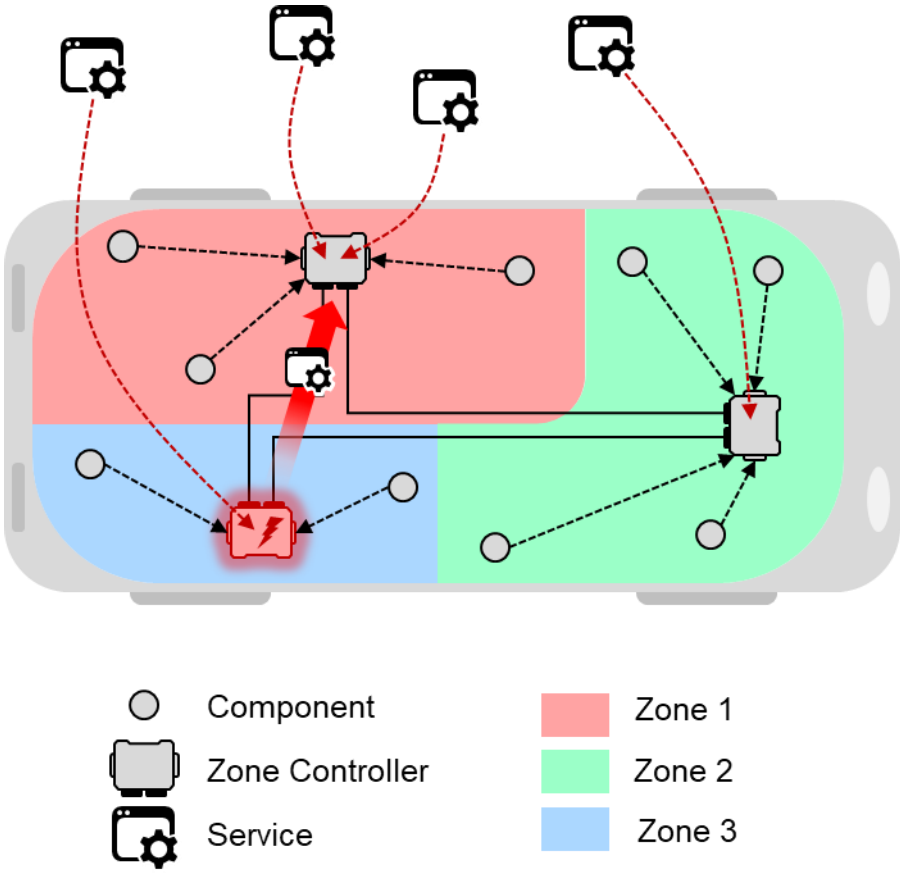

Fig. 13 Methods for design and operation of reliable zonal E/E architectures in AUTOtech.*agil*

7.4. Predictive controllable power electronics using the live digital twin

Power electronics (PEs) are used increasingly in electric vehicles and there is a growing need for them to operate reliably and safely throughout their designed service life. This requirement drives development of more robust control algorithms, along with implementation of intelligent monitoring and improved lifetime models for more accurate estimation of state-of-health (SoH) or Remaining Useful Life (RUL). In AUTOtech.*agil*, a predictive control method is under development for a three-phase inverter, aiming to optimize a chosen target variable, namely the lifetime or efficiency of PE, without negatively affecting the dynamics of the vehicle. Lifetime enhancing control involves prediction and elimination of the damaging temperature swings in PE, thereby reducing thermo-mechanical stress that significantly impacts the operational lifespan of semiconductor components. The future load profile of the PE including expected temperature swings is obtained from the planned trajectory data of the vehicle. Additionally, a live degradable digital twin of the inverter is being constructed, enabling fault diagnostics and fault prediction, as well as virtual testing and simulation of various scenarios for evaluating different control strategies and operating conditions. This digital twin will be developed using real-time measurements of the relevant Temperature Sensitive Electrical Parameters (TSEPs) of semiconductor devices, ensuring its real-time adaptability. The combination of different TSEPs can be used to better correlate them to degradation state and for estimating the junction temperature T_j excluding TSEPs aging effects. Furthermore, as part of the predictive control strategy, a hybrid lifetime model will be developed, integrating both data-driven (TSEPs data) and model-based methods to enhance the accuracy of lifetime estimation and optimize the control strategy.

8. Consistent safety and security concept and algorithms

Safety assurance and approval has proven to be one of the biggest challenges in the pursuit of automated driving. The task of developing a safe and secure automated vehicle gets even more challenging given the fact that in AUTOtech.*agil* a service-oriented architecture is used, that is said to change frequently in order to enable the system to learn and improve throughout the vehicle's life. Moreover, not all services of the driving system are deployed in the vehicle, but also, e.g., in roadside units and control rooms. In AUTOtech.*agil* the impact of these properties is addressed by the development of an overall consistent safety consideration, a modular safety approval based on component properties, a comprehensive security concept and a system monitoring framework.

8.1. Consistent safety consideration

Safety is a desirable emerging property of vehicles equipped with an SAE level 4 automated driving system (ADS). In accordance with relevant automotive safety standards [35, 36], we define safety as the absence of unreasonable risk. In AUTOtech.*agil*, we address challenges of designing an overall architecture as well as software and service architectures that consider safety as a key design aspect. Based on use cases of the transport system, we describe relevant operating conditions as well as an operational design domain (ODD). [37] This allows us to investigate the risk that an ADS poses when operating in the open context. Risk is defined as a function of the probability of occurrence of harm and the severity of harm. A method for managing risk by adapting the target behavior specification of an ADS is presented in [38]. In AUTOtech.*agil*, we aim to extend this approach with a decompositional strategy to represent risk acceptance criteria at design time. A rigorous representation of risks needs to be established. A major challenge lies in the representation and decomposition of non-quantified risks. A suitable method is to be established in order to aggregate the results of scenario-based analyses into a holistic risk assessment that is sufficiently valid for an ODD. At the system level, further aspects such as redundancy, integrity, and uncertainty are to be considered. The holistic safety consideration aims at generating a safety concept that includes operable safety requirements for the design and architecture of the transport system. Furthermore, compatibility with methods for modular validation and safety assurance of artificial intelligence components is to be ensured.

The acceptance of transport systems as developed in AUTOtech.*agil* remains a matter of research. Publications discussing this topic, such as [39], indicate that perceived safety may be an influencing factor on the acceptance of the acceptance object (the transport system, of which is the ADS is a part) by the acceptance subject, such as the general public, vehicle users, and others. Perceived safety depends on the individual definition of safety and the perception of the transport system by these individual stakeholders. Assuming that safety is relevant to acceptance, translation work is needed in order to contribute to the acceptability of the transport system by the stakeholders based on the results of the holistic risk assessment, but also based on other aspects. For this reason, we focus on the stakeholders and how they benefit from operational use cases of the transport system. Finally, a consistent and rigorous safety case will serve as a means to communicate and reason about the achieved level of safety to both internal and external stakeholders. The safety case shall take into account the limited validity of test environments, especially as the project focuses on simulation-based approaches to testing.

A number of stakeholders from industry, science, politics and society have recognized the need to discuss and communicate the issues of safety, risk, and acceptance of CAVs. The Round Table of Autonomous Driving, hosted by the German Federal Ministry for Digital and Transport, provides a platform to seek a common understanding of the open challenges and remaining needs for advancing the field of socially acceptable autonomous mobility.

8.2. Homologation processes and methods for modular validation

Established test and validation processes in the automotive domain substantially rely on large-scale system level release tests to validate safety and performance requirements. This approach not only yields enormous efforts, but also becomes infeasible, when parts of the system such as ML-based services should be updated or even be added to system during the system's life cycle. Thus, we develop test strategies and methods for safe and efficient validation of systems featuring a modular service-oriented architecture with updatable components. We discuss and evaluate methods for system decomposition that allow for shifting test efforts from system level to component level tests. In this context, we especially address the question of safety argumentation for very frequent updates of components. Furthermore, we address the scalable validation of updatable ML- and rule-based components.

The results of the project UNICAR*agil* showed limitations and uncertainties of established knowledge-based decomposition methods. Limitations concern the achievable validity of current test environments, while uncertainties mandatorily arise from a high complexity of highly automated vehicles and their environment. These uncertainties affect the specification of software components and the derived component tests. Untested component behavior can negatively impact the system behavior. In UNICAR*agil* these insufficiencies in the component specification are addressed by a novel interface definition [34]. However, a gap to a complete specification remains due to its knowledge-based approach. Here, we develop data-driven methods to extend the module specification and therefore increase completeness. Furthermore, we use simulation-based approaches to assure system properties on component level.

8.3. Cyber security for Cooperative Intelligent Transport Systems

To ensure the safety of passengers, we continue developing our comprehensive security concept that encompasses the entire vehicle life cycle. This holistic approach enables us to react quickly to a continuously changing attack surface by adjusting risks and security requirements and thus protecting the vehicles against a variety of IT (Information Technology) and OT (Operation Technology) attacks. Our security solution aligns with our project's design philosophy of strongly decoupled and optionally isolated software components, and we aim to minimize management complexity. To this end, we enable developers to specify security requirements interactively and visually in a central tooling, eliminating the need to manually distribute and update security attributes such as keys and certificates for secure communication. For this purpose, we design and specify an Automotive-PKI (Public Key Infrastructure) adapted to the specific requirements of the automotive domain and the vehicles. The processes of certificate issuance, revocation, renewal and validation are transparently linked to the vehicle gateway and the on-board security components for management. The central tooling also provides functionality to annotate in-vehicle security related information (from hosts and network traffic) for direct consideration in the Vehicle Security Monitoring. Furthermore, the system takes responsibility for secure software roll-out and enforces the specified security objectives during runtime, guaranteeing software integrity and communication security.

Our security process is not limited to in-vehicle hardware and software components but also considers the external infrastructure. We specify interfaces allowing for secure interaction with possibly untrusted external components.

8.4. Monitoring framework for holistic self-awareness

Numerous publications discussing the safety of automated vehicles address the necessity of self-awareness [20][21][22] and aspects such as monitoring the operation domain and its limits [23], starting from monitoring simple system errors, such as those arising from sensor and actuator degradation. To enable the system's self-awareness, it is argued in the literature [24] that knowledge about the system itself must be made accessible to the system. This requires a self-representation of the system, encompassing the modeling of its characteristics and properties. It should further include information about the system's architecture, allowing for the explicit mapping of dependencies, such as functional, hardware, or software architecture, e.g. [25]. In this context, we benefit from the well-defined interfaces introduced through the service-oriented architecture from the UNICAR*agil* project.

Lastly, the information inferred from the accumulation of quality information across architectures informs the system with respect to the quality of its own capabilities [26]. Capabilities – often depicted using capability graphs as in [26] – serve as fundamental building blocks for comprehending complex systems, as introduced in [27]. Their concept aims at bridging the gap between desired behavior, requirements and functions in the system’s architecture. Modeling capabilities in an abstract manner promotes technology-neutrality while supporting discussions among various stakeholders [27]. Considering a system’s capabilities and defining requirements accordingly ensures that a system achieves its intended purpose while meeting specified performance criteria.

One objective of this project is to contribute to the overall architecture of automated vehicles and its technical realization by leveraging the quality assessment incorporated, e.g., in the robust probabilistic perception, planning and control approaches to assess the system’s capabilities at runtime. The monitoring framework that we develop in the AUTOtech.*agil* project represents a system-wide monitoring function. Given previous research results from the UNICAR*agil* project, we find that the systematic collection of quality data and the evaluation of the interdependencies within the system are two of the major challenges that we hence aim to address in our work. Our approach involves analyzing requirements and designing a system-wide monitoring framework in the first step – which is not necessarily limited to an automated vehicle such as the vehicle prototypes used in AUTOtech.*agil*. The second step then focuses on developing, integrating, and testing the framework for system-wide monitoring.

9. Acknowledgement

This research is accomplished within the project “AUTOtech.agil” (FKZ 01IS22088). We gratefully acknowledge the financial support for the project by the Federal Ministry of Education and Research of Germany (BMBF).

10. Abbreviations

AD	Automated Driving
ADS	Automated Driving System
AFGBV	Autonome-Fahrzeuge-Genehmigungs-und-Betriebs-Verordnung (Autonomous Vehicle Approval and Operation Ordinance)

AMR	Autonomous Mobile Robot
API	Application Programming Interface
ASAM	Association for Standardization of Automation and Measuring Systems
ASOA	Automotive Service-oriented Software Architecture
AV	Automated Vehicle
AVP	Automated Valet Parking
CAV	Connected Automated Vehicle
CC	Control Center
CR	Control Room
CV	Connected (non-automated) Vehicle
C-ITS	Cooperative Intelligent Transport System
C-VRU	Connected Vulnerable Road User
DDS	Data Distribution Service
ECU	Electronic Control Unit
E/E	Electrical/Electronic
ETSI	European Telecommunications Standards Institute
GLSF	Generalized Low-Speed Function
GNSS	Global Navigation Satellite System
HEC	Holistic Experience Centred
HMI	Human Machine Interface
IT	Information Technology
ITS	Intelligent Transport System
ITS-S	Intelligent Transport System Station
LET	Logical Execution Time
MEC	Multi-access Edge Computing
ML	Machine Learning
ODD	Operational Design Domain
OT	Operation Technology
PE	Power Electronics
PKI	Public Key Infrastructure
REST	Representational State Transfer
ROS	Robot Operating System
RSU	Roadside Unit
RUL	Remaining Useful Life
SAE	Society of Automotive Engineers
SoH	State of Health
SOTIF	Safety Of The Intended Functionality
SOVD	Service-Oriented Vehicle Diagnostics
TSEP	Temperature Sensitive Electrical Parameter
TSN	Time-Sensitive Networking
UAB	Universal Automotive Bus
UAV	Unmanned Aerial Vehicle
UDS	Unified Diagnostic Services
USB	Universal Serial Bus

UX	User Experience
VRU	Vulnerable Road User

11. References

[1] Woopen, Timo; Lampe, Bastian; Böddeker, Torben; Eckstein, Lutz; Kampmann, Alexandru; Alrifaee, Bassam; Kowalewski, Stefan; Moormann, Dieter; Stolte, Torben; Jatzkowski, Inga; Maurer, Markus; Möstl, Mischa; Ernst, Rolf; Ackermann, Stefan; Amersbach, Christian; Leinen, Stefan; Winner, Hermann; Püllen, Dominik; Katzenbeisser, Stefan; Becker, Matthias; Stiller, Christoph; Furmans, Kai; Bengler, Klaus; Diermeyer, Frank; Lienkamp, Markus; Keilhoff, Dan; Reuss, Hans-Christian; Buchholz, Michael; Dietmayer, Klaus; Lategahn, Henning; Siepenkötter, Norbert; Elbs, Martin; v. Hinüber, Edgar; Dupuis, Marius; Hecker, Christian, 2018.
UNICARagil - Disruptive Modular Architectures for Agile, Automated Vehicle Concepts.
In: 27. Aachener Kolloquium Fahrzeug- und Motorentechnik: October 8th - 10th, 2018 - Eurogress Aachen, Germany, pp. 663-694.
ISBN: 978-3-00-057468-9

[2] Statistisches Bundesamt, 2023.
Anzahl der Verunglückten bei Straßenverkehrsunfällen in Deutschland nach Art der Beteiligung von 1980 bis 2022.
Statista GmbH. Online [Access: 10-08-2023] https://de.statista.com/statistik/daten/studie/183/umfrage/verunglueckte-bei-unfaellen-nach-art-der-strassenverkehrsbeteiligung/

[3] Statistisches Bundesamt, 2023.
Anzahl der Getöteten bei Straßenverkehrsunfällen in Deutschland nach Art der Beteiligung von 1991 bis 2022.
Statista GmbH. Online [Access: 10-08-2023] https://de.statista.com/statistik/daten/studie/193074/umfrage/getoetete-bei-unfaellen-nach-art-der-strassenverkehrsbeteiligung/

[4] Böhlen, Boris; Meyer, Oliver; Alrifaee, Bassam; Beerwerth, Julius; Kampmann, Alexandru; Kowalewski, Stefan; Konersmann, Marco; Rumpe, Bernhard; Steinfurth, Felix, 2023.
Software-Defined Vehicle – Herausforderungen in der Diagnose dienstorientierter Fahrzeugarchitekturen.
In: Diagnose in mechatronischen Fahrzeugsystemen XVI.

[5] International Organization for Standardization, Geneva, 2020.
ISO 14229-1: Road vehicles — Unified diagnostic - Part 1: Application Layer.
Online [Access: 01-08-2023]: https://www.iso.org/standard/72439.html

[6] Service-Oriented Vehicle Diagnostics 1.0.0 (SOVD). 2022. Online [Access: 01-08-2023]: https://www.asam.net/standards/detail/sovd/

[7] Buchholz, Michael; Gies, Fabian; Danzer, Andreas; Henning, Matti; Hermann, Charlotte; Herzog, Manuel; Horn, Markus; Schön, Markus; Rexin, Nils; Dietmayer, Klaus; Fernandez, Carlos; Janosovits, Johannes; Kamran, Danial; Kinzig, Christian; Lauer, Martin; Molinos, Eduardo; Stiller, Christoph; Ackermann, Stefan; Homolla, Tobias; Winner, Hermann; Gottschalg, Grischa; Leinen, Stefan; Becker, Matthias; Feiler, Johannes; Hoffmann, Simon; Diermeyer, Frank; Lampe, Bastian; Beemelmanns, Till; Kempen, Raphael; Woopen, Timo; Eckstein, Lutz; Voget, Nicolai; Moormann, Dieter; Jatzkowski, Inga; Stolte, Torben; Maurer, Markus; Graf, Jürgen; v. Hinüber, Edgar; Siepenkötter, Norbert, 2020.
Automation of the UNICARagil vehicles.
In: 29. Aachen Colloquium Sustainable Mobility Band2, p. 1531-1560, Oktober 2020.
DOI: 10.18725/OPARU-34024
Online: https://oparu.uni-ulm.de/xmlui/handle/123456789/34086

[8] Keilhoff, Dan; Niedballa, Dennis; Reuss, Hans-Christian; Buchholz, Michael; Gies, Fabian; Dietmayer, Klaus; Lauer, Martin; Stiller, Christoph; Ackermann, Stefan; Winner, Hermann; Kampmann, Alexandru; Alrifaee, Bassam; Kowalewski, Stefan; Klein, Fabian; Struth, Michael; Woopen, Timo; Eckstein, Lutz, 2020.
UNICAR*agil* - New Architectures for Disruptive Vehicle Concepts.
In: 19th Stuttgart International Symposium - Automotive and Engine Technology, Michael Bargende, Hans-Christian Reuss, Jochen Wiedemann (Eds.), Springer Fachmedien Wiesbaden, 2019.

[9] Henning, Matti; Müller, Johannes; Gies, Fabian; Buchholz, Michael; Dietmayer, Klaus, 2023.
Situation-Aware Environment Perception Using a Multi-Layer Attention Map.
In: IEEE Transactions on Intelligent Vehicles, vol. 8, no. 1, pp. 481-491, Jan. 2023.
DOI: 10.1109/TIV.2022.3164236

[10] Henning, Matti; Buchholz, Michael; Dietmayer, Klaus, 2022.
Situation-Aware Environment Perception for Decentralized Automation Architectures.
In: IEEE Intelligent Vehicles Symposium (IV), Aachen, Germany, 2022, pp. 1087-1092.
DOI: 10.1109/IV51971.2022.9827081

[11] Henning, Matti; Buchholz, Michael; Dietmayer, Klaus, 2023.
The Impact of Frame-Dropping on Performance and Energy Consumption for Multi-Object Tracking.
In: IEEE Intelligent Vehicles Symposium (IV), Anchorage, AK, USA, 2023, pp. 1-6.
DOI: 10.1109/IV55152.2023.10186653

[12] Rittger, Lena; Engelhardt, Doreen; Schwartz, Robert, 2022.
Adaptive User Experience in the Car – Levels of Adaptivity and Adaptive HMI Design.
In: IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 5, pages 4866-4876.
DOI: 10.119/tits.2021.3124990

[13] Kratschmayr, Melissa Lee, 2020.
Entwicklung einer Methode zur Erhebung von kundenseitigen Produktanforderungen angewand im Kontext des autonomen Fahrens.
Dissertation.
DOI: 10.18154/RWTH-2020-07413

[14] Keutgens, Thomas; Klas, Christoph; Zlocki, Adrian; Kwade, Michael; Böttcher, Thomas; Eckstein, Lutz; Strobel, Michael, 2019.
Sound.AI – Teaching Vehicles How to Hear.
In: 28th Aachen Colloquium Automobile and Engine Technology, Aachen, Oktober 2019.

[15] Kwade, Michael; Erraji, Armin; Böttcher, Thomas; Eckstein, Lutz, 2021.
The Hearing Sense of Vehicles – Recognition of Sound Patterns in the Vehicle Environment for Enhanced Situation Awareness.
In: 12. Aachen Acoustics Colloquium, Aachen, November 2021.

[16] Bandur, Victor; Selim, Gehan; Pantelic, Vera; Lawford, Mark, 2021.
Making the Case for Centralized Automotive E/E Architectures.
In: IEEE Transactions on Vehicular Technology 70 (2) (2021) 1230–1245.
DOI: 10.1109/TVT.2021.3054934.

[17] Maul, Mario; Becker, Gerhard; Bernhard, Ulrich, 2018.
Service-oriented EE zone architecture key elements for new market segments.
In: ATZelektronik worldwide 13 (1) (2018) 36-41.
DOI: 10.1007/s38314-017-0092-4

[18] Zaheri, Dorsa; Niedballa, Dennis; Leuffen, Marc; Bilkei-Gorzo, Gergely, 2023.
Practical Implementation of a Zonal E/E Architecture for Autonomous Vehicles in UNICARagil.
In: ATZelectronics worldwide 18 (1-2) (2023) 8–12.
DOI: 10.1007/s38314-023-1436-x

[19] Maier, Jonas; Reuss, Hans-Christian, 2023.
Handling System Complexity in Zonal E/E Architectures.
In: Transportation Engineering, Volume 13, 2023, 100195.
ISSN: 2666-691X, DOI: 10.1016/j.treng.2023.100195

[20] Gregory, Irene M.; Leonard, Charles;Scotti, Stephen J., 2016.
Self-aware vehicles: Mission and performance adaptation to system health.
In: NASA Technical Report 2016.

[21] Lewis, Peter R.; Platzner, Marco; Rinner, Bernhard; Tørresen, Jim; Yao, Xin, 2016.
Self-Aware Computing Systems: An Engineering Approach.
In: Springer International Publishing, 2016.
DOI: 10.1007/978-3-319-39675-0

[22] Carré, Matthieu, 2019.
Autonomic Framework For Safety Management In The Autonomous Vehicle.
Dissertation, Université de Pau et des Pays de l'Adour, 2019.

[23] Colwell, Ian; Phan, Buu; Saleem, Shahwar; Salay, Rick; Czarnecki, Krzysztof, 2018.
An Automated Vehicle Safety Concept Based on Runtime Restriction of the Operational Design Domain.
In: 2018 IEEE Intelligent Vehicles Symposium (I V), Changshu: IEEE, 2018, pp. 1910-1917.

[24] Nolte, Marcus; Jatzkowski, Inga; Ernst, Susanne; Maurer, Markus, 2020.
Supporting Safe Decision Making Through Holistic System-Level Representations & Monitoring -- A Summary and Taxonomy of Self-Representation Concepts for Automated Vehicles.
DOI: 10.48550/arXiv.2007.13807

[25] Bagschik, Gerrit; Nolte, Marcus; Ernst, Susanne; Maurer, Markus, 2018.
A System's Perspective Towards an Architecture Framework for Safe Automated Vehicles, 2018.
In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 2438-2445.

[26] Reschka, Andreas Christian, 2017.
Fertigkeiten- und Fähigkeitengraphen als Grundlage des sicheren Betriebs von automatisierten Fahrzeugen im öffentlichen Straßenverkehr in städtischer Umgebung.
Dissertation, Technische Universität Braunschweig, 2017.

[27] Hofmann, Tamara, 2022.
Capability-based architecture for automated vehicles in urban environment.
Presentation, VM Mid-Term Presentation, 15/16 March, 2022.

[28] Kraftfahrt-Bundesamt, 2022.
KBA erteilt erste Genehmigung für fahrerlos einparkendes Fahrzeug.
Online [Access: 18-08-2023]:
https://www.kba.de/DE/Presse/Pressemitteilungen/Allgemein/2022/pm45_2022_AVP_erste_Genehmigung.html

[29] International Organization for Standardization, Geneva, 2021.
ISO 22737:2021: Intelligent transport systems — Low-speed automated driving (LSAD) systems for predefined routes — Performance requirements, system equirements and performance test procedures.

[30] Ackermann, Stefan; Winner, Hermann, 2020.
Systemarchitektur und Fahrmanöver zum sicheren Anhalten modularer automatisierter Fahrzeuge.
In: 13. Workshop Fahrerassistenz und automatisiertes Fahren - FAS 2020, S. 144-153, Darmstadt, Uni-DAS e. V., 13. Workshop Fahrerassistenzsysteme und automatisiertes Fahren, Walting, Germany, 01. – 03.04.2020.
ISBN: 978-3-941543-47-8

[31] Lecun, Yan; Bottou, Leon; Bengio, Yoshua; Haffner, Patrick, 1998.
Gradient-based learning applied to document recognition.
In: Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.
DOI: 10.1109/5.726791

[32] Krizhevsky, Alex, 2009.
Learning Multiple Layers of Features from Tiny Images.

[33] Davies, Mike; Wild, Andreas; Orchard, Garrick et al., 2021.
Advancing Neuromorphic Computing With Loihi: A Survey of Results and Outlook.
In: Proceedings of the IEEE, vol. 109, no. 5, pp. 911-934, May 2021.
DOI: 10.1109/JPROC.2021.3067593

[34] Klamann, Björn; Winner, Herrmann, 2023.
Introducing the Detailed Sematic Interface Description to Support a Modular Safety Approval of Automated Vehicles - S^2I^2.
In: Safety and Reliability(Accepted).

[35] International Organization for Standardization, Geneva, 2018.
ISO 26262:2028: Road vehicles — Functional Safety.

[36] International Organization for Standardization, Geneva, 2022.
ISO 21448:2022: Road vehicles — Safety of the Intended Functionality.

[37] Gyllenhammar, Magnus et al., 2020.
Towards an operational design domain that supports the safety argumentation of an automated driving system.
In: 10th European Congress on Embedded Real Time Systems, Toulouse, France, 2020.

[38] Salem, Nayel Fabian et al., 2023.
Risk Management Core – Towards an Explicit Representation of Risk in Automated Driving.
DOI: 10.48550/arXiv.2302.07715

[39] Lenz, Barbara; Fraedrich, Eva, 2015.
Gesellschaftliche und individuelle Akzeptanz des autonomen Fahrens.
In: Autonomes Fahren, M. Maurer, J. C. Gerdes, B. Lenz, and H. Winner, (Hrsg.), Berlin, Heidelberg: Springer Berlin Heidelberg, 2015., pp. 639–660.
DOI: 10.1007/978-3-662-45854-9_29.

[40] Kampmann, Alexandru; Alrifaee, Bassam et al., 2019
A Dynamic Service-Oriented Software Architecture for Highly Automated Vehicles.
In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC).
DOI: 10.1109/ITSC.2019.8916841

[41] Lampe, Bastian; van Kempen, Raphael; Woopen, Timo; Kampmann, Alexandru; Alrifaee, Bassam; Eckstein, Lutz, 2020.
Reducing Uncertainty by Fusing Dynamic Occupancy Grid Maps in a Cloud-based Collective Environment Model.
In: 2020 IEEE Intelligent Vehicles Symposium (IV).
DOI: 10.1109/IV47402.2020.9304816

[42] Lampe, Bastian; Woopen, Timo; Eckstein, Lutz, 2019.
Collective Driving: Cloud Services for Automated Vehicles in UNICARagil.
In: 28. Aachen Colloquium Automobile and Engine Technology.
DOI: 10.18154/RWTH-2019-10061

[43] Lampe, Bastian; Reiher, Lennart; Zanger, Lukas; Woopen, Timo; van Kempen, Raphael; Eckstein, Lutz, 2023.
RobotKube: Orchestrating Large-Scale Cooperative Multi-Robot Systems with Kubernetes and ROS.
DOI: 10.48550/arXiv.2308.07053

[44] Cloud Native Computing Foundation, 2018.
CNCF Cloud Native Definition.
Online [Access: 30-08-2023]:
https://github.com/cncf/toc/blob/937e2a808c4b3243974b969668e4a1129f71d923/DEFINITION.md

[45] Lampe, Bastian; Reiher, Lennart; Woopen, Timo; Eckstein, Lutz, 2022.
Cloud Intelligence and Collective Learning for Automated and Connected Driving.
In: ATZelectronics worldwide 17.
DOI: 10.1007/s38314-022-1405-9

[46] Kampmann, Alexandru; Lüer, Maximilian; Kowalewski, Stefan; Alrifaee, Bassam 2023.
Optimization-based Resource Allocation for an Automotive Service-oriented Software Architecture.
In: 2022 IEEE Intelligent Vehicles Symposium (IV), Aachen, Germany, 2022, pp. 678-687
DOI: 10.1109/IV51971.2022.9827429

[47] ETSI, 2014.
Final draft ETSI EN 302 637-2 V1.3.1 (2014-09); Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service.
Online [Access: 04-09-2023]: https://www.etsi.org/deliver/etsi_en/302600_302699/30263702/01.03.01_30/en_30263702v010301v.pdf

[48] Busch, Jean-Pierre; Reiher, Lennart; Eckstein, Lutz, 2023.
Enabling the Deployment of Any-Scale Robotic Applications in Microservice-Based Service-Oriented Architectures through Automated Containerization.
DOI: 10.48550/arXiv.2309.06611

[49] Nicholson, Jeanine; Luttropp, Darius; Jones, Nicole; Friedlander, Julia, 2023.
CPUC Status Conference: Safety Issues Regarding Driverless AV Interactions with First Responders.
Online [Access: 06-09-2023]: https://www.sfmta.com/sites/default/files/reports-and-documents/2023/08/2023.08.07_cpuc_status_conference_8.7.2023_final.pdf

[50] Feiler, Johannes; Hoffmann, Simon; Diermeyer, Frank, 2020.
Concept of a Control Center for an Automated Vehicle Fleet.
In: IEEE 23rd International Conderence on Intelligent Transportation Systems (ITSC), 2020.
DOI: 10.1109/ITSC45102.2020.9294411

[51] Hank, Peter; Müller, Steffen; Vermesan, Ovidiu; Van Den Keybus, Jeroen, 2013.
Automotive ethernet: in-vehicle networking and smart mobility.
In: IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE), 2013, pp. 1735-1739.

[52] Alparslan, Onur; Arakawa, Shin'ichi; Murata, Masayuki, 2021.
Next generation intra-vehicle backbone network architectures.
In: IEEE 22nd International Conference on High Performance Switching and Routing (HPSR), 2021, pp. 1-7.

[53] Rumez, Marcel; Grimm, Daniel; Kriesten, Reiner; Sax, Eric, 2020.
An overview of automotive service-oriented architectures and implications for security countermeasures.
In: IEEE access 8, 2020, pp. 221852—221870.

[54] Open Alliance, 2023.
Automotive Ethernet Specifications.
Online [Access: 13-09-2023]: https://opensig.org/Automotive-Ethernet-Specifications/

[55] USB Implementers Forum, 2023.
USB2.0 Specification.
Online [Access: 13-09-2023]: https://www.usb.org/document-library/usb-20-specification

[56] USB Implementers Forum, 2023.
USB3.2 Rev. 1.1 Specification.
Online [Access: 13-09-2023]: https://www.usb.org/document-library/usb-32-revision-11-june-2022

[57] USB Implementers Forum, 2023.
USB4 Specification.
Online [Access: 13-09-2023]: https://www.usb.org/document-library/usb4r-specification-v20

[58] Pischinger, Stefan; Seiffert, Ulrich, 2016.
Vieweg Handbuch Kraftfahrzeugtechnik.
In: ATZ/MTZ-Fachbuch. Wiesbaden:Springer Vieweg, 2016.
DOI: 10.1007/978-3-658-09528-4

[59] Brunner, Stefan; Roder, Jurgen; Kucera, Markus; Waas, Thomas, 2017.
Automotive E/E-architecture enhancements by usage of ethernet TSN.
In: 2017 13th Workshop on Intelligent Solutions in Embedded Systems (WISES)
DOI: 10.1109/WISES.2017.7986925

[60] AUTOSAR. AUTOSAR - Specification of Timing Extensions.
Standard. AUTOSAR GbR, Nov. 2022.
Online [Access: 13-09-2023]: https://www.autosar.org/fileadmin/standards/R22-11/CP/AUTOSAR_TPS_TimingExtensions.pdf

[61] Henzinger, Thomas A.; Horowitz, Benjamin; Kirsch, Christoph Meyer, 2001.
Giotto: A time-triggered language for embedded programming.
In: Embedded Software: First International Workshop, EMSOFT 2001 Tahoe City, CA, USA, October 8–10, 2001 Proceedings 1. Springer Berlin Heidelberg, 2001.

[62] Kirsch, Christoph M.; Sokolova, Ana, 2012.
The logical execution time paradigm.
In: *Advances in Real-Time Systems*, pp. 103-120, 2012.

[63] Hennig, Julien et al., 2016.
Towards parallelizing legacy embedded control software using the LET programming paradigm.
In: 2016 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), IEEE Computer Soc., 2016.

[64] Gemlau, Kai-Björn, et al., 2021.
System-level logical execution time: Augmenting the logical execution time paradigm for distributed real-time automotive software.
In: *ACM Transactions on Cyber-Physical Systems* 5.2, pp. 1-27, 2021.

[65] Schräder, Tobias et al., 2021.
Development of an Autonomous Family Vehicle using a Scenario-Based Design Approach.
In: 30. Aachen Colloquium Sustainable Mobility, 2021.
DOI: 10.24355/dbbs.084-202110271543-0

Imprint

Publisher

Institute for Automotive Engineering (ika), RWTH Aachen University
Univ.-Prof. Dr.-Ing. Lutz Eckstein Steinbachstraße 7, 52074 Aachen Phone: +49 241 80-25600
Fax: +49 241 80-22147
E-Mail: office@ika.rwth-aachen.de

Chair of Thermodynamics of Mobile Energy Conversion Systems (TME), RWTH Aachen University
Univ.-Prof. Dr.-Ing. Stefan Pischinger Forckenbeckstraße 4, 52074 Aachen Telefon: +49 241 80-48000
Telefax: +49 241 80-22995
E-Mail: office@tme.rwth-aachen.de

Distribution

Aachener Kolloquium Fahrzeug- und Motorentechnik GbR Steinbachstraße 7, 52074 Aachen
Phone: +49 241 88-61251 Fax: +49 241 88-61266
E-Mail: info@aachen-colloquium.com

1st Edition October 2023
www.aachen-colloquium.com

ISBN 978-3-00-076758-6