Available online at: https://ijact.in

Date of Submission	09/10/2020
Date of Acceptance	11/03/2021
Date of Publication	29/03/2021
Page numbers	3960-3968 (9 Pages)

ISSN:2320-0790

An International Journal of Advanced Computer Technology

SEMANTIC REVIEW ON SOFTWARE ARCHITECTURES FOR WEB-BASED APPLICATIONS

[1]Ravi Kiran Mallidi, [2]Manmohan Sharma, [3]Sreenivas Rao Vangala

[1]School of Computer Applications Lovely Professional University Punjab, India
E-Mail: ravikiran.mallidi@gmail.com,
[2]School of Computer Applications Lovely Professional University Punjab, India
E-Mail: manmohan.sharma71@gmail.com,
[3]Solution Architect, Hyderabad, Telangana
E-Mail: sreenivangala@gmail.com

Abstract: Different Software Architecture evolves for the last two to three decades. Organizations depend on Architects to provide the Solutions given business requirements. The Architect job is to provide the Solutions to suit the Business needs and follows the industry standards. Architects have to understand the requirement clearly and provide the Solution. Some of the Solutions that exist in the IT industry for the last two to three decades are N-Tier Architecture, Single Page Architecture, Microservices, and Serverless Architecture. Based on the requirement, any one of the Architecture would propose taking considerations of functional and non-functional requirements. Last three years, most of the projects executed in Single Page along with Microservices Architectures used. This paper elaborates on the different Architectures and their advantages and disadvantages. Prepared questioners and send them to fifty Architects, responded of twenty-five (50% response). Survey collected the information on how the Organizations adopting different Architectures in their project needs. Participants are working from different geographic locations, Organizations, and domains. A survey result clearly shows that 50% growth towards the Microservices based architectures for their significant advantage and benefit.

Keywords: Architecture, N-Tier, Single Page Architecture, Microservices, Serverless.

I. INTRODUCTION

Blueprint of the systems is core for deriving the System Architectures. Business strategy, quality, human dynamics, design, and IT environment are part of Software architecture and design. Architectures describes the major components of the systems to communicate, coordinate, and how it works. Architecture defines all technical components that provide the Solutions to particular requirements provided by the business. The Architecture helps to define a solution to meet functional and non-functional requirements like performance, scalability availability, and security for a given requirement. The Solution impacts the overall success of the product/project development. Software Architecture performs many activities typically discuss requirements with stakeholders, architecture and evaluate a design, communicates, and documentation. The code Architecture design consists of Architecture Analysis, Design, Evolution, and Maintaining existing architecture. Software Architect is the person who provides the entire Solution to the team to develop the design of the system. The Software Architect is expertise in Domain, System Design, Technology, and Software Development Life Cycle (SDLC) process. Do's and Don'ts while doing Architecture are described below

Do's:

- Understand the Requirements well

- Understand and can cater to all constraints along with the functional and non-functional requirements
- Always understand and remember the larger picture while architecting for the Solution
- Remember that architecting is an iterative process
- Architecting is a group activity. Do as much as whiteboarding at the early stages of solution definition with experienced people
- Always compare the solutions to industry standards wherever applicable.
- Be transparent and honest with the customer. Many appreciate that, rather than having last-minute surprises.

Don'ts:

- Be afraid to ask questions until get questions answered
- Re-invent the wheel.

Different Architectures are evolved from the last 20-25 years to develop Web Based Applications. Web-Based application Architectures are evolving from N-Tier to Microservices. For each architecture, there may be advantages and disadvantages. Nowadays, all applications are building using Microservices and migrating legacy applications to Microservices due to the significant advantages of the Microservice Architecture. Microservices architectures are an efficient way of building applications in various Information Technologies Domains.

Along with the various benefits in microservices architectures, there are having still challenges to generate Platform bases Microservices architectures. In this paper, various Architectures are elaborated, along with advantages and disadvantages. The reference architecture has demonstrated and how the different architectures help in different domains.

This paper organized as follows: Section II we describe Literature Review, Section III we describe Different Architectures, Section IV describes Research Methodology, Section V describes Results, and Section VI Concludes the paper.

II. LITERATURE REVIEW

This section describes the literature review of the work done in this area. The research carried out [1] is mostly to do with microservices performance in the context of SOA, API Gateway, and containers. Microservices are still immature to identify the success factors. The work carried out by authors [2,3] is related to performance modeling and refactoring. Notations, Methodologies and tools introduced to refactoring the architectures to achieve performance [2]. Developed performance model and prediction tools for microservices experimental and simulation results [3]. The authors [4,5,6] try to bring out the evolution of microservices architecture in the Information Technology(IT) industry and research context. Microservices impact is high for software quality and maintainability, whereas issues for security feature [4]. Microservice architectures fall under 5 significant categories includes Quality, Architecture Quality, Economic valuation, Knowledge Management, and Modelling techniques [5]. A survey conducted for the adoption of microservices concluded that existing tools and techniques are sufficient to support microservices extraction [6]. In [7,11,13], importance has been given to microservices with Cloud computing in the context of Platform As A Service (PAAS). Shown experimentally significant saving to use Cloud infrastructure with a small impact in Quality [7]. Proposed Knowledge-driven Microservices architecture to enhance plug-and-play components model without much programming efforts [11]. Existing robust CI/CD pipeline used is not suitable for microservices applications, thus adjusted the pipeline to accommodate the new architecture [13]. The research papers [8,14,37] is more to do with microservices, agile and DevOps framework. Microservices architecture challenges associated with an increase in services, communication among services, and diversified technologies addressed using service boundary refactoring and alternative design styles [14]. Developed project to identify the architectural smell detection for refactoring [37]. The concept of General Data Protection (GPDR) has articulated in [10] and data-driven in [39]. REST, RESTFUL API, and distribution microservices architecture has covered in [18] and [25], respectively. Intentions, Strategies, and Challenges for microservices migration has covered in [15]. The theory of auto-scaling and auto performance of microservice architectures carried out in [16,20]. Proposed a tool for continuously monitoring the application and provides statistics about system performance and degradation issues [20]. The microservices extraction procedure from a given monolithic architecture has covered in [12,21]. Proposed a migration strategy from monolithic to microservices composed on five phases called, Functional Analysis, Business function identification, Business function analysis, Business function assignment, and microservice creation [12]. Developed software clustering algorithm SArF for migrating monolithic to microservices [21]. The reference architecture and enterprise architecture styles have covered in [22,23]. Analysis based microservice architectures built-in [6,32,46]. Patterns based microservice architectures have built-in [29,44]. Identified the different patterns and the Advantages and Disadvantages are classified for each pattern [44]. Presented different patterns used in Open Sources microservices projects [29]. A comparison of Layered architecture has presented [39]. Multi-tenant SAAS based microservices architecture articulated [40]. The applications web services, mobility, IoT, and microservices can be referred to [41]. Different Architecture Runtime verification microservice architectures have covered in [42,43]. Developed LTL-based constraint language to monitor architectural elements at runtime [43]. Microservices implementations for Blockchain have covered in [47]. Microservices and their security aspects are in [48]. Interoperability and communications of containers of microservices are in [53]. Online mobile application microservices architecture covered in [55]. Tradeoff and sensitivity analysis for microservices addressed in [56].

III. DIFFERENT ARCHITECTURES

A. N-Tier Architecture

N-Tier Architecture or multitier/multilayered architecture is a client-server architecture in which the presentation, application, and database functions separated. Web-based development N-tier architecture widely used for more than a decade. Architecture majorly separated into three layers called Client Layer, Business Layer, and Persistence Layer. The user interacts with the application using a Browser / Mobile Browser called the Client Layer. Server Layer (Application Layer) sits between the Client and Storage layers. Server Layer divided into Web, Business, and Persistence sub-layers. The entry point of the application is Web Layer and responsible for user interactions on the server-side, Business Layer is where business logic implemented as a carefully composed and well-defined API. Persistence Layer is accountable for abstracting the date to Storage Layer's via Object Relational Mapping (ORM) tools like Hibernate. Storage Layer retains relevant data into the relational database system like MySQL, Oracle, or SQL Server.

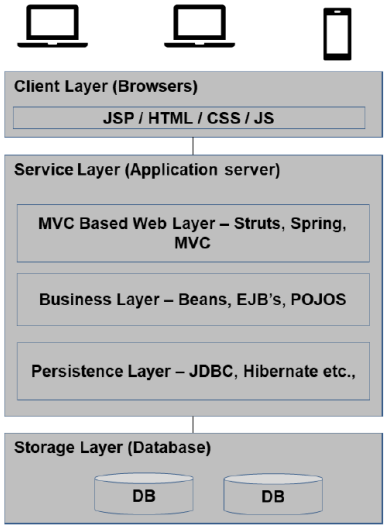

Figure 1: N-Tier Architecture

B. Single Page Architecture (SPA)

Single-Page Applications (SPAs) load a single HTML web application dynamically, and the user interacts with the Server-side Component/Application using Services. SPAs use JavaScript frameworks and HTML5 to create UI, and responsive Web Application without constant page reloads. Much of the work happens on the client-side, in JavaScript. All the front-end pages connected to REST / SOAP services. JavaScript frameworks like Angular, Reactjs used for developing single-page applications The REST services interact with the Business layer for business operations. The Storage Layer retains relevant data into the relational database system like MySQL, Oracle, or SQL Server.

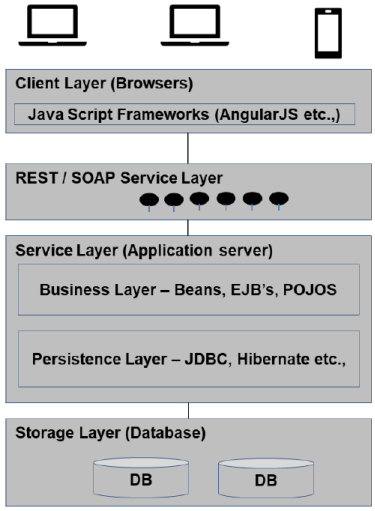

Figure 2: Single Page Architecture

C. Microservice Architecture

Microservice is designed the application as a small subset of the functionalities to full fill the partial requirement. Leverage the Restful API's to connect to third-party hosted services for cross-cutting concerns. They are creating a more significant number of microservices increases the complexity in distributed management systems. There are frameworks like Spring can help to implement microservices quickly. Microservice Architecture allows horizontal scaling and functional change independently without impacting other components.

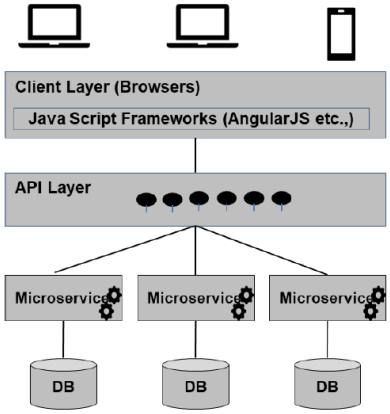

Figure 3: Microservice Architecture

D. Serverless Archietcure

Serverless architecture is favored while the applications are deploying in Cloud environments. Cloud computing execution is a model called as Serverless technology. The cloud provider provides the server inbuilt with the cloud platform. Developers simply write the code and deploy the piece of code to the cloud platform for executions. The Functions can develop by the developer and deploys in the cloud computing services. A Serverless application can be written in Lamda functions to fulfill the requirement.

Serverless architectures if way of executing applications and services without managing infrastructure.

E. Raise of Architecture

Software Architectures raise from Client-Server to Microservices. From 2018 onwards, business is transforming the Architectures to Microservices. All new applications are developing using Microservices.

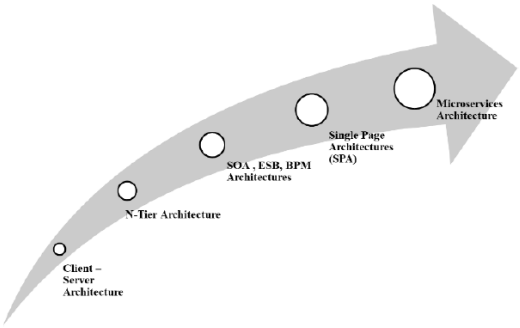

Figure 4: Raise of Architecture

IV. METHODOLOGY

Our methodology conducted a sampling survey on various Architectures defined above for Web-based design. Formulated standard survey questions and send them to different participants for their feedback. Consolidated the survey feedback resulted in the results section and concluded.

A. Participants Selections

Selection of participants according to the roles and position of an experience of 14+ years. Experience in Architecture and Design in software development. Considering Architects, Project Manager with technical skills, and Scrum Master with technical skills. Specific demographic questions put to the practitioners like years of experience, nature of projects, and role.

B. Research Questions

There are 15 questions formulated for this survey in the area of Architecture and Agile methodology. The categories as below:
Category-1: General questions related to Experience, Domain, and Role of the Participant.
Category-2: Solutions developed in Microservices, Legacy Architectures, and Cloud deployments. Increase in Microservices Architecture usage
Category-3: Projects implementations in Agile methodology
Each section has multiple questions, and the question has multiple-choice options. Each participant spends 10-15 minutes to complete the survey process.

C. Survey Execution/Process

The survey is created in Google Forms and sends the link to 50 Architects, Project managers, and Scrum masters working in various Organizations with different sizes of the Organization. The survey conducted between the period of July and Aug 2020. 25 participants responded with their feedback on the survey. The survey designed using multiple-choice questions, based on four levels scale, some questions allow other choice options for unexpected answers.

V. RESULTS

The survey results populated in this section as below based on the category described above.

A. Participants Background and Geographical Disposing

Participants average experience above 18+ years and working as Architects, Project Managers, and Scrum Managers. The participants are from India, the USA, and Canada working with different Organizations. Out of 25 participants, 10 Architects, 2 Director / Chief Technical Officers, 4 Project Managers, and 3 Scrum Masters. Software Domain includes Banking Financial Services, Health Care, Telecom, and Government projects. The geographical disposing of participants concentrated in India (18 participants), followed by the USA (05 participants) and Canada (02 participants). 90% of participants working from Larger Organizations size of > 2000 employees, 05% of participants working from Mid and Small Organization sizes 200-2000 employees, and < 200 employees accordingly.

Research Question 1: Years of experience in the IT industry.
Participants are having different years of experience ranging from 12 to 30 years. The respondent's average years of experience is about 18+ years.

Figure 5: Participants Experience Level

Research Question 2: Participants working Domain.

Participants working domains include Banking Financial and Insurance (BFSI), Telecom and Retail, Health Care. Significant respondents worked on BFSI, followed by

Telecom and Healthcare. Some of them worked on multiple Domains for the last 2 years.

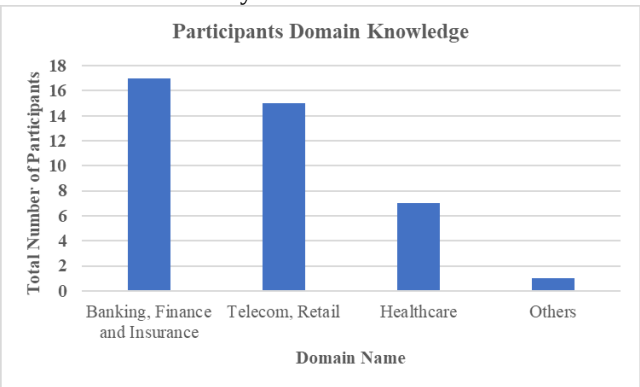

Figure 6: Participants Domain Knowledge

Research Question 3: Participants Current Role.

The majority of the participant's role is Architect, Project Manager, and Scrum Masters. The questions related to Architecture are more relevant to the group targeted.

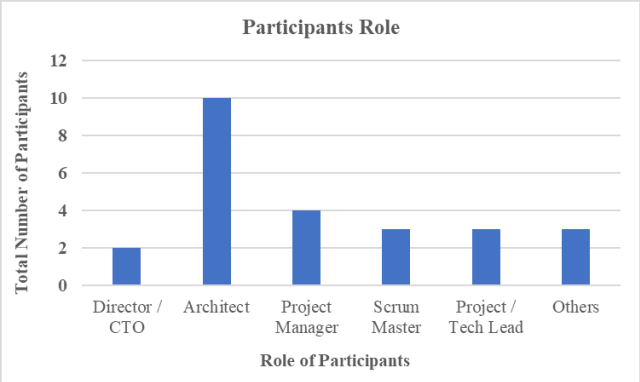

Figure 7: Participants Role

B. Architectures Adoption/Usage

The participants provided a minimum of 05 solutions from 2019-20. 52% participants provide 05-10 solutions and 24% provides < 05 solutions and 24% provided > 05 solutions. Questions about how many of them used Microservices in their total solutions. 70% participants provided < 05-10 microservice solutions, 20% provided 05-10, and 10% provided more than 10 solutions.

Research Question 1: How many total solutions provided in 2019-20, out of which how many of them are in Microservices architecture.

Below responded graph clearly says that < 05 solutions provided responders provided more Microservices architects. Seris1 represents total solutions provided in the year 2019-20. Series2 represent % of microservices architectures provided in total solutions.

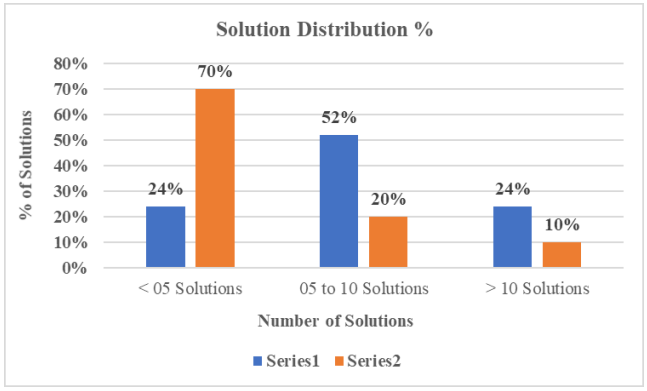

Figure 8: Solution Distribution

Research Question 2: % of growth in Microservices architectures for the past 2-3 years.

Out of 25 participants, 09 observed 20-50% growth, 07 observed > 80% growth, 05 observed 50-80% growth, and 04 observed 20% growth in Microservices architectures in their Organizations.

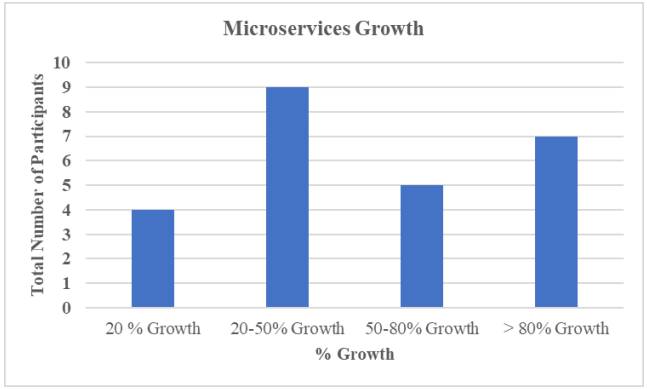

Figure 9: Microservice Growth

Research Question 3: Technologies prefeed to implement Microservices.

Participants use the below technologies for developing Microservices Architecture – Java Spring Boot, RedHat OpenShift with Kubernetes, Micronaut, Docker, AWS EKS, Aquasec for security, and Public Cloud Foundry (PCF). Spring Boot is the most opted one to develop microservices for Java bases projects.

C. Agile and DevOps Adoption

Survey results clearly show that Organizations are adopting Agile as 48%. Hybrid Agile as 44%, and another uses Waterfall and Mixed agile for 08%. 50-80% Microservices architecture adopted Agile/Hybrid agile for project execution and delivery, and 72% projects sprint cycle is 2-4 weeks.

Research Question 1: Software Life Cycle Methodologies used for Microservices Projects in Organization.

Responders uses Agile is the primary methodology for SDLC followed by Hybrid Agile, and Safe Agile (for scrum of scrums)

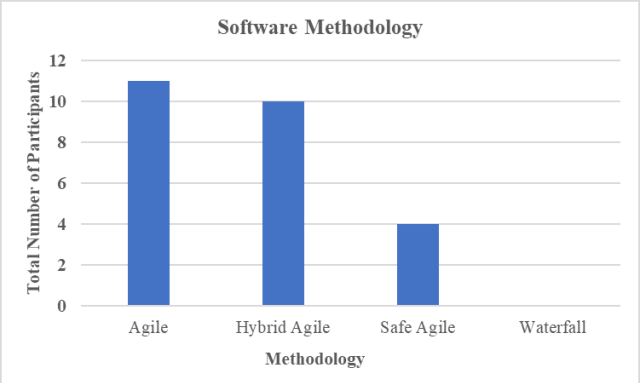

Figure 10: Software Methodology

Research Question 2: Agile / Hybris Agile release cycle frequency.

Most of them responded 2-4 weeks as a sprint release cycle followed by Monthly and Quarterly.

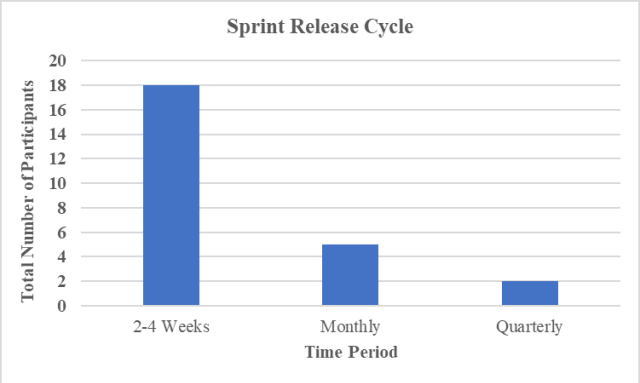

Figure 11: Sprint Release Cycle

Research Question 2: Estimation techniques used in Scrum projects.

Most of the respondents use Planning poker, and the T-shirt used for Scrum estimations.

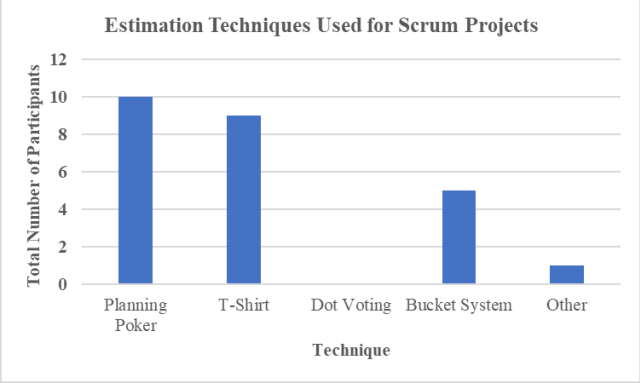

Figure 12: Estimation Techniques Used For Scrum Projects

As one of the questions asked, how many are using Open source and commercial frameworks like IBM. 20% are choosing microservice frameworks like IBM ClodPak, and others are mostly using Open source technologies.

VI. Conclusion

Presented on various Architectures for Web-based applications and discussed on N-Tier, Single Page, Microservices, and Serverless Architectures. Survey conducted with 25 experts majorly from four Organizations. The survey focused on three aspects: different Architectures proposed, Cloud usage, and Agile usage. The results revealed that practitioners tend to use Microservices as one of the primary architecture. More than 50% growth seen in Microservices architectures from the last 2-3 years due to microservices architecture allows application flexibility and performance. Organizations have to address the Microservices adoptability challenges very fast and quickly. Business users have to train on microservices adaptability for better decision making. Continues Integration and Deployment (CI/CD) is the primary factor for Microservices development and deployment. According to the survey, all the Microservices projects are using CI/CD. Spring Boot, OpenShift, and Kubernetes are the most preferred programming languages used. Agile and Hybrid agile used for the SDLC process with 2-4 weeks of sprint cycle. As per requirements, monolithic architectures also play a crucial role in Architecture design. Most of the Microservice projects are deploying in the Cloud platform. The banking and financial industry is moving towards Microservices adoption without moving their projects to the Cloud platform due to Government norms. For the next 3-5 years, the growth has seen in Microservices adoption on a broader range of projects. There is scope to study on new Architecture models like Functional Microservices Functional programming as an analogy of Microservices. The architecture is decomposing by business and domain-driven.

VII. References

[1] Akbulut, A., & Perros, H. G. (2019). Performance Analysis of Microservices Design Patterns. IEEE Internet Computing.

[2] Arcelli, D., Cortellessa, V., & Di Pompeo, D. (2018, April). Performance-Driven Software Architecture Refactoring. In 2018 IEEE International Conference on Software Architecture Companion (ICSA-C) (pp. 2-3). IEEE.

[3] Bao, L., Wu, C., Bu, X., Ren, N., & Shen, M. (2019). Performance Modeling and Workflow Scheduling of Microservice-Based Applications in Clouds. IEEE Transactions on Parallel and Distributed Systems, 30(9), 2114-2129.

[4] Bogner, J., Fritzsch, J., Wagner, S., & Zimmermann, A. (2019, March). Microservices in industry: insights

into technologies, characteristics, and software quality. In 2019 IEEE International Conference on Software Architecture Companion (ICSA-C) (pp. 187-195). IEEE.

[5] Breivold, H. P., Crnkovic, I., & Larsson, M. (2012). A systematic review of software architecture evolution research. Information and Software Technology, 54(1), 16-40.

[6] Carvalho, L., Garcia, A., Assunção, W. K., de Mello, R., & de Lima, M. J. (2019, May). Analysis of the criteria adopted in industry to extract microservices. In 2019 IEEE/ACM Joint 7th International Workshop on Conducting Empirical Studies in Industry (CESI) and 6th International Workshop on Software Engineering Research and Industrial Practice (SER&IP) (pp. 22-29). IEEE.

[7] Chapman, C., Emmerich, W., Márquez, F. G., Clayman, S., & Galis, A. (2012). Software architecture definition for on-demand cloud provisioning. Cluster Computing, 15(2), 79-100.

[8] Chen, L. (2018, April). Microservices: Architecting for continuous delivery and devops. In 2018 IEEE International Conference on Software Architecture (ICSA) (pp. 39-397). IEEE.

[9] Cojocaru, M., Uta, A., & Oprescu, A. MicroValid: A Validation Framework for Automatically Decomposed Microservices.

[10] Colesky, M., Demetzou, K., Fritsch, L., & Herold, S. (2019, March). Helping Software Architects Familiarize with the General Data Protection Regulation. In 2019 IEEE International Conference on Software Architecture Companion (ICSA-C) (pp. 226-229). IEEE.

[11] Dai, W., Wang, P., Sun, W., Wu, X., Zhang, H., Vyatkin, V., & Yang, G. (2019). Semantic Integration of Plug-and-Play Software Components for Industrial Edges Based on Microservices. IEEE Access, 7, 125882-125892.

[12] De Lauretis, L. (2019, October). From Monolithic Architecture to Microservices Architecture. In 2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW) (pp. 93-96). IEEE.

[13] Debroy, V., & Miller, S. (2019). Overcoming Challenges with Continuous Integration and Deployment Pipelines When Moving From Monolithic Apps to Microservices: An experience report from a small company. IEEE Software.

[14] Diaz-Pace, J. A., & Bianchi, A. J. (2019, March). High-Level Design Stories in Architecture-Centric Agile Development. In 2019 IEEE International Conference on Software Architecture Companion (ICSA-C) (pp. 137-144). IEEE.

[15] Fritzsch, J., Bogner, J., Wagner, S., & Zimmermann, A. (2019). Microservices Migration in Industry: Intentions, Strategies, and Challenges. In 2019 IEEE International Conference on Software Maintenance and Evolution (ICSME) (pp. 481-490). IEEE.

[16] Gias, A. U., Casale, G., & Woodside, M. (2019, July). ATOM: Model-Driven Autoscaling for Microservices. In 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS) (pp. 1994-2004). IEEE.

[17] Gidey, H. K., Marmsoler, D., & Eckhardt, J. (2017, April). Grounded Architectures: Using Grounded Theory for the design of software architectures. In 2017 IEEE international conference on software architecture workshops (ICSAW) (pp. 141-148). IEEE.

[18] Haupt, F., Leymann, F., Scherer, A., & Vukojevic-Haupt, K. (2017, April). A framework for the structural analysis of REST APIs. In 2017 IEEE International Conference on Software Architecture (ICSA) (pp. 55-58). IEEE.

[19] Iftikhar, M. U., & Weyns, D. (2017, April). Activforms: A runtime environment for architecture-based adaptation with guarantees. In 2017 IEEE International Conference on Software Architecture Workshops (ICSAW) (pp. 278-281). IEEE.

[20] Janes, A., & Russo, B. (2019, October). Automatic performance monitoring and regression testing during the transition from monolith to microservices. In 2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW) (pp. 163-168). IEEE.

[21] Kamimura, M., Yano, K., Hatano, T., & Matsuo, A. (2018, December). Extracting Candidates of Microservices from Monolithic Application Code. In 2018 25th Asia-Pacific Software Engineering Conference (APSEC) (pp. 571-580). IEEE.

[22] Karkhanis, P., van den Brand, M. G., & Rajkarnikar, S. (2018, April). Defining the C-ITS reference architecture. In 2018 IEEE International Conference on Software Architecture Companion (ICSA-C) (pp. 148-151). IEEE.

[23] Kleehaus, M., & Matthes, F. (2019, October). Challenges in Documenting Microservice-Based IT Landscape: A Survey from an Enterprise Architecture Management Perspective. In 2019 IEEE 23rd International Enterprise Distributed Object Computing Conference (EDOC) (pp. 11-20). IEEE.

[24] Klock, S., Van Der Werf, J. M. E., Guelen, J. P., & Jansen, S. (2017, April). Workload-based clustering of coherent feature sets in microservice architectures. In 2017 IEEE International Conference on Software Architecture (ICSA) (pp. 11-20). IEEE.

[25] Limón, X., Guerra-Hernández, A., Sánchez-García, A. J., & Arriaga, J. C. P. (2018, October). SagaMAS: a software framework for distributed transactions in the microservice architecture. In 2018 6th International Conference in Software Engineering Research and Innovation (CONISOFT) (pp. 50-58). IEEE.

[26] Lotz, J., Vogelsang, A., Benderius, O., & Berger, C. (2019, March). Microservice Architectures for Advanced Driver Assistance Systems: A Case-Study. In 2019 IEEE International Conference on Software Architecture Companion (ICSA-C) (pp. 45-52). IEEE.

[27] Malavolta, I., & Capilla, R. (2017, April). Current research topics and trends in the software architecture community: Icsa 2017 workshops summary. In 2017 IEEE International Conference on Software Architecture Workshops (ICSAW) (pp. 1-4). IEEE.

[28] Manjunath, A., Bhat, M., Shumaiev, K., Biesdorf, A., & Matthes, F. (2018, April). Decision making and cognitive biases in designing software architectures. In 2018 IEEE International Conference on Software Architecture Companion (ICSA-C) (pp. 52-55). IEEE.

[29] Márquez, G., & Astudillo, H. (2018, December). Actual use of architectural patterns in microservices-based open source projects. In 2018 25th Asia-Pacific Software Engineering Conference (APSEC) (pp. 31-40). IEEE.

[30] Masvosvere, D. J. E., & Venter, H. S. (2016). Using a standard approach to the design of next generation e-Supply Chain Digital Forensic Readiness systems. SAIEE Africa Research Journal, 107(2), 104-120.

[31] Mazlami, G., Cito, J., & Leitner, P. (2017, June). Extraction of microservices from monolithic software architectures. In 2017 IEEE International Conference on Web Services (ICWS) (pp. 524-531). IEEE.

[32] Meloca, R. M., Ré, R., & Schwerz, A. L. (2018, October). An Analysis of Frameworks for Microservices. In 2018 XLIV Latin American Computer Conference (CLEI) (pp. 542-551). IEEE.

[33] Mendonça, N. C., Jamshidi, P., Garlan, D., & Pahl, C. (2019). Developing Self-Adaptive Microservice Systems: Challenges and Directions. IEEE Software.

[34] Naab, M., & Rost, D. (2018, April). How to Evaluate Software Architectures: Tutorial on Practical Insights on Architecture Evaluation Projects with Industrial Customers. In 2018 IEEE International Conference on Software Architecture Companion (ICSA-C) (pp. 6-7). IEEE.

[35] Nehme, A., Jesus, V., Mahbub, K., & Abdallah, A. (2019). Securing Microservices. IT Professional, 21(1), 42-49.

[36] Nord, R. L., & Tomayko, J. E. (2006). Software architecture-centric methods and agile development. IEEE software, 23(2), 47-53.

[37] Rachow, P. (2019, March). Refactoring Decision Support for Developers and Architects Based on Architectural Impact. In 2019 IEEE International Conference on Software Architecture Companion (ICSA-C) (pp. 262-266). IEEE.

[38] Salvadori, I. L., Huf, A., & Siqueira, F. (2019, July). Semantic Data-Driven Microservices. In 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC) (Vol. 1, pp. 402-410). IEEE.

[39] Savolainen, J., & Myllarniemi, V. (2009, September). Layered architecture revisited—Comparison of research and practice. In 2009 Joint Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture (pp. 317-320). IEEE.

[40] Song, H., Nguyen, P. H., Chauvel, F., Glattetre, J., & Schjerpen, T. (2019, July). Customizing Multi-Tenant SaaS by Microservices: A Reference Architecture. In 2019 IEEE International Conference on Web Services (ICWS) (pp. 446-448). IEEE.

[41] Song, Z., & Tilevich, E. (2019, July). Equivalence-Enhanced Microservice Workflow Orchestration to Efficiently Increase Reliability. In 2019 IEEE International Conference on Web Services (ICWS) (pp. 426-433). IEEE.

[42] Sotomayor, J. P., Allala, S. C., Alt, P., Phillips, J., King, T. M., & Clarke, P. J. (2019, July). Comparison of Runtime Testing Tools for Microservices. In 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC) (Vol. 2, pp. 356-361). IEEE.

[43] Stockmann, L., Laux, S., & Bodden, E. (2019, March). Architectural Runtime Verification. In 2019 IEEE International Conference on Software Architecture Companion (ICSA-C) (pp. 77-84). IEEE.

[44] Taibi, D., Lenarduzzi, V., & Pahl, C. (2018). Architectural patterns for microservices: a systematic mapping study. SCITEPRESS.

[45] Tang, A., Razavian, M., Paech, B., & Hesse, T. M. (2017, April). Human aspects in software architecture decision making: a literature review. In 2017 IEEE International Conference on Software Architecture (ICSA) (pp. 107-116). IEEE.

[46] Tennage, P., Perera, S., Jayasinghe, M., & Jayasena, S. (2019, August). An Analysis of Holistic Tail Latency Behaviors of Java Microservices. In 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS) (pp. 697-705). IEEE.

[47] Tonelli, R., Lunesu, M. I., Pinna, A., Taibi, D., & Marchesi, M. (2019, February). Implementing a microservices system with blockchain smart contracts. In 2019 IEEE International Workshop on Blockchain Oriented Software Engineering (IWBOSE) (pp. 22-31). IEEE.

[48] Torkura, K. A., Sukmana, M. I., & Kayem, A. V. (2018, December). A cyber risk based moving target defense mechanism for microservice architectures. In 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing,

Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom) (pp. 932-939). IEEE.

[49] Valipour, M. H., AmirZafari, B., Maleki, K. N., & Daneshpour, N. (2009, August). A brief survey of software architecture concepts and service oriented architecture. In 2009 2nd IEEE International Conference on Computer Science and Information Technology (pp. 34-38). IEEE.

[50] Verdecchia, R. (2018, April). Architectural technical debt identification: Moving forward. In 2018 IEEE International Conference on Software Architecture Companion (ICSA-C) (pp. 43-44). IEEE.

[51] Werner, C. M. L. (2019, November). A Survey on Microservices Criticality Attributes on Established Architectures. In 2019 International Conference on Information Systems and Software Technologies (ICI2ST) (pp. 149-155). IEEE.

[52] Wohlrab, R., Eliasson, U., Pelliccione, P., & Heldal, R. (2019, March). Improving the Consistency and Usefulness of Architecture Descriptions: Guidelines for Architects. In 2019 IEEE International Conference on Software Architecture (ICSA) (pp. 151-160). IEEE.

[53] Yuan, E. (2019, May). Architecture interoperability and repeatability with microservices: an industry perspective. In 2019 IEEE/ACM 2nd International Workshop on Establishing the Community-Wide Infrastructure for Architecture-Based Software Engineering (ECASE) (pp. 26-33). IEEE.

[54] Zhang, H., Li, S., Jia, Z., Zhong, C., & Zhang, C. (2019, March). Microservice Architecture in Reality: An Industrial Inquiry. In 2019 IEEE International Conference on Software Architecture (ICSA) (pp. 51-60). IEEE.

[55] Zhao, Y., & Medvidovic, N. (2019, May). A microservice architecture for online mobile app optimization. In 2019 IEEE/ACM 6th International Conference on Mobile Software Engineering and Systems (MOBILESoft) (pp. 45-49). IEEE.

[56] Zhu, L., Aurum, A., Gorton, I., & Jeffery, R. (2005). Tradeoff and sensitivity analysis in software architecture evaluation using analytic hierarchy process. Software Quality Journal, 13(4), 357-375.