Research Project Proposal

This research proposal is shaped around two main interdisciplinary fields: Data Science and Microservice Architectures.

Data science uses scientific methods to extract knowledge, accurate and deep understanding from structured and unstructured data. Data science spans different domains and encopasses data mining, machine learning as well as big data [1]. Data science is playing a prominent role in production across many technology sectors with data scientists providing production teams with software systems for modeling and analysing data that supply valuable feedback on gathered data to improve the product [2]. In turn, applied data science concerns applying data science principles to business needs and creating scalable systems [7,8] that collect, analyze, process, interpret, disseminate and display quantitative or quantifiable data of an organization for analytical, predictive or strategic purposes, hence for making product-relevant predictions and data-oriented business processes of interest.

Microservices enable designing complex software applications as suites of independently deployable services. Microservice architectures promote isolation and autonomy. Each microservice is conceived and implemented independently, and then usually deployed onto containers and serverless functions. Each microservice must contain everything it needs to function (business logic, data, user interface), and must be substitutable without impacting the operation and performance of other microservices. This new way of designing software systems and of blending development and operation, paves the ground to a number of research challenges. A first challenge that we want to face consists of identifying systematic solutions to automatically identify the microservices that compose a complex software system. This challenge emerges both when migrating monolithic systems to the microservice paradigm and when implementing new microservice systems from a global specification of their business goals. Solutions to this challenge should (semi-)automatically identify components that are autonomous and whose composition can guarantee safety, stability, and performance. They should include algorithms and techniques for the a-priori analysis of foreseen assemblies.

The project aims to study, define and implement data-oriented methods and tools for efficiently and (semi-)automatically migrating monolithic systems into scalable microservice systems and for designing new microservice systems in a systematic way [7,8].

The central question that this project asks, then, is twofold: (i) how to systematize the engineering and development of microservice systems in order to guarantee the system requirements by performing a general assessment approach across different microservice-relevant metrics (e.g., loosely coupling, cohesion, independency, technology-neutrality, autonomous deployment, fine-grained functions)? (ii) How to decompose the business functionalities of a monolithic or legacy system into microservice-based clusters by basing on both design-time data (e.g., architectural, modeling and input data) and run-time data (e.g., coming from possible execution traces, log files)?

Methodology

This project is methodologically innovative in the ways it seeks to propose innovative solutions to (1) migrate monolithic systems into microservice ones and (2) design new microservice systems from functional and non-functional requirements specification. The project will be organized into two main Research Streams (RSs) as follows.

RS1: system migration will study automated and data-oriented methods to identify microservices from a monolithic system. It will exploit static and dynamic models, as well as design- and run-time data, to express the interaction between components, and derive a microservice architecture. The coupling between the system components will be limited by imposing a partition of related business capabilities into single microservices, while keeping non-functional requirements guaranteed.

RS2: system synthesis will study automated methods to distribute a global specification of the system, including functional and possibly non-functional requirements, into well delimited business capabilities, to ensure the specified requirements via a general assessment of microservice-specific metrics. To this end, incremental synthesis methods will be devised to start defining broader microservice boundaries that will be then refined to smaller ones.

Put in context within existing literature

The identification of microservices plays a key role both in the migration of existing systems and in the realization of new ones. Systematic approaches to reason on the alternatives to decompose a system into components have been devised. Some of them [3,4] consider static models of the system and are focused on decomposing object-oriented systems into component-based ones. The notions of decomposition that they define can be investigated and adapted to microservice architectures. Other approaches [5,6] consider dynamic models and are limited to find grouping abstractions with no explicit identification of the components and their transformation into code. These methods lack metrics to precisely define the bound context of a microservice, hence preventing automation. This proposal aims to advance these approaches to allow application of data-science to the systematic development of microservices architectures, and enable automation and code generation.

References

[1] F. A. Rocha Silva, "Analytical Intelligence in Processes: Data Science for Business," in IEEE Latin America Transactions, vol. 16, no. 8, pp. 2240-2247, Aug. 2018, doi: 10.1109/TLA.2018.8528241.
[2] L. Cao. 2017. Data Science: A Comprehensive Overview. ACM Comput. Surv. 50, 3, Article 43 (October 2017), 42 pages. DOI: 10.1145/3076253.
[3] A. Shatnawi et al. Reverse engineering reusable software components from object-oriented APIs. Journal of Systems and Software, vol. 131, pp. 442-460, 2017.
[4] H. Washizaki et al. A technique for automatic component extraction from object-oriented programs by refactoring. Science of Computer Programming, 56 (1-2), pp. 99-116, 2005.
[5] H. de Brito et al. On-the-fly extraction of hierarchical object graphs. Journal of the Brazilian Computer Society, vol. 19, pp. 15–27, 2013.
[6] Y. Labiche et al. Combining Static and Dynamic Analyses to Reverse-Engineer Scenario Diagrams, IEEE International Conference on Software Maintenance, pp. 130-139. 2013.
[7] M. Autili et al. A Hybrid Approach to Microservices Load Balancing in Microservices: Science and Engineering, Springer International Publishing, pp. 249–269, 2020.
[8] A. Avritzer et al. Scalability Assessment of Microservice Architecture Deployment Configurations: A Domain-based Approach Leveraging Operational Profiles and Load Tests. Journal of Systems and Software, vol. 165, 2020, doi: 10.1016/j.jss.2020.110564.