2017 IEEE International Conference on Software Architecture Workshops

Architecting Microservices

Paolo Di Francesco
Gran Sasso Science Institute, L'Aquila, Italy
paolo.difrancesco@gssi.it

***Abstract*—Inspired by Service Oriented Architecture (SOA), and from the convergence of Cloud Computing and Web 2.0, Microservice Architecture (MSA) has recently emerged as an architectural style particularly suitable to the cloud infrastructures.**

The MSA 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. Although the set of MSA principles aim for high degree of flexibility, modularity and evolution, adopting, operating and maintaining microservice architectures in practice is challenging and time consuming.

This paper reports on a PhD research project addressing three different challenges concerning MSA: (*i*) the identification of the key properties of microservice architectures, (*ii*) the identification and investigation on a description language for designing and analyzing architectures, (*iii*) the identification of the factors that impact the process of migrating existing applications towards MSA. The initial contributions of this project are: (*i*) a systematic mapping study on architecting microservices performed in order to understand the state of the research and the possible gaps in the area, (*ii*) an approach for architecture recovery of microservice-based systems named MicroART, and (*iii*) the implementation of the MicroART first prototype.

***Index Terms*—Microservices, Software Architecture, Systematic Mapping Study, Architectural Language**

I. Introduction

Netflix, Amazon, The Guardian and other companies [15] have evolved their applications toward the promising and challenging style of microservice architectures (MSAs).
Fowler and Lewis define the microservice architectural style as an approach for developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API [15]. MSA arises from the broader area of Service Oriented Architecture (SOA) and focuses on specific aspects, such as componentization of small services, application of agile practices for development, deployment and testing of services, usage of infrastructure automation with continuous delivery features, decentralized data management and decentralized governance among services. Even though the design principles around the microservice architecture style have been identified, many aspects are still unclear or unexplored. The need to understand the current state of the art on architecting microservices is confirmed by the fact that three different mapping studies [2, 13, 36] have been published recently and are providing important characterization to the field.

The purpose of this paper is to describe a PhD research project that aims at investigating the key aspects of microservice architectures, and realizing a comprehensive understanding of the MSA style and its implications. Three different challenges in this field will be investigated. The first challenge concerns the identification of the key properties of microservice architectures, mainly focusing on the software qualities (e.g., security, performance, maintainability). The second challenge is the identification and definition of appropriate instruments for supporting the design of microservice architectures. The third challenge concerns the migration process of existing application towards the possibility to adopt a microservice-based architecture.

The rest of the paper is organized as follows. In Section II we present the research questions we are addressing during this PhD project, in Section III we present the proposed solution and the current progress of the research, in Section IV the impact on the industry area is discussed, in Section V and VI related work and the conclusions are presented.

II. Research Objectives

Although the microservice architectural style is gaining popularity in both academia and the industry [14], the research in the field is far from mature [36]. From the analysis of the state of the art (see Section V), we have identified three main challenges: *(i)* unclear identification of the software qualities, *(ii)* absence of an architectural language for describing microservice architectures, *(iii)* high complexity involved in the migration of existing software systems towards the adoption of microservice-based architectures. On the basis of these challenges, we have defined the following research questions.

- **RQ-1** - What are the key properties of MSA?
- **RQ-2** - How to describe MSA to guide architectural analysis?
- **RQ-3** - How to estimate the impact of MSA migration?

In **RQ-1**, by *key properties* we mean the inherent properties of a software product, which are classified as either functional properties or quality properties. Functional properties determine what the software is able to do. Quality properties determine how well the software performs [21]. Since software qualities requirements drive the architecture design [34], it is of great importance to understand how MSA affects software qualities. There are many reasons why it is important to address **RQ-1**. First, it is necessary to understand what are the software qualities that MSA can satisfy globally, and locally. Second, understanding the implications of different architectural designs can be useful to discover and identify specific architectural patterns useful for solving recurring problems, or for avoiding poorly designed solutions (e.g., antipatterns).

978-1-5090-4793-2/17 $31.00 © 2017 IEEE
DOI 10.1109/ICSAW.2017.65

Third, identifying the factors which impact the architecture's trade offs can be useful for supporting the reasoning towards the desired direction.

By addressing **RQ-2** we want to provide support to the architectural design activities. An *architectural language* (AL) specific for this domain might be the proper abstraction for modeling these architectures. Indeed, by means of an architectural language, software architects could be able to describe, in a clear, unambiguous and precise manner the architecture of a system. An interesting investigation on the characteristics of next-generation ALs [26], and an industrial empirical study on ALs [30] can provide the basis for an initial evaluation of how existing ALs, including *UML* [40], can be helpful for designing microservice architectures. In this direction, we will investigate existing ALs tailored for service-based systems and on the support they may offer for MSA. Depending on the results that we will gather in this phase, we will likely continue our work by exploiting the most promising ALs for satisfying microservice requirements, either trying to extend or customize the ALs to the domain's needs. If none of the selected ALs is able to provide the necessary capabilities for architecting microservices, then we will try to design a new AL based on MSA peculiarities (e.g., small size, communication only thorough interfaces, teams) and proceed with its implementation, if possible. Many advantages come from using models in software architecture, especially if *Model-driven engineering* (MDE) methodologies and tools are used. MDE is an area of software engineering that promotes models as first class citizens of the software development process [3]. A major benefit of MDE is its ability to shift the development focus from code centric techniques to models expressed in proper domain-specific modeling languages. MDE uses models to simplify and automate activities, as for example automatic code generation of modules and interfaces, or generation of deployment descriptors [9]. Models can also be used to perform different type of analysis, as performance analysis, model checking, simulations, architectural analysis. In addition, models are recently being investigated at runtime in order to cope with the dynamic aspects of ever-changing software and its environment [41].

The third important challenge of this research is related to the industrial adoption of microservice architectures on the large scale. As increasing number of companies are considering cloud as a target platform for migrating their applications [22], the migration process will increase in importance. A number of approaches for migrating towards MSA are starting to arise [1, 24], however the factors that impact the overall migration process have not yet been investigated in detail. Being able to recognize and estimate the impact of the factors involved in the migration process could be of significant advantage for all the companies willing to adopt MSA as they could use estimations to evaluate the feasibility of a product migration within specific effort and budget constraints. In addition, they could be able to analyze different design options for their applications (for example considering different levels of service granularity).

III. Proposed Solution

This section describes the current progress and the planned steps for addressing the research questions introduced in Section II. In Figure 1 an overview of the research plan is reported.

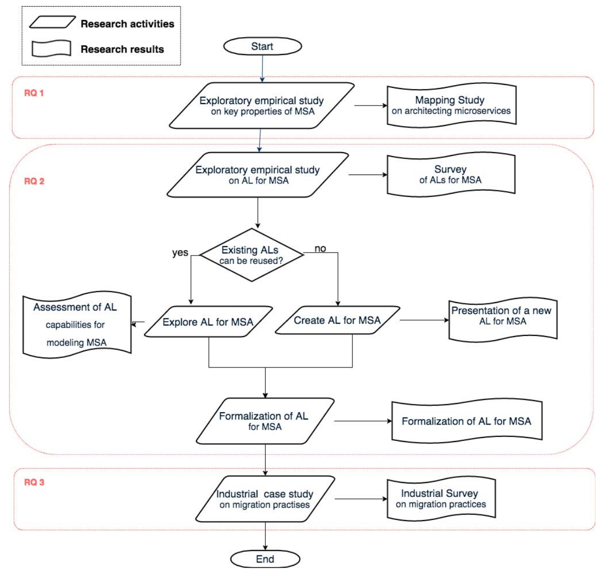

Fig. 1. Research activities and expected results

A. Preliminary Results

The first research question (**RQ-1**) has been addressed by performing a systematic mapping study on architecting microservices [13]. The mapping study has been designed following well-established guidelines [25, 37, 43] and has addressed the following research questions:

1) What are the publication trends of research studies about architecting microservices?
2) What is the focus of research on architecting microservices?
3) What is the potential for industrial adoption of existing research on architecting microservices?

By answering these research questions, the mapping study has characterized many aspects of MSA and also revealed some possible research gaps. For example, the focus on the quality attributes of microservice architectures (**RQ-1**) has revealed that *performance* and *maintainability* are among the most investigated quality attributes, while possible research gaps could exist in the field of security, reliability and portability for MSA [13].

The mapping study has also confirmed that describing microservice architectures (**RQ-2**) is a significant challenge. Indeed, the architectural languages used for describing microservices architectures are mainly *informal architectural languages*, while in few other cases *UML* or domain specific languages have been used or proposed. In this research project, finding the proper methods and techniques for describing and designing microservice architecture is a main concern, which is currently under investigation.

In this direction, we have presented: *(i)* an approach for recovering microservice architecture named MicroART [17], and *(ii)* the implementation of the MicroART first prototype [16]. The approach for recovering microservice architecture presented in [17] was realized by using Model-Driven Engineering techniques, which allowed us to leverage a suitably defined domain-specific language (DSL) for representing the key aspects of the architecture of a microservice-based system, and also to provide a tool-chain for automatically extracting architecture models of the system. We evaluated MicroART on a publicly available benchmark system, with promising results. We present the first prototype of MicroART in [16]. The tool is able to generate models of the software architecture of a microservice-based system that can be managed by software architects for multiple purposes.

It is very difficult to have a clear understanding of the overall architecture of a microservice-based software system, specially when the deployment and operation of the involved microservices evolves at run-time, and by proposing MicroART we aim to move our first steps targeting this need.

B. Ongoing Research

Currently, the research is focusing on the analysis of the concerns arising when architecting a microservice-based system. This preliminary step is required in order to define a suitable set of characteristics that an architectural language for microservices needs to support.

To find the proper level of abstraction, we are proceeding our investigation towards two different directions: *(i)* reverse architecting of AcmeAir[1], an open-source benchmark system for benchmarking microservices-based systems, *(ii)* analysis of the existing architectural languages for modeling service-oriented systems. AcmeAir is a web-based system available in two different architectures (i.e., monolithic service and microservice) and in two different languages (i.e., Node.js and Java), thus providing to researchers an useful benchmark for evaluating, measuring, and comparing their own solutions over a common reference system [13].

The analysis of existing architectural languages for modeling service-oriented systems is only at the beginning, as we are now selecting the initial candidates capable of supporting modeling of microservice architectures. Hints to the necessity of a suitable AL for describing MSA are confirmed in [13] where the authors noted that the majority of the analysed primary studies used *informal* architectural languages for this purpose, and in [2] where the authors state that the lack of consistency in diagrammatic presentations may indicate a need to propose a comprehensive modeling view/language that best covers and describes a microservices-based architecture. *UML* is a promising candidate, even though the lack of a predominant architectural language in the MSA field can indicate difficulties in the description and modeling of microservice architectures [13].

[1]https://github.com/acmeair/acmeair

In this phase, our goal is to find one or more suitable architectural languages that can be reused or extended for the microservices needs. On the most promising architectural languages we will investigate how Model-Driven Engineering techniques can be applied. If no candidates will be selected, we will proceed to define in details the requirements necessary to design a new Domain Specific Language (DSL) for MSA, and possibly proceed with its implementation.

C. Planned Steps

There are three main steps planned for the completion of this research project. The first goal is the realization of a *survey on the architectural languages* that can be used for supporting microservice architectures activities. The second phase will be focusing on the *assessment of the architectural languages capabilities* needed for modeling and designing microservice architectures. If we did not succeed on finding a suitable architectural language for MSA we will design and formalize a DSL for this purpose. The third part of this research will be focusing on *industrial case studies*, upon which we will investigate the experiences, tools and practices used during the migration process in order to identify the factors affecting this process. The targeted output of this process will be an industrial survey on the practices used and a study of the overall impact of the *migration process* in industry. We will also define attributes on models (defined using the architectural language resulting from the previous step) to perform an automatic analysis on the estimation of the impact of the migration process, likely measuring the estimation in terms of effort and time.

IV. Industrial Impact

A first contribution of this research has been provided by the mapping study performed on architecting microservices [13]. Indeed, the mapping study has investigated the *potential for industrial adoption of existing research on architecting microservices*, revealing that: *(i)* the low technology readiness level scores of most studies suggest that industrial transferability is far away, and that *(ii)* the balanced involvement of industrial and academic authors is however promising for knowledge co-creation and cross-fertilization [13].

The second contribution to the industrial field we aim to achieve is the definition of an appropriate and agile tool for designing microservice-based architectures. Our first work in this direction is MicroART [16, 17]. There are many purposes we are targeting for this tool, such as: modeling, documenting, analysis, support to development and deployment.

A third contribution could be the knowledge coming from the analysis of industrial case studies (**RQ-3**), upon which we will investigate the experiences, tools and practices used during the migration process.

Other goals we aim to achieve are: architecture recovery from the application of reverse engineering techniques, and qualitative and quantitative analysis on the architecture models of existing systems in order to provide an estimation for the migration effort involved in the adoption of microservice-based architectures.

V. Related Work

In this section, the related work on the area of interest of this PhD project is discussed.

A. Software architecture for MSA

Microservice architecture is an emerging research topic. The state of the art in the field is composed of a limited amount of papers, each of them providing (preliminary) solutions for the main concerns faced when architecting a microservice-oriented system, such as availability, scalability, etc.
This trend is also certified by a systematic mapping study performed by Pahl and Jamshidi [36] that reported a set of 21 primary studies only. This systematic mapping study is a first classification of the research directions on the field which highlighted that the relevant perspectives considered by researchers are: *(i)* a clear focus on cloud and container technologies when considering the deployment of the system, *(ii)* software migration for modernising monolithic legacy systems towards a microservice-based architecture, *(iii)* long-term software evolution and maintainability of microservice-based systems.

The focus on cloud and container technologies [7, 8, 10, 27, 42] is due to the fact that microservices leverage the elasticity of resources provided by these types of infrastructures. As reported in [4, 35], among the major barriers to cloud adoption there are *security* and *vendor lock-in* and *quality assurance*. Vendor lock-in leads to a decrease of application's portability and represents a major concern for MSA which needs to be cloud agnostic in order to avoid infrastructure constraints. The systematic mapping study has further highlighted a strong perspective on architecture of microservices, and in this direction we have noticed that there is not a significant focus on the quality properties and their impact.

Dragoni et al. performed a survey on microservices in general [14]; one of the main results of the survey is that MSA is intrinsically related and has direct impact on specific quality properties at the system level, specifically: availability, reliability, maintainability, performance, security and testability qualities. Availability and scalability are largely discussed in multiple resources and books [33, 39], but little investigation has been performed on the other named quality properties. Hassan and Bahsoon in a recent study [19] have addressed some design problems about estimating the proper level of granularity of services. More specifically, they considered the existing trade offs between the following design concerns: *size* versus the *number* of services, and *local* versus *global* satisfaction of quality properties. O' Brien et al. have investigated the positive and negative effects that SOA has on quality properties of a system [34], providing also a classification of the degree of the impact on these qualities.

In conclusion, in the state of the art of MSA, many quality properties (with their trade offs), design tactics, and architectural analysis and reasoning techniques must still be investigated and explored in order to identify and take full advantage of the potential benefits that MSA can offer.

B. Architectural languages for MSA

The state of the art does not seem to currently offer a architectural language for specifically describing and designing microservice architectures. The most promising languages for MSA are the ALs used to describe service-based architectures. Among the most relevant we have identified the following SoaML, SOMA, SOADL and StratusML.

SoaML [11, 12] is a modeling language for SOA systems and was adopted in 2009 by the OMG. SOAML is an UML profile for modeling services on business and system levels, service contracts and interfaces, choreography, and more. SoaML metamodel extends the UML2 metamodel to support an explicit service modeling in distributed environments and can be used by most of UML tools.

Service Oriented Modeling and Architecture (SOMA) [5] is a modeling technique proposed by IBM which focuses on service identification, specification and realization.

SOADL [23] specifies the interfaces, behavior, semantics and quality properties of services, providing mechanisms to model and analyze the dynamic and evolving architecture, and supports architecture-based service composition.

StratusML [18] is a technology agnostic modeling framework for cloud applications which allows to define services, configuration, and runtime behavior of applications through a set of adaptation rules, and estimate cost under diverse cloud platforms and configurations.

More details on the comparison of service modeling methods are discussed in [32], while modeling service oriented architectures with UML is presented in [29]. A detailed analysis of these modeling languages will be performed during this thesis work, in order to evaluate if these languages can be suitable to model and analyze microservice architecture.

C. Impact of MSA migration

The migration of systems toward microservices is a major concern in adopting MSA. The difficulties involved are multiple. First, the migration process to adopt cloud architectures is a multi-dimensional problem and a non-trivial task [22]. Second, being microservices an emergent topic, not enough material on the migration techniques is yet available. Third, it is not currently possible to know the impact of migrating an existing application.

Several research papers have addressed the challenges of migration to cloud platforms. Jamshidi et al. have performed a systematic review on cloud migration research [22], in which it emerges that many companies are considering cloud as a target platform for migration and a set of legacy-to-cloud solutions already exists, even though the research is still in an early stage.

On a more specific perspective on migration towards microservices, fewer results have been published. Lin et al. [28] propose a migration method that adopts microservice architectures to support an effective migration of Web application to the clouds. Balalaie et al. [7] confirms that, in general, the application architectures are not ready to fully take advantage of cloud platform, and their adaptation to the cloud environment is a non-trivial task. They discuss their experience on the migration of a monolithic software to MSA.

Ardagna et al. propose a model-driven approach named MODACLOUDS [4] claiming that model driven development allows to design software systems in a cloud-agnostic way and to be supported by model transformation techniques into the process of instantiating the system into specific, possibly, multiple clouds. MODACLOUDS analyzes the impact of cloud adoption, evaluating multiple clouds for the same system, enables risk analysis for the selection of cloud providers. A runtime environment for observing the system under execution is also provided.

Under a reuse of knowledge perspective, Balalaie et al. have defined an initial repository of microservices migration patterns [6] with the goal of building the basic ground for a Situational Method Engineering [20] in this area.

Being the migration towards microservices an incremental and iterative process, the decomposition process of splitting an application into multiple independent smaller services has its challenges. Decomposition can be based on service boundaries [33], on data driven as the database-is-the-service pattern [31], on approaches as SMART [1] or Entice [24], etc. Concerning the area of SOA migration, both academic and industrial approaches have been investigated by Razavian in [38].

The process of migrating application towards MSA involves many risks [4] and although there are many works on the migration process, there are not clear approaches on how to provide a quantitative or qualitative estimation of the migration impact.

VI. Conclusions

This paper presents the motivation and initial steps of an investigation on microservice architectures. The main goal of this project is to realize a comprehensive understanding of the MSA style and its architecture properties, while also providing support for the architecture design of new applications.

The initial contributions to the field are: (*i*) a systematic mapping study on architecting microservices, (*ii*) an approach for architecture recovery of microservice-based systems named MicroART, and (*iii*) the implementation of the MicroART first prototype. Currently the research is progressing towards the analysis of the requirements necessary to describe and design microservice architectures.

A goal of this research is to contribute to industry by providing the tools for supporting the activity of architecting microservices, but also to provide the means for the evaluation of the effort involved in migrating existing applications towards the adoption a microservice-based architectures.

VII. Acknowledgment

I would like to thank my supervisors, Prof. Patricia Lago and Dr. Ivano Malavolta for their guidance and support during this PhD research project.

References

[1] Smart: Analyzing the reuse potential of legacy components in a service-oriented architecture environment.

[2] N. Alshuqayran, N. Ali, and R. Evans. A Systematic Mapping Study in Microservice Architecture. In *Proc. of the 9th International Conference on Service-Oriented Computing and Applications*. IEEE, IEEE, 2016.

[3] M. Amrani, B. Combemale, L. Lúcio, G. Selim, J. Dingel, Y. Le Traon, H. Vangheluwe, and J. R. Cordy. Formal verification techniques for model transformations: A tridimensional classification. *The Journal of Object Technology*, 14(3):1–1, 2015.

[4] D. Ardagna, E. Di Nitto, G. Casale, D. Petcu, P. Mohagheghi, S. Mosser, P. Matthews, A. Gericke, C. Ballagny, F. D'Andria, et al. Modaclouds: A model-driven approach for the design and execution of applications on multiple clouds. In *Proceedings of the 4th International Workshop on Modeling in Software Engineering*, pages 50–56. IEEE Press, 2012.

[5] A. Arsanjani, S. Ghosh, A. Allam, T. Abdollah, S. Ganapathy, and K. Holley. Soma: A method for developing service-oriented solutions. *IBM systems Journal*, 47(3):377–396, 2008.

[6] A. Balalaie, A. Heydarnoori, and P. Jamshidi. Microservices migration patterns. Technical report, Tech. Rep. TR-SUTCE-ASE-2015-01, Automated Software Engineering Group, Sharif University of Technology, Tehran, Iran, 2015.

[7] A. Balalaie, A. Heydarnoori, and P. Jamshidi. Migrating to cloud-native architectures using microservices: An experience report. In *European Conference on Service-Oriented and Cloud Computing*, pages 201–215. Springer, 2015.

[8] A. Balalaie, A. Heydarnoori, and P. Jamshidi. Microservices architecture enables devops: Migration to a cloud-native architecture. *IEEE Software*, 33(3):42–52, 2016.

[9] K. Balasubramanian, A. Gokhale, G. Karsai, J. Sztipanovits, and S. Neema. Developing applications using model-driven design environments. *Computer*, 39(2):33–40, 2006.

[10] L. Bass, I. Weber, and L. Zhu. *DevOps: A Software Architect's Perspective*. Addison-Wesley Professional, 2015.

[11] C. Casanave. Service oriented architecture using the omg soaml standard. *Model Driven Solution*, 2009.

[12] C. Casanave. Enterprise service oriented architecture using the omg soaml standard, a model driven solutions. *ModelDriven. org*, 2012.

[13] P. Di Francesco, P. Lago, and I. Malavolta. Research on architecting microservices: Trends, focus, and potential for industrial adoption. *IEEE International Conference on Software Architecture (ICSA)*, 2017.

[14] N. Dragoni, S. Giallorenzo, A. L. Lafuente, M. Mazzara, F. Montesi, R. Mustafin, and L. Safina. Microservices: yesterday, today, and tomorrow. *arXiv preprint*

arXiv:1606.04036, 2016.

[15] M. Fowler and J. Lewis. Microservices a definition of this new architectural term. *URL: http://martinfowler.com/articles/microservices.html*, Last accessed: Feb 2017.

[16] G. Granchelli, M. Cardarelli, P. Di Francesco, I. Malavolta, L. Iovino, and A. Di Salle. Microart: A software architecture recovery tool for maintaining microservice-based systems. *IEEE International Conference on Software Architecture (ICSA)*, 2017.

[17] G. Granchelli, M. Cardarelli, P. Di Francesco, I. Malavolta, L. Iovino, and A. Di Salle. Towards recovering the software architecture of microservice-based systems. *First International Workshop on Architecting with MicroServices (AMS)*, 2017.

[18] M. Hamdaqa and L. Tahvildari. Stratus ml: A layered cloud modeling framework. In *Cloud Engineering (IC2E), 2015 IEEE International Conference on*, pages 96–105. IEEE, 2015.

[19] S. Hassan and R. Bahsoon. Microservices and their design trade-offs: A self-adaptive roadmap. In *Services Computing (SCC), 2016 IEEE International Conference on*, pages 813–818. IEEE, 2016.

[20] B. Henderson-Sellers, J. Ralyté, P. J. Ågerfalk, and M. Rossi. *Situational Method Engineering*. Springer, 2014.

[21] I. Iso. Iec25010: 2011 systems and software engineering–systems and software quality requirements and evaluation (square)–system and software quality models. *International Organization for Standardization*, 34:2910, 2011.

[22] P. Jamshidi, A. Ahmad, and C. Pahl. Cloud migration research: a systematic review. *IEEE Transactions on Cloud Computing*, 1(2):142–157, 2013.

[23] X. Jia, S. Ying, T. Zhang, H. Cao, and D. Xie. A new architecture description language for service-oriented architec. In *Sixth International Conference on Grid and Cooperative Computing (GCC 2007)*, pages 96–103. IEEE, 2007.

[24] G. Kecskemeti, A. C. Marosi, and A. Kertesz. The entice approach to decompose monolithic services into microservices. In *High Performance Computing & Simulation (HPCS), 2016 International Conference on*, pages 591–596. IEEE, 2016.

[25] B. Kitchenham and P. Brereton. A systematic review of systematic review process research in software engineering. *Information and software technology*, 55(12):2049–2075, 2013.

[26] P. Lago, I. Malavolta, H. Muccini, P. Pelliccione, and A. Tang. The road ahead for architectural languages. *IEEE Software*, 32(1):98–105, 2015.

[27] V. D. Le, M. M. Neff, R. V. Stewart, R. Kelley, E. Fritzinger, S. M. Dascalu, and F. C. Harris. Microservice-based architecture for the nrdc. In *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, pages 1659–1664. IEEE, 2015.

[28] J. Lin, L. C. Lin, and S. Huang. Migrating web applications to clouds with microservice architectures. In *Applied System Innovation (ICASI), 2016 International Conference on*, pages 1–4. IEEE, 2016.

[29] M. López-Sanz, C. J. Acuña, C. E. Cuesta, and E. Marcos. Modelling of service-oriented architectures with uml. *Electronic Notes in Theoretical Computer Science*, 194(4):23–37, 2008.

[30] I. Malavolta, P. Lago, H. Muccini, P. Pelliccione, and A. Tang. What industry needs from architectural languages: A survey. *IEEE Transactions on Software Engineering*, 39(6):869–891, 2013.

[31] A. Messina, R. Rizzo, P. Storniolo, M. Tripiciano, and A. Urso. The database-is-the-service pattern for microservice architectures. In *International Conference on Information Technology in Bio-and Medical Informatics*, pages 223–233. Springer, 2016.

[32] M. Mohammadi and M. Mukhtar. A review of soa modeling approaches for enterprise information systems. *Procedia Technology*, 11:794–800, 2013.

[33] S. Newman. *Building Microservices*. " O'Reilly Media, Inc.", 2015.

[34] L. O'Brien, P. Merson, and L. Bass. Quality attributes for service-oriented architectures. In *Proceedings of the international Workshop on Systems Development in SOA Environments*, page 3. IEEE Computer Society, 2007.

[35] J. Opara-Martins, R. Sahandi, and F. Tian. Critical analysis of vendor lock-in and its impact on cloud computing migration: a business perspective. *Journal of Cloud Computing*, 5(1):1–18, 2016.

[36] C. Pahl and P. Jamshidi. Microservices: A Systematic Mapping Study. In *Proceedings of the 6th International Conference on Cloud Computing and Services Science, Rome, Italy*, pages 137–146, 2016.

[37] K. Petersen, S. Vakkalanka, and L. Kuzniarz. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1–18, 2015.

[38] M. Razavian. *Knowledge-driven Migration to Services*. PhD thesis, Vrije Universiteit, 2013.

[39] M. Richards. Microservices vs. service-oriented architecture, 2015.

[40] J. Rumbaugh, I. Jacobson, and G. Booch. *Unified modeling language reference manual, the*. Pearson Higher Education, 2004.

[41] M. Szvetits and U. Zdun. Systematic literature review of the objectives, techniques, kinds, and architectures of models at runtime. *Software & Systems Modeling*, 15(1):31–69, 2016.

[42] G. Toffetti, S. Brunner, M. Blöchlinger, F. Dudouet, and A. Edmonds. An architecture for self-managing microservices. In *Proceedings of the 1st International Workshop on Automated Incident Management in Cloud*, pages 19–24. ACM, 2015.

[43] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering*. Computer Science. Springer, 2012.