Microservices-based Business Process Model Execution

Agnes Koschmider[1]

Abstract: Recently, microservices evolved as a new software architecture paradigm allowing to build independently deployable software systems. In the context of Business Process Management microservices could be used to automate the execution of highly distributed business process models as required within the digital transformation. This paper sketches a microservices-based business process model execution and discusses challenges.

Keywords: Microservices, business process automation, business process execution

1 Introduction

WSBPEL [WS] is the widely-used standard for business process model execution. It lays the foundation for a process engine automating the execution of business processes specified as web services. Recently, microservices [LeFo14] evolved as a new software architecture paradigm allowing a flexible execution and an independent deployment of software systems. Microservice (architecture style)…"*is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms*" [LeFo14]. Initial discussions can be found how microservices might support business scenarios. Particularly, their use for an agile development of software systems has been attested [DDEF16]. In the context of Business Process Management (BPM) microservices could complement or even replace WSBPEL for business process model execution.

Contrary to WSBPEL the characteristics of microservices support the execution of highly distributed business process models and do not require any centralized management. Particularly, process model activities or events could be specified as a microservice addressing its own technology stack. A microservices-based business process model execution would be performed upon a container technology (e.g., Docker[2]) and thus would replace or complement a process engine-based execution. Such a microservices-based business process model execution would have several advantages. From a technological perspective it would allow a scalable and reliable execution of business process models. From a customer perspective it would support a real-time individualized and personalized production of products. However, several challenges must be solved in order to exploit these benefits.

[1] Institute AIFB, KIT, 76128 Karlsruhe, agnes.koschmider@kit.edu

[2] https://www.docker.com/

This paper sketches a microservices-based business process model execution and discusses the challenges to be met.

2 Microservices

Microservices are characterized by services that are built around business capabilities, they are independently deployable, have little centralized management of services, and can use its own technology stack (the programming language and data storage are not centrally specified). Often microservices are illustrated by means of a comparison with a monolithic style. A monolithic style (on the left hand side of **Fig. 1**) is considered as a large "stone" where all functions run in a single process. An application includes all modules, libraries and dependencies, which are necessary to ensure the proper function. Changes on the application require to test and compile the entire application. In contrast, a microservices style consists of several independent services that together build the application. The modifications of a service do not directly affect the properties and functions of other services thus ensuring independence of services. The main difference between web services and microservices is the automatic deployment. Generally, a deeper understanding is required which service is large (micro*service* vs. web *service*).

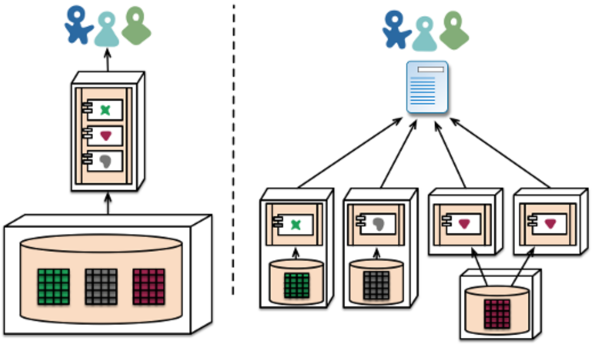

Fig. 1: Monolithic style (left) versus Microservices style (right) [Fowl15]

The main benefits of microservices are (1) improved scalability: services are capable of scaling independent of the use of other services, (2) improved agility: changes on services can be performed without affecting other services, (3) improved availability of the application: a partial failure of a service might not directly affect the entire application, (4) continuous delivery: changes of services do not set the entire application to the maintenance mode.

These benefits of microservices can be exploited for business process model execution allowing more agile and scalable application as discussed in the next section.

3 Microservices-based Business Process Model Execution

A microservices-based business process model execution is performed on the microservices framework (i.e., in a container on an orchestrator). This means that a container technology (e.g., Docker) is required to execute the application code (any programming language can be used to specify the service). Then an orchestrator (e.g., kubernetes[3]) is necessary to automate the deployment of containerized applications. Finally, a monitoring tool can be used to control the distributed environment. Applying this framework to a microservices-based business process model execution means that the orchestration of services is no longer exclusively performed by a process engine but (also) by containerized application orchestrator, which is set on a different abstraction level. This raises two challenges

Challenge 1: *What are appropriate strategies for business process model orchestration* (e.g., routing of business process activities). The control-flow of process models could make the unpredictable flow of microservices instances more transparent.

Challenge 2: *How to deal with processes containing "automatic tasks" and "semi-automatic tasks" and thus how to combine process-based engines and containerized application orchestrator?*

Also, it is open:

Challenge 3: *How to specify the dependencies of microservices executed within distributed business process models*? Microservices require only a minimum of central control, which also raises the challenge of

Challenge 4: *Who takes care of the process*? Microservices communicate with lightweight mechanisms but have so called "dumb pipes“ (i.e., only the endpoints are smart and pipes only forward information) raising the challenge of:

Challenge 5: *How to implement processes according to „smart endpoints and dumb pipes“ characteristic?*

Finally it remains to address:

Challenge 6: *What is an appropriate visualization and assignment of microservices in a business process model?* Fig. 2 shows one alternative. Two microservices (depicted as MS events) are specified in a book flight BPMN diagram. It remains to investigate if microservices are specified as events (see **Fig. 2**) or as a new type of process activities (then an icon is attached to the activity) or if a process model fragment defines a microservice.

Fig. 2 also uses choreography (between Client and System). If microservices are used in such setting it remains to investigate if timer event would implement the routing of

[3] https://kubernetes.io/

activities addressed by the challenge:

Challenge 7: *What is an efficient solution for choreography in microservices-based business process model execution?*

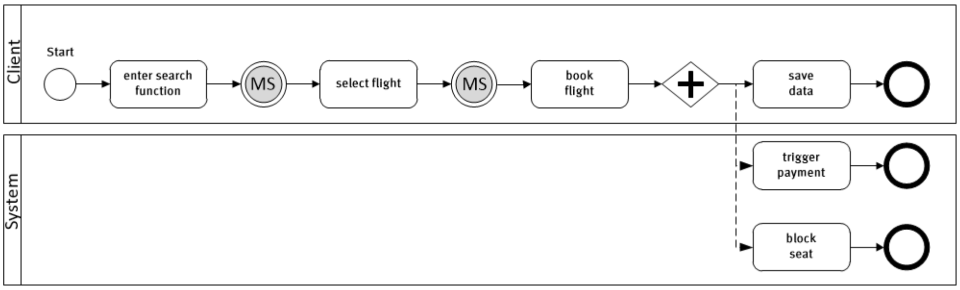

Fig. 2: One option of visualization and assignment of microservices in a business process model

4 Conclusion

Microservices can be seen as an architectural style for the design of distributed software systems having several benefits in contract to common architecture styles. This paper illustrated the use of microservices for business process model execution. A research agenda should cover the following points (1) elaborate the difference between microservices and web services for business process model automation, (2) discuss if a microservices-based business process model execution is complementary to existing solutions or even might dominate them, (3) discuss use cases for a microservices-based business process model execution, and (4) provide a testbed for experiments.

Bibliography

[DDEF16] Daya, S.; Van Duy, N.; Eati, K.; Ferreira, C.M.; Glozic, D.; Gucer, V.; Gupta, M.; Joshi, S.; Lampkin, V.; Martins, M.; Narain, S.; Vennam, R.: Microservices from Theory to Practice: Creating Applications in IBM Bluemix Using the Microservices Approach. IBM Redbooks, 2016

[Fowl15] Fowler, M.: Microservices: Nur ein weiteres Konzept in der Softwarearchitektur oder mehr? OBJEKTspektrum, vol. 01, 2015.

[LeFo14] Lewis, J.; Fowler, M.: Microservices: a definition of this new architectural term. https://martinfowler.com/articles/microservices.html

[WS] OASIS Standard WS-BPEL 2.0: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel