# REFERENCES


[1] B. Shafabakhsh, "Research on Interprocess Communication in Microservices Architecture Master thesis performed in collaboration with IBM," *Degree Proj. Comput. Sci. Eng.*, 2020.

[2] P. Johansson, "Effcient Communication With Microservices," *Umea Univ.*, no. 1108, p. 45, 2017, [Online]. Available: http://www8.cs.umu.se/education/examina/Rapporter/PetterJohansson2017.pdf

[3] G. K S and P. P. T, "A Better Solution Towards Microservices Communication In Web Application: A Survey," *Int. J. Innov. Res. Comput. Sci. Technol.*, vol. 7, no. 3, pp. 71–74, 2019, doi: 10.21276/ijircst.2019.7.3.7.

[4] A. Smid, R. Wang, and T. Cerny, "Case Study on data communication in microservice architecture," *Proc. 2019 Res. Adapt. Converg. Syst. RACS 2019*, no. June 2020, pp. 261–267, 2019, doi: 10.1145/3338840.3355659.

[5] J. Rudy, "Runtime software adaptation: approaches and a programming tool," *J. Theor. Appl. Comput. Sci.*, vol. 6, no. 1, pp. 75–89, 2012, [Online]. Available: http://www.jtacs.org/archive/2012/1/7

[6] R. M. Jösch, "Managing Microservices with a Service Mesh An implementation of a service mesh with Kubernetes and Istio," *Degree Proj. Comput. Sci. Eng.*, 2020, [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-280407

[7] S. Busanelli, S. Cirani, L. Melegari, M. Picone, M. Rosa, and L. Veltri, "A sidecar object for the optimized communication between edge and cloud in internet of things applications," *Futur. Internet*, vol. 11, no. 7, 2019, doi: 10.3390/fi11070145.

[8] M. Söylemez, B. Tekinerdogan, and A. K. Tarhan, "Feature-Driven Characterization of Microservice Architectures: A Survey of the State of the Practice," *Appl. Sci.*, vol. 12, no. 9, 2022, doi: 10.3390/app12094424.

[9] P. Jamshidi, C. Pahl, N. C. Mendonca, J. Lewis, and S. Tilkov, "Microservices: The journey so far and challenges ahead," *IEEE Softw.*, vol. 35, no. 3, pp. 24–35, 2018, doi: 10.1109/MS.2018.2141039.

[10] P. Di Francesco, I. Malavolta, and P. Lago, "Research on Architecting Microservices: Trends, Focus, and Potential for Industrial Adoption," *Proc. - 2017 IEEE Int. Conf. Softw. Archit. ICSA 2017*, pp. 21–30, 2017, doi: 10.1109/ICSA.2017.24.

[11] M. Baboi, A. Iftene, and D. Gîfu, "Dynamic microservices to create scalable and fault tolerance architecture," *Procedia Comput. Sci.*, vol. 159, pp. 1035–1044, 2019, doi: 10.1016/j.procs.2019.09.271.

[12] P. Jamshidi, M. Ghafari, A. Ahmad, and C. Pahl, "A framework for classifying and comparing architecture-centric software evolution research," *Proc. Eur. Conf. Softw. Maint. Reengineering, CSMR*, pp. 305–314, 2013, doi: 10.1109/CSMR.2013.39.

[13] Y. Zhou, "A runtime architecture-based approach for the dynamic evolution of distributed component-based systems," *Proc. - Int. Conf. Softw. Eng.*, pp. 979–982, 2008, doi: 10.1145/1370175.1370217.

[14] J. W. Liu and X. J. Mao, "Towards Dynamic Evolution of Runtime Variability Based on Computational Reflection," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 28, no. 3, pp. 259–285, 2018, doi: 10.1142/S0218194018500092.

[15] G. Huang, H. Mei, and F. Yang, "Runtime software architecture based on reflective middleware," *Sci. China, Ser. F Inf. Sci.*, vol. 47, no. 5, pp. 555–576, 2004, doi: 10.1360/03yf0192.

[16] C. Parra, X. Blanc, A. Cleve, and L. Duchien, "Unifying design and runtime software adaptation using aspect models," *Sci. Comput. Program.*, vol. 76, no. 12, pp. 1247–1260, 2011, doi: 10.1016/j.scico.2010.12.005.

[17] R. Capilla, A. Valdezate, and F. J. Díaz, "A runtime variability mechanism based on supertypes," *Proc. - IEEE 1st Int. Work. Found. Appl. Self-Systems, FAS-W 2016*, pp. 6–11, 2016, doi: 10.1109/FAS-W.2016.16.

[18] J. Gustavsson, "A Classification of Unanticipated Runtime Software Changes in Java," *IEEE Int. Conf. Softw. Maintenance, ICSM*, pp. 4–12, 2003, doi: 10.1109/ICSM.2003.1235401.

[19] T. Würthinger, C. Wimmer, and L. Stadler, "Dynamic code evolution for Java," *Proc. 8th Int. Conf. Princ. Pract. Program. Java, PPPJ 2010*, no. September, pp. 10–19, 2010, doi: 10.1145/1852761.1852764.

[20] Y. Wang, D. Conan, S. Chabridon, K. Bojnourdi, and J. Ma, "Runtime models and evolution graphs for the version management of microservice architectures," *Proc. - Asia-Pacific Softw. Eng. Conf. APSEC*, vol. 2021–Decem, pp. 536–541, 2021, doi: 10.1109/APSEC53868.2021.00064.

[21] D. A. Hahn, D. Davidson, and A. G. Bardas, "Security Issues and Challenges in Service Meshes -- An Extended Study," 2020, [Online]. Available: http://arxiv.org/abs/2010.11079

[22] A. O. Duque, C. Klein, J. Feng, X. Cai, B. Skubic, and E. Elmroth, "A Qualitative Evaluation of Service Mesh-based Traffic Management for Mobile Edge Cloud," *Proc. - 22nd IEEE/ACM Int. Symp. Clust. Cloud Internet Comput. CCGrid 2022*, no. Cc, pp. 210–219, 2022, doi: 10.1109/CCGrid54584.2022.00030.

[23] Zhang, L., Pang, K., Xu, J., & Niu, B. (2023). High performance microservice communication technology based on modified remote procedure call. Scientific Reports, 13(1). https://doi.org/10.1038/s41598-023-39355-4

[24] Karabey Aksakalli, I., Çelik, T., Can, A. B., & Tekinerdoğan, B. (2021). Deployment and communication patterns in microservice architectures: A systematic literature review. Journal of Systems and Software, 180. https://doi.org/10.1016/j.jss.2021.111014

[25] Saleh Sedghpour, M. R., Klein, C., & Tordsson, J. (2022). An Empirical Study of Service Mesh Traffic Management Policies for Microservices. ICPE 2022 - Proceedings of the 2022 ACM/SPEC International Conference on Performance Engineering, 17–27. https://doi.org/10.1145/3489525.3511686

[26] Saxena, D., Zhang, W., Tummala, M., Goel, S., & Akella, A. (2023, June 19). Invited Paper: Towards Efficient Microservice Communication. Proceedings of the 5th Workshop on Advanced Tools, Programming Languages, and PLatforms for Implementing and Evaluating Algorithms for Distributed Systems, ApPLIED 2023. https://doi.org/10.1145/3584684.3597267

[27] Exploring_Analyzing_and_Tuning_Service_Mesh_Perfor. (n.d.).

[28] Sedghpour, M. R. S., & Townend, P. (2022). Service Mesh and eBPF-Powered Microservices: A Survey and Future Directions. Proceedings - 16th IEEE International Conference on Service-Oriented System Engineering, SOSE 2022, 176–184. https://doi.org/10.1109/SOSE55356.2022.00027

[29] Fourtounis, G., Kastrinis, G., & Smaragdakis, Y. (2018). Static analysis of Java dynamic proxies. ISSTA 2018 - Proceedings of the 27th ACM SIGSOFT International Symposium on Software Testing and Analysis, 209–220. https://doi.org/10.1145/3213846.3213864

[30] Liang, S., & Bracha, G. (n.d.). Dynamic Class Loading in the JavaTM Virtual Machine.

[31] A Comparative Analysis of Static and Dynamic Code Analysis A Comparative Analysis of Static and Dynamic Code Analysis Techniques Techniques. (n.d.). https://doi.org/10.36227/techrxiv.22810664.v1

[32] Leines-Vite, L., Pérez-Arriaga, J. C., & Limón, X. (2021). Confidentiality and Integrity Mechanisms for Microservices Communication. 01–16. https://doi.org/10.5121/csit.2021.111701

[33] Weerasinghe, S., & Perera, I. (2023). Optimized Strategy for Inter-Service Communication in Microservices. In IJACSA) International Journal of Advanced Computer Science and Applications (Vol. 14, Issue 2). www.ijacsa.thesai.org

[34] Michael Ayas, H., Leitner, P., & Hebig, R. (2023). An empirical study of the systemic and technical migration towards microservices. Empirical Software Engineering, 28(4). https://doi.org/10.1007/s10664-023-10308-9

[35] Ntentos, E., Zdun, U., Plakidas, K., Schall, D., Li, F., & Meixner, S. (n.d.). Supporting Architectural Decision Making on Data Management in Microservice Architectures.

[36] Buono, V., Petrovic Author Vincenzo Buono Petar Petrovic, P., & Fredrik Stridh Examiner, S. (n.d.). Enhance Inter-service Communication in Supersonic K-Native REST-based Java Microservice Architectures Title Enhance Inter-service Communication in Supersonic K-Native REST-based Java Microservice Architectures.
[37] Ponomarev, E. S., & Arsentieva, K. Ev. (2023). Evolution of dynamic architecture. Urban Construction and Architecture, 13(3), 120–125. https://doi.org/10.17673/vestnik.2023.03.15
[38] Perry, D. E. (2010). Statically Defined Dynamic Architecture Evolution. https://www.researchgate.net/publication/221474621
[39] Jain, & Raj. (n.d.). Service Mesh: Architectures, Applications, and Implementations. http://www.cse.wustl.edu/~jain/cse574-22/ftp/svc_mesh/index.html1
[40] Gomes, F., Rego, P., Trinta, F., de Souza, J., A Gomes, F. A., L Rego, P. A., M Trinta, F. A., & de Souza, J. N. (2023). Comparative Analysis of Service Mesh Platforms in Microservices-Based Benchmark Applications. 10. https://doi.org/10.48545/advance2023-shortpapers
[41] baeldung, W. by: (2024) Class loaders in Java, Baeldung. Available at: https://www.baeldung.com/java-classloaders.
[42] Nero, J.C.B.R. del and Nero, R. del (2023) All about java class loaders, InfoWorld. Available at: https://www.infoworld.com/article/3700054/all-about-java-class-loaders.html.
[43] Jenkov, J. Java reflection - dynamic class loading and reloading, Jenkov.com Tech & Media Labs - Resources for Developers, IT Architects and Technopreneurs. Available at: https://jenkov.com/tutorials/java-reflection/dynamic-class-loading-reloading.html.
[44] Janssen, T. (2024) Communication between microservices, Stackify. Available at: https://stackify.com/communication-microservices-avoid-common-problems/.
[45] Analyst, M.A.R.I. et al. (2021) Top 10 challenges of using microservices for managing distributed systems - spiceworks, Spiceworks Inc. Available at: https://www.spiceworks.com/tech/data-management/articles/top-10-challenges-of-using-microservices-for-managing-distributed-systems/.
[46] Wickramasinghe, T. (2024) 10 challenges in implementing microservices, Medium. Available at: https://blog.bitsrc.io/microservice-challenges-146badd013e3.
[47] Grover, B. (2023) Unlocking the power of effective microservices communication, Camunda. Available at: https://camunda.com/blog/2023/10/unlocking-the-power-of-effective-microservices-communication/.
[48] Documentation Istio. Available at: https://istio.io/latest/docs/.
[49] What is service mesh and, especially, what is Istio? Is It Observable. Available at: https://isitobservable.io/observability/service-mesh/what-is-servicemesh-and-specially-what-is-istio.

[50] GeeksforGeeks (2023) REST API architectural constraints, GeeksforGeeks. Available at: https://www.geeksforgeeks.org/rest-api-architectural-constraints/.
[51] Apache Kafka Architecture deep dive: Introductory concepts Confluent. Available at: https://developer.confluent.io/courses/architecture/get-started/.
[52] baeldung, W. by: (2024b) Quick intro to spring cloud configuration, Baeldung. Available at: https://www.baeldung.com/spring-cloud-configuration.
[53] Kubernetes (2019). Production-Grade Container Orchestration. [online] Kubernetes.io. Available at: https://kubernetes.io/.

[54] Istio. (n.d.). Istio. [online] Available at: https://istio.io/.