Professional Master Internship—5 to 6 months, at the earliest from January 2022

Executor for deploying microservices-based architectures

Sophie Chabridon and Denis Conan
SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris
9 rue Charles Fourier, 91011 Évry cedex
Contacts: Sophie.Chabridon and Denis.Conan [at] telecom-sudparis.eu

Remuneration: approximately 580 € net per month.

Keywords: distributed systems, distributed algorithms, Docker, Kubernetes, Golang.

Context. We are interested in the self-adaptation of distributed architectures based on microservices (see Figure 1). These architectures are based on the Docker and Kubernetes technology bricks. Currently, we have a microservices-based distributed architecture model as well as a scheduler (AI planner) that allows us to reconfigure a starting architecture to a target architecture [3]. The scheduler gives the sequence of actions for the transition from the starting architecture to the target architecture. The actions are typically ‘create a microservice’, ‘connect two microservices by a connector’, etc. These actions are provided to an executor that applies them on a platform based on Docker and Kubernetes. The prototype is available as open source software (https://gitlab.ev.imtbs-tsp.eu/mimosae/mimosae).

Figure 1 – Reconfiguration of a microservice-based architecture

Issues. Currently, service continuity is not ensured during reconfigurations: The system is supposed to be unused during the reconfiguration phase. But, it should not be necessary to shut down an entire running system to change a part of it. The management system should, from the specification of the change, be able to determine a minimal set of microservices that are affected by the change. The rest of the system must be able to continue running normally. In particular, the implementer should give the affected part of the system the opportunity to reach a consistent state before a change is made, and should not force reconfigurations but wait for microservices to reach a consistent state. This coherent state requires that there is no ongoing communication between the microservices concerned or with their environment. Each microservice is said to be in a quiescent [1] or tranquil [2] state: The microservices must remain quiescent or tranquil while the reconfiguration is executed. This gives the newly created microservices the opportunity to be initialised in a consistent state with the rest of the system and the other microservices involved the opportunity to keep the system in a consistent state.

In our research work, we propose an extension of the work in the field to take into account publish/subscribe systems. Therefore, we invite you to participate in the implementation of the two existing algorithmic solutions ([1] and [2]), and to contribute to the finalisation and evaluation of our solution.

This internship topic is part of the research work of the DiSSEM group of the ACMES team of the SAMOVAR laboratory, Télécom SudParis, Institut Polytechnique de Paris.

References