Model-driven software migration towards fine-grained cloud architectures

Robin Lichtenthäler

Distributed Systems Group, University of Bamberg, Germany
robin.lichtenthaeler@uni-bamberg.de

Abstract. Observing the trend towards building cloud-native applications, there is also demand for migrating existing system to the fine-grained architectures of cloud-native applications. To support such migration efforts, this position paper proposes a model-driven approach to enable a structured and reproducible migration.

Keywords: model-driven, migration, cloud-native, position paper

1 Introduction: Towards fine-grained cloud architectures

Today's software is more and more composed of fine-grained components which are combined to build more complex applications. This trend started with the rise of *Service-Oriented Architectures* (SOA) [9] superseding monolithic architectures. SOA aims towards independently developable and deployable components with a bounded business functionality. The trend towards fine-grained components with a bounded business functionality has in recent years manifested itself in the *microservices architectural style* [3] which can be seen as an evolution of SOA. The microservices architectural style in turn combined with the increasing popularity of cloud computing plus the softwareization of infrastructure lead to the emergence of so-called *Cloud-Native Applications* (CNA). In this paper the CNA definition by Kratzke and Quint [6] is adopted: *"A CNA is a distributed, elastic and horizontal scalable system composed of (micro)services [..]. The application and each self-contained deployment unit of that application is designed according to cloud-focused design patterns and operated on a self-service elastic platform."* Fine-grained architectures are an inherent characteristic of CNAs, because they enable the exploitation of the benefits offered by cloud computing. In this context, the latest effort towards even finer-grained components is the *Function as a Service (FaaS)* paradigm [1] where the deployment unit is a single function deployed to an elastic FaaS platform which inherently enables scalability and abstracts from operational concerns.

Building CNAs is the aim of many new software projects, but also of migration projects where an existing system should be transformed to a CNA. A key aspect of such a migration is the architectural transformation to the fine-grained architecture of a CNA which typically means decomposing the existing system.

S. Kolb, C. Sturm (Eds.): 11^{th} ZEUS Workshop, ZEUS 2019, Bayreuth, Germany, 14-15 February 2019, published at http://ceur-ws.org/Vol-2339

However, such a decomposition is difficult and rarely supported by structured approaches or tools. Therefore, it would be beneficial to explore possible structured approaches for the transformation of existing software systems to the fine-grained cloud architecture of CNAs to support software engineers engaged in such projects. This support will remain valuable in the future, because even today it argued that it is a reasonable procedure to start off with a monolithic architecture [8]. A monolithic architecture can then be gradually transformed to a fine-grained cloud architecture, when complexity grows and demand increases.

A promising field of research in this regard is *Model-Driven Engineering* (MDE) [5]. In MDE, models are considered as a key resource for software development and evolution. For architecture transformations, models can help to focus on the key aspects of the software architecture and enable a structured and tool-supported transformation. Although reducing the migration to a fine-grained cloud architecture to a model-based transformation seems promising, a key challenge in MDE is the alignment between the actual code and the model. The transition from code to a model and vice versa should be automated to a large extent, since manual steps in a MDE-based approach would cause an overhead and hinder reproducibility.

To explore the applicability of MDE for the migration of existing software towards fine-grained cloud architectures, the following research questions are investigated:

- Which model or collection of models can be used to represent an existing software architecture and a fine-grained cloud architecture?
- What are rules and patterns for the transformation to a fine-grained cloud architecture and how can they be applied to the identified model or collection of models?
- How can the generation of models from code, the model-to-model transformation, and the generation of code from models be effectively supported by tools?

2 Related Work

Various research directions can be identified covering different aspects of the proposed research questions. Regarding modeling languages for the cloud, the available languages [2] focus on the required cloud resources and their configurations, but not on the complete software architecture, i.e. the relations between the components and how they interact. These languages can be incorporated when a migrated application should be deployed to different cloud providers, but for the proposed MDE approach there is no suitable modeling language yet. Considering MDE-based approaches for software migration, a noteworthy and comprehensive research project was the ARTIST-project [7] which had the goal of providing a complete methodology and corresponding tools for migrating legacy applications to the cloud. While MDE was already used in this project, their focus was more on the general migration from on-premises deployments to cloud deployments without a substantial software architecture transformation. Furthermore, FaaS offerings enabling fine-grained architectures were not yet available at the time of the project and were not considered. Regarding research on the software architecture transformation, i.e., how to split an existing system into smaller components, only some approaches have been proposed, for example by Gysel et al. [4]. However, the existing approaches do not explicitly employ MDE. Furthermore, the existing approaches are based on exemplary use cases and need to be evaluated regarding their validity and general applicability.

3 The model-driven approach

In order to answer the given research questions, this position paper proposes a model-driven migration of existing software systems to fine-grained cloud architectures. The basic idea is that a model can be generated representing the existing system. This model can then be (semi-)automatically transformed to a model of a fine-grained cloud architecture. Finally, artifacts can be generated from the transformed model to facilitate the implementation of the new architecture. Because implementing this approach as a whole is ambitious, it should be separated into different concerns, as represented by the corresponding research questions. First, a modeling language or collection of languages needs to be defined. It should enable the modeling of an existing application, the identification of possibilities for deriving fine-grained components, as well as representing the resulting fine-grained cloud architecture. Therefore, a key challenge for the modeling language is to find the right level of abstraction. Next, rules and patterns for splitting an application into more fine-grained components should be defined and represented in model transformation rules. Finally, because tools are seen as a key part of MDE, tools should be provided for creating a model from code, transforming the model and generating artifacts for the new architecture.

The approach focuses on the actual execution of a migration. However, it should be noted that before the actual execution, a careful analysis of the necessity and usefulness of a migration should be conducted. Not every existing application should be transformed to a CNA and evaluating the applicability of a transformation is another important research question. Sometimes also hybrid applications are sensible where only parts of the existing system are migrated to be cloud-native. The approach could provide additional insights for this question as well, but to keep the approach focused, it is assumed that a prior evaluation has been done.

That said, there are several further challenges which need to be taken into account. There exists a large technological heterogeneity for existing applications and especially cloud-native applications. The aim for this approach would be to make the models for the existing system and the fine-grained cloud architecture as generic as possible. Technological differences are then addressed inside the steps for generating a model from code and generating artifacts, e.g., through profiles. Furthermore, when thinking about possibilities to split an application into smaller components, a purely technical view might suggest several different possibilities with potentially conflicting outcomes. In such cases the business view rather than the technical view is crucial to make reasonable decisions. The approach could therefore be semi-automatic and suggest different options during the model transformation phase which could then be decided upon using business knowledge.

As a starting point, an exemplary application is manually transformed. During the manual transformation, requirements for the modeling language and first transformation rules can be identified. For this exemplary application a specific technological stack needs to be chosen. The differences with regard to other technological possibilities, are considered afterwards. Further applications or successful migration projects are then considered to iteratively evaluate and improve the approach.

In conclusion, transforming applications into fine-grained CNAs is an ongoing challenge for developers. Providing support for this task through structured approaches and tooling would therefore be beneficial to developers and to research on CNAs as a whole.

References

1. Baldini, I., Castro, P., Chang, K., Cheng, P., Fink, S., Ishakian, V., Mitchell, N., Muthusamy, V., Rabbah, R., Slominski, A., Suter, P.: Serverless Computing: Current Trends and Open Problems. In: Research Advances in Cloud Computing, pp. 1–20. Springer Singapore (Jun 2017)
2. Bergmayr, A., Breitenbücher, U., Ferry, N., Rossini, A., Solberg, A., Wimmer, M., Kappel, G., Leymann, F.: A systematic review of cloud modeling languages. ACM Computing Surveys 51(1), 1–38 (Feb 2018)
3. Dragoni, N., Giallorenzo, S., Lafuente, A.L., Mazzara, M., Montesi, F., Mustafin, R., Safina, L.: Microservices: Yesterday, Today, and Tomorrow. In: Present and Ulterior Software Engineering, pp. 195–216. Springer International Publishing (Sep 2017)
4. Gysel, M., Kölbener, L., Giersche, W., Zimmermann, O.: Service Cutter: A Systematic Approach to Service Decomposition. In: Service-Oriented and Cloud Computing, pp. 185–200. Springer International Publishing (Aug 2016)
5. Kent, S.: Model Driven Engineering. In: Lecture Notes in Computer Science, pp. 286–298. Springer Berlin Heidelberg (Apr 2002)
6. Kratzke, N., Quint, P.C.: Understanding Cloud-native Applications after 10 Years of Cloud Computing - A Systematic Mapping Study. Journal of Systems and Software 126, 1–16 (Jan 2017)
7. Menychtas, A., Konstanteli, K., Alonso, J., Orue-Echevarria, L., Gorronogoitia, J., Kousiouris, G., Santzaridou, C., Bruneliere, H., Pellens, B., Stuer, P., Strauss, O., Senkova, T., Varvarigou, T.: Software modernization and cloudification using the ARTIST migration methodology and framework. Scalable Computing: Practice and Experience 15(2) (Jul 2014)
8. Nakazawa, R., Ueda, T., Enoki, M., Horii, H.: Visualization tool for designing microservices with the monolith-first approach. In: 2018 IEEE Working Conference on Software Visualization (VISSOFT) (Sep 2018)
9. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: Service-oriented computing: a research roadmap. International Journal of Cooperative Information Systems 17(02), 223–255 (Jun 2008)