IoT and Microservices Based Testbed for Connected Car Services

Soumya Kanti Datta‡, Mohammad Irfan Khan‡, Lara Codeca‡, B. Denis†, Jérôme Härri‡ and Christian Bonnet‡
‡EURECOM, 450 route des Chappes, 06904 Sophia Antipolis, France
†CEA-Leti, MINATEC Campus, 17 avenue des Martyrs, F38054 Grenoble, Cedex 9, France
Emails: {dattas, khanm, codeca, haerri, bonnet}@eurecom.fr, benoit.denis@cea.fr

***Abstract*—The Internet of Things (IoT) has accelerated the innovations in the connected car domain. Together with Intelligent Transportation Systems (ITS), the IoT technologies envision unleashing efficient and more sustainable transportation systems. A connected car is already considered as an IoT resource whose capabilities can be exposed through an IoT platform, but few testbeds provide full-stack, cloud-based environment for connected car services experimentation. This paper formulates several research and engineering challenges for developing such testbed. Our proposed testbed exploits the IoT and microservices to provide advanced functionalities such as (i) the integration of heterogeneous sensors and vehicular communication technologies in a complete end-to-end stack and (ii) an Edge Server for vehicular data annotation, local processing with actuation. The testbed exposes the vehicular resources securely through a Cloud based framework and open APIs. They are supported by The Web of Things (WoT) based interoperable descriptions of vehicular devices and interactions among relevant actors. Finally, the Cloud system deploys the connected car services using microservices philosophy. In addition to that, we present a functional IoT architecture of the testbed, describe its operational phases, novel aspects, prototyping and experimentation details.**

***Keywords*-Connected Cars; Internet of Things; Microservices; Testbed; Web of Things.**

I. INTRODUCTION

With the IoT technologies maturing, more and more business scenarios are taking advantage of the IoT systems for remote monitoring, proactive maintenance, and actionable insights. The connected car domain is also actively benefiting from the IoT, Edge, and Cloud systems [1], [2]. Several established car manufacturers and new entrants have already integrated cellular connection in their cars making it possible for the cars to connect to devices, services and platforms external to them. As a result, such vehicles can be viewed as resources for data collection, computation and actuation from an IoT system point-of-view [3]. Additionally, the ongoing wave of modernization of city infrastructures, strict emission standards, and safety are driving the development of next-generation and sustainable transportation systems. Intelligent Transportation Systems (ITS) and connected car initiatives are using the IoT systems to develop more efficient transportation system that minimize the impact on the environment as well.

To achieve this vision, it is important to have an infrastructure where innovative connected car services can be experimented and improved. A testbed is an environment that allows experimentation and testing for the research and development of solutions. It provides a rigorous, transparent and replicable environment for experimentation and testing[1].

Several research and engineering challenges arise to develop a testbed that integrates connected vehicles, ITS, and IoT. Several of such challenges are explained below.

- Due to the presence of many stakeholders (e.g. car manufacturers, end users, platform providers, Telecom service providers, developers), several types of communication technologies (e.g. ITS-G5, cellular, LTE D2D, LTE-V) should be supported for V2X data exchange. It poses a challenge to make connected car services work with such different technologies.
- Connected car resources should be integrated with a secure Cloud framework for real-time data and control information exchange. It requires the resources to be described using a standard format and then exposed to authorized parties through the secure Cloud framework. Such description should express the capabilities (e.g. events, properties, and actions) supported by the car resources. The description should be consumed by services that may interact with the connected cars. It is another challenge to provide a standard and compact description of hundreds of resources available in the connected cars.
- It is widely acknowledged that centralized processing of connected car data is costly in terms of latency and QoS. Given the nature of safety requirements in highly and full autonomous vehicles (a category of connected cars), it is important to consider a powerful Edge server (or a Cloudlet if necessary) on such cars for resource discovery, local data fusion, communication, and actuation. The Edge server should be capable of securely provisioning relevant services from the Cloud framework.
- A scalable, extensible and secure Cloud framework that hosts the connected car services is necessary. This is another challenge as current practices of software development and virtualization techniques do not scale well.
- Seamless integration of heterogeneous devices (car resources, smartphones), communication networks into IoT

978-1-5386-4725-7/18/$31.00

---

[1] https://www.ict-fire.eu/wp-content/uploads/Experimentally_driven_research_V1.pdf

platforms are required for the testbed. This allows easy execution of experiments through the testbed.
- To extend the testbed capabilities as well as experimenting with the current functionalities, the testbed must be designed and developed using standardized common service functions. The third party components added to the testbed should be compliant with the same standard.
- The interactions among the IoT platform components, connected car services and consumer devices do not follow any uniform operational flow. The entry point of available testbeds and services vary widely.
- The final challenge is to develop an easy way to write and submit experiments, and collect the results from the testbed.

This paper aims to describe an IoT and cloud based testbed enabling experimentation on connected car services. The testbed services are running in a virtualized environment and are deployed using microservices. They are the extensions of popular Service Oriented Architecture (SOA) and can be viewed as bundle of small services each running on a container in our EURECOM-IoT cloud. The use of microservices brings several benefits to the testbed: loosely coupled services, lightweight test environment, rapid development, scalability, extensibility, load balancing, event logging, virtualization of services and modularization [4]. Our main contributions are are exploitation of the IoT and microservices to design an advanced architecture for the testbed to solve the mentioned challenges, and the presentation the testbed prototype with one connected car service experimentation. It focuses on provisioning a precision positioning algorithm.

The rest of the paper is organized as follows. Section II summarizes the state-of-the-art testbeds available for the IoT systems. Section III describes the proposed architecture of the testbed along with building blocks and operational steps. Section IV reports our prototype implementation and one connected car service experimentation. Section V concludes the paper.

II. STATE-OF-THE-ART

The current research and industrial initiatives on testbeds supporting various use-cases ranging from generic testing to smart building, parking management, sensor networks, smart city etc. The paper [5] has identified several of such testbeds and simulators currently used in the IoT domains.

The EU FP7 project WISEBED[2] aimed at providing a multilevel infrastructure of connected testbeds. The target of the project has been on the highly interdisciplinary area of wireless sensor networks and distributed systems. The project [6] provides a software platform called TARWIS, a flexible and generic testbed management system.

The EU project SmartSantander [7] provides an open testbed deployed in the Spanish city of Santander. It provides a city-wide experimental platform consisting of 12,000 IoT devices. They include IEEE 802.15.4 compliant nodes, parking sensors, irrigation and environmental monitoring among others. Subsequently, the framework has been extended to other regions in the United Kingdom, Germany and Serbia.

The authors of [8] highlight the need of involving end users of IoT systems in the experimentation life cycle. They present a testbed of a user centric smart building system. It integrates IoT nodes (i.e. devices), embedded gateways, smartphones of users, smart displays and a cloud server. The testbed functionalities are developed using the SmartSantander framework and WISEBED APIs for IoT management related tasks, device registration, collection of statistics etc. Using the testbed, the authors performed several case studies including: (i) energy efficiency in buildings (energy awareness, interactive mobile feedback, load disaggregation), and (ii) human dynamics in office spaces (measurement of activity heat map and social graph).

The EU H2020 FIESTA-IoT[3] proposes experiments as a service. The project provides software tools, best practices and processes enabling IoT testbed and platform providers to interconnect their devices and services. This process also preserves interoperability using semantic web technologies. The project allows submission of experiments for the underlying testbeds. The experiments are designed and executed to be run on a virtualized environment. Two key aspect of the process are the dynamic discovery of available IoT resources and the testbed agnostic IoT data access. Federation of the available testbeds, their challenges, and an unified IoT ontology for testbeds federation have been addressed in [9].

A generic IoT architecture for a testbed as a service is proposed in [10]. It is called IoTbed and uses the IoT devices located in the Edge level. Apart from those, there are coordinator/cluster head, IoT gateways, services and end users. The architecture is composed of several components: (i) management and operation control, (ii) resource manager, (iii) reputation and incentive manager and (iv) experiment manager. The operational model of the testbed includes (i) reputation point calculation, (ii) incentive calculation, (iii) testbed registration and (iv) experiment submission. A comparison of IoTbed with FIESTA-IoT, WISEBED, FIT IoT-Lab shows that the former is the most advanced as it supports heterogeneity, scalability, federation, user interfacing, resource allocation, IoT device registration, reputation scheme, incentive framework, and security policy.

HarborNet, a testbed successfully deployed at one of the seaports Portugal, is described in [11]. It provides abilities to cloud based service deployment, network control and data collection from nearby vehicles and RSUs. The testbed supports wide range of communication technologies (e.g. IEEE 802.11p, Wi-Fi, 3G and LTE) as well as GPS. The proposed services are centralized in a cloud which exposes interfaces to collect data, management, data warehousing, port operation etc.

Apart from that, a smart motor testbed is reported in [12].

---

[2] https://www.uni-muenster.de/Comsys/en/research/projects/project_wisebed.html

[3] http://fiesta-iot.eu/index.php/iot-experiments-as-a-service/

There is also the Japan-wide Orchestrated Smart/Sensor Environment (JOSE)[4] builds upon an Infrastructure as a Service (IaaS) approach. The unique aspect of this testbed is that it allows parallel execution many IoT services. Each of the services has its own physical sensor networks and Cloud.

It is clear from the above discussion that few of the current testbeds support connected car service related full-stack experimentation. That is one of the main motivations of the work presented in the following sections.

III. TESTBED ARCHITECTURE AND BUILDING BLOCKS PROTOTYPING

This section presents the testbed architecture (shown in Fig. 1), its functional building blocks, and their operational steps. Novel aspects in the presented architecture include an Edge Server, virtualization, microservices, WoT based Thing repository, open APIs to extend the offered services, and experimentation capabilities.

A. Connected Car

Our testbed provides access to one connected car. A vehicular On Board Unit (OBU) is used to interface with the CAN bus of the car. With the CAN bus access tools, higher layer applications and services access the CAN signals and messages. For the current prototype, it is possible to access the gear position, status of indicator lights, car windows and doors and change the status of headlights and car windows.

The OBU SDK allows embedding intelligence in the car as well as provides air interfaces with ITS-G5 and Wi-Fi. Currently the OBU supports exchange of GNSS data and CAM messages as required by connected car services. The SDK allows software development in C language.

To include the connected car into the IoT system, a Thing Description (TD)[5] of the car is developed [13] using semantic web technologies. The TD lists the properties, events and actions of the available sensors, and actuators. To ensure interoperability with connected car services, a standardized vocabulary following the ongoing W3C Web of Things initiative is used.

B. V2X Communication

Our testbed is one of the few that successfully integrates ITS-G5 communication stack [14] and provides the ability to experiment on the use of the same in connected car services. An MK5 Road Side Unit (RSU) and an MK5 OBU from Cohda Wireless are used in the testbed. It inherits the same (lower) access layer from IEEE 802.11p while providing new protocols for the upper layers. Integration of this communication technology paves way for Cooperative ITS related connected car services. It is due to the integration of V2X ITS-G5 technology, the testbed can support experimentation on high precision positioning for connected vehicles [15].

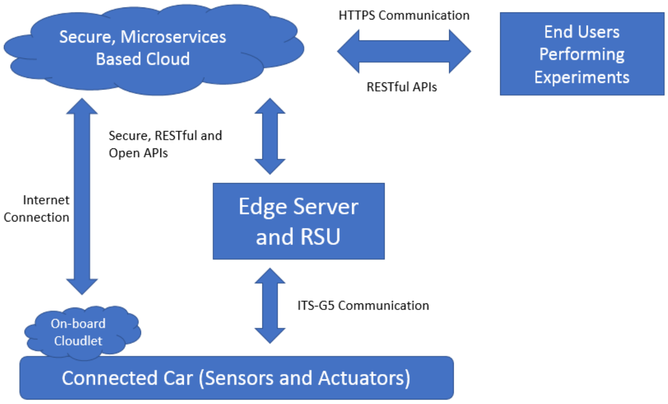

Fig. 1. Functional IoT architecture of the proposed testbed.

C. Edge Server

The Edge Server introduces capabilities for V2X communication (through the RSU), vehicular data validation, annotation, local processing, local actuation, and data exchange with a Cloud system. Our implementation of a vehicular gateway using Fog Computing [2] has been significantly advanced to create the Edge Server. It includes two dedicated interfaces for V2X and Cloud communication, and more robust services for sensor data validation, metadata annotation, local processing, and actuation. These functional elements are shown in Fig. 2. The vehicular subsystems (e.g. GNSS, Lidar, temperature sensor) and the data they generate are heterogeneous. To manage such variety, the Edge Server integrates a data validation and annotation module. Its main purpose is to check if the raw data is valid and then perform annotation. The range of data points of car sensors are used during the data validation phase. For annotation, additional information (e.g. unit of measurement, timestamp, type of sensor, domain of operation, unique identification of sensor) are added creating a metadata. This is important as the raw measurement itself is not suitable for processing. To settle the heterogeneity, we annotate the data following the specifications mentioned in Media Types for Sensor Measurement Lists (SenML)[6], thereby creating a uniform metadata from the raw data. Then the Edge Server processes the metadata locally depending on the use-case and actuate if necessary. The implementation is generic enough to be deployed on a connected car as well as on RSUs.

We describe an example here. The Edge Server deployed on a car can combine the sensor data from ambient temperature, humidity sensor to determine if there is fog outside. If yes, the server will automatically turn on its fog lamp through local actuation. Otherwise, if actuation is not necessary, the processed and encrypted sensor data (using AES-256) can be exchanged with a Cloud system over a secure communication link. The abilities of creating uniform metadata, processing, and actuation from the Edge Server are the novel aspects of the testbed.

Each of the mentioned steps in the Edge Server is accessible

---

[4] https://www.nict.go.jp/en/nrh/nwgn/jose.html

[5] https://github.com/w3c/wot/blob/master/plugfest/2018-prague/TDs/EURECOM-TD/EURECOM_X5_TD.jsonld

[6] https://tools.ietf.org/html/draft-ietf-core-senml-13

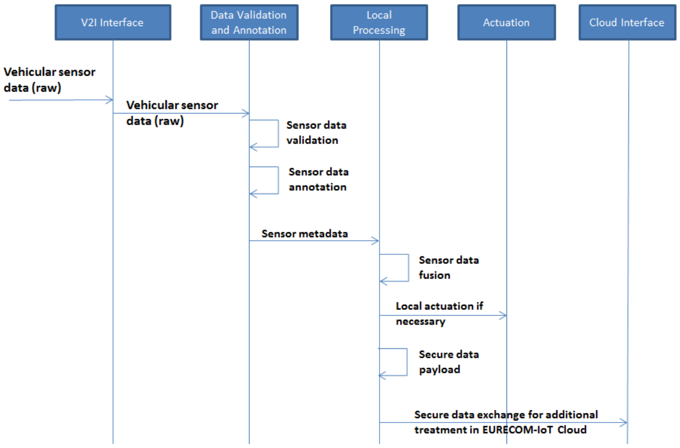

Fig. 2. Software Elements and Operational Steps of the Edge Server.

using a dedicated API. The server is developed using Node JS (with MongoDB used for database) and deployed in a Raspberry Pi alongside the RSU. Depending on the use case and experimentation scenario, the server can be deployed in a virtualized environment in the car, effectively transforming the car into a Cloudlet [16].

D. Secure, Cloud Based Web Services

Our Cloud based connected car services solve several of the identified research and engineering challenges presented at the beginning of the paper. The internal components are depicted in Fig. 3. The software framework deployed in the Cloud is secure by-design. The services exchange data with the Edge Server and connected car using a HTTPS based secure communication channel. The exchanged data is encrypted using AES-256. Each client or third party using the Cloud for experimentation must obtain a secret key (AES-256) and a JSON Web Token (JWT)[7] through the OpenSSL based secure key exchange mechanism. The JWT is utilized for both authentication and authorization which is a novel aspect presented in this work. Both the secret key and JWT are valid for a specific time and must be changed after that. These functionalities are part of the security provisioning service.

The sensors and actuators of the car are described and registered to the Thing Repository in the Cloud. This component follows the principle of CoRE Resource Directory[8] and Web of Things, which advocates for the use of standardized web technologies to counter the fragmentation among the IoT platforms. The vehicles must register their TD directly at the registration interface of the Thing Repository and the TD is then stored in a secure storage in the Cloud. Authorized clients or third parties doing experiments on our testbed must consume the TD to discover the sensors and actuators available in the car for testing as well as their properties, supported events and actions. This is done through the discovery interface.

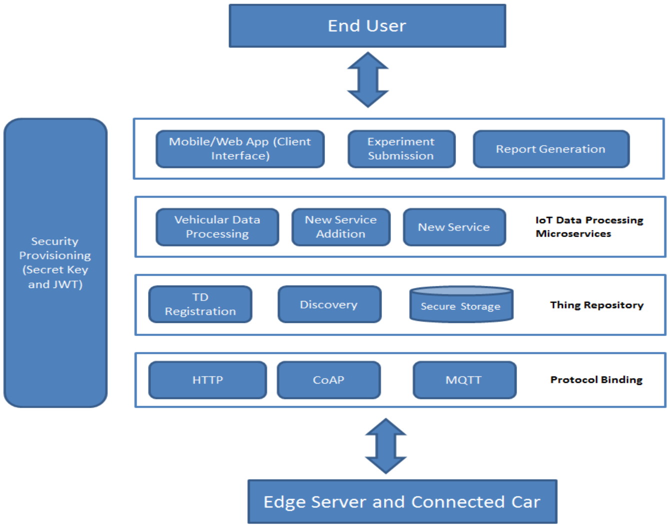

Fig. 3. Functional Elements of Cloud Based Connected Car Services.

In addition to that, many complex algorithms used by different services can be housed in the cloud. For example, semantic reasoning on vehicular data for trajectory mining [17], or tourism recommendation system [18], can be deployed as a service. We introduce the concept of microservices to extend the Cloud capabilities by adding new application logic as a microservice. Such a loosely coupled service can be developed independently, using any software tool, and standard data exchange format. Then the microservice is deployed using a container. This approach is extensible and also supports service scalability, as the replication of the same services become easy. Additionally, this mechanism supports lightweight deployment, rapid development, interoperability, virtualization and strong modularization in the Cloud. Adding a new connected car service is performed using ”New Service Addition” interface.

For seamless integration of new services and microservices deployment, the WoT philosophy of protocol binding is used. It maps abstract interactions between the IoT services (e.g. security provisioning, TD registration, discovery, data processing etc.) onto concrete protocol operations. For example, the vehicular TD registration request is mapped onto a POST request while the same for discovery is mapped onto a GET request.

For client application, we provide an Android application called Connect and Control Things (CCT) which is able to discover and consumer the available TD and visualize the processed data from the connected car resources. There are also web services enabling experiment submission and report generation. The experiments are designed using a descriptive language similar to that used in FIESTA-IoT project[9]. The described services and the architecture drawn in Fig. 3 are deployed in EURECOM-IoT cloud running on a virtual machine in Amazon Web Service.

---

[7] https://jwt.io/

[8] https://tools.ietf.org/html/draft-ietf-core-resource-directory-13

[9] http://moodle.fiesta-iot.eu/mod/book/view.php?id=104&chapterid=37

E. Novel Aspects and Limitations

Compared to the available testbeds, our implementation is secure by-design successfully integrates heterogeneous computing platforms (e.g. Cloud, Edge Server, RSU), ITS-G5, and a real connected car. This opens up the testbed usage to all stakeholders in the IoT and connected car domain. We utilize WoT based standard protocol, interface and description of the car making the resources easy to be consumed by the Cloud based web services. The Edge Server is capable of local computation and actuation which decongests the core network and enables distributing the intelligence among testbed elements. The web services, data exchange and storage are secure by design and novel mechanism like JWT is used for authentication and authorization. Microservices based service deployment is another unique property of the Cloud, and allows rapid extension of new services.

Although the testbed features above novel aspects, it has three main limitations. Due to non-availability of multiple vehicles, the scalability aspect could not be tested. Another limitation is that it lacks an automated tool that logs and displays the performance of connected car services (e.g. consumption of computational, communication resources, latency experienced by vehicles). During the experiment mentioned in the next section, such results are manually collected. Finally, the implemented security features are basic while we need advanced threat modeling and countermeasures.

IV. EXPERIMENT ON PRECISION POSITIONING SERVICE

The EU H2020 HIGHTS project[10] has been making coordinated efforts to offer high precision positioning service for Cooperative-ITS (C-ITS). This project combines the satellite based positioning systems with innovative use of vehicular on-board sensing and infrastructure based wireless communication technologies (Wi-Fi, ITS-G5 etc.) to produce next-generation and high precision positioning information for C-ITS applications. HIGHTS developed three such next-generation positioning algorithms which depend on different infrastructure (e.g. sensor and communication technologies). We derived the following experiment to be developed and tested on the proposed testbed.

A. Experiment Scenarios

In our experiment, a connected car needs precision positioning service while moving through urban canyons or tunnels where GNSS signals are not reliable. The vehicle is equipped with the Edge Server (or a Cloudlet) that detects the available sensors and V2X communication technologies it has access to. The information is securely exchanged with the Cloud. Based on the provided information, the latter would select the best precision positioning algorithm and its grade (higher the grade, more is the location precision) using a decision tree shown in Fig. 4. The positioning service of the Cloud will return the information to the vehicle. We experimented with two scenarios. For the first scenario, GNSS, on-board Lidar and HD maps are available for positioning. The seconds scenario involves GNSS, ITS-G5 and UWB ranging for positioning. It is decided that for the experiment to be successful, the Cloud system must be able to correctly inform the connected car about the positioning algorithms and its grades for both the scenarios.

B. New Microservice Addition

We developed the new microservice called ”positioning service” using Node JS framework and MongoDB for database. Then it is ported into the EURCOM-IoT Cloud. Also, the current OBU in the car lacks the intelligence to detect and inform which sensors and V2X communication stack is available for the calculation of precision positioning. To complement the OBU capabilities, we developed an Android Auto application (AAA) which acts as the on-board Edge Server. The experiment required us to rapidly develop two new software elements and deploy them into the overall testbed.

C. Running the Experiment

To run the experiment, the AAA performs security provisioning at the beginning. For the first scenario, AAA identifies the infrastructure elements (e.g. GNSS, Lidar and HD Map) that the car has access to and securely uploads the information to the newly added ”positioning service” in the Cloud. Following the decision tree, it determines that Ibeo Map based algorithm with grade 5 fits the scenario and securely informs that to the car. Similarly, for the second scenario, the application identifies that the car has access to GNSS, ITS-G5 and UWB ranging but not to Lidar. In this case, the Cloud recommends unconstrained VA-Cloc algorithm for absolute position with grade 3. HTTP based protocol binding is used for both the scenarios in this experimentation.

D. Experiment Results

For both the scenarios, the positioning service return correct algorithms and their grades. We outline early experimental results on computational and communication resources consumption at the AAA and Cloud system. The memory footprint the Node JS files implementing the Cloud system is about 25.8 KB. The runtime CPU load and memory are presented in Table I.

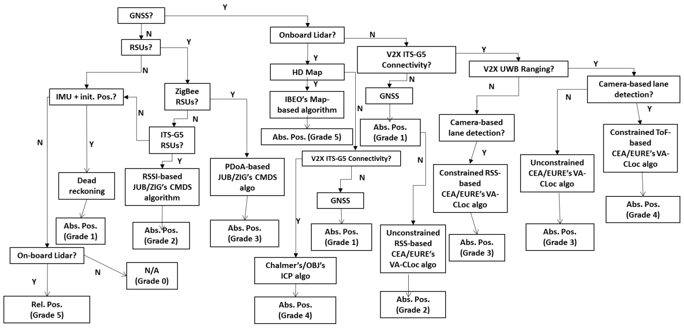

Fig. 4. Decision tree for Cloud based precision positioning service.

---

[10] http://hights.eu/

TABLE I
RESOURCE UTILIZATION AT THE CLOUD SYSTEM

Process name	**CPU Load**	**Memory**
Nodejs (testbed web services)	1.7%	4.8%
mongod (database)	0.3%	4.3%
dockerd	0.3%	4.8%
containerd	0.3%	1.4%

For the AAA, the apk file size is 2.29 MB. On an average, its runtime CPU consumption for user initiated tasks is 10% and that for Kernel is 5% (during the lifetime of the application). We measured the available and allocated Java memory using a Motorola G3. The available memory for AAA has been around 18.06 MB of which 11.24 MB were allocated for most of the operations. Rest of the memory was either free (most of the time) or used during the additional computation (encryption, decryption, network) for Cloud communication. In addition to that, we also noted the size of messages exchanged with the cloud. For the first and second scenario, the AAA transmitted 1.51 KB and 1.65 KB respectively, and received 6.52 KB and 6.48 KB respectively.

The obtained results not only validate the success of the experiment, but also establish the performance of the testbed and Edge Server (e.g. AAA). The testbed services utilize small amount of runtime resources establishing the lightweight aspect of the microservices. Also the on-board Edge Server is lightweight, easily extensible and consumes limited network bandwidth.

V. CONCLUSION

In summary, the paper motivates for a testbed to experiment on innovative connected car services. The testbed architecture and prototype successfully integrates several heterogeneous components. The connected car services are built using several IoT common service functions and deployed using microservices in the EURECOM-IoT Cloud. The Edge Server enables vehicular data validation, annotation, local processing and actuation if necessary. Integration of V2X ITS-G5 technology is another unique feature which allows us to perform a connected car service experimentation on high precision positioning. A new microservice (e.g. positioning service) and an Android auto application (for the on-board Edge Server) are developed and rapidly deployed for this testing. This demonstrates the extensibility of our testbed platform. The experimental results show that the current prototype of the Edge Server and web services are lightweight and extensible. We also identify three main limitations of our testbed. As for future work, we are focusing on addressing these limitations.

ACKNOWLEDGMENT

This work has been performed within the frames of the HIGHTS project, which is funded by the European Commission (636537-H2020), and AFA SCHEIF project. EURECOM acknowledges the support of its industrial members, namely, BMW Group, IABG, Monaco Telecom, Orange, SAP, and Symantec.

REFERENCES

[1] S. K. Datta, R. P. F. D. Costa, J. Härri, and C. Bonnet, "Integrating connected vehicles in Internet of Things ecosystems: Challenges and solutions," in *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 1–6, June 2016.

[2] S. K. Datta, C. Bonnet, and J. Haerri, "Fog Computing architecture to enable consumer centric Internet of Things services," in *2015 International Symposium on Consumer Electronics (ISCE)*, pp. 1–2, June 2015.

[3] S. K. Datta, J. Haerri, C. Bonnet, and R. F. D. Costa, "Vehicles as Connected Resources: Opportunities and Challenges for the Future," *IEEE Vehicular Technology Magazine*, vol. 12, pp. 26–35, June 2017.

[4] M. A. Jarwar, S. Ali, M. G. Kibria, S. Kumar, and I. Chong, "Exploiting interoperable microservices in web objects enabled Internet of Things," in *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 49–54, July 2017.

[5] M. Chernyshev, Z. Baig, O. Bello, and S. Zeadally, "Internet of Things (IoT): Research, Simulators, and Testbeds," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2017.

[6] H. Hellbrck, M. Pagel, A. Kller, D. Bimschas, D. Pfisterer, and S. Fischer, "Using and operating wireless sensor network testbeds with WISEBED," in *2011 The 10th IFIP Annual Mediterranean Ad Hoc Networking Workshop*, pp. 171–178, June 2011.

[7] L. Snchez, V. Gutirrez, J. A. Galache, P. Sotres, J. R. Santana, J. Casanueva, and L. Muoz, "SmartSantander: Experimentation and service provision in the smart city," in *2013 16th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pp. 1–6, June 2013.

[8] M. Nati, A. Gluhak, H. Abangar, and W. Headley, "SmartCampus: A user-centric testbed for Internet of Things experimentation," in *2013 16th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pp. 1–6, June 2013.

[9] R. Agarwal, D. G. Fernandez, T. Elsaleh, A. Gyrard, J. Lanza, L. Sanchez, N. Georgantas, and V. Issarny, "Unified IoT ontology to enable interoperability and federation of testbeds," in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pp. 70–75, Dec 2016.

[10] M. Hossain, S. Noor, Y. Karim, and R. Hasan, "IoTbed: A Generic Architecture for Testbed as a Service for Internet of Things-Based Systems," in *2017 IEEE International Congress on Internet of Things (ICIOT)*, pp. 42–49, June 2017.

[11] C. Ameixieira, A. Cardote, F. Neves, R. Meireles, S. Sargento, L. Coelho, J. Afonso, B. Areias, E. Mota, R. Costa, R. Matos, and J. Barros, "Harbornet: a real-world testbed for vehicular networks," *IEEE Communications Magazine*, vol. 52, pp. 108–114, September 2014.

[12] J. Queiroz, J. Barbosa, J. Dias, P. Leito, and E. Oliveira, "Development of a smart electric motor testbed for Internet of Things and big data technologies," in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, pp. 3435–3440, Oct 2017.

[13] S. K. Datta and C. Bonnet, "Describing things in the Internet of Things: From CoRE link format to semantic based descriptions," in *2016 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, pp. 1–2, May 2016.

[14] N. Xia and C.-S. Yang, "Vehicular Communications: Standards and Challenges," in *Internet of Vehicles. Technologies and Services for Smart Cities* (S.-L. Peng, G.-L. Lee, R. Klette, and C.-H. Hsu, eds.), (Cham), pp. 1–12, Springer International Publishing, 2017.

[15] G. M. Hoang, B. Denis, J. Härri, and D. T. M. Slock, "Cooperative localization in gnss-aided vanets with accurate ir-uwb range measurements," in *2016 13th Workshop on Positioning, Navigation and Communications (WPNC)*, pp. 1–6, Oct 2016.

[16] M. Otomo, G. Sato, and Y. Shibata, "In-vehicle Cloudlet Computing System for Disaster Information Based on Delay Tolerant Network Protocol," in *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*, pp. 737–742, March 2017.

[17] B. Klotz, R. Troncy, D. Wilms, and C. Bonnet, "Generating semantic trajectories using a car signal ontology," in *WWW 2018, The Web conference, April 23-27, 2018, Lyon, France*, April 2018.

[18] B. Klotz, P. Lisena, R. Troncy, D. Wilms, and C. Bonnet, "DrIveSCOVER: A tourism recommender system based on external driving factors," in *ISWC 2018, 16th International Semantic Web Conference, October 21-25, 2017, Vienna, AUSTRIA*, October 2017.