# Collective Driving - Cloud Services for Automated Vehicles in UNICARagil

Bastian **Lampe**, M.Sc., Timo **Woopen**, M.Sc., Prof. Dr.-Ing. Lutz **Eckstein**
Institute for Automotive Engineering, RWTH Aachen University, Aachen, Germany

**Summary**

Future Cooperative Intelligent Transport Systems (C-ITS) require an integrated functional framework that provides cloud-based services to automated vehicles and other traffic participants. The goal is to process, store and share relevant information in order to continually assure and improve the efficiency, safety and comfort of the C-ITS. This paper introduces a first conceptual hypothesis for such a framework that is developed in the project UNICARagil, funded by the German Federal Ministry for Education and Research (BMBF). Three main components of this framework, the Collective Environment Model, the Collective Memory and the Collective Behavior, are presented. Open challenges associated with current and future technology are discussed.

## 1 Introduction

The project UNICARagil conducts research on modular hardware and software architectures for agile automated vehicles with the aim of developing four use-case specific driverless and fully automated prototypes. In the project, the focus lies on the topics of **automation**, **safety**, **security**, **verification & validation**, and **modularization** [1]. Since future automated vehicles will be part of an integrated C-ITS containing various connected traffic participants, one aspect of the project is the development of a cloud-based functional framework that provides various valuable services to traffic participants. The developed framework is designed to support traffic participants in their goal to accurately perceive their environment and to plan safe, efficient and comfortable behavior. It allows traffic participants to receive an estimate of the accuracy of their perception algorithms by evaluating the discrepancy to the environment representations of other traffic participants. Mechanisms are developed that gradually improve perception and planning algorithms by continually learning from data gathered in the field.

The paper is structured in the following way. First, an overview on the functional architecture for the vehicle prototypes developed in UNICARagil is given. Then, the extension of the in-vehicle architecture with the corresponding cloud-based functions is presented. In the next sections, the potential of cloud-based services is derived from the limitations of vehicles that either allow no connectivity or use state-of-the-art Vehicle-to-Vehicle (V2V) technology. In the end, three major components of the functional framework developed in UNICARagil are presented and open challenges are discussed.

## 2 Architectures in UNICARagil

This section presents the functional system architecture developed in the UNICARagil project, into which the cloud concepts introduced in this paper are integrated. Additionally, the service-oriented software architecture, used for the implementation of the in-vehicle software and adapted for the cloud software, is featured.

Fig. 1 shows the schematic overview of the project's overall concept introduced in [1]. It describes a possible configuration of elements in a future C-ITS. For each of these elements, concepts are developed in UNICARagil. There are different vehicle-specific concepts like the dynamic modules, the modular platform and the sensor modules. Additionally, there are infrastructure-based concepts such as the control room, the "Info Bee" and the cloud. Together, they form the integrated C-ITS concept developed in UNICARagil.

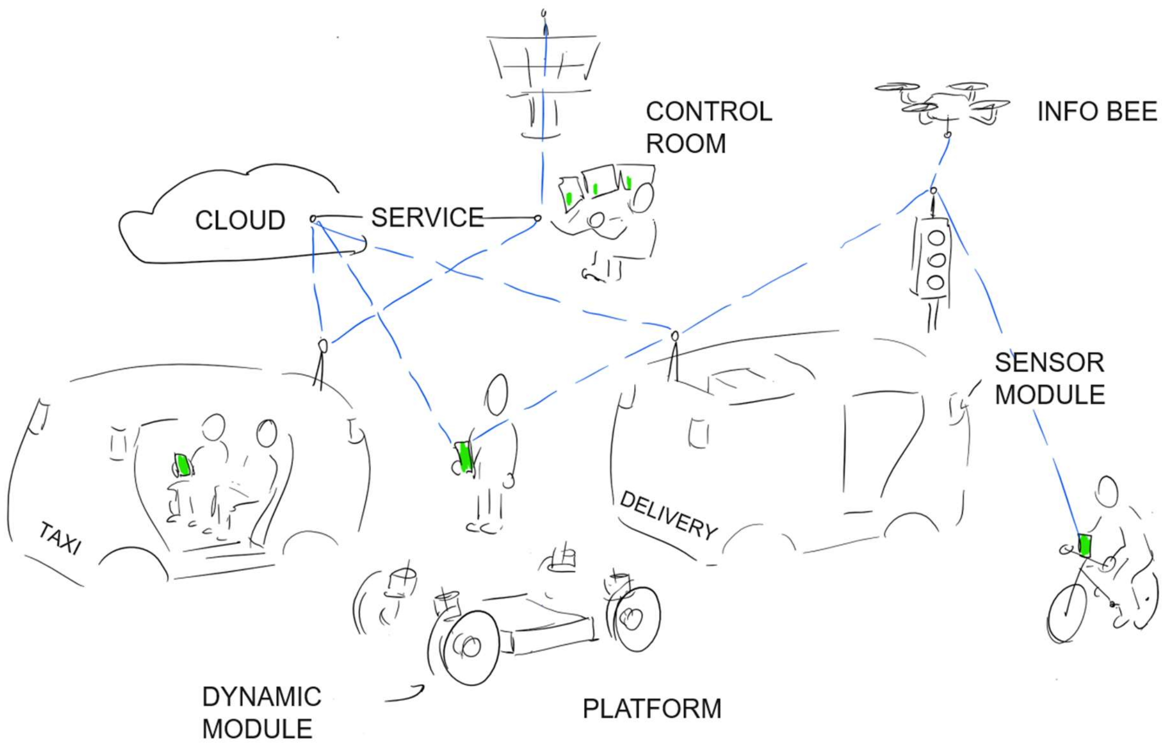


Fig. 1: Sketch of the overall system developed in UNICARagil [1]

The functional architecture represents the necessary functionalities for the realization of the fully automated and driverless operation of the vehicle prototypes. As presented in [1], the "A-Model" outlines the functional architecture of the vehicle itself. Therein, all vehicle specific functions are arranged, from the sensor input as data source to the actuators, which execute the driving tasks. The model is extended by the corresponding cloud functionalities matching the vehicles' functions on the left side of Fig. 2.

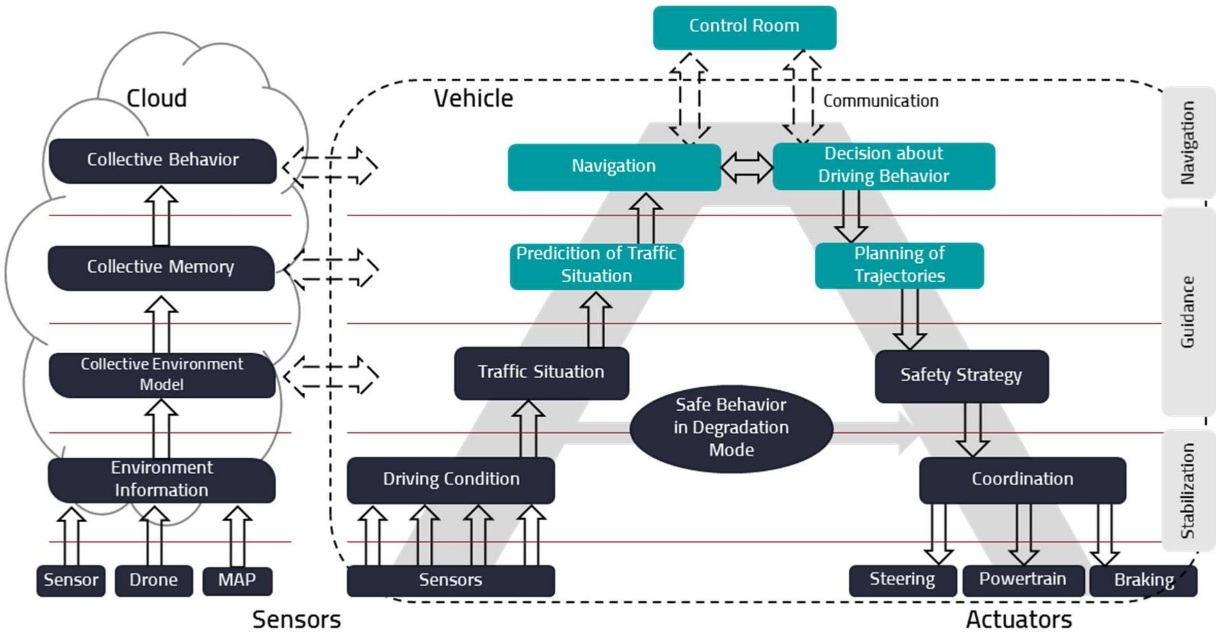


Fig. 2: A-Model for the functional architecture derived from [1]

The A-Model describes the flow of information between different abstract functions of the modular automated driving function. The left side of the underlying "A" delineates the perception functions. Starting at the different environment sensors, whereof LiDAR, Camera, Supersonic and Radar are used in UNICARagil, the information is processed in different layers. It starts with the perception of a vehicle's own driving condition. Then, the traffic situation is analyzed and predicted into the future. This forms the basis for the in-vehicle environment model. Based on the predicted traffic situation and the navigation input, the decision about the driving behavior is positioned on the topmost layer of the right side of the A-Model. From that point, the downwards facing direction of the information flow can be characterized by the classic levels of the driving task according to Donges [2]. On the guidance level, the trajectories are planned and the safety strategy is applied to the planned trajectories. The last abstract function, before the information flow reaches the actuators, is the coordination. Here, the safely planned trajectories are used as input for the different controllers for the connected actuators. The horizontal connection between the guidance and stabilization layer is one of the characteristic A-Model features. This shortcut in the information flow enables the architecture to ensure safe behavior in case of latencies or malfunctions in one of the upper functionalities - similar to a reflex of a human being. Hence, the vehicle itself is also able to behave in a safe way under various degradation modes.

The model described above includes the necessary functions for an automated and driverless vehicle. With the growing number of automated vehicles developed, the usage of external entities that support the automated vehicles in their driving task becomes more relevant, because external entities can provide various valuable services. The arising potential is described in section 5. In UNICARagil, the vehicles are designed such that they may operate automated and driverless without the additional cloud functionality. The external functions support the vehicles in various aspects that are described in section 6.

The cloud functionalities, shown on the left of Fig. 2 enable the vehicle fleet to not only enrich the perception data processed inside the vehicle with external data from different vehicles or other sensors, but also to support the behavior and trajectory planning of the vehicle. The cloud architecture is compatible to the information processing levels of the A-Model such that the different collective functions match the perception and planning functions inside the vehicle.

The so-called Automotive Service-Oriented Software Architecture (ASOA) takes a key role in the later implementation of the introduced functional architecture. It allows for updateability and extensibility of the system. For this purpose, functional modules divide into smaller, loosely coupled services. Each service represents the smallest, not further dividable entity of the necessary software feature. A service transmits and receives the payload data together with its quality data and meta information as shown in Fig. 3.

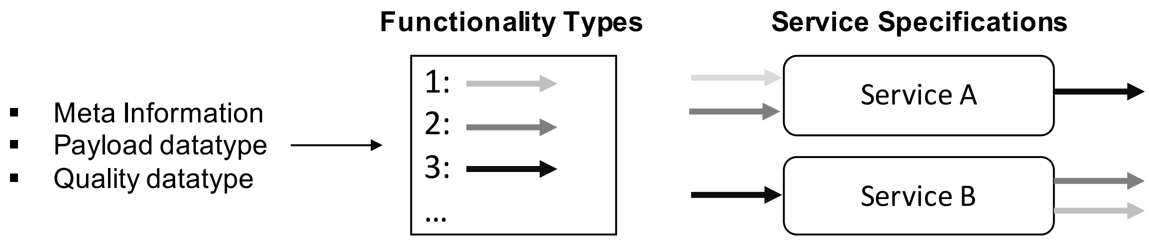


Fig. 3: Unambiguous Formulation of a Service [1]

In comparison to today's systems, which are integrated at design time and thus hard to update or upgrade, the ASOA service composition is integrated at runtime. This allows for interchangeability of the software components even at runtime, e.g. if a new service is able to deliver the same information with a better quality. Thus, a service is formulated in an appropriate, machine understandable framework.

The UNICARagil architecture also incorporates external service providing systems like the Collective Environment Model (CEM), the Collective Memory (CM) and the Collective Behavior (CB) presented in this paper. They are adapted to the ASOA framework in order to be compatible to corresponding in-vehicle services. To introduce the motivation behind the envisioned UNICARagil cloud-based framework, the next sections outline possible limitations of vehicles that either allow no connectivity or are connected using state-of-the-art technology. Afterwards, the potential of the concepts developed in UNICARagil is derived from these limitations.

## 3 Limitation of vehicles with no connectivity

In part, the potential of connected vehicles and the cloud-based framework developed in UNICARagil arises from the limitations of vehicles that do not have the capability to connect to other vehicles or cloud-based systems. This section outlines the restrictions of vehicles with no connectivity and the emerging potential of connected vehicles.

One limitation of vehicles that comprise no connectivity arises from their inability to share data. Modern automated systems often rely on learning from data because processing rules are too complex to directly formulate them. Learning itself does not happen in the vehicle but data from vehicles is needed to enrich the datasets for the offboard learning algorithms. An increase in the amount of data used for training usually leads to better performing models. If only data from test vehicles was used to train the machine learning based algorithms, data of critical situations in the field may be lost due to the limited number of test vehicles and thus limited number of recorded scenarios.

With multiple cameras and LiDAR sensors installed in automated vehicles, data rates increase to hundreds of megabytes per second or even higher depending on the sensor setup. It is not suitable to store this amount of data in every single vehicle, so most of the data would be lost. In contrast to that, cloud servers can be equipped with vast room for data storage. They can continually collect a large portion of the data from connected vehicles over a long period of time. This immensely increases the amount of data that can then be used by the learning algorithms.

One often neglected aspect of automated driving is its impact on the range of the automated vehicle. Electrification is an ongoing development in today's and the future automotive industry. The limited range of vehicles using state-of-the-art battery technology is one of the major challenges of this process. The ongoing trend of automation can lead to an even higher demand for electrical power because of the additionally needed computational resources and can thus limit the range even further. According to [14], computing consumes the most energy of the additional technology in connected and automated vehicles. Many research projects and manufacturers aim at limiting the power consumption of the necessary computing devices by developing more efficient technology. Nevertheless, computational requirements of new algorithms used in automated driving will presumably keep rising, which can result in an increasing energy demand and more dissipated heat. In contrast, cloud services run on permanently running servers with relatively low limitations to their power consumption. Although in the project UNICARagil, cloud-services only support the vehicle functions, the goal to reduce the energy consumption of vehicles represents one reason why it might become reasonable to at least partly substitute vehicle functions with cloud-services in the future.

With the rapid development in automated driving functions, software and even hardware updatability and upgradability become a critical limitation of many vehicles. Vehicles whose software cannot be updated and upgraded over-the-air soon become outdated. Since there exits rapid progress on more efficient computers and control units for automated vehicle as well, hardware can become outdated as well, because it will at some point no longer be capable to sufficiently run the latest software. If this software provides additional safety, it might become necessary to replace the vehicle's hardware or buy a new vehicle. Vehicles whose functionality can be substituted by cloud-based functions cannot become outdated as fast if the cloud keeps up with state-of-the-art technology.

Most of the current automated vehicles are equipped with various different sensors for environment perception. Mostly supersonic, radar and camera sensors are used. First manufacturers already use LiDAR sensors as well. Regarding fully automated driving, the number of sensors needed for sufficient environment perception presumably remains fairly high. As an example, the UNICARagil self-driving prototypes feature over 16 cameras, 4 LiDAR sensors and 10 radars. However, even with these many sensors, the environment perception is subject to occlusions and to the degradation of the resolution and accuracy of sensors with a growing distance. Cooperative cloud-based sensing can combine the sensor data of multiple vehicles and therefore reduce the occluded area and increase the accuracy and range of individual vehicles' perception.

The usage of different sensor technology in current vehicles introduces redundancies within the limits of the used sensors and computing setup. However, degradations of all sensors or other common cause errors are not always detectable within the system. Cloud-based services can detect degradation modes in the vehicles' environment perception by comparing environment representations of multiple vehicles and of infrastructure-based sensors and thus add additional redundancies.

All of the aforementioned limitations may affect the traffic efficiency, energy efficiency, comfort, resource consumption and costs because the overall system's safety level must not be compromised. If the same level of safety is to be achieved, more restricted behavior becomes necessary. This may for example lead to scenarios where

- no overtaking is possible, because the state of the left lane is uncertain,
- early braking before an intersection is necessary, because a lane's occupancy status is not known,
- a bad choice for the turn lane is made, because a blocking vehicle on one of the lanes behind an intersection cannot be considered,
- vehicles may not react to current changes in traffic system (avoid construction, pot holes, traffic jams, …).

Additionally, it might become necessary that

- vehicles need to be regularly checked for proper functioning because no mutual verification exists,
- people need to buy the latest vehicle model in order to be able to use the latest hardware and software,
- manufacturers and suppliers can only make limited use of data gathered in the field to improve hardware and software.

As a result, it is possible that

- uncomfortable and energy inefficient accelerations become necessary,
- time inefficient routes are chosen due to uncertain information on potentially blocked lanes,

- driving comfort is negatively affected due to necessary trajectory updates correcting earlier misjudgments,
- data needs to be acquired in costly field operational tests,
- limitations in software development come up due to the limited availability of data from the field (data of testing fleet might not be representative of data in the field),
- consumption of natural resources increases, because hardware becomes outdated and needs to be replaced more often,
- higher costs arise from upgrading vehicles or hardware components.

Many of the aforementioned limitations can be addressed by enabling vehicles to connect to each other and to suitable cloud-services as described in section 6. The following sections describe the state-of-the-art Vehicle-to-everything (V2X) technologies and outlines the still prevailing limitations.

## 4 Limitations of connected vehicles using state of the art technology

Connected and automated vehicles go together in most statements regarding automated driving in the last years. There are different standardization organizations working on communication standards worldwide. In the U.S., the standardized communication messages are defined in [3]. In Europe, the European Telecommunications Standards Institute (ETSI) defines different specifications on V2X solutions. There are predefined message types that are to be used for connected vehicles. This paper focuses on the European standardizations, since UNICARagil is a publicly funded project in Germany. Nevertheless, both the U.S. and the European V2X standards are driven by the usage of the technology standardized in IEEE 802.11. The amendment IEEE 802.11p describes a specialized wireless LAN band for automated vehicles' communication [4]. It defines protocols for the physical layer and parts of the data link layer in the OSI model [5]. Technology specified to these standards is called ITS-G5 in Europe and WAVE in the U.S. [6]. However, with the further development of cellular communication technologies, a new competitor to the automotive WLAN arose in the last years.

There are multiple technological differences between solutions using WAVE/ITS-G5 and solutions using cellular networks. The former use vehicular ad hoc networks (VaNET). Here, vehicles directly communicate with each other or with roadside communications units (RSUs), respectively. In cellular networks, the data traffic always passes a base station, called evolved Node-B (eNB). From there, it may proceed to the internet, where it can be processed and then transmitted to its intended recipient. Mobile edge computing aims at reducing latency by placing the processing server close to the eNB. Future LTE-V2X and 5G Side-link aim at the realization of direct communication without the necessary usage of the cellular network.

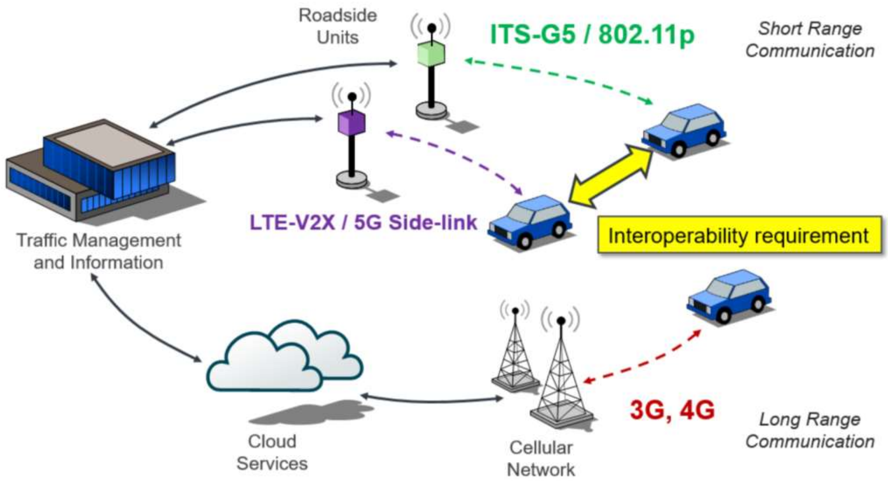


Fig. 4: Visualization of the communication differences of 802.11p and cellular networking [8][7]

In general, V2X communication can be divided into time critical messaging that affects current behavior and non-time critical messaging that affects behavior in a more distant future. Whatever communication technology is used, the latency for time critical information should be as low as possible [8]. Therefore, the ITS-G5 band uses a reduced bandwidth in order to reduce latency and increase robustness. As a result, the defined message types do not extend a certain message size.

With the prevailing message types, vehicles share their own position and vehicle data, report critical traffic situations and receive infrastructure information like traffic light signals or intersection geometries. By the date of this paper, there is no final standardized message type that can be used to share information on the vehicles' environment. However, ETSI is currently working on the standardization of these messages.

With the current state-of-the-art communication protocols and technologies, some of the limitations mentioned in section 3 can already be addressed. Nevertheless, other limitations persist, e.g. due to insufficiently sharing data of the environment representations of vehicles.

In order to address the remaining limitations, the development of communication technologies that are capable of delivering large throughput at low latencies is crucial. Only with these capabilities, the cloud framework presented in this paper can make full use of its potential. First, we take a look at existing technologies. Their shortcomings add to the motivation behind the research conducted in UNICARagil.

Current mechanisms that share data between different vehicles are realized by the usage of VaNETs. These mainly act as a way to exchange already processed data. There is no central cloud server dedicated to processing the transmitted data. Well known applications for the usage of the direct V2X communication are the ETSI standardized messages:

- Cooperative Awareness Message (CAM),
- Decentralized Environmental Notification Messages (DENM),
- Map Data Messages (MAP),
- Signal Phase and Timing Messages (SPaT).

These messages rely on vehicular data or infrastructure data, which is transmitted by an on-board unit (OBU) or roadside unit (RSU), respectively.

CAM messages are used to provide high frequency updates of vehicles' data, like position, speed and more [9]. Within the range of a vehicle transmitting its CAM, other vehicles receive the data and can react to it individually. In a scenario where not all traffic participants are connected, critical information can get lost.

DENM messages are event triggered and communicate hazard cases [10]. With the usage of VaNET communication, these messages are usually transmitted from one detecting vehicle to many different other vehicles in the affected area by multi hop principles. A key challenge is to identify the situation and communicate the necessary data. Yet, there is not always enough information for other vehicles to best deal with this situation because all information may be provided by a single vehicle that may not be able to communicate the necessary details of a situation, either because it did not perceive them or because existing message formats do not provide the necessary means to share them.

MAP and SPaT messages are usually generated by infrastructure based RSUs [11][12]. They report the infrastructure topology and communicate the traffic light status. These messages are always location dependent. With VaNETs, they are communicated to the relevant vehicles that are near the RSU.

A Local Dynamic Map (LDM) is a conceptual data store which is embedded in an ITS station containing topographical, positional and status information within a dedicated geographic area of interest [13]. All data in an LDM is structured into four layers of information, each representing a different degree of dynamics of the data. The LDM can be accessed through the transmission of the various ETSI messages to the corresponding ITS station.

In general, the technologies presented above share the disadvantage that very limited data is transmitted between different traffic participants and the data processing is still done by the vehicles itself. It is not assured that all traffic participants receive all potentially useful data for their current scenario.

## 5 Potential and challenges of connected vehicles in UNICARagil

With the introduction of the cloud-based framework in UNICARagil, we address the issues described in the previous sections. In general, the cloud-based services enable collective processing and storage of large amounts of data from which new insights can be gained. In particular, the cloud system can address the following use cases:

- Collective environment perception with a large number of traffic participants,
- Mutual verification of environment perception of traffic participants,
- Learning from large amounts of data gathered by the cloud-based system,
- Collective route planning and trajectory optimization.

The potential of such a system comes with a number of challenges that need to be addressed. The main challenge for any safety critical real-time cloud-based application is the needed ultra-reliable and low-latency communication (URLLC) to the cloud. There are multiple competing objectives for which a trade-off has to be made. A prominent one is the choice of the communication technology. V2V via ITS-G5 allows low latency communication but there is no central computing entity that may process and store data. There are various cellular solutions that compete with ITS-G5 as described in section 4. Cloud computing allows central data processing and thus a potentially better system performance but comes with higher latencies. Multi-access edge computing (MEC) allows for lower latencies than centralized cloud computing but lacks the central computing unit to combine data from different MEC cells and is therefore only locally usable. Fog computing is a combination of MEC and a centralized approach. It provides a central data processing and data storage backend. It additionally uses servers at the edge of the network for time critical functionalities. The benefits come at the price of a higher system complexity [15]. Communication via ITS-G5 is also compatible to the fog-computing approach but relies on a widely developed infrastructure of RSUs connected to the central server. Depending on an application's requirements regarding latency and reliability, the suitable communication technology needs to be selected. There are additional challenges to be addressed in the UNICARagil cloud concept:

- data selection and reduction,
- data synchronization,
- environment prediction.

By selecting only relevant data, the necessary bandwidth and introduced latency may be reduced. The synchronization of data of multiple traffic participants in the cloud supports the combination of the data. The prediction of the future state of the environment may mitigate the negative effects of latency introduced by the cloud. The cloud concept proposed in UNICARagil provides mechanisms that deal with these challenges. The three modules called Collective Environment Model, the Collective Memory and the Collective Planning are proposed as possible solutions. The functional concept developed in UNICARagil is in principle technology agnostic as long as there is an external processing unit, be it a centralized cloud server or multiple MEC units with a connected central server backend. The next section describes the functional architecture of the UNICARagil cloud concepts and its advantages.

## 6 Cloud in UNICARagil

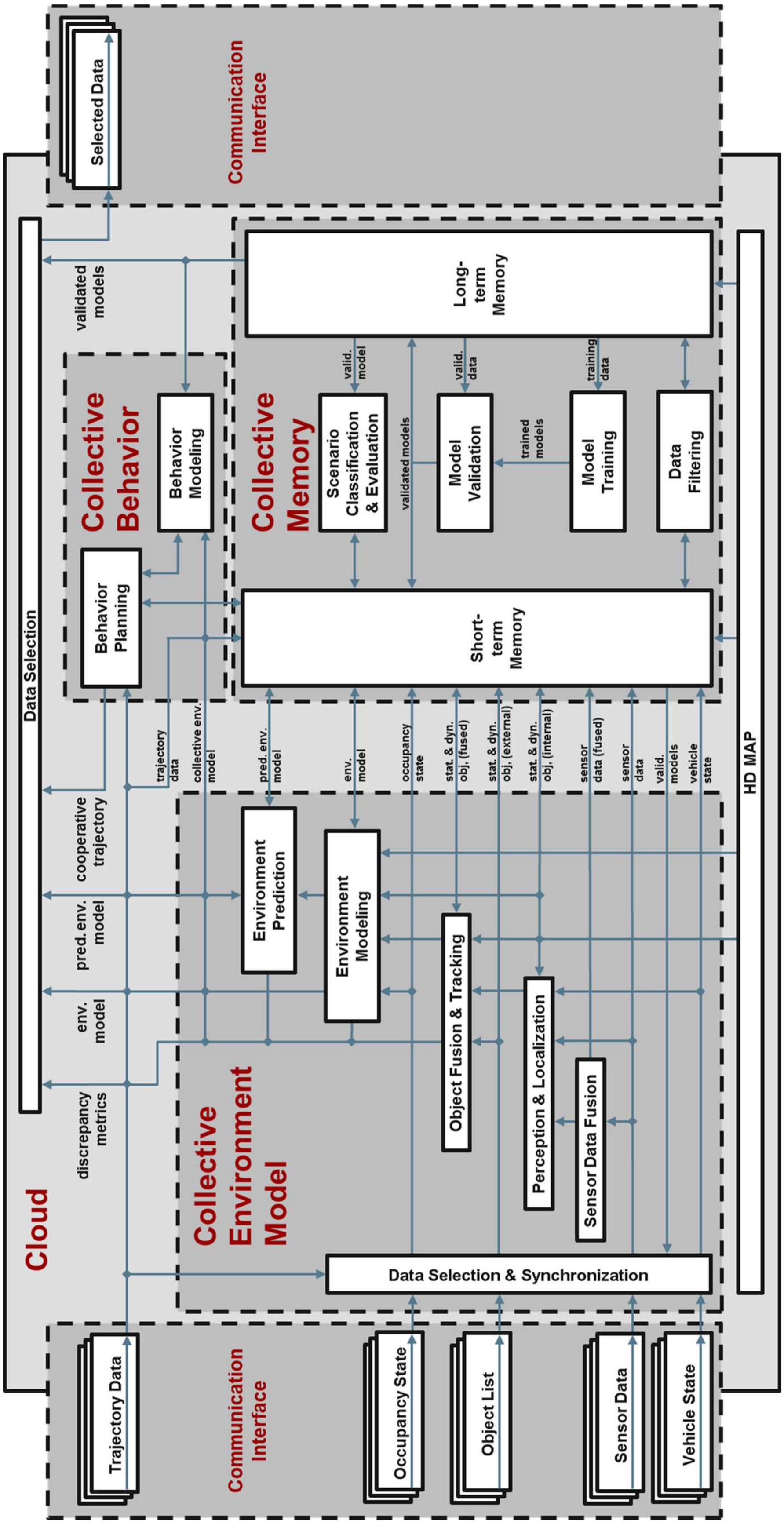


Fig. 5: Functional architecture of the UNICARagil cloud framework

The functional architecture depicted in Fig. 5 contains three major modules of the envisioned cloud-based system in UNICARagil. The complete system is comprised of more subsystems than presented in this paper.

The Collective Environment Model, the Collective Memory and the Collective Behavior Planning are introduced. Together, they aim at overcoming the limitations of vehicles with no connectivity or with connectivity through the most common state-of-the-art V2X technology. In the following, each module of the functional framework depicted in Fig. 5 is described.

## 6.1 Collective Environment Model

The Collective Environment Model (CEM) aims at combining the environment models and when possible the sensor data of multiple vehicles that are located in the vicinity of each other. It then provides the computed collective environment model as a service to traffic participants such that

1. the accuracy of the computed collective environment model exceeds the accuracy of the individual vehicle environment models in order to further reduce the risk associated with driving in a partially observable environment;
2. the range of the collective environment model exceeds the range of the individual vehicle environment models such that behavior planning can take into account more information in order to increase efficiency and comfort;
3. occlusions are minimized compared to the individual vehicle environment models, further enabling connected vehicles to increase efficiency and comfort;
4. vehicles get an estimate of the accuracy of the output of their individual environment modeling algorithms based on the discrepancy between overlapping environment models of multiple traffic participants.

There exist additional interfaces to the Collective Memory and the Collective Behavior which are described in the respective subsections 6.2 and 6.3.

### 6.1.1 External data interfaces of the CEM

Any cloud-based system interacting with multiple vehicles needs to be compatible with the various possible inputs and outputs of the data processing in the vehicle. For the CEM, this means that it needs to take the various forms of representation of the vehicles' environments as input and provide either the same representation or another one that is compatible with further processing steps in the vehicle such as behavior planning.

The vehicle environment models in UNICARagil are comprised of three different representations of the environment. The occupancy state of the vehicles' environment is encoded in a Dynamic Occupancy Grid Map (DOGMa) and a representation of the drivable (free) space. Additionally, object lists are computed that contain all dynamic objects in the vicinity of a vehicle.

### 6.1.2 The information-latency trade-off

Each data processing step in the vehicle can be thought of as a data and information reducing abstraction of input data to a useful output. The sensor raw data containing millions of camera image pixels and hundreds of thousands of radar and LiDAR reflections is in the end reduced to a steering angle and a throttle value. The useful information is "bought" with processing power, time and information loss / data reduction. It is usually impossible to reproduce the input from the output.

This poses a challenge to any cloud-based system that integrates data from multiple sources. For some functionalities of real time application such as automated driving, low latencies are required in order to function safely. On the other hand, sufficient environment information is also required in order to produce an environment model that is sufficient for planning safe behavior in a partially observable environment. In-vehicle processing already poses challenges to the compliance with the real-time requirements. When sending data over a wireless network, this challenge gets even more difficult.

Current technology such as the ETSI Messages introduced in section 4 choose an environment representation that is small in size and therefore requires only little bandwidth and can be sent at low latencies. This comes at the price of information loss in the data.

Object lists usually contain only those objects perceived by a vehicle that the vehicle's perception module is sufficiently confident about with regards to their existence probability. It is not possible to combine multiple uncertain detections from multiple vehicles into one more confident detection that results in a new object. It is also not possible to combine sensor raw data such that new detections can be computed.

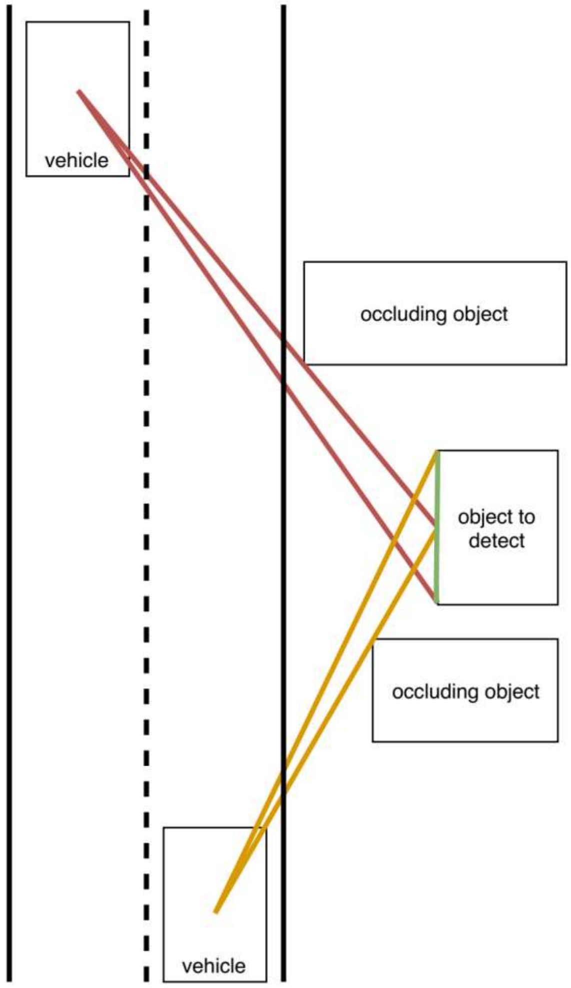


Fig. 6: Example traffic scene with an occluded object partly seen by two other vehicles

In the depicted scene in Fig. 6, the occlusion of the object may lead to

1. the object not being detected;
2. the dimensions of the object incorrectly evaluated;
3. the object class incorrectly assigned;
4. the confidence about any object characteristic being low

in each individual vehicle.

Especially if the object is not included in the global object list of a vehicle, information about it cannot be considered when combining the object lists of multiple vehicles. When sharing data among multiple vehicles, it can therefore be beneficial if low confidence object hypotheses are not discarded but transmitted as well. Since information is lost when computing object hypotheses already, it can also be beneficial to share sensor raw data. It is for example possible to combine the pointclouds of multiple vehicles. If the occupancy state is of interest, it is possible to combine the DOGMa of multiple vehicles as well.

### 6.1.3 Data selection and reduction

As described in the previous section, sharing of data with a high information content that is less abstracted can be beneficial to object detection but poses additional challenges to the compliance with real-time constraints. It can therefore be beneficial to reduce the amount of data sent to the cloud-based system because this can reduce the introduced latency and the required bandwidth. There exist multiple factors that influence whether a part of the collective environment model should be computed from the environment models of multiple traffic participants or whether the vehicle's own environment representation is sufficient.

**Areas of high importance**

Based on the scenario an automated vehicle currently encounters, there exist areas that are presumably more important than others. Example: When performing a lane change, the occupancy state of the adjacent lane plays an important role. It is more reasonable to request data for these areas than for data of other less important areas. It is also more reasonable to publish data of these areas to increase the quality of the CEM. This is assuming that an increase in the amount of data in the CEM is never associated with a degradation of the collective environment model. This necessitates that no processing algorithm overestimates its performance and that all subsequent data processing algorithms base the output computation on the performance of the input delivering algorithm.

**Belief and plausibility**

According to evidence theory as proposed by Dempster and Shafer [16], a belief describes the lower bound of the probability of a proposition to be true. The plausibility describes the upper bound for the probability of the proposition to be true. While the belief function incorporates evidence *for* the proposition, the plausibility function incorporates evidence *against* the proposition, i.e. for the negation of the proposition. In contrast to classical probability theory, this approach is capable to not only quantify

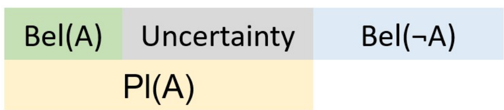


Fig. 7: Belief, plausibility and uncertainty with respect to a proposition A

aleatoric uncertainty but also epistemic uncertainty. The epistemic uncertainty is encoded in the size of the interval between belief and plausibility. Propositions in this context are estimates of the state of the environment of a traffic participant.

There are three objectives with respect to these concepts that traffic participants and the CEM should pursue in coordination. Belief and plausibility in environment representations should be high while the epistemic uncertainty should be low. Since epistemic uncertainty results from a lack of knowledge, new evidence can reduce the epistemic uncertainty. It may also increase belief and plausibility.

Vehicles may subscribe to the CEM to receive additional evidence by other traffic participants. Of course, traffic participants must also publish data to the CEM, otherwise the data is not available for others. There needs to be a mechanism that exchanges information between traffic participants and the CEM on what areas are *currently* associated with low belief, low plausibility and high uncertainty. To increase responsiveness and availability, there can be a supporting mechanism in the cloud that gathers information about what areas are *usually* associated with these low values. Data exchange can subsequently be focused on these areas. Example: An occluded area in front of a vehicle that is planned to be overtaken by the ego-vehicle is associated with high uncertainty because there is little evidence for its occupancy state. The preceding vehicle should be notified and start delivering additional evidence to the CEM whose output can be subscribed by the ego-vehicle.

#### 6.1.4 Data synchronization and matching

The clocks in multiple vehicles can be synchronized with GNSS time but the sensor data acquisition and processing are not synchronized. When fusing data of multiple vehicles in the cloud, it is required that either the data fusion algorithms do not rely on synchronized data or that they are able to synchronize data of multiple vehicles that arrive in the cloud practically non-deterministically. Before synchronizing data, it needs to be decided which data should be synchronized. The frequency of data arriving from one connected traffic participant may vary as well as the frequency of arriving data across different traffic participants as depicted in Fig. 8.

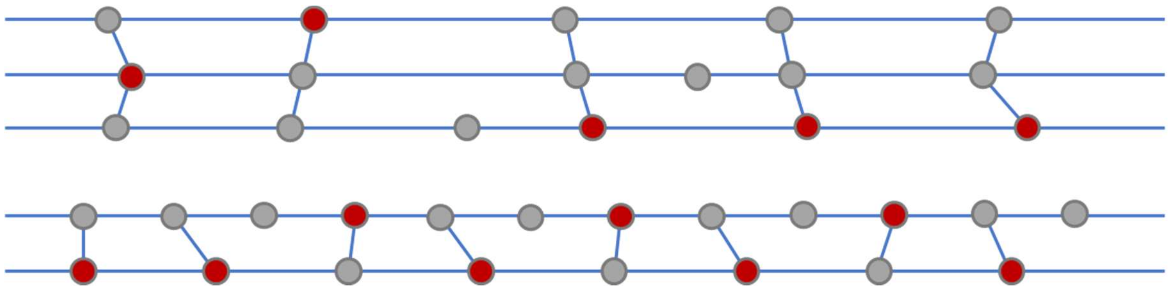

Fig. 8: Visualization of association of data from multiple traffic participants with a varying frequency for each traffic participant (top) and varying frequency across traffic participants (bottom) over time.

In order to apply synchronization, data from at least two traffic participants is required. Presumably, the quality of the data fusion of multiple traffic participants declines with the duration between the timestamps of the data since the older data may be outdated. It can therefore be beneficial to limit the maximum allowed duration. On the other hand, the fusion algorithm can benefit from an increase in the number of inputs. It may therefore be beneficial to allow a longer duration to acquire data. Hence, there is a trade-off to be made that is scenario and data specific.

One possible way of synchronizing data is to predict the future state of old data when new data arrives. A cloud-based system enables multiple new possibilities to perform this synchronization by prediction. In combination, they are capable to handle the synchronization problem for mixed traffic, where automated and connected traffic participants exist next to traffic participants that are not connected, e.g. older vehicles and pedestrians. Both short-term and long-term predictions can be addressed.

**Incorporation of self-reported motion state**

This method is capable of short-term predictions of the state of traffic participants that are connected to the cloud. They do not need to be automated. All traffic participants that are connected to the cloud may share their motion state. The self-reported motion state presumably enables more accurate short-term predictions of the pose of a vehicle than an estimated motion state based on environment sensors of other traffic participants. For vehicles, relatively simple motion models such as the Constant Turn Rate and Velocity (CTRV) can be used.

**Incorporation of self-reported planned trajectories**

This method is capable of longer-term predictions of the state of traffic participants that are connected to the cloud and that are automated, thus being able to share their planned trajectories. For long-term predictions, the past and current motion states, in combination with simple motion models, are insufficient, because the violations of the simplifying assumptions of the motion models become relevant. For automated traffic participants that are connected to the cloud, this problem can be mitigated by incorporating the self-reported planned trajectories of traffic participants. The planned trajectories also only represent an estimate of the future motion state and pose of the traffic participants but this estimate is presumably better than one computed from a simple motion model.

**Incorporation of learned motion models**

This method is capable of short-term predictions of the state of traffic participants that are not connected to the cloud and presumably not automated. Two of the main challenges of simple motion models are that they are generally incapable of reliably predicting the long-term motion of objects but also incapable to reliably predict the short-term motion of objects that change their motion state relatively abruptly such as pedestrians.

Long-term predictions are not possible because the errors introduced by the violation of the motion models' assumptions accumulate over time. Short-term predictions are difficult because the data considered by simple motion models such as the past and current motion state are insufficient. Behavior is highly dependent on context and a more detailed representation of the object.

More complex motion models, even learned ones, can also be used by traffic participants, e.g. automated vehicles. A cloud-based service has several advantages that are described in the following.

**Less limited energy and hardware**

It is to be expected that progress on the prediction performance of neural networks and other machine learning based models has long not ended. This is due to rapidly advancing hardware and software. Yet, automated vehicles are and will be deployed in the meantime. Due to restricted energy consumption and thermal constraints, it can be beneficial to make use of specialized hardware in vehicles. Technology such as ASICs are energy efficient but, compared to FPGAs and GPGPUs, also more difficult or even impossible to update with respect to the models that they run. A more centralized cloud-based system is not as limited in its energy use and can therefore make use of more flexible hardware. Even if latency restrictions lead to ASICs being used in the cloud, it remains presumably easier to update the hardware in a relatively small number of servers than in all automated vehicles. The use of a cloud-based system therefore enables already manufactured vehicles to make use of the latest best performing models for data processing such as the prediction of the state of other traffic participants.

**Less limited data storage**

Another limitation faced in automated vehicles is their limited ability to store the large amounts of data that they receive from their sensors and subsequently process. This data is highly valuable though because it can be used to train machine learning based models that are needed for automated driving. Variants of semi-supervised learning are especially promising because they do not necessarily rely on manually labeled data. Instead of e.g. assigning a class to a pixel in an image, the future state of one of the possible environment representations acts as the label. For this process, it is not necessary to manually label data but only to record the future environment representations and feed them into the training process. This data can be a low-level representation such as a LiDAR pointcloud, image or radar detection but also the higher-level representation such as object lists, occupancy grids or free space representations. All of these may also contain information on the dynamics of the environment.

A cloud-based system allows vehicles to transmit the respective data such that they can be stored in the cloud. Here, there storage of large amounts of data is technically and economically much more viable. Inside the cloud-based systems, machine learning based models can then be trained and validated. These models are potentially better than the models initially deployed with the vehicles because they take into consideration a larger amount and more recent data.

**Less limited over-the-air software updates**

The models trained in the cloud can be deployed to vehicles if the vehicles' computation hardware on which the models are run is compatible with the new model topology. Even vehicle hardware that is flexible with respect to the algorithm it can run such as GPUs may not be suited to run the best models though because the requirements may for example demand more computational power to run a model within a specified time than is available. In this case, only cloud-based systems whose hardware is kept up-to-date are capable of delivering the most accurate model outputs, e.g. predictions about the future motion state of other traffic participants. Since it is easier to replace hardware in a more centralized server, software updates are also less limited in these servers.

### 6.1.5 Functional modules of the CEM

Most of the functions inside the Collective Environment Model are in principle similar to the functions that run in vehicles, which is why they are not described in detail here. The main difference is that they have access to more information because traffic participants may share data such as their location, pose and motion state that can be incorporated in the perception and localization as well as the tracking, fusion and prediction. How this data integration is best achieved is still an open research question.

For machine learning based perception models, attention mechanisms are a promising approach to increase the performance of the inference. Based on information shared by traffic participants, attention masks can be concatenated to inputs for models that can thereby incorporate the prior knowledge in the computation of the output. This way, not only object-based fusion of data is possible, but also the fusion of sensor raw data with object data.

### 6.1.6 Environment prediction and latency reduction

Here, we define prediction as the estimation of the state of some representation at a time different from the timestamp assigned to the representation. Usually, we predict the future (from the past and present). The prediction of the future state of the environment is not only an important component when synchronizing incoming data. It can also help to reduce the virtual latency introduced by a cloud-based service by predicting and providing the state of the environment at the time when the data arrives in a traffic participant that is connected to the cloud. The virtual latency is the duration between the (virtual) timestamp assigned to data based on the point in time when the data is supposed to represent something (e.g. the environment) and a real-time timestamp at which the data is supposed to be used. The (true) latency of a system is the duration between two real-time timestamps during which a system computes output based on some input. As an example, there is latency between a sensor measurement and the output of an object detection algorithm that takes the sensor measurement as input. We can then estimate the state of an object arbitrarily far into the future and assign the future (virtual) timestamp to the object.

The true latency can only be reduced by minimizing the duration of data processing. The virtual latency can be reduced by estimating the state of some data further into the future. The virtual latency is negative as long as the virtual timestamp is further in the future than the real time at which the data is to be used. A behavior planning algorithm is an example for an algorithm that can make use of the estimated environment representation. At any point in time, it can in principle take into consideration the past, present and future representation of the environment.

There are two essential classes of uncertainties associated with predicting the future state of the environment. First, there is aleatoric uncertainty due to elements in the environment that result in a different environment prediction every time they are measured or computed and used for prediction. This class of uncertainty captures the influence of chance on the process.

Second, there is epistemic uncertainty introduced by aspects about the environment that could in principle be known but are not in practice. This class of uncertainty is for example associated with elements in the environment that are neglected because of computational constraints or inaccuracies in the motion models introduced by simplifications. This class of uncertainty can also be introduced through environment representations that are bounded in space or contain occlusions. In addition, elements may move into the bounds or become non-occluded during the time between the timestamp of the input data and the virtual timestamp of the estimated data.

A cloud-based system may reduce epistemic uncertainty in the predicted environment representation. It may be reduced by neglecting fewer elements in the environment representation because more processing power is available in the cloud. Additionally, better motion models may be used in the cloud because of the more flexible hardware that can run up-to-date models trained with a larger amount and more recent data as described above. By combining the environment representations of multiple traffic participants, elements that are either occluded or outside the range for one vehicle's sensors but not for the sensors of another, additional elements become available to both traffic participants.

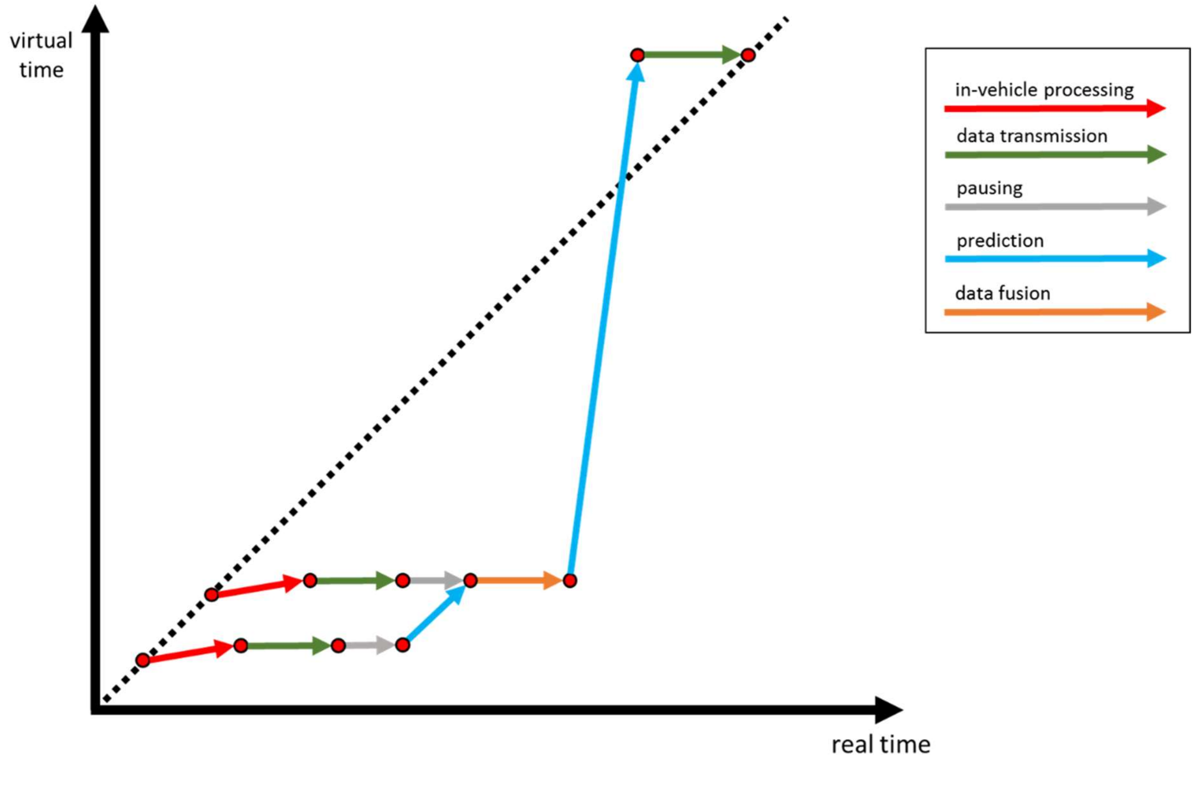


Fig. 9: Example data synchronization and fusion scheme for two vehicles if acquisition rates and transmission durations are not known.

Fig. 9 shows the simplest case of how virtual and real time develop in relation to the processing of data acquired by sensors in two different vehicles that are connected to the cloud. Here, the processing in the vehicles and the transmission of the processed data takes the same amount of time for both vehicles. Only one measurement of each of the vehicles is considered. The vehicles each process the measurement data to compute a more abstract representation of the environment. During this step, they may already predict the future state of the environment, which is indicated by the inclined red arrows. The data from the vehicle that first measured its environment arrives in the cloud first. If it is not known when the next data arrives, nothing can be done yet because for a prediction, the time stamp of the more recent data is needed. As soon as the data of the second vehicle arrives, its timestamp can be analyzed and the data of the first vehicle can be predicted to that timestamp. Now, the data can be fused. In order for the data to have zero virtual latency when it arrives in the vehicles, the fused data is now predicted into the future such that its virtual timestamp lies in the future in comparison with the current time to such an amount that real time exactly catches up during the time that it takes to transmit the data. For this purpose, transmission latency needs to be monitored constantly. When transmission and processing times are known, another fusion scheme becomes possible that is capable to reduce the true latency of the system and the uncertainty introduced by prediction.

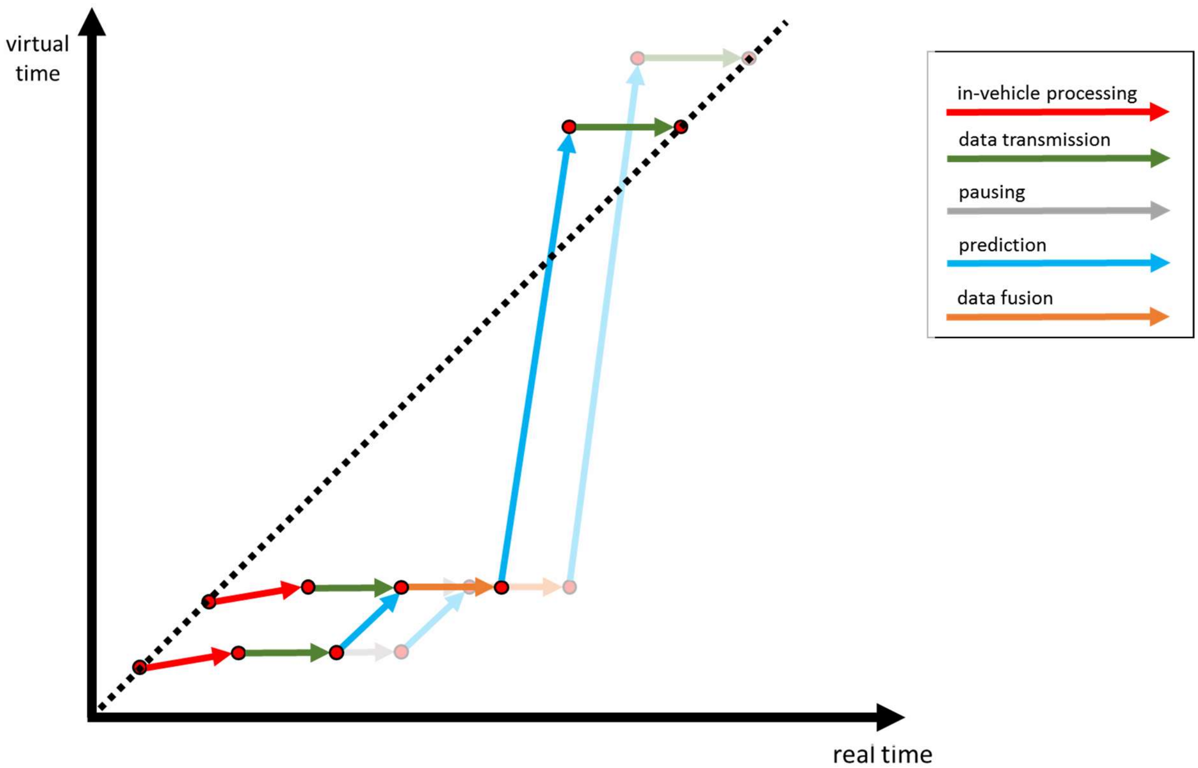


Fig. 10: Improved data synchronization and fusion scheme for two vehicles if acquisition rates and transmission durations are known.

In Fig. 10, it is known when the data of the second vehicle arrives in the cloud and what timestamp it has, so prediction of data that is available in the cloud can already occur. This scheme only works if the prediction step takes less time than there is time between the arrival of the data of the first vehicle and the arrival of the data of the second vehicle. When both data are available and share approximately the same timestamp, they can be fused and their future state can be predicted. Since less real time has elapsed since the sensor information was acquired in comparison to the other scheme, prediction time can be reduced, resulting in smaller introduced uncertainty.

Many additional prediction and fusion schemes are possible but are not discussed in detail here. More research is necessary to determine which schemes are best suited under which circumstances.

### 6.2 Collective Memory

The Collective Memory (CM) acts as the short- and long-term memory of the cloud-based system in UNICARagil. In addition, it provides non time-critical applications that support the Collective Environment Model (CEM) and the Collective Behavior (CB). In a fog-computing setting, it would represent the centralized backend.

The short-term memory of the Collective Memory acts as a data exchange layer. It gathers all data that is transmitted to the CEM. At the same time, it collects all data processed by the CEM. It may exchange non time-critical data with the CEM and the CB such as trained or parametrized models. Since the amount of data shared by vehicles exceeds the storage and processing capabilities of current technology and because not all data is equally valuable, there exists the possibility to filter the data in the short-term memory before it is stored in the long-term memory.

The long-term memory provides data to the algorithms in the CM that train or parametrize models such as artificial neural networks or Markov models. The trained or parametrized algorithms are validated using data that is different from the data used to train or parametrize the models in order to get an unbiased estimate of the models' performance on new data. If the performance exceeds the performance of previous models, it may be deployed to the CEM, CB and compatible traffic participants.

As described earlier, machine learning based models require vast amounts of training data. The data used to train the models should be representative of the data the model encounters when deployed to the user. In a changing world, it is therefore necessary to constantly acquire more data that can be used to train up-to-date, well performing and therefore safe models. The CM allows traffic participants to transmit and store data, thereby allowing this goal to be achieved.

An additional application that a CM may provide is the classification and evaluation of scenarios, e.g. for extracting relevant scenarios for testing automated vehicles. This is a challenging component in the development process of automated vehicles and plays a major role in the assurance their safety.

The following definitions are used in this paper: A scenario is a finite sequence of successive scenes. A scene is a representation of the world as perceived by an entity such as an automated vehicle. A scenario classification is the mapping of a class to a scenario from a list of predefined possible classes. A scenario class is usually represented as a description in language and comes with a set of requirements manually defined by humans. The evaluation of a scenario is a description of the desirability of the actions of traffic participants in a scenario.

The classification of scenarios into classes that are intelligible for humans is a means to reduce the complexity of the vast number of possible scenarios. They become more accessible and therefore usable for human analysis.

**Scenario evaluation and policy learning**

The evaluation of scenarios is a means to extract data that may be used to determine desirable policies for automated agents. A policy is a possibly stochastic model that defines the behavior of an agent based on the belief about the current world state.

An optimal policy can be found through various means such as Value Iteration or Policy Iteration and usually depends on a reward function. A reward function associates a numerical reward with the transition of an agent from one state to another based on a conducted action. The reward can then be used to adjust the agent's policy. Reward functions can be manually defined but also found through apprenticeship learning via inverse reinforcement learning (IRL) [17]. Here, the goal is to find the reward function of expert agents. Traffic does not only consist of expert drivers though. The evaluation of scenarios experienced by a large number of human or computer drivers can help identify the relative experts from whom all others can then learn. The function used for scenario evaluation can be used to pre-select the experts but presumably does not capture their reward functions sufficiently well, which is why it needs to be found by other means such as IRL. When the reward function is sufficiently approximated, one can use it to find a policy via direct reinforcement learning. The approach to first learn the goal of the expert instead of the policy can help to later find a policy that generalizes better to changes in the environment. In addition, the reward function can be analyzed by humans in order to get an understanding of what determines "good" behavior. Another advantage is the approach's ability to adjust the policy over time without the need for additional demonstrations by experts. [18]

Another approach to extract desirable policies is made possible by the evaluation of scenarios and the subsequent identification of expert drivers: All future actions of the expert drivers and the corresponding relevant data processed in the CEM are stored either implicitly or explicitly in the long-term memory. This data can be used to directly learn the experts' policies via (supervised) Imitation Learning. Depending on the availability of data, there are multiple possibilities to do so.

If expert drivers did not transmit any data to the cloud, their actions and beliefs of world states can only be inferred from data sent by other traffic participants that contains sufficient information about the expert driver's actions and beliefs.

Actions of the expert drivers are encoded in their sequence of motion states as perceived by other traffic participants. A belief about the environment of the expert driver can be inferred from the environment perception of other traffic participants. When actions and beliefs are known, supervised learning can be used to learn the experts' policy. This option is suitable for expert drivers of vehicles that do not possess the ability to connect to the cloud.

If the expert drivers transmit performed actions, they can be directly used in combination with a corresponding environment representation provided by the CEM to learn the experts' policy. This option is suitable for expert computer drivers operating connected vehicles.

A third possibility is to compute experts' actions from their transmitted vehicle state in combination with a corresponding environment representation provided by the CEM to learn a desirable policy via imitation learning. This option is suitable for connected vehicles that are not automated and therefore cannot transmit actions because these are performed by human drivers.

The aforementioned mechanisms can also be applied to non-expert drivers resulting in the extraction of probable behavior instead of desirable behavior. Here, the evaluation of a scenario is not needed to identify experts.

### 6.3 Collective Behavior

The Collective Behavior module consists of two submodules. The behavior modeling uses the models provided by the CM and computes the desired future behavior and the probable future behavior. It takes the collective environment model as input. Since it is possible that not all traffic participants are able or willing to follow suggested trajectories, the desired behavior may not be feasible. The behavior planning corrects the desired behavior in accordance with the probable behavior of those traffic participants that do not directly cooperate via the cloud. Since the result can lead to new optimal behavior, there needs to be an iterative process between behavior modeling and behavior planning. The goal is to compute a trajectory for each traffic participant that leads to a scenario that is evaluated as positively as possible. In order to validate and adjust the trajectory suggestions, they are fed back into the CM, where they can be used to evaluate the current models and to train new, better ones.

## 7 Conclusion and outlook

In this paper, we have presented a cloud-based functional framework that provides three major services to traffic participants such as automated vehicles. The Collective Environment Model (CEM) combines the environment representations of multiple traffic participants and provides the result as a service. Automated vehicles may incorporate the CEM into their behavior planning. They may also get an estimate of their perception performance through a discrepancy metric that describes the consistency of their perception with that of other traffic participants. The Collective Memory (CM) acts as the data und processing backend and provides up-to-date perception, environment modeling and behavior planning models to the CEM, to the Collective Behavior (CB) and to compatible traffic participants. For this purpose, it may automatically classify and evaluate scenarios that are contained in the vast amount of data transmitted by traffic participants. With the extracted data, the models can be trained and validated. The CB makes use of the large amount of collected data in the CM in order to extract and provide behavior recommendations as a service. These support traffic participants to find cooperative trajectories that are efficient, comfortable and safe.

Designing an integrated functional framework that provides cloud-based services in a C-ITS consisting of various traffic participants that may or may not be automated and may or may not be connected is not an easy task. The list of research questions arising is far beyond the capacity of this paper. The functional architecture can hardly be conceived without the technology enabling the functions. Since enabling technologies such as 5G and ITS-G5 are still in early stages of development, not all questions can be answered yet. It remains to be seen whether these new technologies will be capable of transmitting the large amounts of data under the restrictive requirements applying to automated driving. Reliably providing high quality data at low latencies is not only a question of the communication technology though. The development of suitable hardware and software that processes the data is just as important. In UNICARagil, the first step in this direction is made.

## 8 Acknowledgement

This research is accomplished within the project “UNICARagil” (FKZ EM2ADIS002). We acknowledge the financial support for the project by the Federal Ministry of Education and Research of Germany (BMBF). We also thank our project partners at the Chair of Computer Science 11 - Embedded Software of RWTH Aachen University for their support in the cloud architecture.

## 9 References

[1] WOOPEN, Timo, et. al., 2018.
UNICARagil - Disruptive Modular Architectures for Agile, Automated Vehicle Concepts
In: 27th Aachen Colloquium. Aachen, October 8th-10th, 2018
ISBN: 978-3-00-057468-9

[2] DONGES, Edmund, 1982.
Aspekte der Aktiven Sicherheit bei der Führung von Personenkraftwagen.
In: Automobil-Industrie 27,183–190

[3] SAE INTERNATIONAL Dedicated Short Range Communications (DSRC) Message Set Dictionary, SAE International, Warrendale, USA, 2009

[4] IEEE - INSTITUTE OF ELECTRICAL AN ELECTRONICS ENGINEERS IEEE standard for Information technology-- telecommunications and information exchange between systems-- local and metropolitan area networks-- specific requirements Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications; Amendment 6: Wireless access in vehicular environments Institute of Electrical and Electronics Engineers, New York, USA, 2010

[5] ISO - INTERNATIONALE ORGANISATION FÜR NORMUNG
ISO/IEC 7498-1: Information Technology - Open Systems Interconnection - Basic Reference Model: The Basic Model
ISO, Genf, Schweitz, 1994

[6] FESTAG Andreas, 2014.
Cooperative intelligent transport systems standards in Europe
In: IEEE Communications Magazine

[7] Initiativen der Europäischen Komission
ITS-G5 technology – A Fact Sheet.

[8] HAMEED MIR, Z., FILALI, F.
LTE and IEEE 802.11p for vehicular networking
A performance evaluation
EURASIP Journal on Wireless Communications and Networking, S. 89ff, 2014

[9] ETSI - EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE
Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service
ETSI, Valbonne, France, 2014

[10] ETSI - EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE
Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service
ETSI, Valbonne, France, 2014

[11] ETSI - EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE
Intelligent Transport Systems (ITS); Testing; Conformance test specifications for Signal Phase And Timing (SPAT) and Map (MAP); Part 1: Test requirements and Protocol Implementation Conformance Statement (PICS) pro forma
ETSI, Valbonne, France, 2015

[12] SAE INTERNATIONAL
Dedicated Short Range Communications (DSRC) Message Set Dictionary,
SAE International, Warrendale, USA, 2009

[13] ETSI - EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE
Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM); Rationale for and guidance on standardization
ETSI, Valbonne, France, 2011

[14] GAWRON, James H. et. al.
Environmental Science & Technology 52 (5), 3249-3256, 2018
DOI: 10.1021/acs.est.7b04576

[15] BROGI, Antonio, FORTI, Stefano
QoS-Aware Deployment of IoT Applications Through the Fog
IEEE Internet of Things Journal Volume: 4 , Issue: 5, Oct. 2017

[16] SHAFER, Glenn
A Mathematical Theory of Evidence
Princeton University Press, Apr. 1976

[17] ABBEEL, Pieter, NG, Andrew
Apprenticeship Learning via Inverse Reinforcement Learning
Computer Science Department, Stanford University, Stanford, CA 94305, USA, 2004

[18] PIOT, Bilal, GEIST, Matthieu, PIETQUIN, Olivier
Bridging the Gap between Imitation Learning and Inverse Reinforcement Learning
IEEE Transactions on Neural Networks and Learning Systems Volume: 28 , Issue: 8 , Aug. 2017