Topic 2 – “Developing successful cloud software”
A peek on the main trends and challenges about microservices design and development

Nicola Meneghetti
Email: menego1983@gmail.com
Course Id: 876a5df21ca

Abstract— **This essay takes into consideration the main aspects that characterize cloud computing. In particular it provides an overview on how the classical software development process is impacted. It then narrows down the research by analyzing some aspects related to the architectural methodologies and the tool-chain that come with the creation of microservices. After a brief overview on the possible future of automatic software development and its implications, it brings attention to the fact that security is a topic not addressed enough by expert developers and the actual literature available on the subject. Based on these outcomes, three research questions are provided in order to better define design techniques for microservices architecture, security enactment and essential tooling needed to aid programmers in the realization of these aspects. Finally, some future research is taken into consideration in order to implement such findings in the standard procedures for cloud development.**

I. Introduction

During the last decade we experienced a big change in the way software is developed and distributed. In this regard, cloud computing plays a crucial part where all the rules of classical software development are evolving to address the ubiquity and the faster time to market demands. An example of new architecture that leverages cloud computing is given by microservices, in which applications are essentially a composition of independent, uncoupled entities that communicate between each other[1]. New roles emerged, such as DevOps, where development and operation converge into a single figure with the ability to implement and coordinate the high level of automation required by this new approach. Consolidated techniques like object oriented programming are shifting to more abstracted paradigms such as MDE (Model Driven Engineering) where models are the building blocks of software systems, separating design and logic from implementation details [2], [3]. All these aspects are still growing towards maturity and they bring a whole new set of challenges and considerations.

As a matter of fact, there seems to be still some unclarity in what best defines techniques and methodologies for the creation of microservices. According to Sam Newman[1], *“microservices should be small, autonomous services that work together”*. Although this definition is clear and simple there is not an officially established way to address size and degree of independence if not by taking portions of classical methodologies that cover some of these aspects [2]. For example the modularity and the loose coupling principles from SOA (Service Oriented Architecture)[1] or the core concept of bounded context from DDD (Domain Driven Design)[2] that constitutes the foundation of microservices[4].

Nonetheless, many efforts are being made in the creation of an ecosystem able to assist and facilitate developers in these processes. As reported by G. Fylaktopoulos et al. [2], we already have at our disposal a whole set of integrated development environments (IDE) that offer a series of functionalities to create, test and deploy cloud software. These platforms are able to support many programming languages together with the relative debugger and testing framework. They can be used as web application without the need of an on-premise installation. They offer versioning capabilities such as git[3]. Few of them also provide code collaboration tools (e.g. AWS Cloud9[4]) similar to Google docs[5]. However these products, despite showing that some steps are being taken in the right direction, are still far from completion and in general there is still no feature parity with their desktop-based counterparts [2].

Furthermore, considering the increasing abstraction that comes with cloud computing, there are no environments able to leverage multi-layer programming and MDE, hence simplifying and automating, in part, software development [2]. Nonetheless, there is already some promising work in this regard. In their paper *Cloud Automatic Software Development*, H.Benfenatki et al. [5] theorized and showcased a paradigm in which it is possible to create cloud software without a deep programming knowledge if not at all. This new kind of service, called Automatic Software Development as a Service (ASDaaS), leverages all the core concepts of cloud computing. Based on simple business requirements and through service discovery[6], the final application is the result of a composition of existing microservices and the automatic creation of the missing ones. The whole purpose of this approach is to free the software programmer from the task of developing functionalities required by the client and to focus more in the enactment of other important activities that are usually left aside or given less priority due to time

[1] `https://www.ibm.com/support/knowledgecenter/SSMQ79_9.5.1/com.ibm.egl.pg.doc/topics/pegl_serv_overview.html`

[2] `https://en.wikipedia.org/wiki/Domain-driven_design`

[3] `https://git-scm.com/`

[4] `https://docs.aws.amazon.com/cloud9/`

[5] `https://www.google.com/docs/about/`

[6] `https://www.nginx.com/blog/service-discovery-in-a-micros`

pressure.

As a consequence of the aforementioned case, security is considered one of the less treated topics in microservices related research and the first source of concern among software developers [6], [7]. In their work, M.S.Hamzehloui et al. [6] analyze the main areas of interest about microservices in order to highlight the key characteristics and to bring out important topics that still require additional study. In more than one occasion they analyze a series of systematic mappings comprising tents of papers about microservices and the outcome stresses out how security is one of the least discussed attributes. Also in the survey conducted by J.Ghofrani et al. [7], becomes apparent how security is the main concern when experts needs to design and develop microservices. This provides a clear direction in which developers should put more effort.

II. Areas of investigation

Following the considerations exposed in the introduction, I identified 3 areas of interest that require further investigation in order to improve the quality of processes and products related to cloud software and microservices in particular.

A. Boundaries

In this new scenario, characterized by the convergence towards 'everything as a service' and the increasing abstraction of processes [5], there is the need to determine the boundaries that define the behaviors of the various units that compose a cloud software solution [2]. There is still much unclarity and besides some best practices [6], a clear set of methodologies is not completely defined yet. So, **what techniques are utilized to define the boundaries of a microservice in the architectural design phase?**

A systematic review should be conducted in order to provide an answer to the question that arises from these considerations.

B. Security

As pointed out by the existing bibliography and as a result of the previous inquiries, it is safe to assert that one of the matters that requires more focus is security. It should be imperative then, that all the activities related to cloud software development are intertwined by practices that involve security assessment. In order to reach this result we need to provide an answer to the following question. **How can we improve the awareness and standardize the adoption of security methodologies in microservices?**

A qualitative research will provide the necessary insights to understand the reasons why security is not addressed properly and how to mitigate such behavior.

C. Tool-chain

In light of the knowledge derived from the previous findings and considering what the actual landscape of IDE providers offers to cloud engineers, we need to retrieve more information for answering the following research question: **what are the changes in the software development tool-chain that are essential for the production of cloud applications but are not offered yet by the existing means in terms of microservices definition and security enforcement?**

For this purpose, an additional survey should be conducted in order to gather critical requirements that are still missing and compare the outcomes with the features offered by the current solutions.

III. Future work

In this essay, after a generic introduction about the cloud development landscape, I analyzed the main challenges that architects and developers usually face when it comes to the design and creation of microservices. I highlighted some critical aspects regarding architectural principles, tooling and practices that still lack maturity. Lastly, I provided three research questions in order to gather information about techniques used to define the boundaries of microservices, methodologies to enforce security in all the development phases and available tools that can help in the realization of the previous two aspects.

Additional work should be conducted in order to understand the process implementation able to introduce the outcome of these results into the various phases of a project development. How can we show the value of defining the correct bounded contexts? What advantages are reflected then in term of security? What are the main flaws in which the development incurs when it overlooks such procedures? Is there a need of additional tools to aid us on these matters?

The evolution triggered by cloud computing has drastically changed the way to conceive conventional development methodologies but this phenomena is still in its early stages and much work should be done in order to produce an adequate set of officially recognized procedures for creating safe and qualitative cloud products.

References

[1] S. Newman, *Building Microservices, Designing Fine-Grained Systems*. O'Really, 2015.

[2] G. Fylaktopoulos, G. Goumas, M. Skolarikis, A. Sotiropoulos and I. Maglogiannis, "An overview of platforms for cloud based development," 2016, https://www.researchgate.net/publication/290788490_An_overview_of_platforms_for_cloud_based_development.

[3] A. Fuggetta and E. Di Nitto, "Software process," in *Proceedings of the on Future of Software Engineering*. ACM, 2014, pp. 1–12.

[4] E. Schäffer, H. Leibinger, A. Stamm, M. Brossog, and J. Franke, "Configuration based process and knowledge management by structuring the software landscape of global operating industrial enterprises with Microservices," *Procedia Manuf.*, vol. 24, no. 2, pp. 86 –93, 2018, https://www.sciencedirect.com/science/article/pii/S2351978918305304.

[5] Hind Benfenatki, Hamza Saouli, Nabila Benharka, Parisa Ghodous, Okba Kazar, Youssef Amghar, "Cloud Automatic Software Development," *20th ISPE International Conference on Concurrent Engineering*, 2013, https://pdfs.semanticscholar.org/860e/69b88b4dc1f01c1e61c3385f57df70fd7b72.pdf.

[6] Mohammad Sadegh Hamzehloui, Shamsul Sahibuddin and Ardavan Ashabi, "A Study on the Most Prominent Areas of Research in Microservices," *International Journal of Machine Learning and Computing*, vol. 9, no. 2, pp. 242 –247, April 2019, https://www.researchgate.net/publication/333033011_A_study_on_the_most_prominent_areas_of_research_in_microservices.

[7] Javad Ghofrani and Daniel Lübke, "Challenges of Microservices Architecture: A Survey on the State of the Practice," February 2018, https://www.researchgate.net/publication/328216639_Challenges_of_Microservices_Architecture_A_Survey_on_the_State_of_the_Practice.