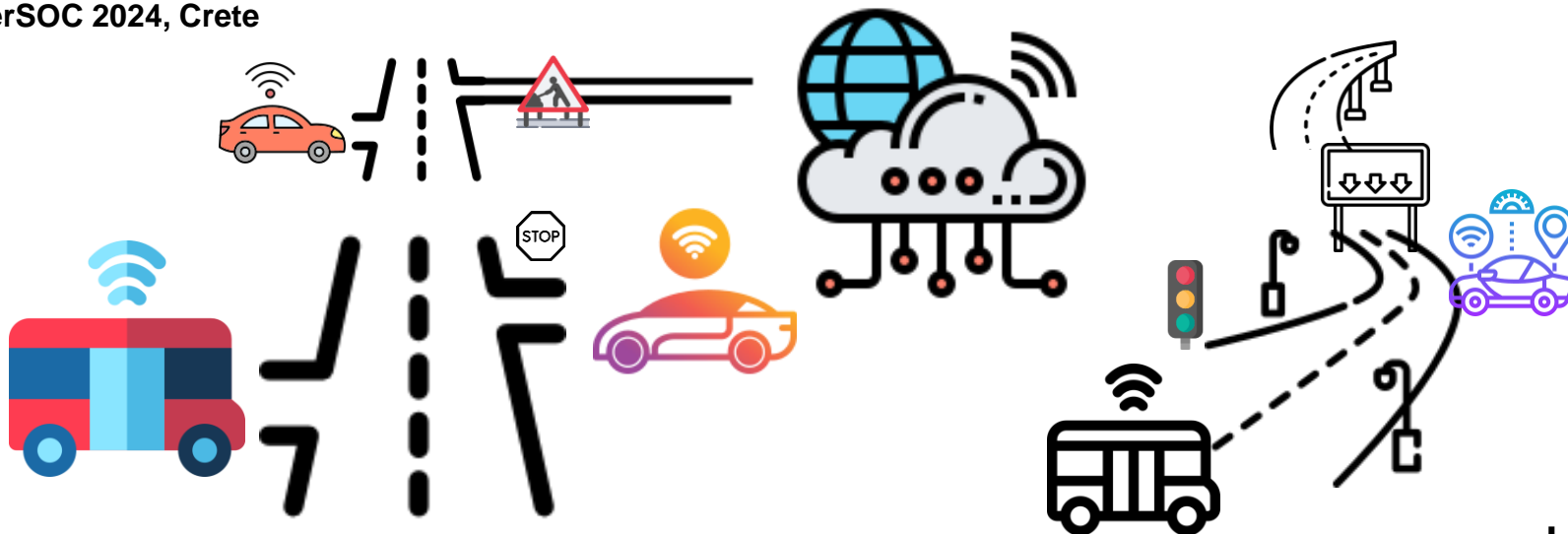



An Orchestrator for the Dynamic Extension of Automotive E/E Architectures to the Cloud


Martin Sommer, Housseem Guissouma, Marc Schindewolf and Eric Sax

SummerSOC 2024, Crete



 Definition of automotive cloud-based Software Components

 Motivation for cloud-based Software Components (SWC)

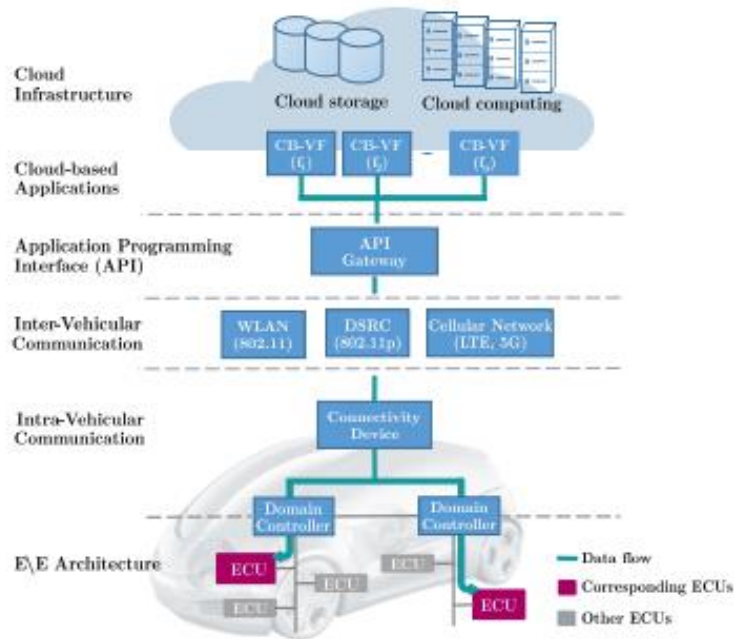
 Use Cases & Application models of cloud-based SWC

 The SW-Orchestrator as central part for cloud-based SWC

 Concept of a cloud-based SWC

 Prototypical Implementation & Evaluation

What are automotive cloud-based SWC?



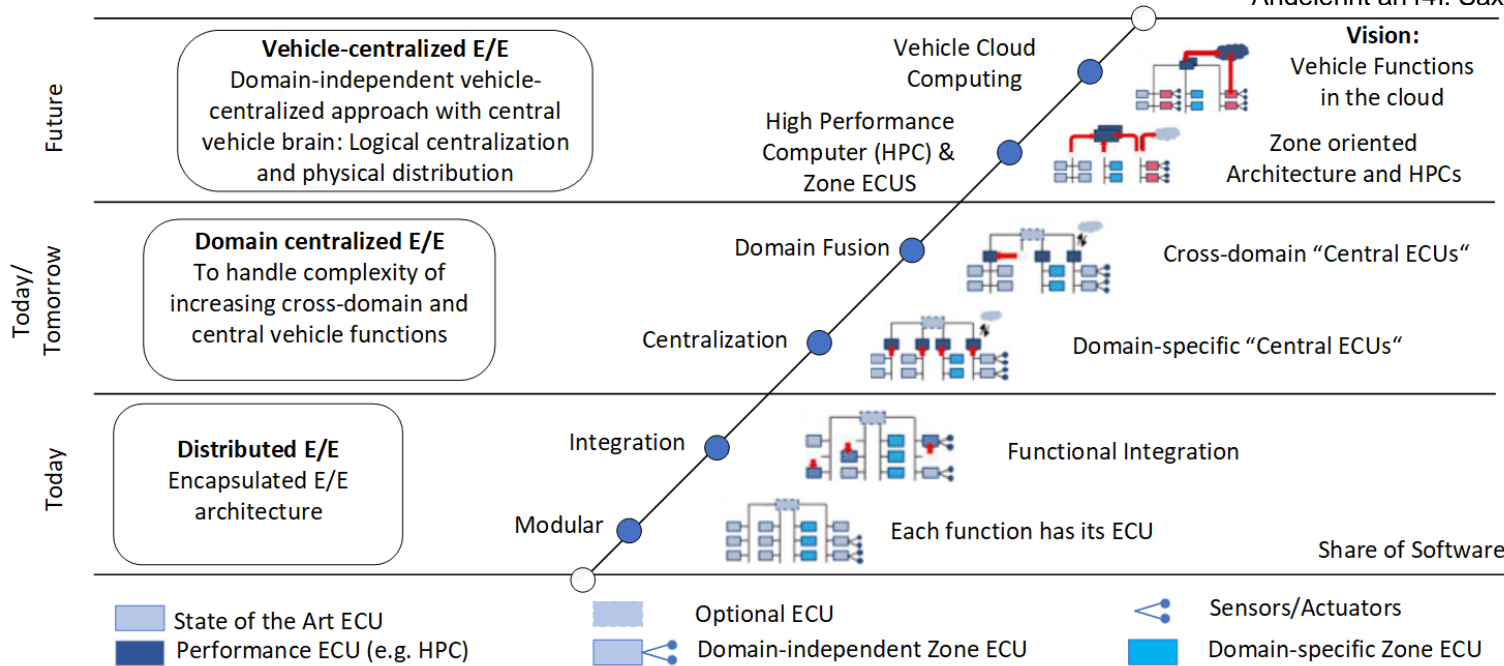
■ Definition:

- Regulation, control or monitoring tasks
- CB SWC use cloud computing and storage capacities instead of available vehicle resources
- The input from CB SWC can come from both the vehicle and the cloud
- The output is intended either to optimize existing functions in the vehicle or to replace them

[1]: Farzaneh Milani, Dissertation, 2020

Evolution of Electric/Electronic (E/E) Architectures

Analehnt an I4I: Sax et al, 2020



[2]: V. Navale et al., "(r)evolution of e/e architectures," SAE International Journal of Passenger Cars, 2015

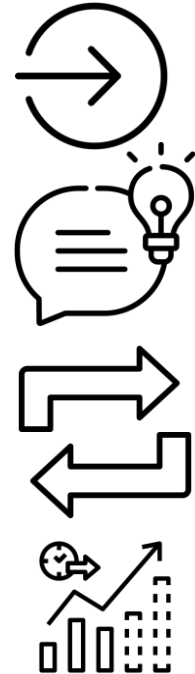
Motivation for cloud-based vehicle SWC

- Reduce resource restrictions: Resource restrictions can be overcome in the cloud. The available resources in the vehicle can be used for SWC that absolutely have to be carried out in the vehicle.
- Reduce energy consumption in the vehicle: By transferring the calculation of a SWC to the cloud, the energy consumption of the previous control units (ECUs) is partially or completely eliminated (depending on the application model, see next slides).
- Reduce (development) costs: A SWC that is partially or fully executed in the cloud can be further developed, parameterized and tested there much more easily without any access to the control unit in the vehicle. Control units can be completely or partially eliminated.
- Creating databases and fleet learning: A SWC in the cloud can benefit from fleet knowledge and feed or access databases.



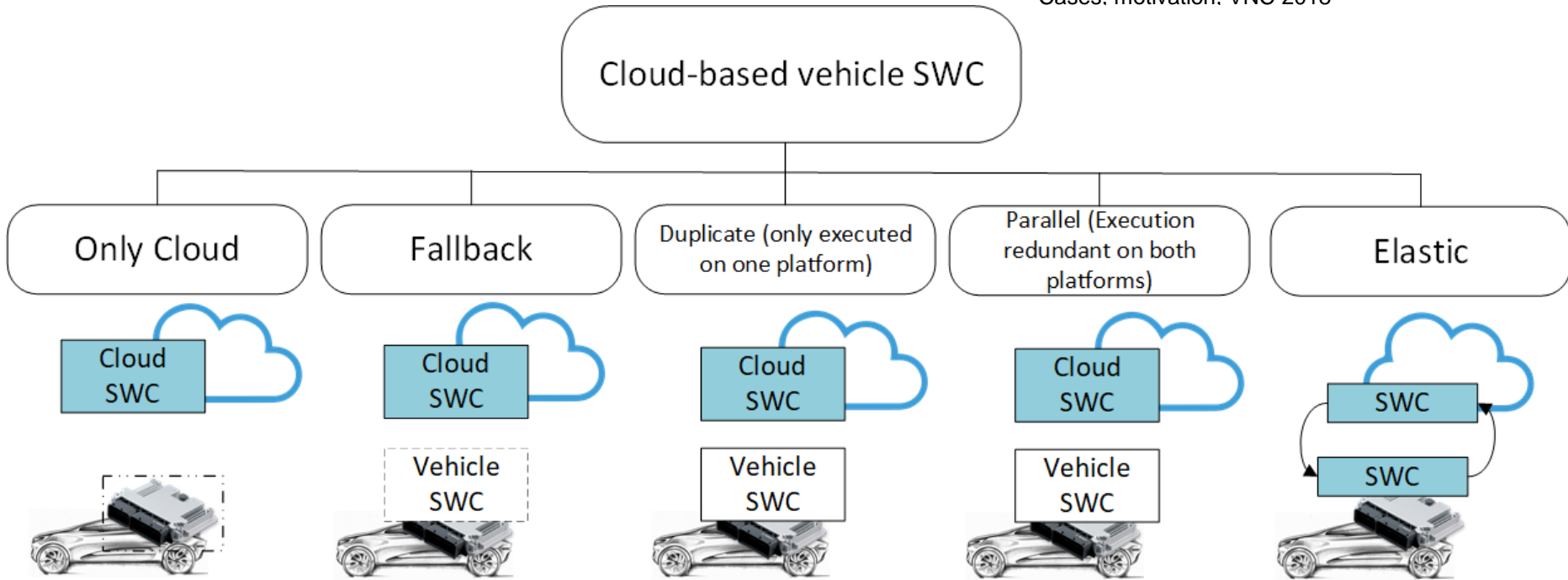
Use Cases for cloud-based SWC

- Input data for On-Board Functions
 - Cloud as a data pre-processing center for functions in the vehicle
- Suggestions for On-Board Functions
 - Cloud as a data pre-processing center for functions in the vehicle
- Control strategy (Control over the air (COTA))
 - Control loops from the vehicle to cloud and back
- Prediction models
 - Compute intensive Functions (e.g. lifetime models for vehicle batteries) on the basis of many input sources



Application models of cloud-based vehicle SWC

Adapted from [3]: Farzaneh Milani and Christian Beidl, Cloud based vehicle functions: Definition, Use Cases, motivation, VNC 2018



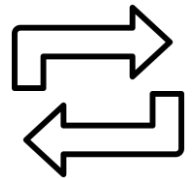
Software Orchestrator Definition

Definition of the Orchestrator

An orchestrator is a SOA module that oversees and directs the automatic selection and management of multiple resources and services in a dynamic environment to meet certain objectives.

Note: Objectives could, among others, include time-efficiency, consistency, security, and cost-effectiveness.

→ In our work the Orchestrator serves as central component for cloud-based vehicle SWC in order to make the Use Case Control over the air (COTA) (slide 6) possible with a service oriented architecture



Orchestrator Requirements

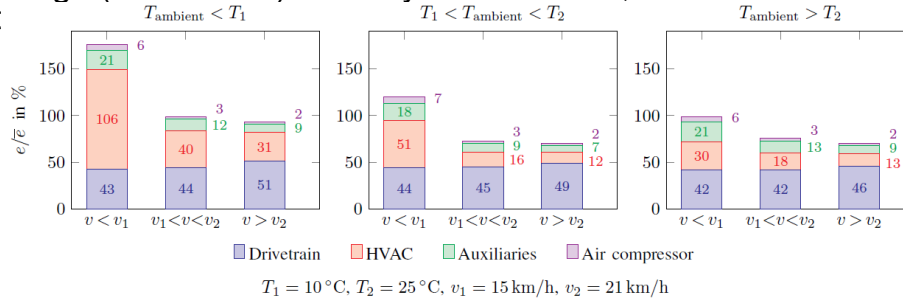
- Req-1: The orchestrator must have access to **an internal service database** containing an overview of all services and their status.
- Req-2: The orchestrator must be able to **check the availability of each service** and adjust its status if necessary.
- Req-3: The **error case, where an original available service cannot be reached, must be detected** and handled accordingly. The orchestrator must decide whether to adjust the corresponding status or to carry out further measures, such as directing a service replacement.
- Req-4: The orchestrator should be able to **discover new services** during runtime and integrate them into the existing system. Therefore, the service database shall be dynamically expandable.
- Req-5: The orchestrator must be able to define service execution orders in a database, as services can have interdependencies.
- Req-6: The orchestrator must be able to **establish a connection to external networks**, manage them, and make their services available to the vehicle.
- Req-7: The connection to networks other than the internal vehicle network must be secure. Networks must be authenticated before the connection is established, and communication must be encrypted.
- Req-8: Safety-relevant functions should be given the opportunity to be prioritized so that it can be ensured that they are completed on time.
- Req-9: The orchestrator shall know the requirements of the services in order to optimally distribute them to the electronic Control Units (ECUs) in terms of priority, dependencies, and resource requirements.

Use Case

■ Cloud-based Heating, ventilation and air Conditioning (HVAC) control of an electric city bus

■ Reducing total energy consumption in the vehicle

- The contribution of HVAC in city buses can exceed that of the drivetrain. Potential energy savings (see slide 5) not only at ECU level, but the function itself must be more intelligent



[4]: Tobias Rösch et al. , Multi-layer approach for energy consumption optimization in electric buses, VTC 2023 Spring, Florence, Italy

■ Reducing development cost

- Cost-efficient access and further development of cloud-based functions for OEMs
- Providing different versions (e.g. tailored to climate regions) in the cloud is more cost-efficient than flashing on the production line



The type of the cloud-based HVAC control

Criteria	Weight	Only Cloud	Fallback	Parallel	Duplicate	Elastic
		Points/weighted points	Points/weighted points	Points/weighted points	Points/weighted points	Points/weighted points
Necessary on-board Hardware	1	9/9	7/7	5/5	5/5	4/4
Capital Expenditures, Operating Expenses	2	4/8	6/12	2/4	3/6	4/8
Availability/Robustness	2	4/8	10/20	10/20	9/18	5/10
Security	0.5	4/2	9/4.5	5/2.5	5/2.5	5/2.5
Sum		26	43.5	31.5	31.5	24.5

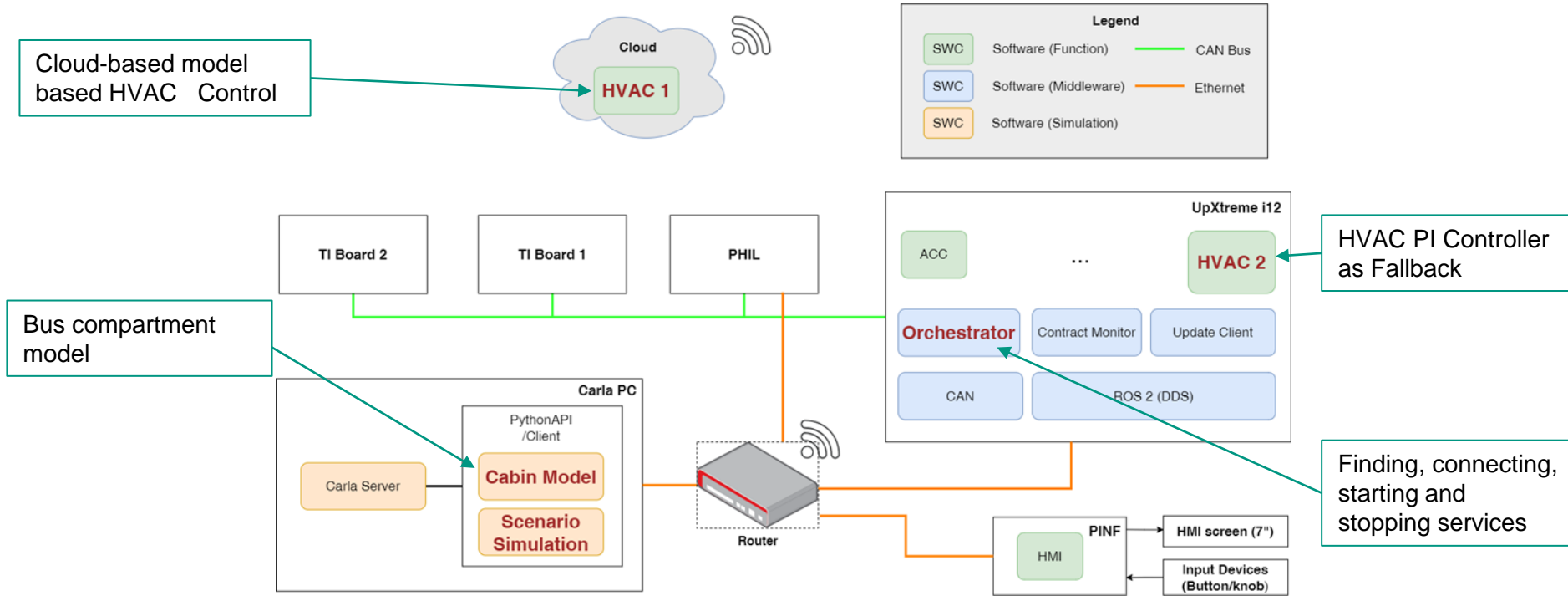
The demonstrator platform ATLAS

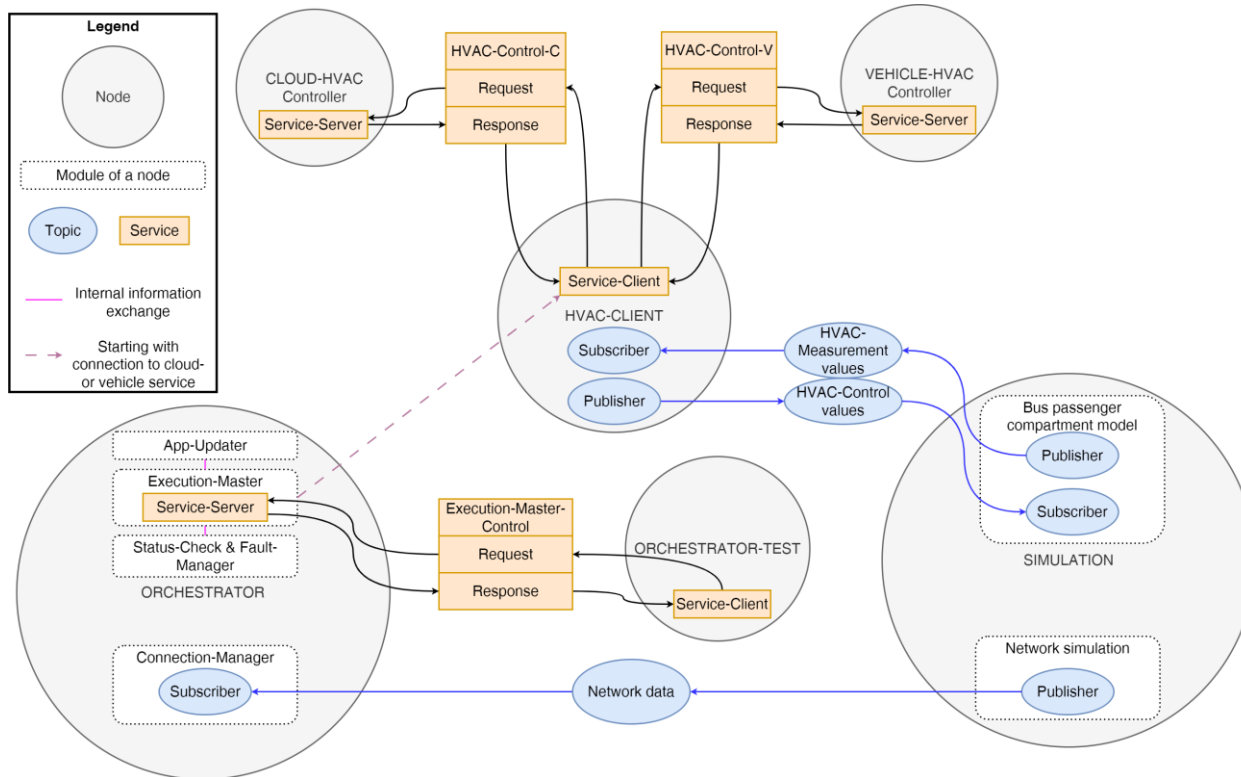
- Service oriented or signal oriented?
 - Service oriented (Robot Operating System 2 (ROS2)) due to the higher expandability of the SW architecture at runtime
 - Development of a model predictive HVAC control (MPC) that predicts disturbance variables (door openings, passengers, etc.) and whose control targets can be defined with a cost function

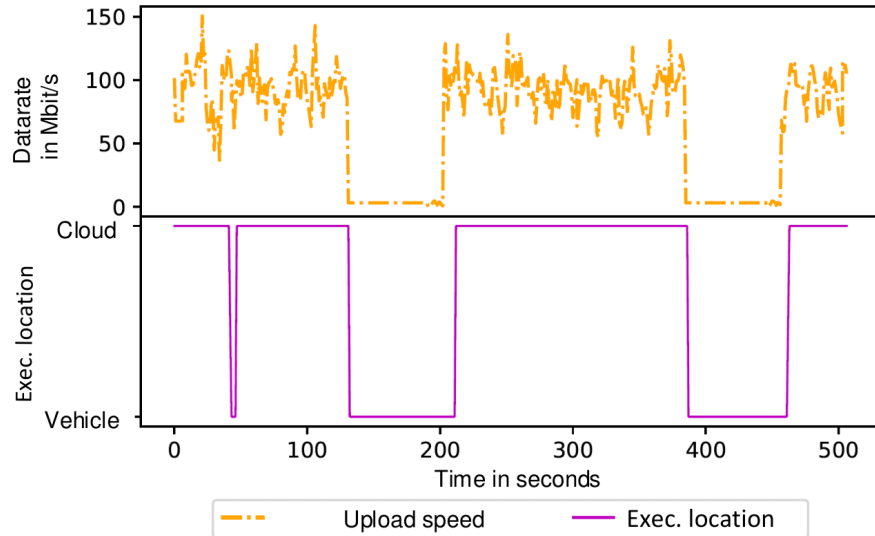


Concept of a cloud-based SWC

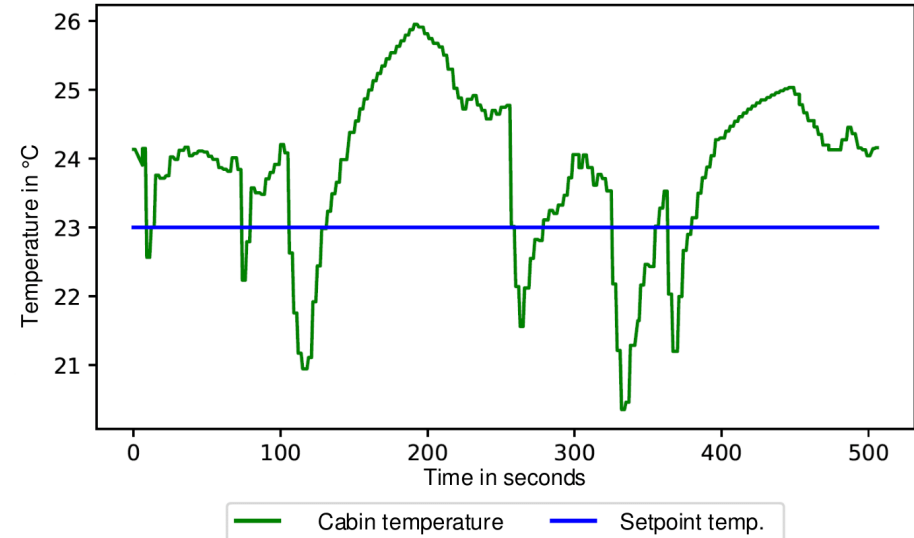
Software and Hardware Architecture of the ATLAS Demonstrator







Evaluation with both (Cloud and Fallback) controllers being PI controllers as MPC is under development



Next steps

- Analysis of energy reduction potentials through the usage of the cloud and the advanced model predictive controller
- Analysis of latency effects of cloud connection (with cellular 4G and 5G) control performance

Part of the SofDCar Project funded by the EU

SummerSoc
Service Oriented Computing

SofDCar



Finanziert von der
Europäischen Union
NextGenerationEU

Gefördert durch:



aufgrund eines Beschlusses
des Deutschen Bundestages

BACKUP

Entwicklung einer statischen cloudbasierten HLK Funktion

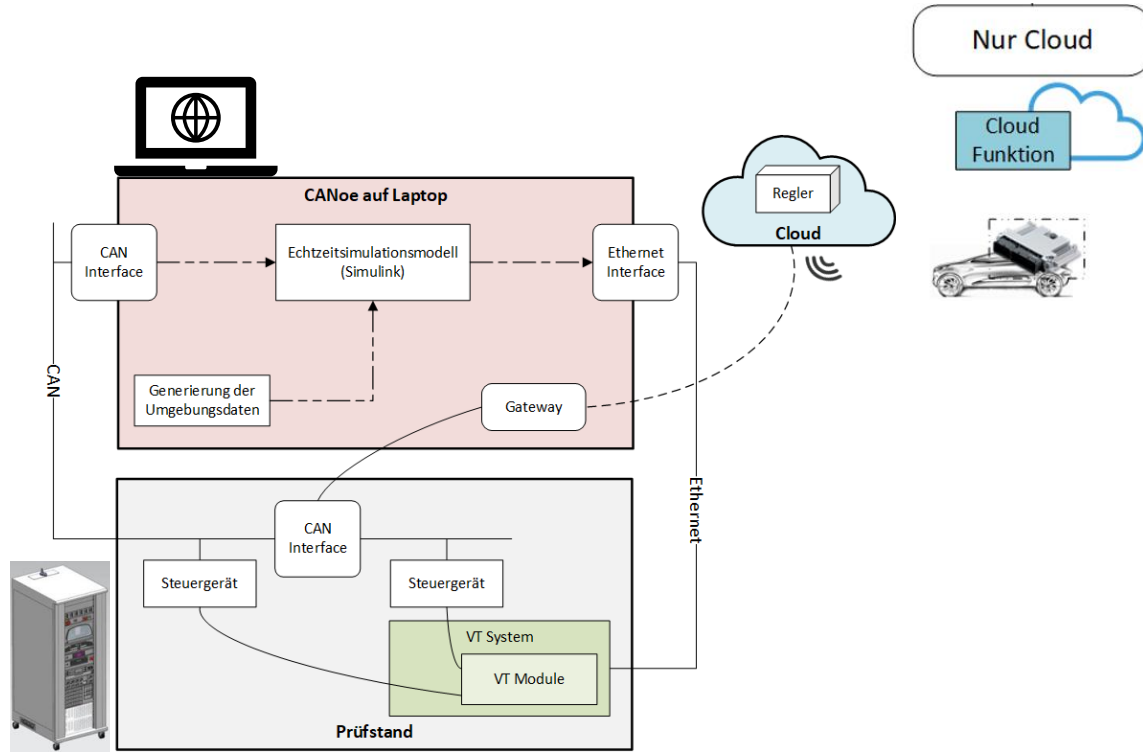
- Entwicklung und Test eines modellprädiktiven HLK Reglers in Matlab/Simulink
- Simulativer Vergleich des neuartigen Reglers mit dem bestehenden PID Regler
 - Verwendung der Fahrzeugmodelle des Tools GT Suite der EvoBus GmbH
 - Vergleich des Energieverbrauchs und des thermischen Komforts der beiden Regler

		Kühlszenario			Heizzenario		
Metrik	Einheit	PI/PID	MPC	Abweichung	PI/PID	MPC	Abweichung
Abweichung der Kabinentemperatur (MAE)	°C	1,008	0,931	-7,6 %	1,309	0,969	-25,9 %
Gesamtenergieverbrauch	kWh	3,19	3,10	-2,8	1,97	1,99	+1 %

[9]: Sommer et al, 2021

Test der statisch verlagerten HLK Funktion

- Die neu entwickelte Funktion (Regler) wurde als AUTOSAR Adaptive Applikation in Azure Cloud verlagert
- Test der cloudbasierten Funktion an HLK Prüfstand von EvoBus in Neu-Ulm
- Die verlagerte HLK Adaptive Applikation hat erfolgreich Stellwerte anhand generierter Umgebungsdaten berechnet und auf den CAN Bus am Prüfstand gesendet/geschrieben



Klassifikation von Steuergeräten nach geschätzter Leistungsaufnahme

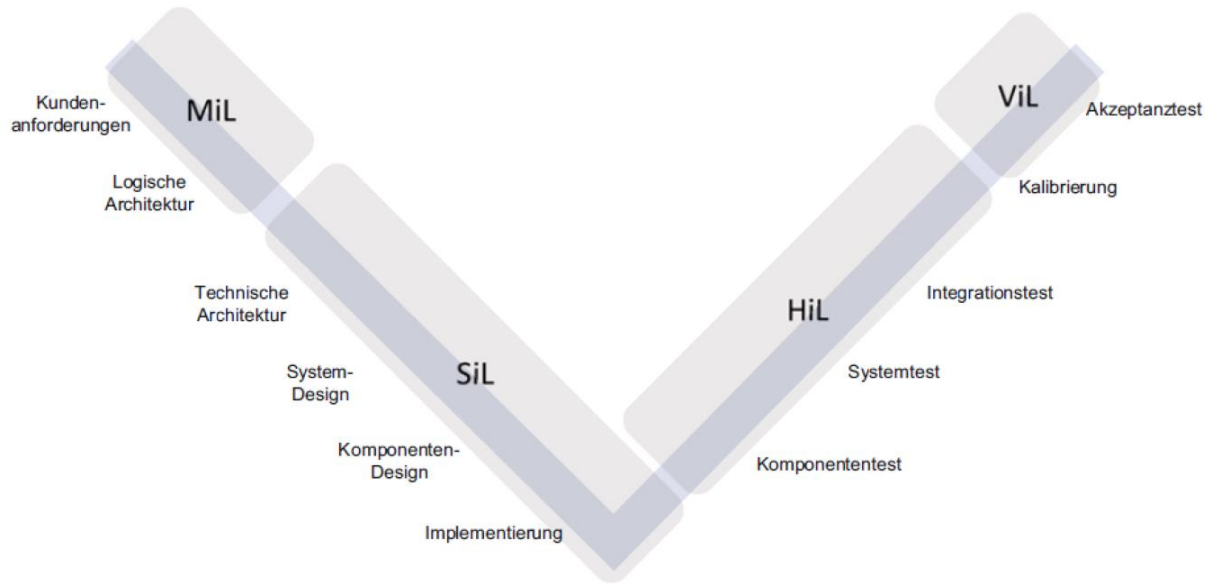
Leistungsklasse	Microcontroller	Peripherie	Netzwerk	Wirkungsgrad	
Hoch	2.5 W	2 W	0.5 W	50 %	10 W
Mittel	0.8 W	1 W	0.12 W	50 %	4 W
Gering	0.5 W	0.5 W	0.04 W	50 %	2 W

[3]: Norbert Balbierer. Energiemanagement Ethernet-basierter Fahrzeugnetze, Dissertation, TU Ilmenau, 2018

Definitionen

- **Allokation:** Die Allokation (engl.: allocation) stellt den Prozess der Verteilung von Anforderungen, Ressourcen oder anderen Entitäten auf die Komponenten eines Systems oder Programm dar (ISO 24765)
 - Entspricht dem Verständnis unserer Diskussion im Arbeitskreis Funktionsverteilung
- **Dynamische Verlagerung:** Die dynamische Verlagerung (engl. dynamic relocation) ist die Verlagerung eines Computerprogramms zur Laufzeit (ISO 24765)
 - Entspricht dem Verständnis unserer Diskussion im Arbeitskreis Funktionsverteilung
- **Funktionsverlagerung (gekürzt):**
 - Fahrzeugapplikation, die Cloud-Rechen- & Speicherkapazitäten anstelle der verfügbaren Rechenkapazitäten des Fahrzeugs nutzen
 - Sowohl Informationen aus dem Fahrzeug (z.B. On-Board-Sensoren) als auch Daten aus der Cloud können als Eingabe verwendet werden
 - Die Ausgaben der verlagerten Funktionen sind für die Verbesserung von bereits vorhanden Funktionen im Fahrzeug oder zur vollständigen Ersetzung ebendieser bestimmt
 - Funktionen in der Cloud **ohne Einfluss auf die bordeigene Fahrzeugaktork sind nicht Bestandteil des Konzepts der Funktionsverlagerung**

V-Modell und XiL Zuordnung



Quelle: Szenariobasierte simulationsgestützte funktionale Absicherung hochautomatisierter Fahrfunktionen durch Nutzung von Realdaten, Dissertation, Raphael Pfeffer, 2020

Batteriebusse im Reichweitenvergleich

Typ: Standardbus, 12 Meter

- Problem: Die Reichweite von Batteriebussen erfüllt weiterhin nicht alle Anforderungen der Verkehrsbetriebe
- Eine alleinige Vergrößerung der Batterie löst das Problem nicht
- Eine Analyse der Einflussfaktoren auf die Reichweite offenbart den hohen Beitrag der Klimatisierung

Hersteller	Modell	Speicherkapazität	Max. Reichweite laut Hersteller
Alstom	Aptis	350 kWh	200 km
Ebusco	Ebusco 2.1	311 kWh	300 km
Mercedes Benz	eCitaro	243 kWh	150-200 km
Solaris Bus & Coach	Urbino 12 electric	300 kWh	190 km
BYD	BYD 12m	380 kWh	300 km

[1]

Sankey Diagramm der Energieaufteilung im Elektrobus

Speicher/Senken

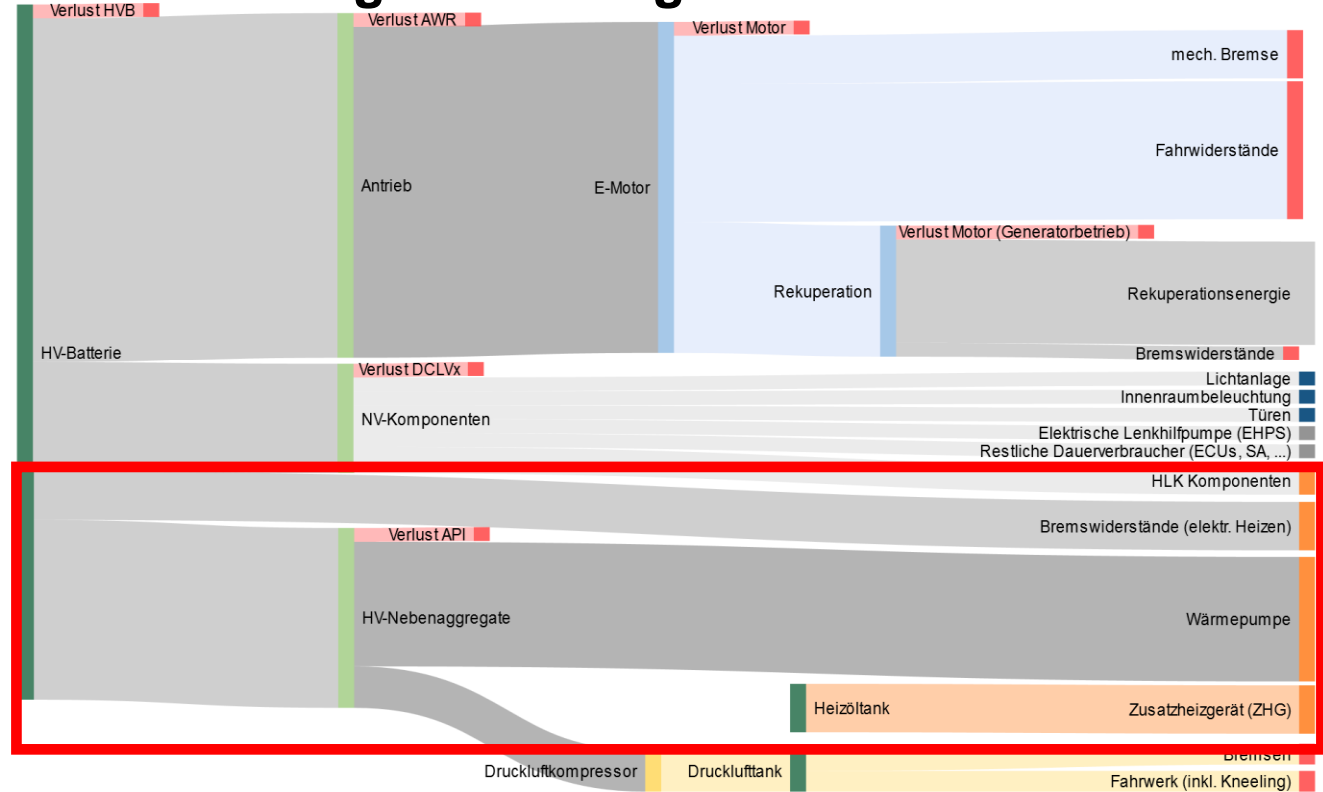
- Energiespeicher
- Bedarfsverbraucher
- Dauerverbraucher
- "Verlorene" Energie
- Wirkungsgradverluste

Wandler

- Elektrisch
- Kinetisch
- Pneumatisch
- Thermisch

Energieformen

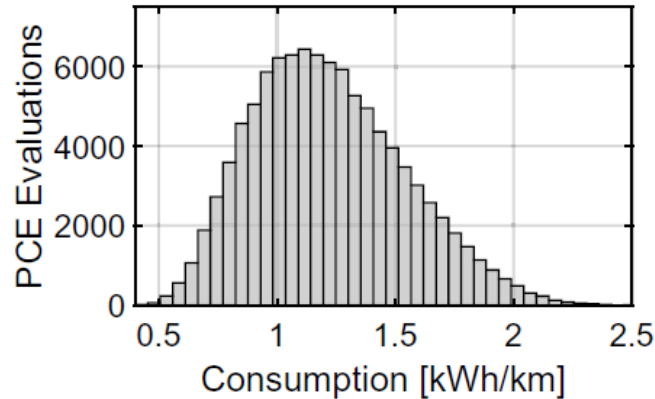
- dreiphasig
- Hochvolt
- Niedervolt
- Pneumatische
- Fossile



[2]

~30%

Energetischer Beitrag der Klimatisierung im elektrischen Stadtbus



- Bei einem durchschnittlichen Energieverbrauch eines 12 Meter Standard-Elektro-Stadtbusses von 1,2 kWh/km fallen bis zu 0,36 kWh/km für die Klimatisierung an

Modellierung des Energieverbrauchs für Elektrobusse (12m Standardbus) in 100.000 Polynomial Chaos Expansion (PCE) Modelldurchläufen

Metrik	Verbrauch
Durchschnittswert	1,20 kWh/km
Standardabweichung	0,32 kWh/km

[3]

XiL Komponenten

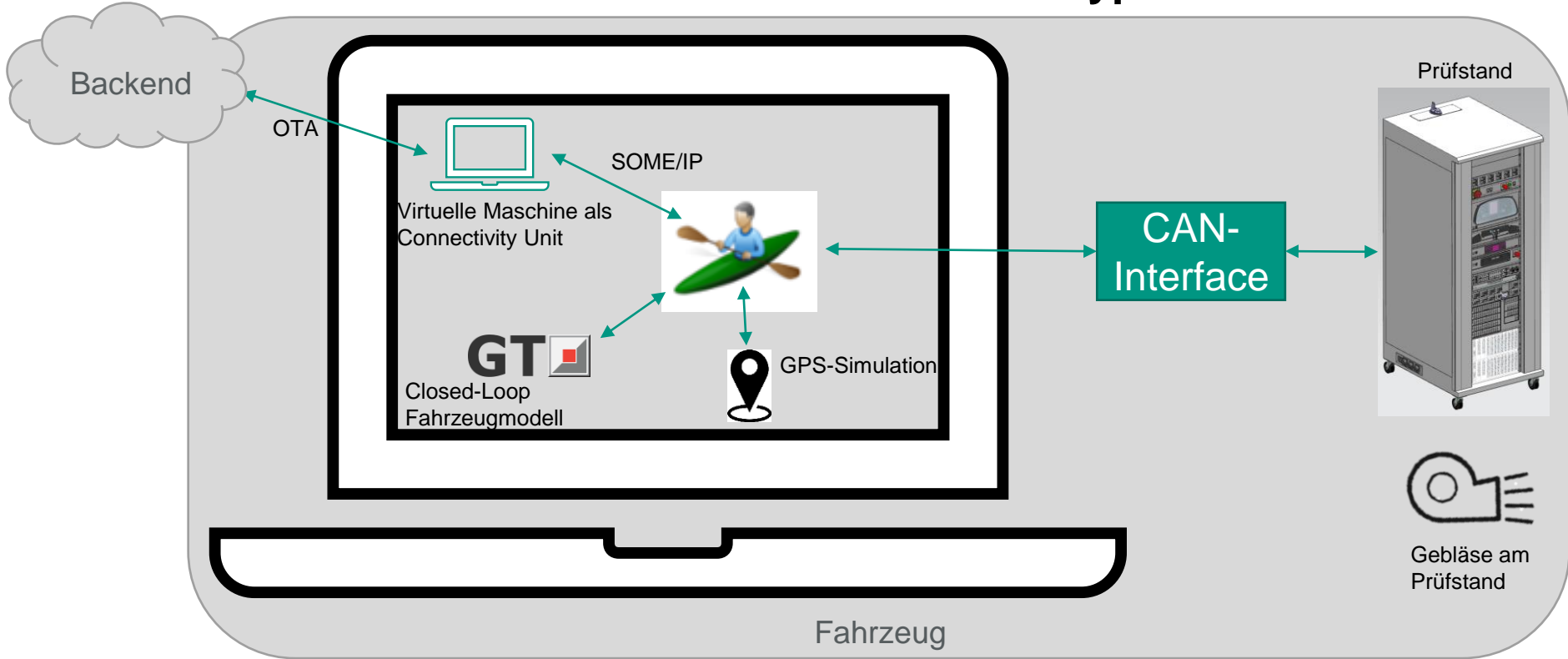
■ Virtuelle und reale Komponenten verschiedener Testmethoden

	MiL	SiL	ECU-HiL	System-HiL	Rollenprüfstand	ViL	Fahrversuch
Funktions-Code	V	R	R	R	R	R	R
Steuergerät	V	V	R	R	R	R	R
System	V	V	V	R	R	R	R
Fahrzeug	V	V	V	V	R	R	R
Fahrer	V	V	V	V	V/R	V/R	R
Fahrdynamik	V	V	V	V	V	R	R
Erlebbarkeit	V	V	V	V	V	R	R
Fahrbahn	V	V	V	V	V	R	R
Verkehr/Umfeld	V	V	V	V	V	V	R

V: virtuell, R: real

Quelle: Szenariobasierte simulationsgestützte funktionale Absicherung hochautomatisierter Fahrfunktionen durch Nutzung von Realdaten, Dissertation, Raphael Pfeffer, 2020

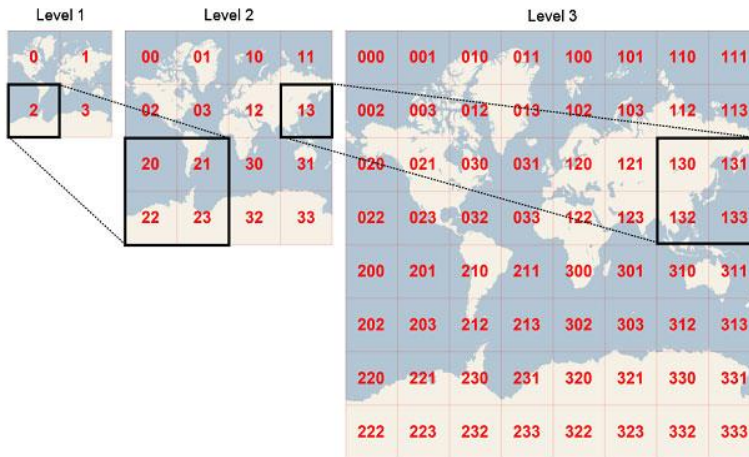
Hardware und Software für den Prüfstand Prototyp



Routenerkennung & JSON Datenbank

Geo_FindTrip

- Die Routenerkennung wurde mithilfe eines QuadTile basierten Systems entwickelt



QuadTiles

Schwartz, J. (2018). Bing Maps Tile System - Bing Maps | Microsoft Docs. Retrieved November 26, 2019, from <https://docs.microsoft.com/en-us/bingmaps/articles/bing-maps-tile-system>

Level	# Tiles	Tile width (° of longitudes)	m / pixel (on Equator)	~ Scale (on screen)	Examples of areas to represent
0	1	360	156 412	1:500 million	whole world
1	4	180	78 206	1:250 million	
2	16	90	39 103	1:150 million	subcontinental area
3	64	45	19 551	1:70 million	largest country
4	256	22.5	9 776	1:35 million	
5	1 024	11.25	4 888	1:15 million	large African country
6	4 096	5.625	2 444	1:10 million	large European country
7	16 384	2.813	1 222	1:4 million	small country, US state
8	65 536	1.406	610 984	1:2 million	
9	262 144	0.703	305 492	1:1 million	wide area, large metropolitan area
10	1 048 576	0.352	152 746	1:500 thousand	metropolitan area
11	4 194 304	0.176	76 373	1:250 thousand	city
12	16 777 216	0.088	38 187	1:150 thousand	town, or city district
13	67 108 864	0.044	19 093	1:70 thousand	village, or suburb
14	268 435 456	0.022	9 547	1:35 thousand	
15	1 073 741 824	0.011	4 773	1:15 thousand	small road
16	4 294 967 296	0.005	2 387	1:8 thousand	street
17	17 179 869 184	0.003	1 193	1:4 thousand	block, park, addresses
18	68 719 476 736	0.001	0 596	1:2 thousand	some buildings, trees
19	274 877 906 944	0.0005	0 298	1:1 thousand	local highway and crossing details
20	1 099 511 627 776	0.00025	0 149	1:5 hundred	A mid-sized building

Quelle: Open Street Maps

Routenerkennung & JSON Datenbank

Geo_UpcomingData

- Routenerkennung rein auf Basis der GPS Daten
- GPS Signal wird einer QuadTile zugeordnet
- Zu jeder QuadTile (zoom level 17) sind in einer JSON Datenbank eine Vielzahl an Informationen zugeordnet (aus Google Transfer und OpenStreetMap)

```
{  
  "ID": 2238473365,  
  "center_latitude": 48.7586975,  
  "center_longitude": 8.8357544,  
  "amenities": 1,  
  "crossing": 0,  
  "maxSpeed": 30,  
  "trips": [  
    "1.T0.76-670-j21-7.1.H",  
    "104.T0.76-670-j21-7.39.R",  
    "1.T0.76-880-j21-2.1.H",  
    "22.T0.76-880-j21-2.16.R"  
  ]  
}
```

- Berechnete Tile ID
- Zentrale QuadTile Koordinaten
- Einrichtungen: # Cafes, Banken, Zivilgebäude
- Max. Geschwindigkeit
- Trips, die dieses QuadTile durchqueren

- Jeder Trip wird zudem in einer gesonderten JSON Struktur gespeichert
- Dadurch kann schnell auf Informationen, wie nächste Haltestellen etc zugegriffen werden
- Ein Trip Node, beispielsweise eine Haltestelle eine Kurve etc., wird wie folgt innerhalb einer Trip JSON Datei abgespeichert

```
{  
  "tile": 2238486072,  
  "latitude": 48.9433836,  
  "longitude": 9.264515,  
  "azimuth": -133,  
  "stop": false,  
  "distance": 35.74  
}
```

- Zugehörige Tile ID
- Zentrale QuadTile Koordinaten
- Azimut: Azimut Winkel des Trips zwischen letztem und jetzigem Node
- Haltestelle
- Gesamtdistanz [m] gefahren seit Beginn des Trips

Datengetriebene Vorhersagemodelle

predict_disturbances

- Die datengetriebenen Modelle sind alle auf Basis von Long-Short-Term-Memory (LSTM) Netzen entstanden
- Die Input Parameter der einzelnen Netze wurden auf Basis von Korrelationsmatrizen erarbeitet
- Die Hyperparameter dieser Netze wurde mithilfe Bayes'scher Optimierung angepasst
- Es stellte sich heraus, dass eine Vorhersage der Motordrehzahl mit den gegebenen Input Parametern nicht realistisch umsetzbar ist
 - Das trainierte Netz wird aber verwendet
 - Andere datengetriebene Vorhersagemodelle zeigen eine gute Performance

Test der Routenerkennung

Geo_UpcomingData

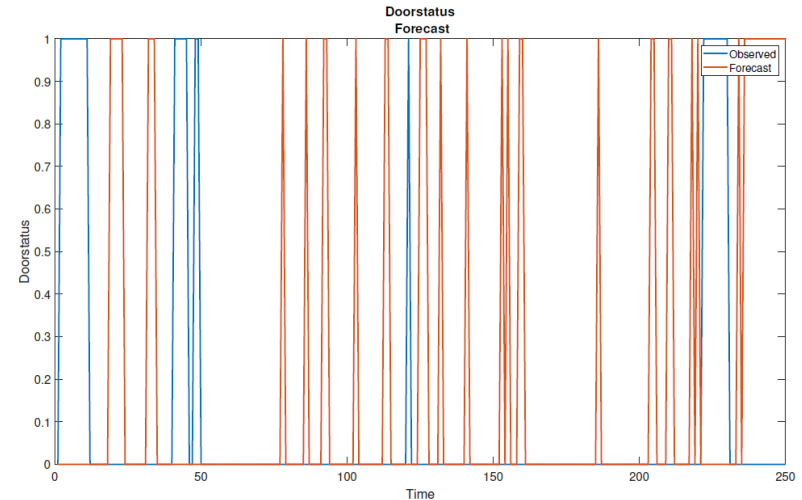
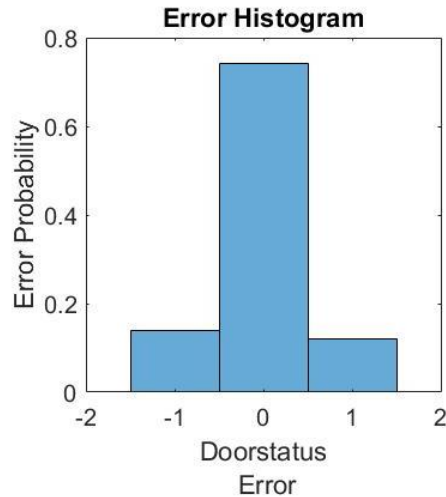
- Routenerkennung auf Basis der SSB Daten
- Datensatz: PrioOneDump_17-03-2021.csv
- Skript liest GPS Daten ein und ordnet diese zunächst QuadTiles und anschließend Routen zu
- Ergebnis bei Test der Zeitschritte: 4450-4550 (09.03.2021 12:40 Uhr GMT) → Route 44
 - Oftmals liegen Routen aufeinander → Aussage über Route ist dann schwierig. Eine klare Unterscheidung zwischen Nachtbus und Normalbetrieb ist nicht allgemein durchführbar

```
>> TestRoute  
Route ID: 30-44-j21-1  
Route Short Name: 44  
Route Long Name: Killesberg - Hbf - Charlottenpl. - Westbahnhof
```


Test der Türöffnungsprädiktion

Geo_UpcomingData

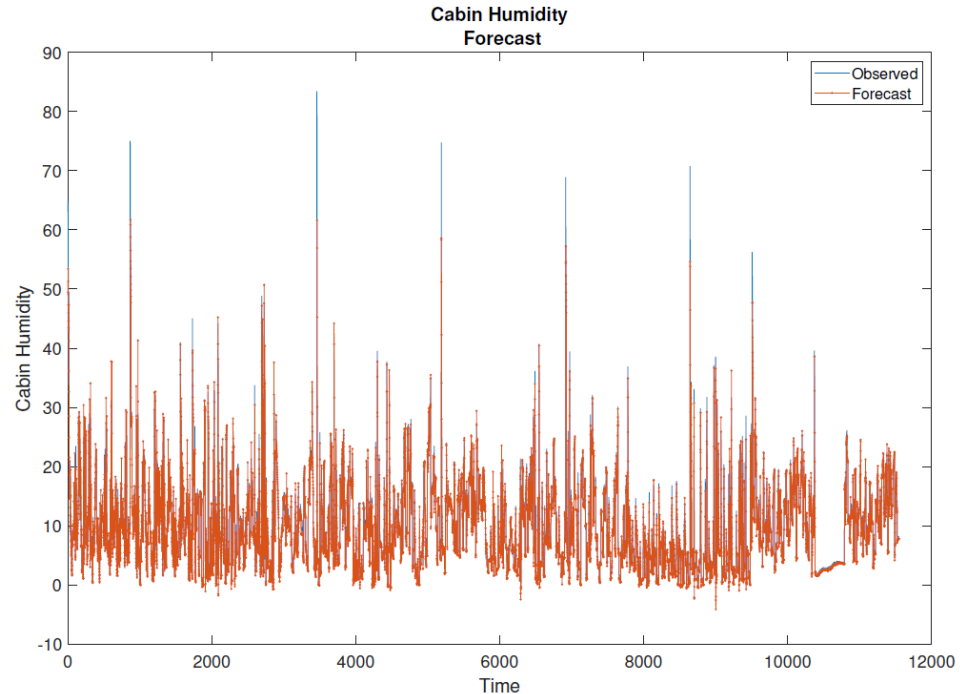
- Eine Vorhersage von Haltestellen wurde auf Basis der aufgezeichneten SSB Daten durchgeführt
- Ein Halt des SSB Busses wurde über ein stehendes GPS Signal definiert (+Offset)
- Ein Vergleich wurde auf Basis von geplanten Halten auf Route (distanzbasiert) und dem stehenden GPS Signal des Fahrzeugs (interpretiert als Haltestelle) durchgeführt



Test des datengetriebenen Modells der Kabinenluftfeuchtigkeit

predict_disturbances

- Die Input Kanäle des Kabinenluftfeuchtigkeitnetzes lauten:
 - Kabinenluftfeuchtigkeit (Vergangenheit)
 - Passagiere
 - Türöffnungen
 - Äußere Luftfeuchtigkeit
 - Frischluftklappe (Dach)
- Das Netz zeigt einen Fehler von unter 1% Luftfeuchtigkeit



SOMEIP Funktionsaufrufe

Req/Response	Signalname	Signalfluss
Request	DateTimeSecond....Year	Rx from Vehicle
	GPS Longitude/Latitude	Rx from Vehicle
	Vehicle Weight	Rx from Vehicle
	ambientTemperature	Rx from Vehicle
	CabinTemperature	Rx from Vehicle
	Outblow Temperature	Rx from Vehicle
	Ambient Humidity	Rx from Vehicle
	Cabin Humidity	Rx from Vehicle
	Cooling Water Temperature	Rx from Vehicle
	Engine RPM	Rx from Vehicle
	Cabin Recirculation Flap	Rx from Vehicle
	Vehicle Speed	Rx from Vehicle
	Door Status	Rx from Vehicle
Response	Cabin outblow Temp Set Value	Tx to Vehicle
	Cabin Blower PWM	Tx to Vehicle

Data Logs SSB - Extracts

Latitude	Longitude	Speed	Weight	Door Status
4875203	924162	0	0	3
4875222	924128	0	0	3
4875279	924047	0	0	3
4875295	923928	0	0	3
4875320	923927	0	0	3
4875346	924083	0	0	3
4875393	924240	0	0	3

■ Ssb_20210923_1

- Speed + Weight immer 0

Latitude	Longitude	Speed	Weight	Door Status
4876190	926890	4238	1689	0
4876187	926892	3530	1686	0
4876185	926895	2517	1686	0
4876183	926896	1049	1686	0
4876183	926896	465	1688	0
4876183	926896	255	1688	0
4876183	926896	0	1688	1
4876183	926896	0	1687	1
4876183	926896	0	1687	1
4876183	926896	0	1687	1
4876183	926896	0	1687	1

■ PrioOneDump_22_07_2021

- Vehicle weight bleibt nur bei geöffneter Tür gleich, ansonsten stark schwankend

Vorteile einer Model Predictive Control

Vorteil	Ausprägung
Stellgrößenbeschränkungen können vorgegeben werden	<ul style="list-style-type: none">• Wärmepumpe der Aufdachanlage (Kompressor und Lüfter)• Umluftklappe
Zustandsgrößenbeschränkungen können vorgegeben werden	<ul style="list-style-type: none">• Temperatur- und Frischluftvorgaben
Vorhersagen von Störgrößen können für eine optimale Stellgrößenstrategie verwendet werden	<ul style="list-style-type: none">• Wetter• Passagiere• Routeninformationen
Formulierung konkurrierender Regelziele in Form einer Kostenfunktion	<ul style="list-style-type: none">• Geringe Leistungsaufnahme• Einhaltung des Komfortbereichs

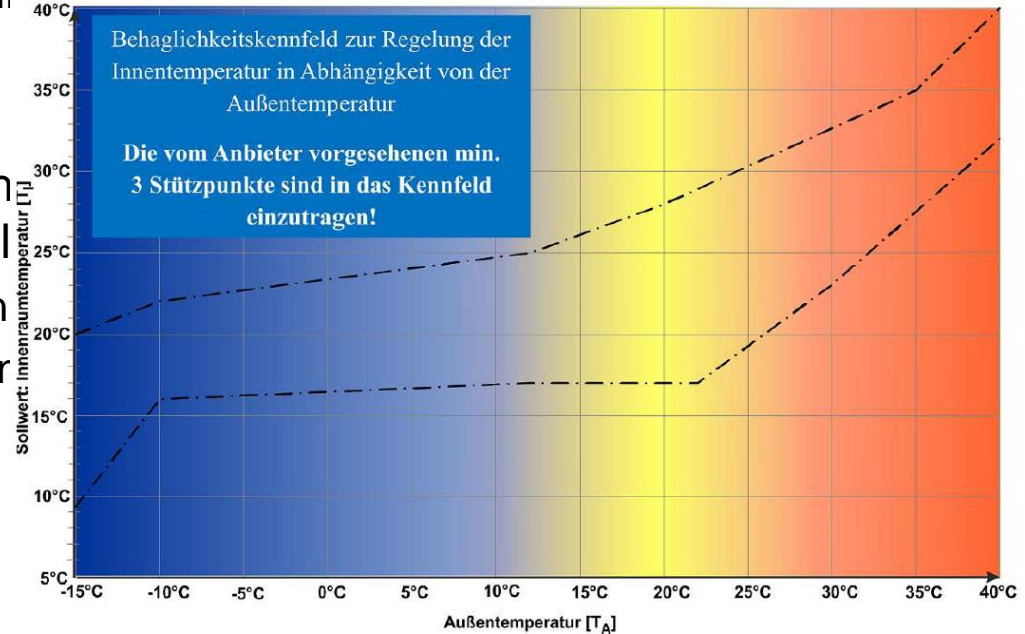
3 Klimatisierungsvorgaben des Verbands der deutschen Verkehrsunternehmen (VdV)

■ Beschreibung von zwei Behaglichkeitskennlinien

- Economy
- Komfort

■ Die Economy-Kennlinie ist für Fahrgäste und E-Mobilität sinnvoll

- Heizen im Bereich der unteren Kennlinie
- Kühlen im Bereich der oberen Kennlinie

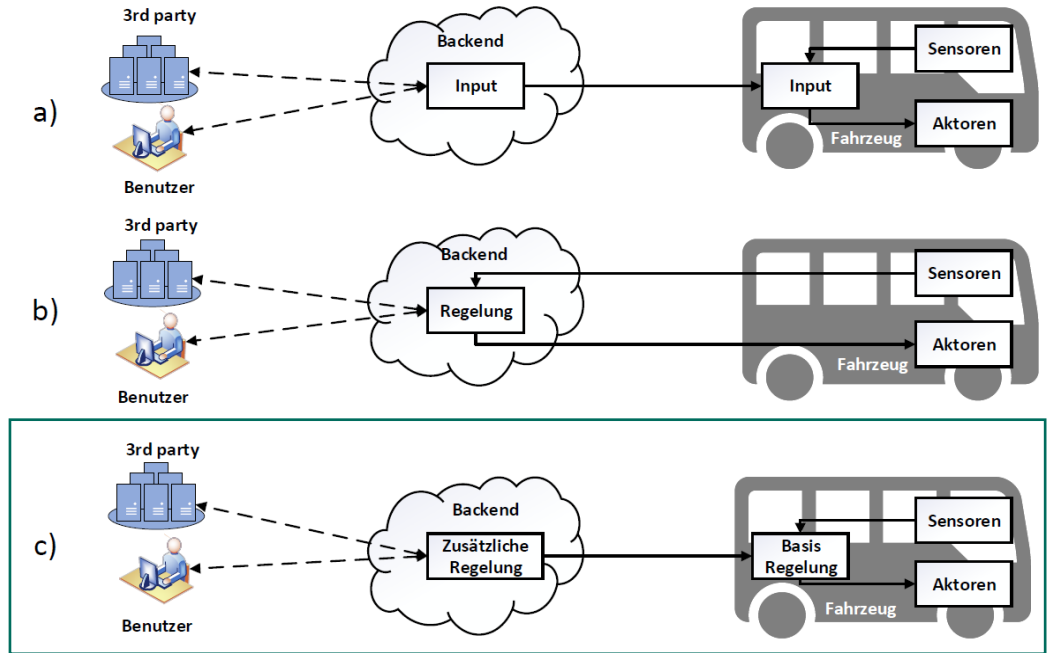


[3]

Verteilte Fahrzeugfunktionen

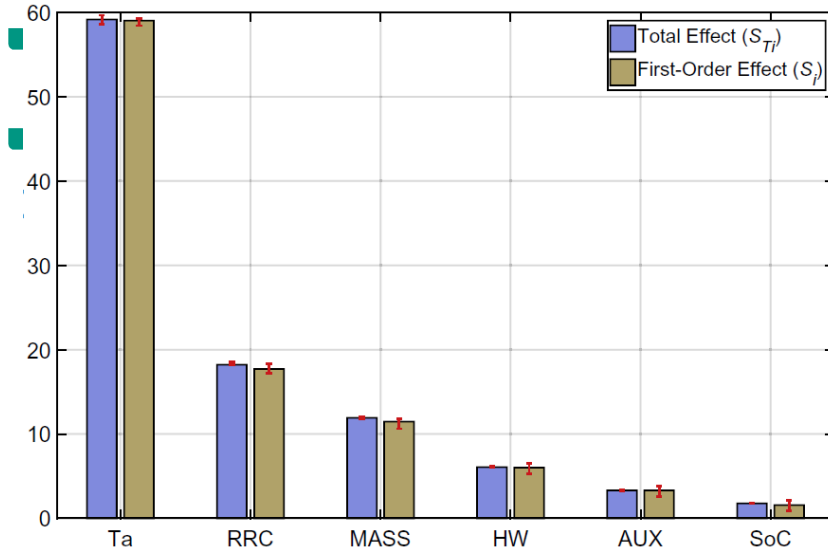
■ Verteilte Fahrzeugfunktionen lassen sich in drei wesentliche Typen unterscheiden:

- a) Backend stellt zusätzlichen Input
- b) Die Fahrzeugfunktion wird kor
- c) Verfügbarkeit einer zusätzliche



[6]

Einflussgrößen auf den Energieverbrauch von Elektrobussen



[9]

Die Masse des Elektrobusses haben einen
 den größten Einfluss auf den Energieverbrauch des Fahrzeugs.

Der Energieverbrauch des Fahrzeugs ist auf den hohen
 Werten

Faktor	Beschreibung	Wertebereich	Einheit
Ta	Außentemperatur	[-30, 35]	°C
RRC	Rollwiderstands- koeffizient	[0.006, 0.02]	-
MASS	Masse des Fahrzeugs	[8500, 15000]	kg
HW	Gegenwind	[-10, 10]	m/s
AUX	Nebenverbraucher (Hydraulik etc.)	[2, 7]	kW