Microservices for Multimobility in a Smart City

Cristian Lai, Francesco Boi, Alberto Buschettu and Renato Caboni

CRS4, Center for Advanced Studies, Research and Development in Sardinia,
Pula Italy
Email: {cristian.lai, francesco.boi, alberto.buschettu, renato.caboni}@crs4.it

Abstract—**In this paper, we discuss our thoughts of microservice architecture and the way to build Internet of Things (IoT) services for multimobility in a smart city. We briefly introduce a draft architecture that we have used to develop an experimental Web application, designed to be used in a real case study and capable of interfacing with a wide range of heterogeneous IoT devices and services.**

Keywords–IoT; Smart City; Microservices; Multimobility.

I. Introduction

Information and Communication Technology (ICT) is evolving toward more scalable architectures, and those based on microservices are paving the way to modern applications [1]–[3]. The number of connected devices available in real life has significantly grown [4] as crucial parts of the IoT. In this paper, we discuss how we figure out microservice architectures and efficient applications specifically designed for the IoT field in front of a large number of users and a high demand of resources, as opposed to monolithic architectures. We introduce the draft of an architecture used to develop an experimental Web application for multimobility services in a smart city. This architecture is capable of integrating both physical and logical devices as well as of managing typical operations, such as device registration, data storage, and data retrieval.

In a smart city, multimobility combines different modalities of transportation [5], e.g., private cars, bus, carsharing, and bikesharing. The shift from homogeneous to multimodal mobility is growing in popularity, especially in urban centers with recurring problems associated with congestion, parking, and an overall lack of space [6]. Drivers moving by car towards an area with high traffic volume, such as the city centre, have the available information necessary to elude high traffic intensity areas avoiding time and fuel wasting besides preventing the increase of traffic. Every single element, such as a parking area, a bus stop, a car or bike sharing station, takes part in an extremely sophisticated network of miscellaneous connected IoT devices. Within a scalable system, a single device must be managed independently from the others. With the microservice paradigm, each device is managed by a dedicated microservice.

This paper is organized as follows. Section II reviews state of the art service-oriented architectures. Section III describes our microservice architecture. Finally, Section IV provides conclusions and future perspectives.

II. State of the Art

In IoT systems, a centralized architecture is responsible for offering one or more services to the user while the necessary data is produced by a set of devices deployed in different locations. These devices generate an amount of data readily available, used to create vertical applications. One way to access these data is usually using cloud-based platforms [7]–[9] (service-oriented centralized architectures). These platforms provide Application Programming Interface (API) for storing and retrieving data and interfacing with existing systems using the most common IoT protocols, such as HyperText Transfer Protocol (HTTP) and Message Queuing Telemetry Transport (MQTT) [10]. In this kind of architecture, device management is centralized, and this can generate an overload and bottlenecks. To date, the centralized architectures are successful state of art technologies but built as monolithic solutions, not offering the flexibility required to deal with heterogeneous devices efficiently. Often, they consist of three main parts: a client-side user interface, a database and a server-side application. Changes to the system necessitate building and deploying a new version of each component.

On the contrary, microservice architectures are divided into some small independent services, each of which implements a specific feature. A scalable architecture achieves more efficient management of available resources by allocating more only to those modules that are overstressed, rather than unconditionally to all the sub-components [2] [11]. To overcome the previously discussed limitations, we have been adopting the microservice paradigm and developing an architecture that provides independently deployable and loosely coupled basic services.

III. Microservice Architecture

Our concept of microservice architecture is a collection of independently deployable and loosely coupled basic services. Each microservice runs in its process, communicates with lightweight mechanisms, such as HTTP resource API [12] and manages significant operations. Our microservice architecture, namely *CRS4 Microservice Core for Iot* (CMC-IoT), is composed of basic microservices: i) **Cmc Auth** is a token generator that protects microservices from unauthorized access using a token-based authentication technique; ii) **Cmc App** manages resources and applications sign-up and the subsequent sign-in phase; iii) **Cmc User** manages user access to protected microservices. Basic microservices have been used for developing a specialized set for the IoT, including: i) **Cmc Devices** manages the device functionalities providing the Representational State Transfer (REST) create, read, update and delete (CRUD) operations; ii) **Cmc History** stores and retrieves historical data produced by devices; **Cmc Persistence** is a scheduler for general-purpose devices that do not directly provide their data. Once a device has been added to CMC Devices, it can send its data through a token authorized write operation. Then, data can be read by other authorized devices, applications, etc. Cmc Devices uses the basic microservices to check if tokens are valid and authorized to query a specific device (see Figure 1). CMC-IoT is the foundation of our

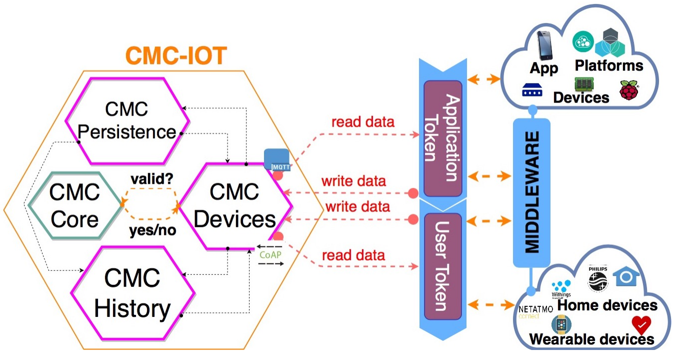

Figure 1. CMC IoT.

first Web application designed for citizen multimobility in a smart city, called SmartMobility. Smartmobility is the name of the application built on top of the presented architecture. It presents on a User Interface (UI) smart city data belonging to diverse mobility services. Currently, the contemplated services are parking areas, traffic sensors, public bus transportation, and car and bike sharing services. The application is supposed to be used in the following scenario. Before entering the city centre, the driver, using SmartMobility, can check the availability of free parking spots in the monitored parking areas close to his destination. In this way, she does not have to drive around the city looking for a free parking spot. Once the parking lot has been chosen, she can plan the fastest path to reach it according to real-time traffic information in the main city roads shown in SmartMobility. From the parking area, the driver can again check on the application the availability of mobility services, such as bus stops and sharing services, and their reliability so that she can choose the one most suitable for his needs or alternatively walk to his final destination. All the devices, managed by CMC-IoT and composing the considered mobility services, are shown on a map in the UI. Each graphical element shows additional information. For example, the parking lot shows in real-time not only the overall parking capacity but also further information about other mobility services in the area close to it. The bus stop provides the lines, the timeschedules and the service reliability. Carsharing and bikesharing show the number of available vehicles.

IV. Conclusion

Our application approaches the effectiveness of the drafted microservice architecture. The potential of the adopted microservice architecture allows designing modular systems. They can grow incrementally without continuous redesign, development, and deployment of the entire application. The system is divided into small and lightweight services, purposely built to perform a very cohesive business function. Every single element, i.e., a parking area, a bus stop, a car or bike sharing station could be added as well as removed independently, while the system can be further enriched with new services. Both these actions can be performed by modifying only the directly interested modules without affecting the others. Services independence allows reaching for an extremely sophisticated network of connected IoT devices. As a follow-up, we will develop a real case scenario demonstrating the implementation of the microservice architecture. Several tests will be performed to determine performances under an increasing number of requests per unit of time.

Acknowledgment

This work has been partially supported under the ERDF (European Regional Development Fund), POR Sardegna FESR 2007-2013, PIA 2013 (project: n.295 PIA ENTANDO).

References

[1] S. Baskarada, V. Nguyen, and A. Koronios, “Architecting microservices: Practical opportunities and challenges,” Journal of Computer Information Systems, 09 2018, pp. 1–9.

[2] N. Dragoni et al., “Microservices: How to make your application scale,” Perspectives of System Informatics, 2018, pp. 95–104.

[3] M. Kalske, N. Mäkitalo, and T. Mikkonen, “Challenges when moving from monolith to microservice architecture,” in Current Trends in Web Engineering, I. Garrigós and M. Wimmer, Eds. Cham: Springer International Publishing, 2018, pp. 32–47.

[4] M. Hung, “Leading the IoT,” Gartner, Tech. Rep., 2017. [Online]. Available: http://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf [accessed: 2019-06-17]

[5] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, “Internet of things for smart cities,” IEEE Internet of Things Journal, vol. 1, no. 1, Feb 2014, pp. 22–32.

[6] S. Shaheen, A. Stocker, and A. Bhattacharyya, “Multimobility and Sharing Economy,” Transportation Research Board, 500 Fifth Street, NW, Washington, DC 20001, Tech. Rep. E-C120, 2016.

[7] A. I. Abdul-Rahman and C. A. Graves, “Internet of things application using tethered msp430 to thingspeak cloud,” in SOSE. IEEE Computer Society, 2016, pp. 352–357.

[8] C. Lai, A. Pintus, and A. Serra, “Using the web of data in semantic sensor networks,” in Complex, Intelligent, and Software Intensive Systems CISIS 2017. Advances in Intelligent Systems and Computing, vol 611. Springer, Cham, L. Barolli and O. Terzo, Eds., 2018, pp. 106–116.

[9] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, “Sensing as a service model for smart cities supported by internet of things,” Transactions on Emerging Telecommunications Technologies, vol. 25, no. 1, 2013, pp. 81–93.

[10] A. H. Ngu, M. Gutierrez, V. Metsis, S. Nepal, and Q. Z. Sheng, “Iot middleware: A survey on issues and enabling technologies,” IEEE Internet of Things Journal, vol. 4, no. 1, Feb 2017, pp. 1–20.

[11] A. Krylovskiy, M. Jahn, and E. Patti, “Designing a smart city internet of things platform with microservice architecture,” in Proceedings of the 2015 3rd International Conference on Future Internet of Things and Cloud, ser. FICLOUD ’15. Washington, DC, USA: IEEE Computer Society, 2015, pp. 25–30.

[12] M. Fowler and J. Lewis, Microservices, 2014. [Online]. Available: http://martinfowler.com/articles/microservices.html [accessed: 2019-06-17]