

Collection and Identification Of Microservices Patterns And Antipatterns

Rafik Tighilt*, Manel Abdellatif**, Nader Abu Saad***, Naouel Moha*
Yann-Gaël Guéhéneuc****

*Université du Québec à Montréal, Montréal, Québec, Canada
tighilt.rafik@courrier.uqam.ca

**Polytechnique Montréal, Montréal, Québec, Canada.

***Bethlehem University, Bethlehem, Palestine

****Concordia University, Montréal, Québec, Canada

Abstract. Microservices architectural style is becoming more and more popular in academia and industry. However, a lack of understanding of its core concepts and the absence of ground-truth leads to a lot of misconceptions and development mistakes. In our research work, we aim to clarify the academic knowledge on microservices through the collection and the automatic identification of microservices patterns and anti-patterns. To this end, we aim to (1) introduce an exhaustive collection of microservices (anti-)patterns, and (2) propose an automatic approach for the identification of (anti-)patterns in microservice based systems. The continuous integration and continuous delivery for microservices can introduce anti-patterns that may affect the maintainability of the system and decrease its quality. Thus, we searched for re-engineering tools used to identify (anti-)patterns in microservice based systems. The results of our analysis showed that there is no fully-automated identification approach in the literature. This motivates us to propose (anti-)patterns for the identification process as a first step and then investigate how we can automatically identify them from the artifacts of microservice based systems.

1 Introduction

A microservice is defined as a small service, having a single responsibility, running on its own process and communicating with lightweight mechanisms Lewis and Fowler (2019). Microservices are generally built around business capabilities and independently deployable by a fully automated deployment machinery with a minimum centralized management of these services Newman (2015). An illustration of a microservice based system could be an e-commerce application. Each microservice of this system fulfils a single business function (Inventory microservice, Shipping microservice, Cart microservice, etc.), and has its own database. These microservices are loosely coupled, and communicate through lightweight REST APIs.

The popularity of microservice architectures has grown in the last few years due to the inability of monolithic applications to handle applications scalability and development cycles