M2 internship

Automated reverse-engineering of a Cloud API

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
Supervisor 3: Djamel Eddine Khelladi
Email: djamel-eddine.khelladi@irisa.fr

Keywords.

Software engineering, Model-driven engineering, Cloud computing, Google Cloud Platform, Natural Language Processing

Context and Problem.

Cloud computing witnesses a great growth in industry because it ensures the needs of companies in terms of resources (virtual machines, network, storage, etc.), while minimizing their cost.

However, today there are a multitude of cloud providers that offer similar services but are accessible in various manners (CURL scripts, Shell scripts, Java or Python code, etc.).

In order to overcome such a limitation, the scientific community explored the power of Model-Driven Engineering (MDE) and software language engineering, which allow an abstraction of technical details and automate a large number of tasks: code generation, deployment, monitoring, etc.

Model-driven engineering consists in creating dedicated Domain-Specific Modeling Languages (DSML) to capture the concepts (meta-model) and behavioral semantics of a system.

Such DSMLs can have a huge return on investment for industry by enabling better management of a system.

However, proposing DSMLs to manage a Cloud system is a difficult and costly task because one must have a great knowledge of the Cloud services and handle all their concepts.

To date, a large number of DSMLs for the Cloud are proposed manually, and the automation of such a task can be profitable for cloud engineers.

Recent work [2] has shown that it is possible to **automatically infer a DSML from a description of Cloud Application Programming Interfaces (APIs)**. This work focuses on a particular Cloud provider: Google Cloud Platform (GCP) in order to help cloud engineers deploy their applications with less errors and less effort.

Objectives.

You will be able to strengthen your expertise in software engineering practices (Git, Object-Oriented Programming, textual and code analysis) and more particularly in Model-Driven Engineering which is becoming widely adopted in industry.

After a short bibliographical study of retro-engineering techniques which exploit APIs (for instance [1,3,4], as well as Natural Language Processing (NLP) techniques, you will:

1. Propose an approach for inferring GCP DSML. For this, you will refer to a past research prototype from [2], which builds a DSML from GCP HTML documentation.
2. Implement a model-based framework that relies on this approach to design and deploy GCP applications.
3. Propose an automated approach that would automatically handle the evolution of GCP API. This can be done by experimenting NLP techniques to extract knowledge from GCP documentation. The approach needs to incrementally detect streaming modifications, by calculating and modifying only the differences between the initially processed version and the newly modified one.

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 centres 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 9 faculty members, 20 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
- Taste for software development
- Strong expertise in Java
- Knowledge in Cloud Computing, Natural Language Processing and Model-Driven Engineering is a plus
- Ability to write and communicate orally, preferably in English since DiverSE hosts many foreign PhD students, postdocs and visiting researchers

References

[1] Hanyang Cao, Jean-Rémy Falleri, and Xavier Blanc. Automated generation of REST API specification from plain HTML documentation. In Michael Maximilien, Antonio Vallecillo, Jianmin Wang, and Marc Oriol, editors, Service-Oriented Computing, pages 453–461, Cham, 2017. Springer International Publishing.

[2] S. Challita, F. Zalila, C. Gourdin, and P. Merle. A Precise Model for Google Cloud Platform. In 2018 IEEE International Conference on Cloud Engineering (IC2E), pages 177–183, 2018.

[3] F. Haupt, F. Leymann, A. Scherer, and K. Vukojevic-Haupt. A Framework for the StructuralAnalysis of REST APIs. In 2017 IEEE International Conference on Software Architecture (ICSA), pages 55–58, 2017.

[4] J. Zhai, J. Huang, S. Ma, X. Zhang, L. Tan, J. Zhao, and F. Qin. Automatic Model Generation from Documentation for Java API Functions. In 2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE), pages 380–391, 2016.