

Internship offer for Master 2 students or final year computer engineering students - 2023

DSML for design, orchestration and adaptation of microservices

DiverSE team: <http://www.diverse-team.fr>

Supervisor 1: Stéphanie Challita

Email: stephanie.challita@irisa.fr

Supervisor 2: Benoît Combemale

Email: benoit.combemale@irisa.fr

Keywords.

Microservices, Cloud Computing, DSML, Orchestration, Adaptation

Context and Problem.

The microservices architecture consists of using multiple very small services that perform one and only one task instead of having a large and monolithic service [1, 2].

This architecture is more and more adopted because it allows decoupling responsibilities and eases the maintenance, evolution, and replacement of a particular service.

However, new challenges arise with this architecture because having multiple services working together requires much more effort in inter-service communication, deployment, the orchestration, etc.

Objectives.

Based on the review of the literature [3, 4, 5], the objective is to propose a Domain Specific Modeling Language (DSML) to model microservices applications.

The DSML will help developers to develop stateless and stateful microservices, proposing different solutions for handling heterogeneous microservices in terms of communications, the persistence of the data and synchronization.

The DSML must be able to check the correctness of the current designed application, according to the basic requirements of microservices, such as required values for deployment, scripts, etc. The DSML should also provide all the subsequent operations for the automated deployment and orchestration of the modeled application.

Existing solutions done by large groups such as Spotify [6] and Uber [7] can be used as the basis and inspiration for the definition of the DSML.

The second objective is to be able to recommend and guide the developers during the conception. Using a prediction model, this guidance will help the developer to decide which microservice should be stateful or stateless, how the microservices should communicate, which platform seems to be the best for the deployment, etc.

Last but not least, you can generate, by relying on the DSML, an autonomic loop to ensure the dynamic reconfiguration, i.e., to help the developer to monitor the microservices application while it is in production, and to adapt the microservices according to events that occur during production, or to the environment.

These adaptations will be taken into account by dynamically changing the choices that have been made during conception, such as state management, communications, etc, and the implementation that has been deployed.

Environment.

The candidate will work at Inria in the DiverSE team. Inria is the French national institute for research in computer science. There are 8 Inria research centers located throughout France, hosting more than 200 research teams. The DiverSE team is located in Rennes. DiverSE's research is in the area of software engineering. The team is actively involved in European, French and industrial projects and is composed of 12 faculty members, 10 PhD students, 2 post-docs and 4 engineers. The candidate will work in the context of one of the main topics currently explored in the team, involving various professors and students.

Required skills.

- Ability to work autonomously
- Strong skills in object-oriented programming
- Writing and speaking in English since the DiverSE team host foreign researchers
- Knowledge in model-driven engineering approach

References

[1] J. Thones. Microservices. IEEE Software, 32(1):116–116, 2015.

[2] James Lewis on Microservices
<https://www.se-radio.net/2014/10/episode-213-james-lewis-on-microservices/>

[3] F. Rademacher, J. Sorgalla, and S. Sachweh. Challenges of domain-driven microservice design: A model-driven perspective. IEEE Software, 35(3):36–43, 2018.

[4] P. Di Francesco. Architecting microservices. In 2017 IEEE International Conference on Software Architecture Workshops (ICSAW), pages 224–229, 2017.

[5] R. Petrasch. Model-based engineering for microservice architectures using enterprise integration patterns for inter-service communication. In 2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE), pages 1–4, 2017.

[6] Spotify and Backstage - <https://engineering.atspotify.com/2020/04/21/how-we-use-backstage-at-spotify>

[7] Uber, IDL, Apache Thrift - <https://eng.uber.com/service-oriented-architecture/>

[8] Chris Richardson, Microservices.io - <https://microservices.io/patterns/index.html> and <https://microservices.io/patterns/microservices.html>