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

Philipp Heisig[1] and Sabine Sachweh[1]

IDiAL — Dortmund University of Applied Sciences & Arts,
philipp.heisig@fh-dortmund.de

Abstract

Connectivity and digitization have turned vehicles into smart devices which continuously increasing their functionality by interacting with their environment and sharing multi-modal mass data among people, businesses, service providers, or OEMs within the context of the Internet of Things. While microservices features scalability and flexibility, they are especially suitable for realizing mobility services that are deployed in cloud infrastructures. However, as connected vehicle scenarios have to consider environmental conditions, such as a changing connectivity, the according cloud-based mobility services must be extensively tested at different stages of the development process to get continuous feedback on how well-designed the software architecture and the service itself is. Nevertheless, testing requires mass data from real world scenarios to ensure both the proper functionality of the service and that the architecture is robust and scales with a varying number of vehicles. This paper addresses this problem by proposing a model-driven test framework for the virtual validation of cloud-based mobility services. Therefore, different simulators will be coupled within a co-simulation environment to generate massive amounts of vehicle-specific data and simulate the environmental conditions in which vehicles operate. This allows a scenario-driven validation if a service react flexibly (scalability), reliably (robustness), and with the intended behavior to service requests from vehicles.

1 Introduction

Technological advances, digitization, and area-wide mobile Internet have transformed vehicles into software-based high tech products with built-in connectivity and autonomous driving features. The rapid growth and the tremendous number of connected vehicles on the road makes the connected vehicle domain a major element of the Internet of Things (IoT) [3]. In general, these vehicles are characterized by the massive amounts of multi-modal data provided by various sensors, the behavior of consumers, and the interaction with the environment. Edge computing combined with smart infrastructures allows to even generate more informative and synthetic data in this context. By collecting, exchanging, fusing, analyzing, and processing this mass data in multiple and simultaneously operating applications within the Cloud, innovative and data-driven mobility service can be realized, spanning from road safety over smart, efficient, and green transportation to location-dependent services [7]. Thus, future connected vehicles will exhibit a cloud-based vehicle architecture allowing a much higher level of integration and an increased number of use cases [4]. However, vehicles operate in a safety-critical and time-sensitive environment with changing conditions. Thus, unreliable vehicle connectivity with changing data transfer rates must be expected and real-time processing of the resulting data may be necessary for autonomous driving features. Furthermore, cloud-based mobility services have to scale with the high number of vehicles on the road, while the architecture has to process also a variance in data stemming from different types of vehicles.

While the Microservice Architecture (MSA) style denotes a promising solution for the design and implementation of cloud-based mobility services [8], testing and validation of the deployed microservices poses several new challenges that go beyond the requirements of other IoT domains [10]. As such services running in the Cloud and serving multiple and a varying number of vehicles on demand, they must be tested with a broad range of connected vehicles including their interaction with the environment [6]. Thus, a massive amount of vehicle-specific data need to be fed into the services for a validation regarding their proper functionality and the fulfillment of non-functional requirements, in particular scalability and reliability. However, such a validation cannot be accomplished by occasionally applied approaches: Test drives based on a vehicle fleet or setting up a large number of hardware and vehicle nodes to generate vehicle-specific data are not feasible from an economic and operational perspective [2] as they are cost-intensive and require special domain knowledge. Especially in early development phases, the risks and legal requirements stand in the way of extensive test drives. While the use of so-called dummy data would allow to validate the scalability and robustness of software architectures to a certain extent, the lack of semantics and variance in the data sets does not allow to validate the functionality of the services. In addition to the provision of meaningful data sets, also environmental conditions like a changing connectivity have to be taken into account for the validation process. Thus, novel and open testing methodologies are required that do not exhibit any real hardware components, but rather foster a virtual test environment that can be easily set up, replicated, and used for various connected vehicle scenarios.

2 Virtual Testing Cloud-native Mobility Services

One way to enable a virtual validation of such mobility services is the usage of simulators. Simulators have been a widely used tool within the automotive sector for decades as they allow for a proof-of-concept design and evaluation by spanning both virtual and physical domains [1]. Not only they are cheaper than real tests, they can also be carried out much earlier in the development process to provide valuable feedback to service developers. Nevertheless, simulators are usually specialized in reproducing certain aspects, e. g. simulating vehicular networks. Thus, different kind of simulators have to be interconnected in a co-simulation to support all of the previously described validation aspects. Setting up such a co-simulation environments is, however, a complex and time-consuming task, that requires a lot of domain knowledge. Developers that are experienced in implementing MSAs or developers that come from a different domain, such as insurance companies or breakdown services, may do not have this domain knowledge or want to focus on the actual application and possible innovations. Figure 1 proposes a model-based approach for abstracting the complexity by generating an adequate simulation environment for testing mobility services and their Cloud architectures via a model-based scenario description. In this way, service providers can be supported with the realization of innovative mobility services by getting continuous feedback along the development process (feedback loop), especially in early phases when software architectures are designed. The following steps describe the approach and the integration at any stage of a development process:

1. *Scenario Specification:* The first step is the definition of a connected vehicle scenario (*Scenario Specification*) including its requirements, which is usually done by a domain expert in natural language. Based on this, developers can start to define a first software architecture sketch and implement basic functionality. Likely, the architecture will be a MSA to feature scalability and flexibility.
2. *Modeling and Configuration:* Within the second step, the *Scenario Specification* will be formalized via a Domain-Specific Modeling Language (DSML) that is designed for describ-

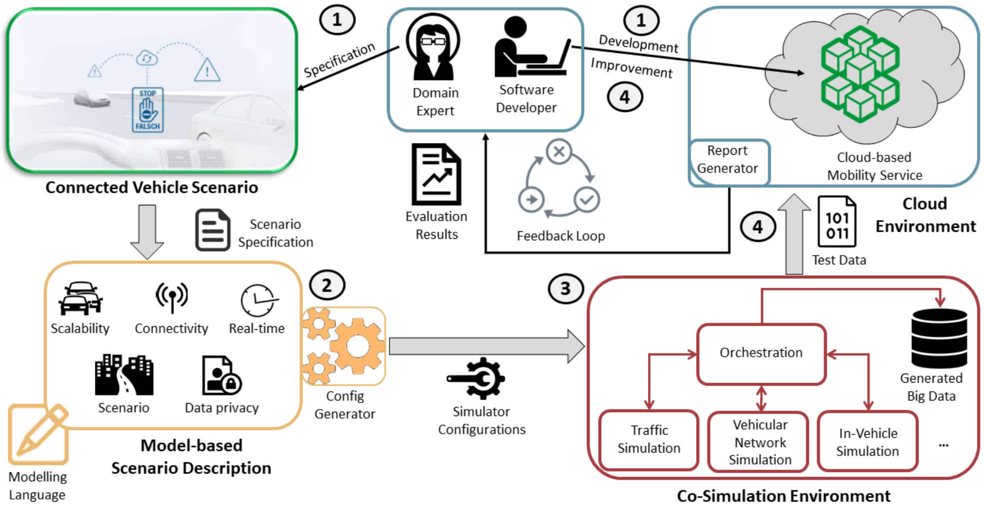

Figure 1: Model-driven co-simulation framework for the virtual validation of mobility services

ing non-functional requirements towards cloud-based mobility services, e. g. how many and what types of vehicles should be simulated. In addition, the DSML can be extended to capture domain-independent testing aspects like data privacy. The resulting *Scenario Model* act as input for a set of *Config Generators*, which automatically generate configurations for each simulator used in the scenario. Depending on the simulation tool and in which way it can be configured, model-to-model or model-to-text transformations can be applied for the generation process.

3. *Co-simulation:* While the configurations allows to set up each simulator independently, they still need to be integrated into a co-simulation environment including a component that is responsible for orchestrating and controlling the simulation flow of each simulator to enable interoperability among the simulators. The simulation environment then allows to generate large amount of semantically enriched vehicle data on different level of detail. For a replication of tests or to carry out tests at any time, the generated data will be persisted in a data base.

4. *Feedback Loop:* In the last step, test data from the simulation will be contentiously fed into the mobility service to test both the service functionality and the software architecture behind. Predefined metrics asses the architecture against the different non-functional requirements defined in the *Scenario Model*, such as response time or the amount of vehicles that have been simultaneously served. But also metrics specifically developed for MSAs can be integrated to identify, for example, microservices anti pattern [9] such as API versioning or hardcoded endpoints. Another example for such a MSA metric would be to measure the cyclic dependency, i. e. the amount of inter-service communication. The test results are then generated by a *Report Generator* and allow the developer to improve the architecture and service implementation and test it again, either via new data sets or based on the previous one to establish a (*Feedback Loop*).

3 Conclusion & Future Work

In this paper, we proposed an approach for a scenario-driven validation of cloud-based mobility services that can be applied at different phases in the development process to continuously asses and improve the software architecture as well as ensure the correct behavior of the service functionality. Especially the technical debt can be reduced when applying testing already at early stages of the development process. A first proof of concept with one traffic simulator for the previously described approach has been published in [5]. The results demonstrated, that traffic simulators are in general suitable for testing functional and non-functional requirements of cloud-based mobility services, but that connected vehicle specific aspects need more thorough investigations. Although the focus in this paper is on testing MSAs in the automotive domain, the approach can be also applied to other domains by using a subset of the simulators or providing support for additional simulators from other domains.

Among the implementation of the different components in Figure 1, we are planning for the future to study existing MSAs from both automotive and other domains to identify potential metrics for assessing the architecture and services quality. This also gives further insights into the definition of an appropriate DSML. In addition, mechanisms for enabling Command & Control from the application under test to the simulations have to be developed.

References

[1] Ian Briggs, Martin Murtagh, Robert Kee, Geoffrey McCulloug, and Roy Douglas. Sustainable non-automotive vehicles: The simulation challenges. *Renewable and Sustainable Energy Reviews*, 68:840–851, 2017.

[2] Maxim Chernyshev, Zubair Baig, Oladayo Bello, and Sherali Zeadally. Internet of things (iot): Research, simulators, and testbeds. *IEEE Internet of Things Journal*, 5(3):1637–1647, 2017.

[3] Juan Contreras-Castillo, Sherali Zeadally, and Juan Antonio Guerrero-Ibañez. Internet of vehicles: architecture, protocols, and security. *IEEE Internet of Things Journal*, 5(5):3701–3709, 2017.

[4] T. Haberle, L. Charissis, C. Fehling, J. Nahm, and F. Leymann. The connected car in the cloud: A platform for prototyping telematics services. *IEEE Software*, 32(06):11–17, nov 2015.

[5] Philipp Heisig, Sven Erik Jeroschewski, Johannes Kristan, Robert Höttger, Ahmad Banijamali, and Sabine Sachweh. Bridging the gap between SUMO & kuksa: Using A traffic simulator for testing cloud-based connected vehicle services. In Melanie Weber, Laura Bieker-Walz, Robert Hilbrich, and Michael Behrisch, editors, *Simulating Connected Urban Mobility, SUMO User Conference 2019, May 13-15, 2019, Berlin-Adlershof, Germany*, volume 62 of *EPiC Series in Computing*, pages 213–229. EasyChair, 2019.

[6] Márton Tamás Horváth, Tamás Tettamanti, Balázs Varga, and Zsolt Szalay. The scenario-in-the-loop (scil) automotive simulation concept and its realisation principles for traffic control. 2019.

[7] Ning Lu, Nan Cheng, Ning Zhang, Xuemin Shen, and Jon W Mark. Connected vehicles: Solutions and challenges. *IEEE internet of things journal*, 1(4):289–299, 2014.

[8] Tobias Schneider and A Wolfsmantel. Achieving cloud scalability with microservices and devops in the connected car domain. In *Software Engineering (Workshops)*, pages 138–141, 2016.

[9] Davide Taibi, Valentina Lenarduzzi, and Claus Pahl. *Microservices Anti-patterns: A Taxonomy*, pages 111–128. Springer International Publishing, Cham, 2020.

[10] Ibrar Yaqoob, Ejaz Ahmed, Ibrahim Abaker Targio Hashem, Abdelmuttlib Ibrahim Abdalla Ahmed, Abdullah Gani, Muhammad Imran, and Mohsen Guizani. Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges. *IEEE wireless communications*, 2017.