We Need to Talk about Microservices: an Analysis from the Discussions on StackOverflow

Alan Bandeira, Carlos Alberto Medeiros, Matheus Paixao, Paulo Henrique Maia
{alan.bandeira,carlos.fernandes}@aluno.uece.br, {matheus.paixao, pauloh.maia}@uece.br
State University of Ceara - Fortaleza, Brazil

Abstract—**Microservices are a new and rapidly growing architectural model aimed at developing highly scalable software solutions based on independently deployable and evolvable components. Due to its novelty, microservice-related discussions are increasing in Q&A websites, such as StackOverflow (SO). In order to understand what is being discussed by the microservice community, this work has applied mining techniques and topic modelling to a manually-curated dataset of 1,043 microservice-related posts from StackOverflow. As a result, we found that 13.68% of microservice technical posts on SO discuss a single technology: Netflix Eureka. Moreover, buzzwords in the microservice ecosystem, e.g., blue/green deployment, were not identified as relevant subjects of discussion on SO. Finally, we show how a high discussion rate on SO may not reflect the popularity of a certain subject within the microservice community.**

Keywords-**Microservice; StackOverflow; Topic Modelling**

I. INTRODUCTION

In the traditional monolithic architecture, a software system is constructed as a single logic unit that aggregates several services that share the same computational resources (e.g., memory space, CPU processing and database) in order to provide business functionalities [1]. Although monolithic applications are simple to be developed, they present limitations, such as difficulties to maintain and evolve, followed by inefficient scaling of computational resources, which may cause an over(under)provisioning of those resources [2].

With the advance of cloud computing and containerization technologies, *microservices* arise as a novel architectural style to develop a single application as a collection of independent, well-defined, and intercommunicating services [1][3]. Since microservices can be individually scaled, they provide an efficient manner to allocate computational resources, enabling flexible horizontal scaling in cloud environments [4].

For this reason, this new architectural style has been adopted by notorious tech companies, such as Netflix, Uber and LinkedIn [5], which, in turn, is pushing both industrial and open source software development through continuous contributions of open source tools backed up by large cloud computing providers such as Microsoft and Amazon [6]. Furthermore, recent studies have proposed processes and frameworks to migrate legacy monolithic systems to a microservice-based architecture, thus benefiting companies that desire to reuse and evolve their current systems [7][8][9][10].

Due to its novelty, the microservice architecture has brought new (and evolved existing) concepts, technologies and tools from both software engineering and distributed systems [6], which is reflected in the rapid increase of discussions about the subject in Q&A websites, such as StackOverflow (SO). Although SO has been successfully used in studies regarding different domains, such as mobile applications [11][12], cloud computing [13], software architecture [14], and web-based communication systems [15], to the best of our knowledge, there is no work that investigates how microservices are being debated in SO. This analysis is important since it may guide both academic and industrial research towards solutions that have the potential to impact real-world software engineering practitioners with respect to this new architectural style.

Therefore, this paper presents an analysis that aims at answering two research questions: "what microservice-related subjects are developers discussing on StackOverflow?" (RQ1) and "what microservice-related subjects receive the most attention from developers?" (RQ2). The contributions of this paper are threefold: (i) a categorization of microservice-related discussions in SO; (ii) a replicable framework for extracting technical and conceptual subjects from discussions in SO; (iii) a manually curated dataset of microservice discussions in SO.

II. BACKGROUND AND DEFINITIONS

Our goal is to study discussions on SO, i.e., exchanges of ideas between two or more people concerning a certain subject. Therefore, questions that received no answers, or questions in which all the answers were given by the question's author are not interesting to our study. Moreover, since comments on SO do not consist of a response to the question [16], we have not considered them in our study. Hence, we define a discussion on SO as being a *combination of a question and one (or more) answers, where there is at least one answer whose author is not the author of the question*.

On StackOverflow, a user has the choice of selecting one or more tags to describe a question. The set of tags is mostly used to indicate the question's subjects and to identify other SO users that might be able to provide an answer to the question. For this study, we consider as **microservice-tagged** the discussions in which the question's author employed the 'microservices' tag. Note that SO enforces that variations on the spelling of microservices will always be mapped to the same tag. Moreover, we noticed that not all microservice-tagged discussions concern microservice subjects. Thus, we consider as **microservice-related** the subset of discussions that actually discuss ideas, notions and concepts related to the microservice ecosystem. More details on the identification of microservice-related discussions are provided in Section III-B.
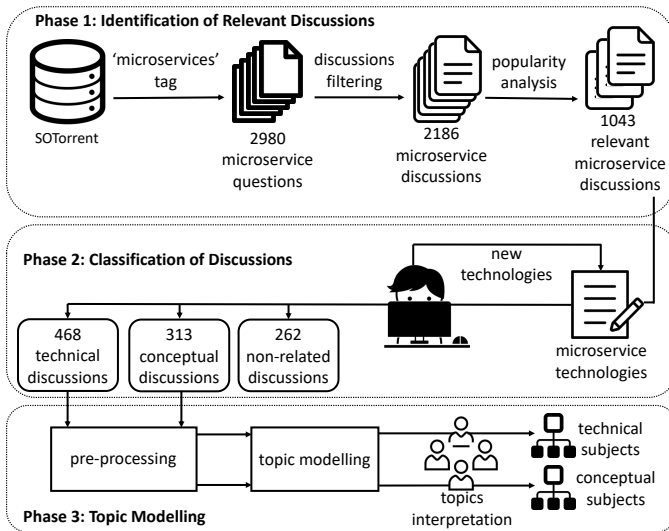
Fig. 1: Framework for extracting technical and conceptual subjects from discussions on SO. Phase 1 identifies relevant microservice discussions, and phase 2 performs a manual analysis to identify technical and conceptual discussions. Phase 3 employs topic modelling to group the discussions into subjects.

III. EMPIRICAL STUDY DESIGN

Our goal is to study microservice-related discussions on StackOverflow. Hence, we ask two research questions:

RQ1: What microservice-related subjects are developers discussing on StackOverflow?

RQ2: What microservice-related subjects receive the most attention from developers?

Our supporting website [17] presents a complete replication package for our study, including the raw data and outcomes for all steps of our experimental methodology and analyses.

A. Identification of relevant discussions on StackOverflow

In this paper, we make use of the SOTorrent dataset [18] (release 2018_09_23). The framework we employed to identify microservice-related discussions from SO is displayed in Figure 1. The framework consists of three phases. On the first phase, we retrieved all questions from SOTorrent in which the author used the *microservices* tag, resulting in 2,980 microservice-tagged questions. Next, we filtered the discussions, as described in Section II. This resulted in a total of 2,186 microservice-tagged discussions.

SO employs five different proxies for popularity: *Answer-Count*, *ViewCount*, *CommentCount*, *FavoriteCount*, and *Score*. For each of these metrics, we ranked the microservice-tagged discussions and extracted all discussions above the 3rd quartile of the distribution. Our rationale is that popular questions in the SO community will less likely be affected by noise [19][20]. Finally, we collected a corpus of 1,043 relevant microservice-tagged discussions according to each of the five popularity proxies employed by SO.

B. Classification of microservice discussions

The second phase of our experimental framework aims at classifying the microservice-tagged discussions into separate subsets. Consider the discussion whose question is *"How to stop a spring boot service from command line?"*, for example. As one can see, this discussion is much more related to Spring Boot than to microservices. Hence, we defined this type of discussions as **Non-Related Discussions**. In addition, we noticed that some discussions debated microservice subjects at different levels. While some dealt with low-level technical issues, such as *"Host WepAPI on Service Fabric"*, others debated high-level concepts, e.g., *"Microservices: what are pros and cons?"*. These two types of discussions are inherently different and need to be studied in a separate fashion. Thus, we defined the first type of discussions as **Technical Discussions** and the second one as **Conceptual Discussions**. Note that the union of technical and conceptual discussions forms the set of microservice-related discussions.

We employed a manual analysis procedure to classify each microservice-tagged discussion into one of the three discussion types defined above. In the manual analysis, two authors of the paper separately served as reviewers for the same subset of discussions. In case of disagreement, the two reviewers debated the classification until an agreement was reached. In case of no agreement after debating, a third author was involved to settle the classification.

Initially, we considered technical discussions to be the ones mentioning microservice technologies. Thus, we leveraged the list of 42 microservice technologies provided by Jamshidi et al. [6]. Through a mixed method of token matching and regular expressions, we identified a preliminary set of technical discussions. To cover for microservice technologies not included in the list, we manually reviewed the discussions not initially classified as technical to identify additional technologies. Finally, we identified 27 new microservice technologies, resulting in a final set of 468 technical discussions. For this manual analysis, the inter-reviewer agreement rate was 94.4%. Next, we manually analyzed the remaining 575 discussions to identify the conceptual and non-related discussions. As a result, we reached the final subset of 313 conceptual discussions and 262 non-related discussions. For this manual analysis, the inter-reviewer agreement rate was 74.9%.

C. Topic Modelling

On the third phase, we employed topic modelling to group both technical and conceptual discussions into different subjects and categories. First, we performed three pre-processing steps according to surveys and guidelines on the application of topic modelling in software engineering [21][22]: stop words removal, lemmatization and pruning. For stop words removal and lemmatization, we employed a curated dataset of English stop words [23] and the SpaCy library [24], respectively. For pruning, we removed the words that appeared in more than 80% and less than 2% of the documents [21].

For the topic modelling, we employed Latent Dirichlet Allocation (LDA) [25], and the mallet tool [26]. The LDA settings were configured based on guidelines [21], followed by empirical experimentation on our own dataset. We used $k = 20$ topics, being each topic described by 15 words. The same
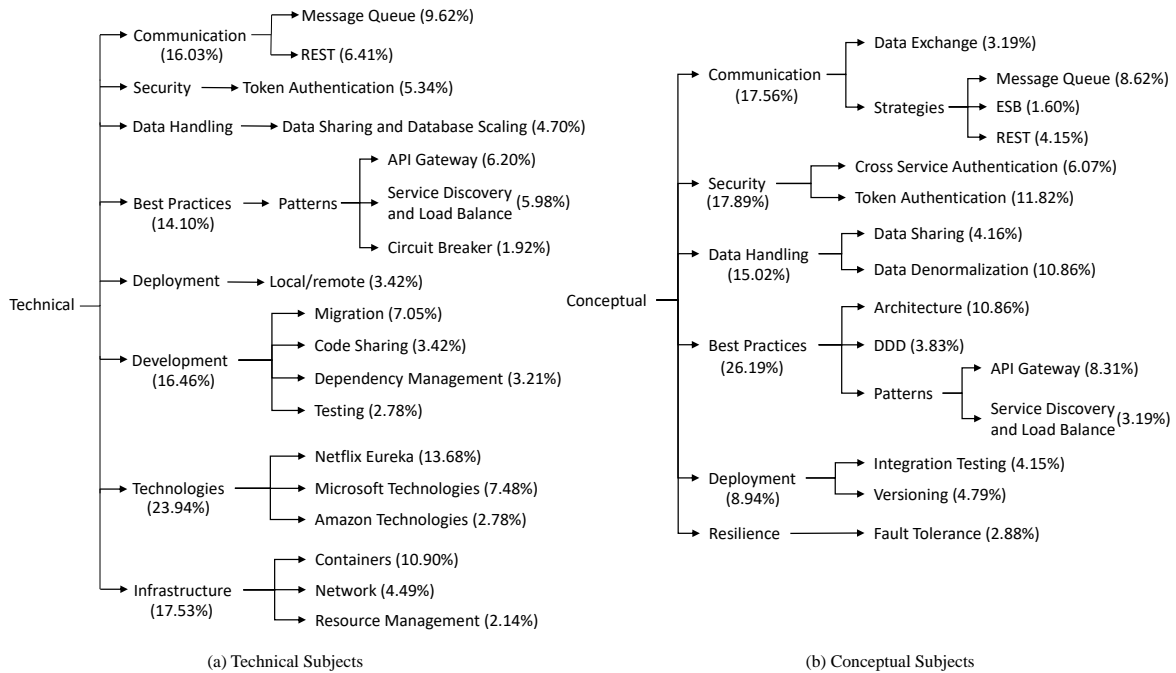
Fig. 2: Categorization of microservice-related subjects discussed in technical and conceptual discussions on StackOverflow, as identified by LDA topic modelling. In addition, we report the proportion of discussions in which each subject is discussed.

settings were used for the LDA execution on both technical and conceptual discussions.

Finally, the topics provided by LDA were interpreted by all authors of the paper in group sessions, where we created a categorization for each subset of discussions. Furthermore, LDA provides not only the set of topics but also the similarity of each discussion to each extracted topic. Hence, we leveraged this information to identify the most discussed subjects.

IV. EMPIRICAL STUDY RESULTS

A. RQ1: What microservices-related subjects are developers discussing on StackOverflow?

Our interpretation for LDA's extracted topics for the technical and conceptual discussions are displayed in Figures 2a and 2b, respectively. Each of the 20 topics extracted by LDA was interpreted as a single subject of discussion. These are represented by the endpoints in the categorization. In addition, we merged related topics into the same subject. For technical discussions, we merged two topics into Message Queue and other two into Microsoft Technologies, obtaining 18 technical subjects. For conceptual discussions, we merged two topics into Message Queue, two topics into DDD and two topics into Data Sharing. Finally, we interpreted one of the conceptual topics as inconclusive, resulting in 15 conceptual subjects.

The categories were designed by us to group related subjects into high-level concerns. Furthermore, for each subject and category, we display the percentage of discussions described by the subject. Consider the REST subject, for example. The figure indicates that one of the topics extracted by LDA for the technical discussions is related to the REST technology. In addition, 6.41% of technical discussions are discussing REST.

Due to space constraints, we cannot delve into all the details of our categorization. Nevertheless, we discuss a few key points next. First, consider the Message Queue and Token Authentication subjects, for example. These concepts arose as solutions and patterns for new challenges presented by the microservice architecture. As one can see, both are highly discussed subjects, as 9.62% and 8.62% of technical and conceptual discussions debate Message Queue, respectively. For Token Authentication, the results are 5.34% and 11.82%. The high rate of discussion for these two subjects in both technical and conceptual discussions indicate that these subjects may still be considered as 'open problems'. Even though developers discuss the tools that implement these subjects, SO's users are still discussing the basic concepts around these subjects in order to obtain a better understanding of these concepts.

Differently, consider the Containers subject. Although representing 10.90% of technical discussions, Containers are not discussed in conceptual discussions. This may indicate that this is a well-defined and understood concept within the microservices community, in which developers mostly discuss the tools and technologies that implement it.

Although similar, microservices are not a variation or an evolution of SOA [27]. Nevertheless, we noticed that the difference between microservices and SOA was a cross-cutting concern within microservice discussions on SO. This is exemplified by the Architecture subject, which was discussed in 10.86% of conceptual discussions, where developers mostly discussed the differences between these architectural styles.

Moreover, one may notice that 1.60% of conceptual discussions are concerned with the ESB pattern. ESBs were a dominant communication strategy in SOA applications, yet its usage in the microservice architecture is highly inad-

visable [1][28]. Hence, ESBs do not appear as a subject in technical discussions, which indicates that the microservices community indeed avoids this communication strategy. A similar scenario concerns the Fault Tolerance subject in conceptual discussions. Although it is a relevant and serious issue in SOA applications, its lack of discussion in technical discussions may indicate that this a 'solved' problem in the microservice ecosystem. In fact, fault tolerance is seemingly transparent for microservice developers nowadays as most cloud providers, such as Microsoft and Amazon, provide fault tolerance capabilities [29][30].

We also highlight subjects that are not listed in either technical nor conceptual categorizations. Consider canary testing and blue/green deployment, for example. Both subjects received a great amount of attention from both academic [31][32] and industrial [33][34] practitioners due to its applicability to the microservice architecture, yet these do not appear as subjects of discussion on SO. On the one hand, this may indicate that microservice developers are still mostly unaware of such deployment strategies. On the other hand, canary testing and blue/green deployment might simply not appeal to the wider microservice developer community.

As an answer to RQ1, we report developers discussing subjects related to Communication, Security, Data Handling, Best Practices and Deployment in both technical and conceptual discussions. Subjects concerned with Development, Technologies and Infrastructure are only discussed in technical discussions while Resilience is only discussed in conceptual discussions. Moreover, subjects discussed in both technical and conceptual discussions may indicate open problems in the microservice ecosystem. Differently, the absence of certain subjects from technical discussions may indicate either 'solved problems' or concepts that need a wider uptake by microservices practitioners. Finally, our observations are based on our interpretation of the categorization, where additional studies are necessary for a full comprehension of these phenomena.

B. RQ2: What microservice-related subjects receive the most attention from developers?

In RQ1, we presented the microservice-related subjects being discussed on StackOverflow, followed by the number of discussions concerned to each subject. In this context, one may assume that the amount of discussions on SO directly reflects the attention a subject receives from the community. To investigate this assumption, for both technical and conceptual discussions, we ranked the subjects according to each of the five popularity metrics employed by SO, as described in Section III-A. Due to space constraints, we limit our discussion to the *ViewCount* and *AnswerCount* metrics, as these reflect the most different aspects of SO's popularity.

For technical discussions, the discussions related to Netflix Eureka present a total of 116,485 views and 102 answers on SO, being ranked 1st for both metrics. In this case, we assessed that the subject with the largest number of discussions has also received most of the attention from the community. However, this is not the case for other subjects in technical discussions.

Containers, for example, represent the second biggest amount of discussions, yet it is ranked 7th and 4th in number of views and answers, respectively. This indicates that although constantly asked, containers questions neither attract many developers nor encourage users' interaction.

When considering conceptual discussions, we noticed that the two subjects with most discussions (Token Authentication and Data Denormalization) ranked 1st and 2nd in both number of views and number of answers. However, DDD only represents 3.83% of conceptual discussions, yet it ranked 4th in number of answers. This indicates that particular subjects may attract more discussion than others.

Finally, we highlight that popularity and attention on StackOverflow do not reflect the quality and/or amount of developers concerned to a certain subject or technology. As an example, our data indicates that Microsoft Technologies are more discussed than Amazon Technologies, while also being ranked above for both number of views and answers, yet Amazon holds the biggest market share in the cloud computing ecosystem [35]. Similarly to RQ1, further studies and analyses are necessary for a complete understanding of this issue.

V. THREATS TO THE VALIDITY

External: Our observations may not generalise for the entire microservice development ecosystem since we consider the data from a Q&A website as a representative sample of the community. To mitigate this threat, we used the StackOverflow data, which is the biggest computer science Q&A platform, being employed in many other studies [11][12][13][14][15].

Internal: Our manual classification of discussions and interpretation of topics may present personal biases. To alleviate this threat, the manual classification involved three authors of the paper, in which we achieved up to 94% agreement rate. The topics' interpretation was performed in open-debate sessions involving all authors of the paper. In addition, there might be microservice-related discussions that did not employ the 'microservices' tag, and therefore, were not included in our study. Nevertheless, we identified and analysed 1,043 discussions, which we consider to be a representative sample to base our observations upon.

VI. CONCLUSION

This paper presented an analysis of microservice-related discussions on StackOverflow. We classified 1,043 microservice-tagged posts into technical (44.87%), conceptual (30%) and non-related (25,13%) discussions and, taking into account the first two groups, we categorized the discussions using a topic modelling approach. We found 18 and 15 subjects for technical and conceptual discussions, respectively. Our findings indicate that Netflix Eureka is the most viewed and answered subject within technical discussions, while its conceptual counterpart is Token Authentication.

As future work, we plan to perform a detailed analysis regarding subjects' popularity on SO and its connection to usage in the microservice ecosystem. More specifically, we intend to share and discuss our findings with the microservice developer community in order to validate and enrich our study.

REFERENCES

[1] M. Fowler and J. Lewis, "Microservices," 2014. [Online]. Available: http://martinfowler.com/articles/microservices.html

[2] N. Dragoni, S. Giallorenzo, A. L. Lafuente, M. Mazzara, F. Montesi, R. Mustafin, and L. Safina, *Microservices: Yesterday, Today, and Tomorrow*. Cham: Springer International Publishing, 2017, pp. 195–216.

[3] S. Newman, *Building Microservices*, 1st ed. O'Reilly Media, Inc., 2015.

[4] I. Nadareishvili, R. Mitra, M. McLarty, and M. Amundsen, *Microservice Architecture: Aligning Principles, Practices, and Culture*, 1st ed. O'Reilly Media, Inc., 2016.

[5] M. Villamizar, O. Garcs, L. Ochoa, H. Castro, L. Salamanca, M. Verano, R. Casallas, S. Gil, C. Valencia, A. Zambrano, and M. Lang, "Infrastructure cost comparison of running web applications in the cloud using aws lambda and monolithic and microservice architectures," in *2016 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, May 2016, pp. 179–182.

[6] P. Jamshidi, C. Pahl, N. C. Mendonça, J. Lewis, and S. Tilkov, "Microservices: The journey so far and challenges ahead," *IEEE Software*, vol. 35, no. 3, pp. 24–35, 2018.

[7] A. Balalaie, A. Heydarnoori, and P. Jamshidi, "Microservices architecture enables devops: Migration to a cloud-native architecture," *IEEE Softw.*, vol. 33, no. 3, pp. 42–52, May 2016.

[8] A. Levcovitz, R. Terra, and M. T. Valente, "Towards a technique for extracting microservices from monolithic enterprise systems," *CoRR*, vol. abs/1605.03175, 2016.

[9] A. Bucchiarone, N. Dragoni, S. Dustdar, S. T. Larsen, and M. Mazzara, "From monolithic to microservices: An experience report from the banking domain," *IEEE Software*, vol. 35, no. 3, pp. 50–55, May 2018.

[10] H. Knoche and W. Hasselbring, "Using microservices for legacy software modernization," *IEEE Software*, vol. 35, no. 3, pp. 44–49, May 2018.

[11] L. Guerrouj, S. Azad, and P. C. Rigby, "The influence of app churn on app success and stackoverflow discussions," in *2015 IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, March 2015, pp. 321–330.

[12] C. Rosen and E. Shihab, "What are mobile developers asking about? A large scale study using stack overflow," *Empirical Software Engineering*, vol. 21, no. 3, pp. 1192–1223, jun 2016.

[13] M. Rahman, "Mitigating information disclosure attacks in the cloud by blocking invalid user and figure out problems to solve ddos by analyzing stackoverflow questions," in *2017 2nd International Conference on Electrical Electronic Engineering (ICEEE)*, Dec 2017, pp. 1–4.

[14] M. Soliman, M. Galster, A. R. Salama, and M. Riebisch, "Architectural knowledge for technology decisions in developer communities: An exploratory study with stackoverflow," in *2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, April 2016, pp. 128–133.

[15] S. A. Chowdhury and A. Hindle, "Mining stackoverflow to filter out off-topic irc discussion," in *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*, May 2015, pp. 422–425.

[16] StackOverflow. Stackoverflow's privileges. Accessed: 2019-01-04. [Online]. Available: https://stackoverflow.com/help/privileges

[17] A. Bandeira, C. Alberto, M. Paixao, and P. H. Maia. Replication package and supporting webpage for the paper: We need to talk about microservices. [Online]. Available: https://alanpbandeira.github.io/stackoverservices/

[18] S. Baltes, C. Treude, and S. Diehl, "Sotorrent: Studying the origin, evolution, and usage of stack overflow code snippets," in *Proceedings of the 16th International Conference on Mining Software Repositories (MSR 2019)*, 2019.

[19] D. Kavaler, D. Posnett, C. Gibler, H. Chen, P. Devanbu, and V. Filkov, "Using and asking: Apis used in the android market and asked about in stackoverflow," in *International Conference on Social Informatics*. Springer, 2013, pp. 405–418.

[20] M. Ahasanuzzaman, M. Asaduzzaman, C. K. Roy, and K. A. Schneider, "Mining duplicate questions in stack overflow," in *Proceedings of the 13th International Workshop on Mining Software Repositories - MSR '16*. New York, New York, USA: ACM Press, 2016, pp. 402–412.

[21] S. W. Thomas, A. E. Hassan, and D. Blostein, "Mining Unstructured Software Repositories," in *Evolving Software Systems*, T. Mens, A. Serebrenik, and A. Cleve, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 139–162.

[22] T.-H. Chen, S. W. Thomas, and A. E. Hassan, "A survey on the use of topic models when mining software repositories," *Empirical Software Engineering*, vol. 21, no. 5, pp. 1843–1919, oct 2016.

[23] I. Brigadir. Collection of english stop words. Accessed: 2019-01-04. [Online]. Available: https://github.com/igorbrigadir/stopwords

[24] spaCy. spacy: Industrial-strength nlp. Accessed: 2019-01-04. [Online]. Available: https://spacy.io/

[25] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[26] A. K. McCallum, "Mallet: A machine learning for language toolkit," 2002, http://mallet.cs.umass.edu.

[27] T. Erl, *Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2004.

[28] O. Zimmermann, "Microservices tenets," *Computer Science-Research and Development*, vol. 32, no. 3-4, pp. 301–310, 2017.

[29] M. Szpuszta and S. Vaitinadin. Microsoft azure - fault tolerance pitfalls and resolutions in the cloud. Accessed: 2019-01-04. [Online]. Available: https://msdn.microsoft.com/en-us/magazine/mt422582.aspx

[30] J. Barr, A. Narin, and J. Varia, "Building fault tolerant applications on aws," Amazon, Tech. Rep., 2011.

[31] A. Tarvo, P. F. Sweeney, N. Mitchell, V. Rajan, M. Arnold, and I. Baldini, "CanaryAdvisor: a statistical-based tool for canary testing (demo)," in *Proceedings of the 2015 International Symposium on Software Testing and Analysis - ISSTA 2015*. New York, New York, USA: ACM Press, 2015, pp. 418–422.

[32] R. Heinrich, A. van Hoorn, H. Knoche, F. Li, L. E. Lwakatare, C. Pahl, S. Schulte, and J. Wettinger, "Performance Engineering for Microservices," in *Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering Companion - ICPE '17 Companion*. New York, New York, USA: ACM Press, 2017, pp. 223–226.

[33] D. Sato. Canaryrelease. Accessed: 2019-01-04. [Online]. Available: https://martinfowler.com/bliki/CanaryRelease.html

[34] M. Fowler. Bluegreendeployment. Accessed: 2019-01-04. [Online]. Available: https://martinfowler.com/bliki/BlueGreenDeployment.html

[35] Canalys. Cloud marketshare q4 2018 and full year 2018. Accessed: 2019-01-04. [Online]. Available: https://www.canalys.com/newsroom/cloud-market-share-q4-2018-and-full-year-2018