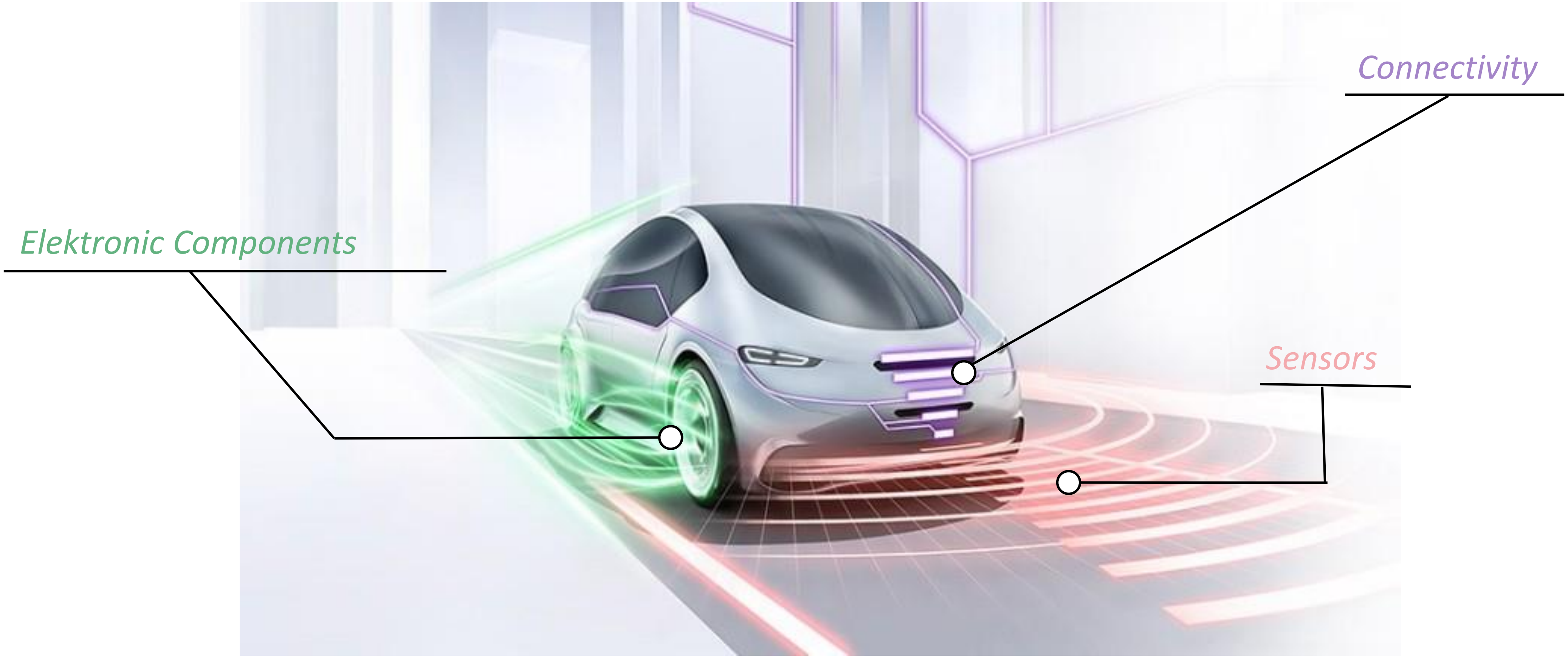
Towards a Model-driven Testing Approach for Microservice Architectures in the Automotive Domain

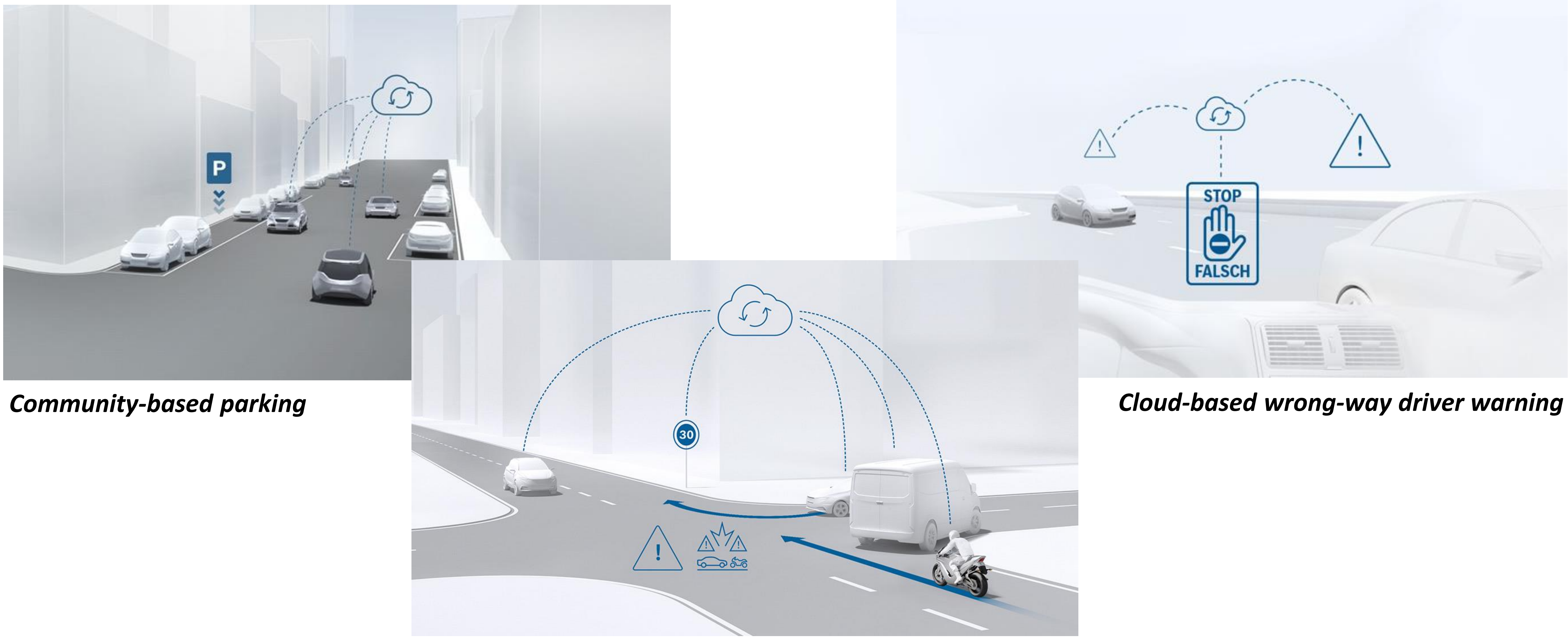
Philipp Heisig (philipp.heisig@fh-dortmund.de) | Sabine Sachweh
Institute for the Digital Transformation of Application and Living Domains (IDiAL)
FH Dortmund - University of Applied Sciences and Arts

Motivation

Electric, Connected & Autonomous Vehicles



Next-generation Mobility Services



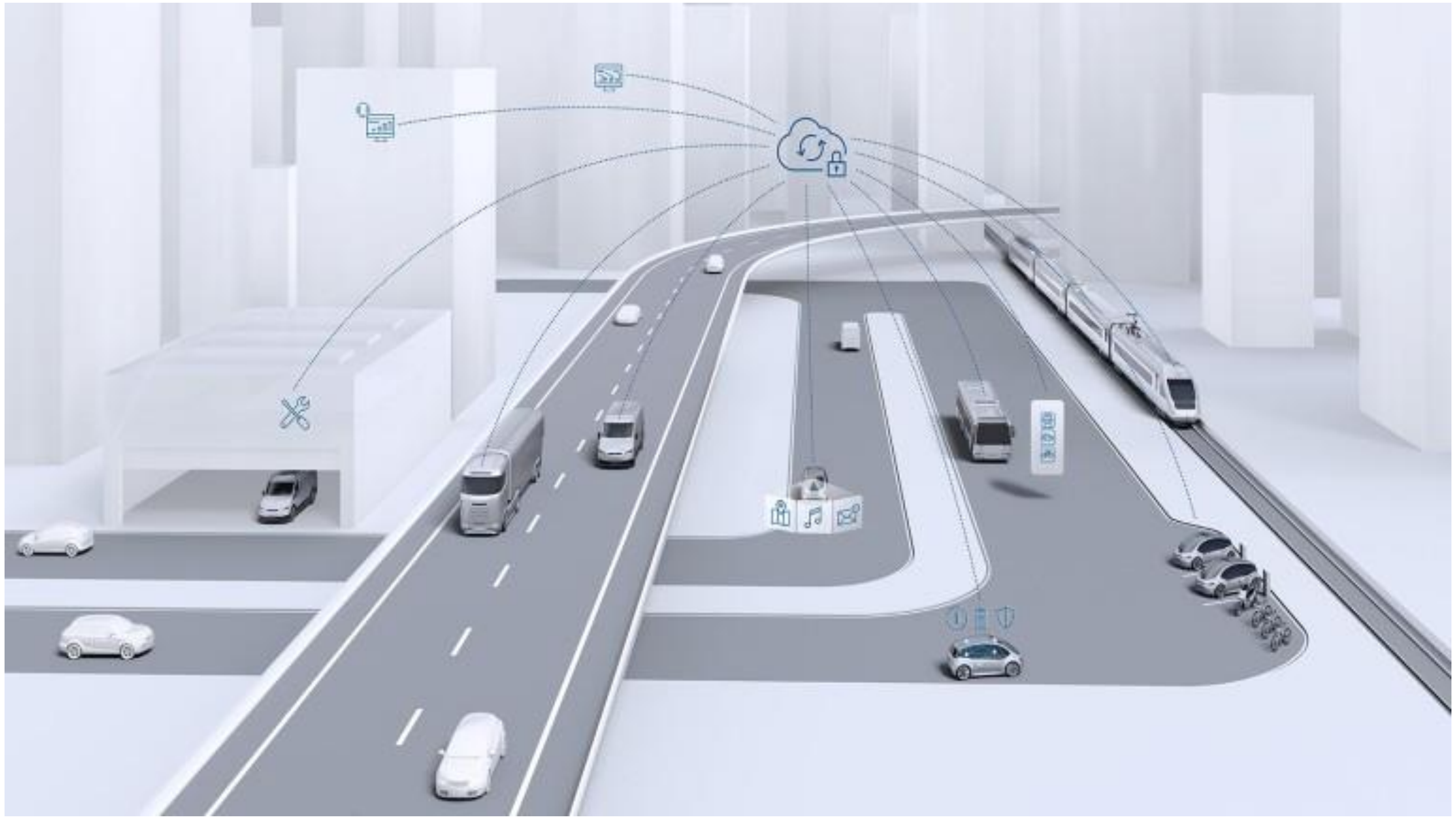
Community-based parking

Cloud-based wrong-way driver warning

Pre-crash warning

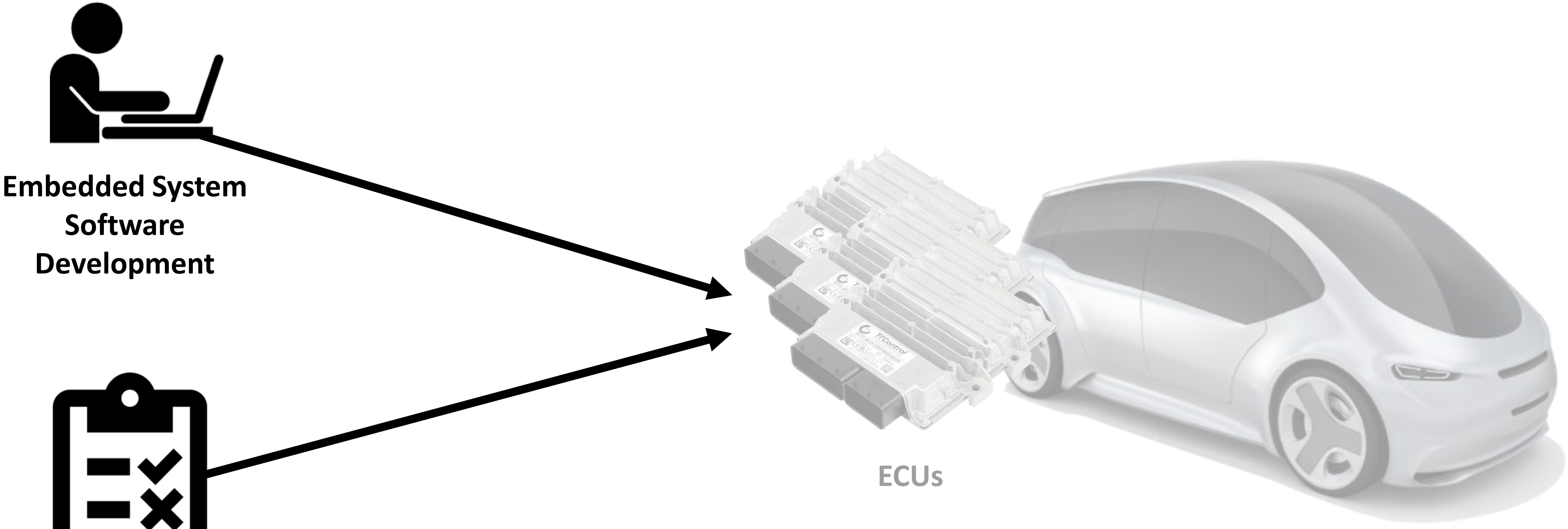
Source: https://www.bosch-mobility-solutions.com/

Major Element of the IoT



Source: https://www.bosch-mobility-solutions.com/

Previous Development of Automotive Software

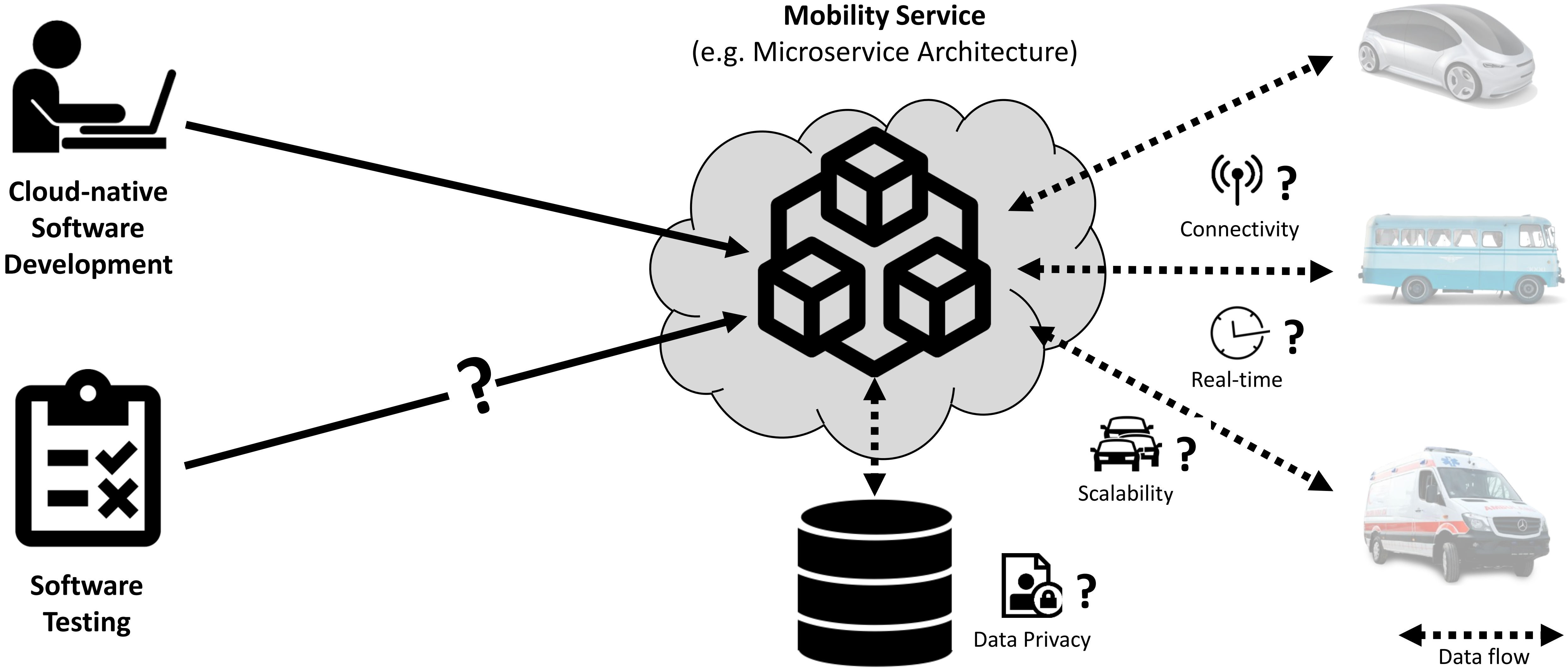


Embedded System Software Development

Software Testing

ECUs

New Dimensions in Automotive Software Development



How to Test Cloud-based Mobility Services?

- **Microservice Architectures** are likely used to realize the cloud counterpart of next-generation mobility services [1, 2, 3, 4]
- **New testing dimension**: Cloud-based software components and the interactive nature of connected mobility domain
  - **Massive amount of vehicle-specific data** need to be fed into the services for a validation of
  - Also **environmental conditions**, such as changing connectivity, must be considered
- **Test drives** for generating real vehicle data are not always possible and cost-intensive
- Setting up many **hardware and vehicle nodes** to generate vehicle-specific data
- **Dummy data** exhibit a lack of semantics and variance in the data

Virtual Test Environment

- Testing methodologies are required that **do not exhibit any real hardware components**

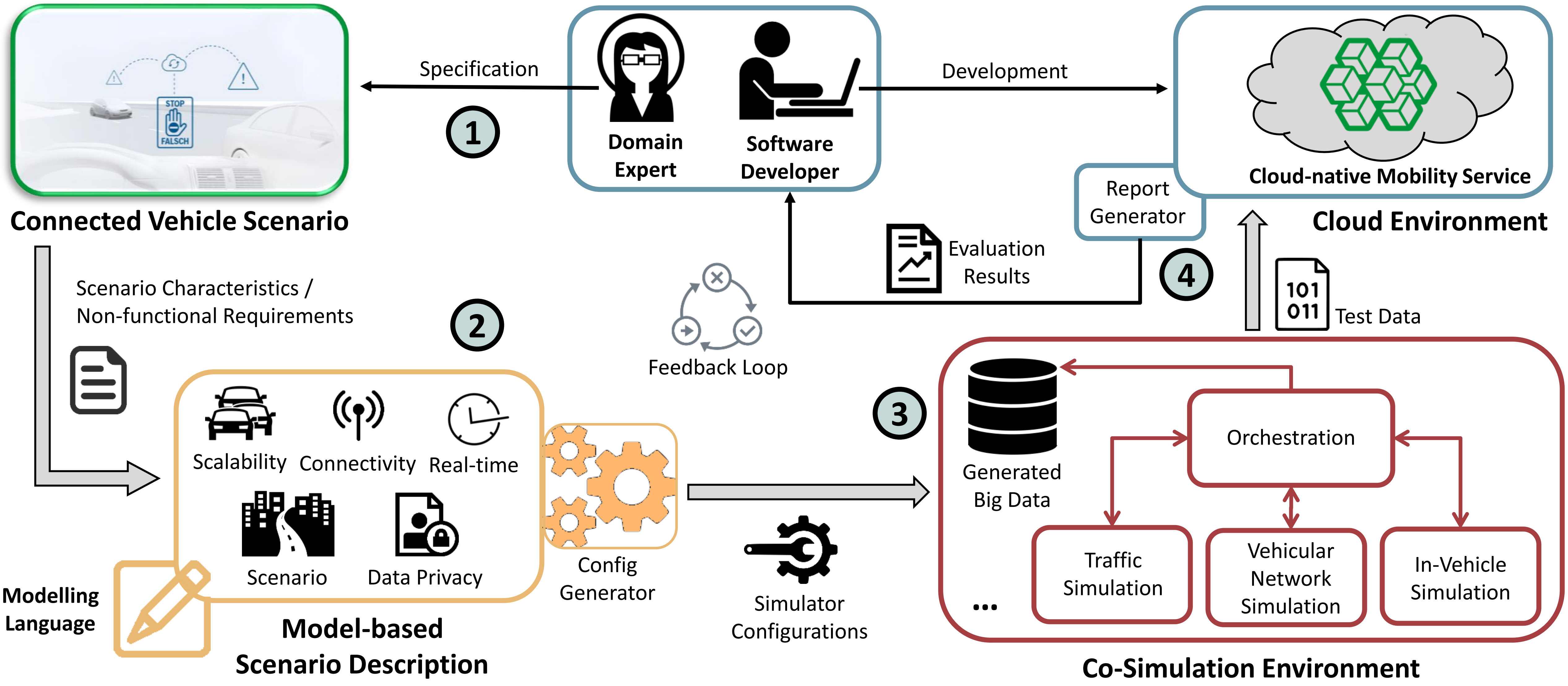
- **Virtual test environment** that can be easily set up and used for various scenarios
- **Simulators** are one way to enable a virtual validation of such mobility services
- Simulation-based tests have several advantages over real tests
  - Cheaper than real tests
  - Can be replicated almost unlimited
  - Allow for a proof-of-concept design and evaluation at early stages in the development process

- **Problem**: Setting up a simulation environment involves a lot of different challenges
  - Extensive domain knowledge necessary
  - Provision of realistic and reusable simulation scenarios
  - Simulators are usually specialized in reproducing certain aspects, e. g. microscopic traffic simulations

➢ Integration of several simulators within a **Co-simulation environment** to create virtual prototypes

Virtual Testing of Cloud-native Mobility Services

Model-driven Co-simulation Framework



Connected Vehicle Scenario

Specification

1

Domain Expert

Software Developer

Development

Cloud-native Mobility Service

Cloud Environment

Report Generator

Evaluation Results

4

Test Data

Scenario Characteristics / Non-functional Requirements

2

Feedback Loop

3

Scalability

Connectivity

Real-time

Scenario

Data Privacy

Config Generator

Simulator Configurations

Modelling Language

Model-based Scenario Description

Generated Big Data

Orchestration

...

Traffic Simulation

Vehicular Network Simulation

In-Vehicle Simulation

Co-Simulation Environment

Goals

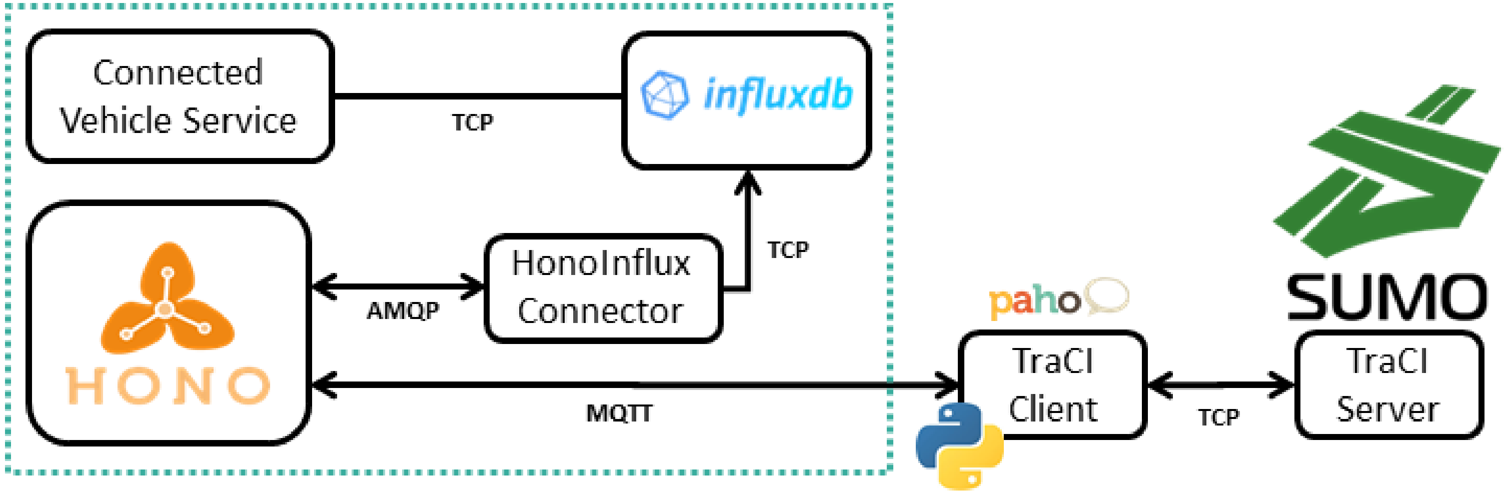
- **Model-based description** of relevant testing aspects from use case specification
- (Semi-) Automatically **deriving co-simulation environment** from test scenario models
  - Simulation orchestration: Open source frameworks like **Eclipse MOSAIC**
- Reusable test scenarios
  - Test data can be persisted and test scenarios reproduced
  - Reusing simulations but varying aspects, e.g. communication technologies such as 5G, LTE-V2X etc.
- **Feedback loops** to continuously enhance software quality
- **MSA-specific metrics** to detect architectural smells and microservices anti pattern [5] as well as security issues
  - e.g. Hard-Coded Endpoints, No API Gateway, Cyclic Dependency, Shared Persistency and Libraries, API versioning
  - Does the service comply with security policies?

https://projects.eclipse.org/proposals/eclipse-mosaic

Conclusion & Outlook

First results

- Using a traffic simulator for generating large-scale vehicle data [6]
  - Eclipse SUMO used to simulate traffic scenarios including microscopic properties like the position or emission
- Feedback about the functionality of the service itself
  - Problems with data storage, GUI, and API-usage
- Testing the scalability of connected vehicle IoT architectures
  - 1.532.783 MQTT messages have been sent to the cloud between 06:00 and 06:15 in the simulation



Takeaway Points

- Scenario-driven validation of cloud-based mobility services presented
- Can be applied at different phases in the development process to continuously…
  - … assess and improve the software architecture
  - … ensure the correct behavior of the service functionality
- The approach is not limited to the automotive domain and could also be also applied to other domains by…
  - … using subset of the simulators
  - … providing support for additional simulators from other domains (via new config generators)
- Several questions regarding the assessment of (automotive) MSAs arises:
  - Can technical debt be reduced when applying testing at early stages of the development process?
  - How to detect and measure anti patterns?
  - How do MSAs in the automotive domain differ from other domains?
  - How to describe security policies and validate them accordingly?

References

[1] Abeck, Sebastian, et al. "A Context Map as the Basis for a Microservice Architecture for the Connected Car Domain." *INFORMATIK 2019: 50 Jahre Gesellschaft für Informatik – Informatik für Gesellschaft* (2019).

[2] Datta, Soumya Kanti, et al. "Iot and microservices based testbed for connected car services." *2018 IEEE 19th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*. IEEE, 2018.

[3] Lotz, Jannik, et al. "Microservice Architectures for Advanced Driver Assistance Systems: A Case-Study." *2019 IEEE International Conference on Software Architecture Companion (ICSA-C)*. IEEE, 2019.

[4] Schneider, Tobias, and A. Wolfsmantel. "Achieving Cloud Scalability with Microservices and DevOps in the Connected Car Domain." *Software Engineering (Workshops)*. 2016.

[5] Taibi, Davide, Valentina Lenarduzzi, and Claus Pahl. "Microservices Anti-Patterns: A Taxonomy." *Microservices*. Springer, Cham, 2020. 111-128.

[6] Heisig, Philipp, et al. "Bridging the Gap between SUMO & Kuksa: Using A Traffic Simulator for Testing Cloud-based Connected Vehicle Services." *SUMO*. 2019.