Critical Success Factors of Microservices Architecture Implementation in the Information System Project

Mochamad Gani Amri, Teguh Raharjo, Anita Nur Fitriani, Nurman Rasyid Panusunan Hutasuhut
Master of Information Technology-Faculty of Computer Science, University of Indonesia, Jakarta, Indonesia

***Abstract*—Microservice Architecture (MSA) promises enhancements in information systems, including improved performance, scalability, availability, and maintenance. However, challenges during the design, development, and operations phases can hinder successful deployment. This research presents a case study of one of the leading telecommunications companies in Indonesia, which encountered a three-month delay in implementing its microservices architecture (MSA). The study aims to provide actionable insights for the company to enhance its MSA deployment and contribute to academic knowledge by offering a structured approach to evaluating critical success factors (CSFs) in similar contexts. Through a literature review, twenty-one factors were identified and categorized into four groups: (1) Organization, (2) Process, (3) Systems & Tools, and (4) Knowledge, Skills & Behavior. The Analytical Hierarchy Process (AHP) was used to evaluate the priority of each factor based on survey data from project executors and software development practitioners. The findings indicate that the Organization category is the most crucial, with (1) Top Management Support, (2) Clear Vision, and (3) Adequate Resources being the top three CSFs for MSA implementation.**

***Keywords*—Microservice architecture; software architecture; critical success factors; analytical hierarchy process**

I. INTRODUCTION

Microservice Architecture (MSA) has emerged as a prevailing trend in software development in recent years [1]. The advent of microservice architecture gained traction in the early 2010s, serving as a countermeasure to the complexities encountered in monolithic application development, deployment, and scalability [2]. Software development teams and organizations across various sectors have embraced microservice architecture for the construction and management of their applications [3]. MSA's adoption spans multiple industries, from technology to e-commerce and banking, and across different development environments, from cloud computing to on-premise solutions [4].

MSA delineates a monolithic application into a collection of smaller, isolated, and interconnected services [5]. Each of these services bears distinct responsibilities and operates independently from others, thereby enabling development teams to work in isolation and reduce the overall system complexity [6]. MSA's popularity stems from its greater flexibility in development, the capability for independent scaling and updating of components, and the facilitation of system management and security [7].

Despite the improvements and benefits offered by MSA implementation, O'Reilly published research [8] that identified potential challenges (pains) arising at various implementation phases of MSA (design, development, and operations) such as overcoming existing mindset, decomposing functionality and integration to legacy system as the top three challenges. Cross-phase issues in design, development, and operation have been identified, including architecture, management, monitoring, testing, and others [9]. Several researchers have also explored the benefits (gains) and challenges (pains) of MSA deployment [9,10].

As a case study, one of Indonesia's leading telecommunications companies is improving its workforce management system to enhance availability, scalability, and maintainability. One of the initiatives being undertaken was migrating from monolithic to microservices-based architecture. However, during the development process, challenges arose, causing the project delivery to be delayed by three months beyond the initial target. Based on the interview conducted with one of the technical managers, the complexity of the current monolithic system and lack of microservice experience at the organizational level made the duration of important aspects such as development and testing take longer than expected. This case problem aligns with research and practical evidence, which suggests that MSA is not a panacea, as it presents several challenges during its implementation phases [11]. About 9% of respondents experienced complete failure (not successful at all), and 37% reported limited success (some success) in their MSA implementation efforts [8].

Previous studies have researched the challenges and benefits that can be referred to as a candidates for the factors that affect MSA implementation. First [6], offers an insightful historical perspective on the evolution of microservices, highlighting key transitions that have shaped modern software engineering practices. Second [10], provides valuable empirical evidence from industry on the practical challenges of adopting microservices, directly supporting my analysis of deployment differences between public cloud and on-premise systems. Third [11], delves into the current obstacles and future directions for microservices, enriching the discussion of ongoing challenges and emerging trends.

The significant benefits offered by MSA implementation, coupled with the challenges faced during its execution, form the basis of this study. Previous researchers have laid a solid foundation of knowledge on the benefits, problems, and challenges of implementing MSA, yet there has been no structured collection of critical success factors (CSFs) within specific case studies, especially in Indonesia. Hence, this study aims to identify and address the research question:

RQ: What are the critical success factors of microservice architecture implementation in the information systems project?

The author also argues that the success of an initiative or program is intrinsically linked to an organization's internal capabilities. These capabilities are comprised of four components: (1) Organization, (2) Processes, (3) Systems & Tools, and (4) Knowledge, Skills & Behaviors [12]. These components form the basis for categorizing the critical factors identified in this study.

II. Research Methods

This study is organized into three stages, as illustrated in Fig. 1. Step 1 involves the identification of Critical Success Factors (CSFs) through a systematic literature review (SLR) and the categorization of these factors based on capability categories. Step 2 encompasses the development and execution of a survey designed to collect quantitative data, serving as the foundation for the prioritization process. Step 3 entails the evaluation of priority levels utilizing the Analytic Hierarchy Process (AHP), a methodology that facilitates multi-criteria decision-making.

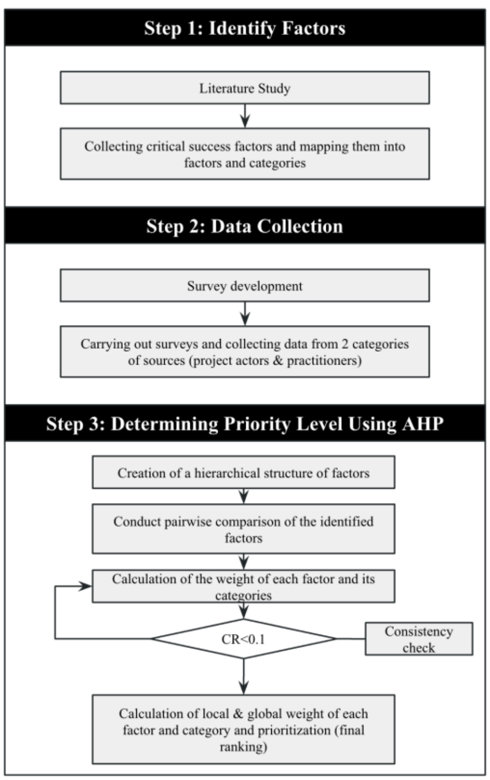

Fig. 1. Research method.

III. Identify Success Factors

A. Literature Study

In the initial stage of this study, the investigators conducted a literature study to identify the Critical Success Factors (CSFs) in the implementation of Microservice Architecture (MSA). Table I illustrates that the literature study commenced with a manual search through the electronic data source (EDS) Google Scholar, resulting in ten relevant studies.

TABLE I. Research Selection

Stages of Literature Study	Electronic Data Source (EDS)			
	Google Scholar (Manual)	Science Direct (SLR)	IEEE Xplore (SLR)	Total
String Execution	-	95	318	413
Study Extraction	-	20	48	68
Study Screening & Selection	10	11	8	29

For further research, two additional EDS, namely ScienceDirect and IEEE Xplore, were selected for systematic literature review (SLR) utilizing a three-step method.

- String Execution: The execution of search strings was performed according to the specifications outlined in Table II. These two data sources were targeted for the SLR and successfully gathered 413 research articles from journals and conferences.
- Study Extraction: The 413 research articles underwent a further filtering process based on titles & keywords to determine their relevance for in-depth reading. This resulted in 68 papers, with the primary aim being to avoid research focused on technical implementations in the form of application solutions that were outside the search for factors.
- Study Screening & Selection: A total of 29 studies, comprising 19 SLR studies (11 from ScienceDirect and eight from IEEE Xplore) plus ten manually identified studies, were advanced for closer examination. This involved quickly reading the abstracts, research findings, discussions, and conclusions to identify whether they contained fsactors relevant to MSA implementation.

TABLE II. SLR Research Specifications

Serarch Key Words	("microservices" OR "microservices architecture") AND ("factors" OR "success" OR "failure" OR "challenges")
Section	Title, Abstract & Author Keywords
Publication Type	Research Articles or Conferences
Publication Year	2019-2024

B. Define the Success Factors

In this section, the researchers elucidate the Critical Success Factors (CSFs) identified from the literature study. A total of 21 critical success factors were discovered for the implementation of Microservice Architecture (MSA), described as follows.

1) Top management support: The support exhibited by top leaders of an organization towards a specific initiative or project encompasses recognition, resource allocation, and active involvement from top management in guiding, encouraging, and supporting the success of the initiative. Researchers concur that the success of information system implementation is inseparable from comprehensive support from top management within an agency or company [33-34].

2) Clear vision: Research [35] adds a clear vision as a critical success factor for information system implementation into its framework. A clear vision includes the goals to be achieved, available resources, timelines, systems to be implemented, vendors, and possible methods needed to achieve these goals.

3) Organizational culture: The implementation of MSA can drive organizational culture change through the division of teams into smaller units, reducing dependencies and communication between teams. This can also lower costs in the management process [10]. A survey conducted [30] mentions that MSA results in better development and deployment with scalability compared to monolithic architecture; however, there are challenges where organizational culture becomes a primary factor in MSA adoption. Organizational culture also significantly influences company performance and effectiveness, employee morale and productivity, and its ability to attract, motivate, and retain talented individuals [36].

4) Adequate resources: Project managers and software architects are the main focus of high development costs in MSA implementation due to developers lacking experience in MSA implementation [22]. The need to design a good MSA architecture to optimize the performance of MSA-based applications requires adequate software architecture resources [29]. In this research context, adequate resources include the technical, financial, and human resources allocated for the implementation process.

5) Project management: Effective project management is a critical need for every information system project initiative [31]. Project management encompasses the processes of planning, organizing, directing, and controlling the resources used to achieve the project objectives within specified time, budget, and scope limits. Requirements Engineering (RE) and non-RE tasks/sub-tasks, such as overseeing the transition of requirements to code, coordination, or project management, are regarded as essential skills in the era of MSA [43]. The synchronization between the teams is also a challenge in developing microservice architecture due to the complex dependencies and communication of services [47].

6) Change management: Managing resistance in the implementation process of a new initiative is an important factor; organizations must have good planning for projects to run effectively [31]. The process of managing change within an organization involves understanding, preparation, execution, and monitoring of changes to achieve the desired outcomes and minimize potential negative impacts.

7) Training and education: The main goal of training is to enhance the skills, knowledge, and competency levels of all users within the organization. Training should be provided and considered as part of the implementation process because it influences collective beliefs about the benefits of the implemented system [31]. Microservices practitioners should have competencies in at least one, ideally multiple, of the core MSA roles (web developer, DevOps engineer, or data engineer). The three main collections of competencies indicate the primary technical skills that practitioners who want to work with microservices should have [45].

8) Vendor selection: The process of selecting vendors that will collaborate with an organization to provide specific products, services, or solutions. This involves the evaluation and selection of vendors that best meet the needs and requirements of the organization. Careful selection of vendors, products, and services is necessary because failure to do so successfully can be very costly, as shown by several reported failure cases [25].

9) Agile methodology: The development of distributed microservice-based systems can adopt the agile approach as a system development method; therefore, communication is crucial for knowledge transfer among teams [24]. The software development process with agile methodology focuses on adaptive team collaboration, responsiveness to changes, and the delivery of high-value products. Organizations found agile development compatible with microservice-based architectures. The common view was that the bounded context of each microservice works well in agile development [47].

10)System modeling: The process of creating models that visually or conceptually represent a system. The goal is to understand, analyze, design, and construct systems in a systematic and structured manner, for example, using UML. Modeling diagrams are used to describe various aspects of MSA-based systems. Results show that Architecture and Functional Flow diagrams are commonly used to depict a high-level view of MSA [28]. A study found that the lack of a formal representation of domain models is a challenge for software modeling and emphasizes even simple and informal diagrams such as UML can be used to represent domain models as long as they help communicate design insights [42].

11)API management and standardization: API management involves managing the API lifecycle, including planning, development, testing, deployment, and maintenance of APIs (API Contract, API Versioning, Open API Standard, etc.). Further research worth conducting is on the pains felt due to exploiting APIs to enable microservice services to communicate [9]. The microservice API evolution process suffers from the loose coupling between services and leads to communication overheads and backward compatibility necessity [46].

12)Service communication: Communication between various microservices within a system (Discovery of services, Replicated Service Instances, Load Balancing, Replicating data, Remote Calls, Relation Between Tables, REST, Event Driven). There are facts that intercommunication between API-based systems also poses real issues [9]. In a microservice-based system, to overcome the challenge of complex communication and dynamicity at runtime in failure detection, distributed tracing data is used [44].

13)Infrastructure automation: When infrastructure automation technology is applied, the deployment of microservices becomes as straightforward as monolithic systems [10]. Activities encompassing infrastructure automation processes include CI/CD, tools (version control, build, test, deploy), and testing tactics (unit, API, integration & contract testing).

14)Monitoring and logging: The application of tools for monitoring, information gathering, and data analysis to ensure optimal performance, detect issues, and support accurate decision-making. A major challenge of implementing microservices is the intrinsic complexity of monitoring applications composed of a large, dynamically evolving, and heterogeneous number of components [9]. There are also challenges to detecting failures in microservice-based systems due to the inherent characteristics of such systems, including complex communications, frequent updates, dynamicity at runtime, and complex log management [44]. Monitoring is of paramount importance to continuously drive development and testing, starting from the feedback collected from the field [48].

15)Cloud computing adoption: Data storage, processing, and resource management over the internet using infrastructure provided by cloud service providers such as IaaS, PaaS, and SaaS. SaaS Cloud-native based on microservices offers customers flexible infrastructure opportunities while leveraging the economies of scale provided by cloud providers and multi-tenancy architecture [27].

16)System architecture design: The expertise in planning the overall structure and components of a system. This involves identifying system needs, selecting appropriate technologies, modeling components, and arranging interactions between those components. In the process, considering microservices-based architecture design identifies parts (partitions) of applications in limited contexts is not easy; microservices yield benefits from implementing this bounded context [9].

17)Database architecture and design: Determining the appropriate database model, designing schema, defining tables and their attributes, and constructing relationships among them. Creating databases that can efficiently store and retrieve data, ensure data integrity, and meet system requirements. The design of a database per service pattern allows for independence among MSA services by equipping each service with its storage (if needed) [9].

18)System security: Knowledge related to practices for protecting computer systems, networks, or software applications from unauthorized access, data breaches & cyber threats. System security aims to ensure the confidentiality, integrity, and availability of system resources and data. Security also poses issues at the design time, especially due to access control (ACL) and the proliferation of endpoints [9] (Table III).

TABLE III. CRITICAL SUCCESS FACTORS BASED ON LITERATURE STUDIES

ID	**Factors**	**Studies**
F1	Top Management Support	L13 [32], L14 [33]
F2	Clear Vision	L15 [34]
F3	Organizational Culture	L9 [10], L11 [30] L16 [35]
F4	Adequate Resources	L2 [22], L10 [29]
F5	Project Management	L12 [31], L24[43], L28[47]
F6	Change Management	L12 [31]
F7	Training and Education	L12 [31], L26[45]
F8	Vendor Selection	L5 [25]
F9	Agile Methodology	L4 [24], L28[47]
F10	System Modeling	L8 [28], L23[42]
F11	API Management & Standardization	L1 [9], L9 [10], L11 [30], L17 [36], L27[46]
F12	Service Communication	L1 [9], L2 [22], L6 [26], L11 [30], L17 [36], L18 [39], L21 [40], L25[44]
F13	Infrastructure Automation	L1 [9], L2 [22], L6 [26], L8 [28], L9 [10], L11 [30], L18 [37], L20 [39], L21 [40]
F14	Monitoring & Logging	L1 [9], L2 [22], L6 [26], L8 [28], L9 [10], L11 [30], L18 [39], L21 [40], L25[44], L29[48]
F15	Cloud Computing Adoption	L7 [27], L10 [29]
F16	System Architecture Design	L1 [9], L2 [22], L8 [28], L11 [30], L17 [36], L18 [37], L21 [40]
F17	Database Architecture & Design	L1 [9], L9 [10], L11 [30], L17 [36], L18 [39], L21 [40]
F18	System Security	L1 [9], L6 [26], L8 [28], L11 [30], L17 [36], L21 [40], L22 [41]
F19	DevOps Culture	L8 [28], L9 [10], L11 [30], L19 [38]
F20	Microservices Experience	L2 [22], L2 [23], L10 [29], L11 [30], L18 [37]
F21	System & Software Design Patterns	L8 [28], L11 [30], L19 [38], L21 [40]

19)DevOps culture: The integration of software development (Dev) practices with IT operations (Ops) to create an efficient, sustainable workflow focused on the rapid delivery of business value. The combination of MSA and DevOps brings several other benefits, including increased software release cadence, system reliability and scalability, resilience in case of failures, and decentralized team management to control application development [28]. In the technological aspect, containerization also enables DevOps to achieve faster deployment implementation than VMs as it is not efficient to run each microservice on a separate VM due to its long startup time and increased resource usage [41].

20)Microservices experience: Knowledge and experience in designing, developing, and managing application architectures using the microservices approach. Research [22] highlights higher development costs, possibly due to developers undertaking microservice development for the first time.

21)System and software design patterns: Several MSA design patterns have been identified. The most recurring design patterns during the MSA implementation process in DevOps are Circuit Breaker and migration patterns, followed by Observer, Load Balancer, Scalability, and Deployment [28].

According to Strategy, and a unit of PwC focused on business strategy consulting, capabilities are comprised of four components as illustrated in Fig. 2. Organization, Process, Tools & System, and Knowledge, Skills & Behaviors [12]. Each component is described as follows:

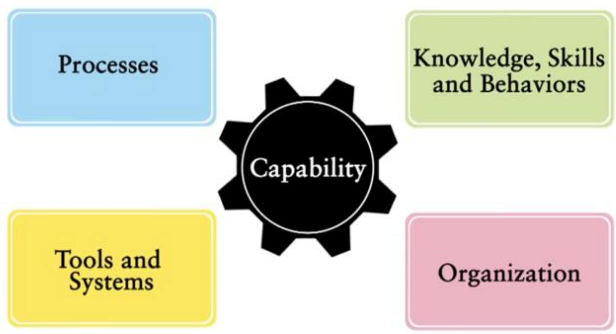

Fig. 2. Four parts of capability according to Strategy and PWC.

a) Organization: The entire structure and design of a company, including governance, management, and its operational model. Organizations with strong capabilities can identify, develop, and leverage competitive advantages in a competitive market. They can transform the knowledge, skills, and behaviors of individuals or groups into sustainable organizational capabilities that create value for all stakeholders involved.

b) Process: The steps or activities used by a company to create value for customers. In the context of an organization, processes aim to achieve efficiency, effectiveness, and desired outcomes. This is crucial to design, manage, and continuously improve processes to enable the organization to operate better, enhance customer satisfaction, and achieve competitive excellence.

c) Tools and system: The right combination of relevant systems and tools can strengthen the capabilities of an organization or individual, facilitating the coordination, efficiency, and monitoring needed to achieve desired goals. It is important to match systems and tools with the specific needs of the organization or individual and ensure they align with established strategies and objectives.

d) Knowledge, skills, and behaviors: In many situations, the success of an individual or group does not solely depend on the knowledge they possess but also on the skills applied in practice and behaviors that support good performance. These interrelated knowledge, skills, and behaviors often become a focus in the development of individuals and human resource management strategies to effectively achieve organizational objectives.

C. Decision Modeling Using AHP

The Analytic Hierarchy Process (AHP) is a widely recognized method for solving complex decision problems developed by Saaty [13]. It breaks down any complex issue into multiple sub-problems using AHP in terms of hierarchy levels, where each level represents a set of criteria or attributes relative to each sub-problem.

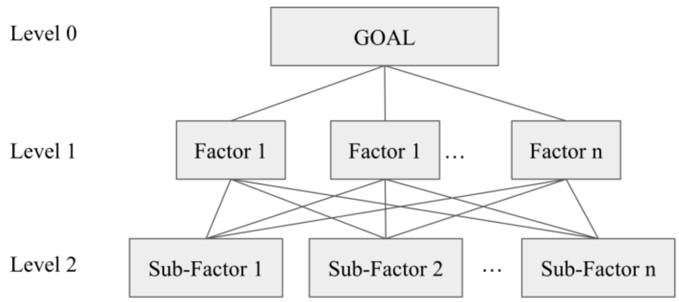

Fig. 3. AHP methodology structure as a framework for complex decision-making problems.

For instance, in an AHP model, Fig. 3 would illustrate the top level (level 0) of the hierarchy representing the goal of the problem, followed by the middle level (level 1) representing strategic and operational factors, and the last level (level 2) typically representing alternative actions or sub-factors that must be considered to achieve the goal. AHP organizes feelings, intuition, and logic in a structured approach to decision-making, which has proven beneficial in environments largely composed of intangible attributes.

AHP enables an individual to organize a system and its environment into interacting elements and then synthesize these by measuring and ranking the impact of these elements on the overall system [13,14]. AHP comprises four phases.

- Phase 1: Problem structuring and designing the hierarchy structure of factors and subfactors.
- Phase 2: Data collection based on pairwise comparisons through expert surveys. In the data collection process, interviews must be conducted with experts actively involved in problem resolution [16]. Subsequently, a pairwise comparison matrix A is formed as follows:

$$A = (a_{ij}), (i, j = 1 \dots n) \quad (1)$$

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} = \begin{pmatrix} 1 & a_{12} & \dots & a_{1n} \\ 1/a_{12} & 1 & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ 1/a_{1n} & 1/a_{2n} & \dots & 1 \end{pmatrix} \quad (2)$$

- Phase 3: Determining the priority weights of each factor and performing a consistency check.

In the third phase, the focus is on identifying the relative importance values of the factors themselves within a specific category rather than comparing them to factors or categories elsewhere [17]. This can be achieved by calculating their local weights using (3) [16].

$$W_{Ai} = \frac{\left(\prod_{j=1}^{n} a_{ij}\right)^{1/n}}{\sum_{i=1}^{n}\left(\prod_{j=1}^{n} a_{ij}\right)^{\frac{1}{n}}} \quad (3)$$

Subsequently, the global weights, which indicate the relative importance of factors among all factors, are calculated as follows: the global weight of a CSF (Critical Success Factor) category is equal to its local weight. For CSFs, the global weight is the product of the local CSF weight and the local weight of the related category. This step concludes with a consistency test through the calculation of the Consistency Ratio (CR) as follows: calculation of the largest eigenvalue.

Calculate the biggest Eigen Value using (4).

$$A\omega = \lambda_{max}\omega \quad (4)$$

Calculate CR value using (5).

$$CR = \frac{CI}{RI} \quad (5)$$

- Phase 4: Analyzing the priority weights and determining solutions for the problem.

In the fourth phase, the priority weights calculated from the pairwise comparisons are analyzed to derive the best solutions for the problem. These weights represent the relative importance of each criterion or alternative, helping to prioritize options based on the decision-maker's preferences.

IV. Result and Discussion

A. Result

After identifying the critical success factors (CSFs) impacting the implementation of MSA, a hierarchical structure was constructed, and the upper categories of these factors were established. The categories are Organization, Process, Tools & System, and Knowledge, Skills & Behaviors, which are specifically explored in this study. The four categories and their respective factors are illustrated in Fig. 4.

Once the hierarchy/categorization was established, comparative assessment was employed to ascertain the relative significance of these factors. A pairwise comparison matrix was then developed to prioritize the comparative assessments into ratio scale measurements. In this study, a nine-point scale was applied to compare the significance of each factor pair, as detailed in Table IV. The evaluation consists of five values: 1, 3, 5, 7, and 9, representing equal importance, slight importance over another, essential or strong importance, demonstrated importance, and absolute/extreme importance, respectively. Intermediate assessments were expressed through the use of even numbers 2, 4, 6, and 8.

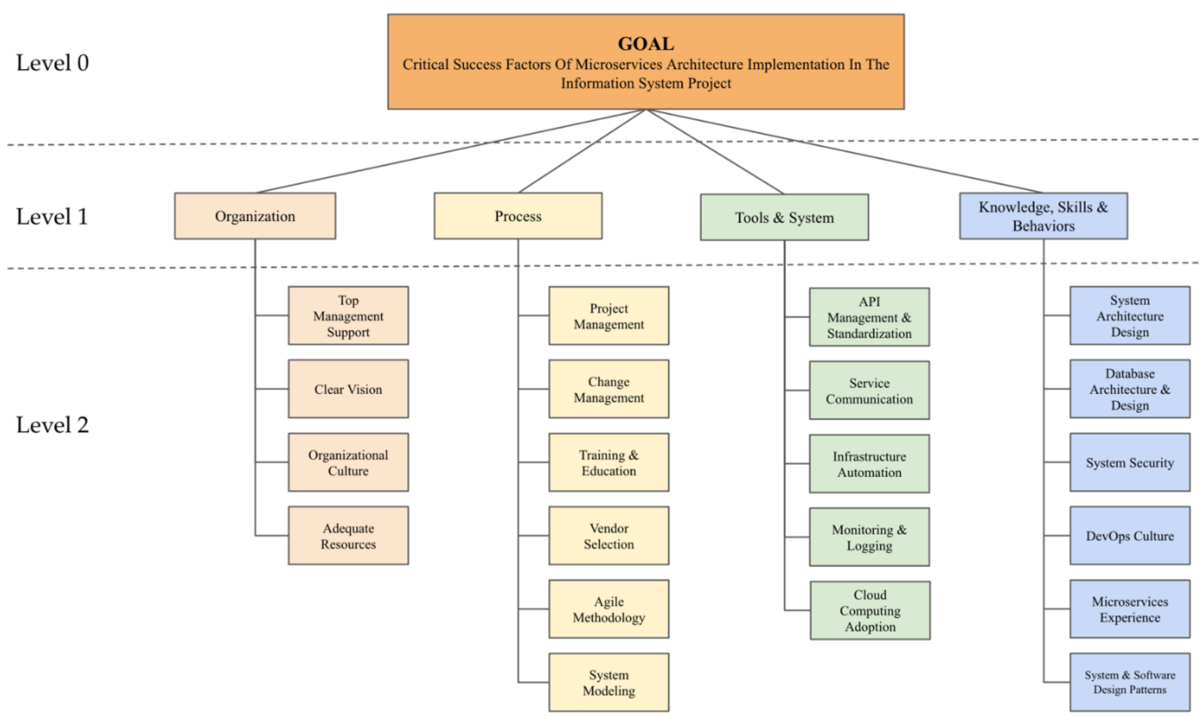

Fig. 4. Microservice architecture implementation success factors hierarchical structure.

Experts were asked to perform pairwise comparisons of factors. These comparisons were then used to create a pairwise comparison matrix. An example of such comparison is management support versus a clear vision, posing the question, "How significant is top management support compared to a clear vision?" If experts responded with "demonstrated importance," a value of 7 was assigned in the matrix. Conversely, if experts indicated that a clear vision holds greater importance than top management support, a value of 1/7 was added to the matrix. The involvement of experts in this prioritization was documented in a pairwise comparison survey.

TABLE IV. SCALE MEASUREMENT

i compare with j	a_{ij}	a_{ji}
Equally Importance (EI)	1	1/1
Moderately Importance (MI)	3	1/3
Strongly More Importance (SMI)	5	1/5
Very Strongly More Importance (VSMI)	7	1/6
Extremely More Importance (EMI)	9	1/7
Equally Importance (EI)	1	1/1

After completing the comparison matrix, researchers proceeded to calculate priority weights, consistency index, random consistency index, and the consistency ratio among the factors using the pairwise comparison matrix. Table V displays the random consistency index provided by [18]. It is crucial to note that when calculating the consistency ratio, a threshold must not be exceeded. To ensure desirable outcomes, it is advised that the consistency ratio is within the range of 0 and 0.1, especially for matrices exceeding a 4x4 dimension, as recommended by Saaty [18]. If the calculated consistency ratio is equal to or less than the acceptable value, this indicates that the comparative assessments represented in the matrix have a consistent level of consistency.

TABLE V. RANDOM CONSISTENCY INDEX

Size of Matrix	**Random Consistency Index**
1	0
2	0
3	0.58
4	0.90
5	1.12
6	1.24
7	1.32

In this study, the process of calculating priorities was divided into four according to the number of categories, and the results from the consistency index check were as follows: Organization (Cat 1) 0.0931, Process (Cat 2) 0.0924, System & Tools (Cat 3) 0.0972, and Knowledge, Skills & Behaviors (Cat 4) 0.0822. These results indicate that the overall process of calculating priorities is consistent.

Table VI displays the results of the priority weights and rankings for each category and the factors associated with them. The Category column contains weight values representing the importance level of each category. The four categories derived from the supporting parts of the capabilities required by any company or institution, in the context of this study, refer to capabilities in implementing MSA. Analysis revealed that the Organization category is most crucial, with a weight of 0.4844 in the implementation of MSA. The Knowledge, Skills & Behaviors category ranks second with a weight of 0.2240, followed closely by the Process and Tools & System categories with weights of 0.1615 and 0.1302, respectively.

In the Factor column, there are two weight values: local weight as a priority within the category scope of the factor and global weight as the basis for the Priority column as the factor's ranking compared to all studied factors. In the Organization category, the Top Management Support factor is deemed most important, with a local weight of 0.5011. For the Process category, the Project Management factor is the main priority with a local weight of 0.3046. In the Tools & System category, the highest local weight of 0.3938 is for the API Management & Standardization factor, and in the last category, Knowledge, Skills & Behaviors, the Microservices Experience factor is considered most important with a local weight of 0.2300. Globally, the top three critical success factors are Top Management Support, Clear Vision, and Adequate Resources.

B. *Discussion*

This study focuses on the process of identifying, defining, and evaluating critical success factors (CSFs) for the implementation of Microservices Architecture (MSA) at one of the leading telecommunications companies in Indonesia. Each factor will be mapped into categories based on an institution's or company's capabilities to implement MSA. This mapping process is crucial as a high-level overview for the management at the company to understand which internal capabilities need to be enhanced for similar initiatives to be carried out effectively and efficiently in the future. A literature study was conducted by combining a manual search of studies deemed relevant to the topic discussed, supplemented with the application of a systematic literature review (SLR) to enrich the determinants of successful MSA implementation.

Table VII displays the demographics of the experts who assisted in the prioritization process of the critical success factors (CSFs) in this study. We were supported by four experts, consisting of two types: those who were involved and those who were not directly involved with the project implementation. The presence of these two types of experts is expected to achieve a balance that supports better objectivity in the prioritization process conducted through surveys. The experts were selected with several selection criteria such as more than five years of experience in the software field, having experience in developing systems based on MSA, and being certified in software architecture or IT service management.

TABLE VI. SUMMARY OF WEIGHTING AND PRIORITY LEVEL OF FACTORS

Categories			**Factors**					
ID	***Name***	***Weight***	***ID***	***Name***	***Local Weight***	***Local Rank***	***Global Rank***	***Priority***
Cat 1	Organization	0.4844	F1.1	Top Management Support	0.5011	1	0.2427	1
			F1.2	Clear Vision	0.2630	2	0.1274	2
			F1.3	Organizational Culture	0.0768	4	0.0372	9
			F1.4	Adequate Resources	0.1591	3	0.0771	3
Cat 2	Process	0.1615	F2.1	Project Management	0.3046	1	0.0492	6
			F2.2	Change Management	0.1700	3	0.0274	14
			F2.3	Training & Education	0.0830	6	0.0134	20
			F2.4	Vendor Selection	0.1557	4	0.0251	16
			F2.5	Agile Methodology	0.1072	5	0.0173	18
			F2.6	System Modeling	0.1795	2	0.0290	13
Cat 3	Tools & System	0.1302	F3.1	API Management & Standardization	0.3938	1	0.0513	5
			F3.2	Service Communication	0.2853	2	0.0371	10
			F3.3	Infrastructure Automation	0.1346	3	0.0175	17
			F3.4	Monitoring & Logging	0.1039	4	0.0135	19
			F3.5	Cloud Computing Adoption	0.0824	5	0.0107	21
Cat 4	Knowledge, Skills & Behaviors	0.2240	F4.1	System Architecture Design	0.1550	4	0.0347	11
			F4.2	Database Architecture & Design	0.1312	5	0.0294	12
			F4.3	System Security	0.2004	2	0.0449	7
			F4.4	DevOps Culture	0.1675	3	0.0375	8
			F4.5	Microservices Experience	0.2300	1	0.0515	4
			F4.6	System & Software Design Patterns	0.1160	6	0.0260	15

TABLE VII. EXPERTS DEMOGRAPHICS

Criteria		**Total**	**Percentage**
Age	20-30	1	25%
	30-50	3	75%
	> 50	0	0%
Education	Bachelor	2	50%
	Master	2	50%
	Doctoral	0	0%
Experience	2-5 years	1	25%
	5-10 years	3	50%
	> 10 years	1	25%
Job Title	Technical Staff	3	75%
	Manager	1	25%
	Executive	0	0%
Project Involvement	Yes	2	50%
	No	2	50%

The Analytic Hierarchy Process (AHP) was employed to derive local and global weights as benchmarks for determining the priority levels of each category and critical success factors (CSFs) for the implementation of Microservices Architecture (MSA). This study produced a taxonomy of critical success factors (CSFs), taking into account both global and local weights. The identified success factors were categorized into four distinct sections; every section offers several insights for enhancing the MSA implementation process at the company.

Fig. 5 presents the outcome of the global weighting process for all determining factors of success using AHP as the method for complex decision-making. It can be concluded that the factor Top Management Support significantly outperforms other factors. Following in second and third order are the factors of Clear Vision and Adequate Resources respectively. An interesting finding is that these three factors fall within the same category of Organization. This justifies that the Organization category, displayed in Fig. 6, also ranks first from a capability perspective and significantly surpasses other categories, such as Knowledge, Skills & Behaviors in second place. The results of this study address the research question.

For future research, we acknowledge several limitations of this study, particularly in determining the critical success factors (CSFs), although the literature review utilized the Systematic Literature Review (SLR) method. The results from these factors have not undergone empirical testing for more objective factor selection. Approaches such as the frequency approach, found in [19,20], could be employed. Furthermore, the implementation of an expert judgment committee to evaluate the factors could be conducted to ensure that each factor is clearly defined [21].

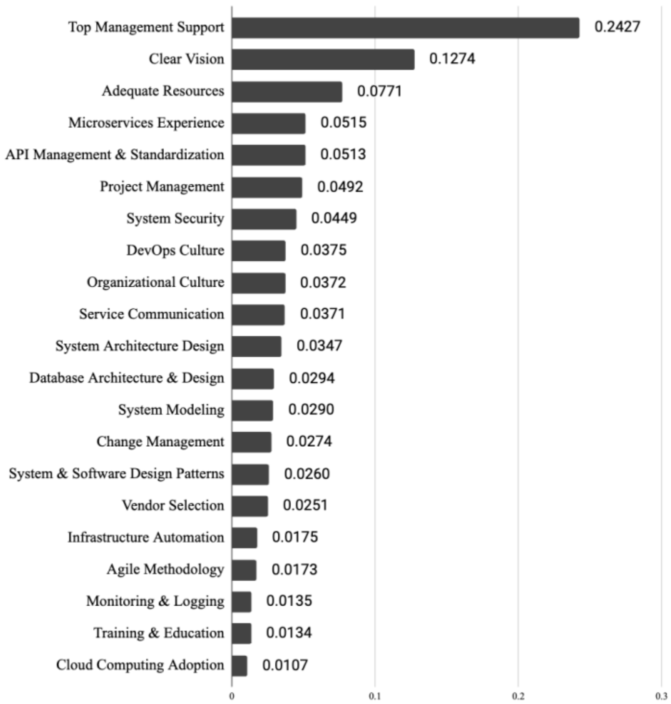

Fig. 5. List of overall implementation success factors ordered by global weight.

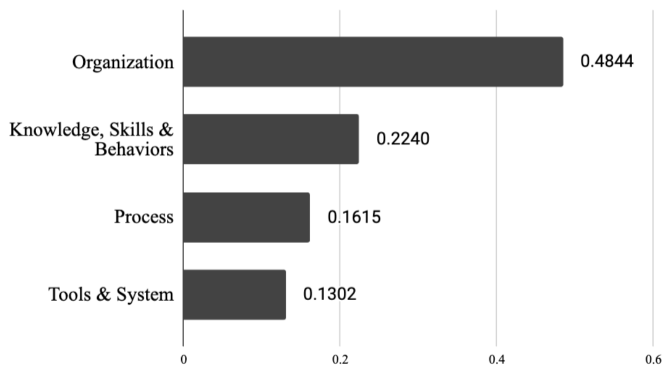

Fig. 6. List of categories based on capabilities ordered by weight.

V. Conclusion and Future Work

In this study, 21 critical success factors (CSFs) were successfully identified from the literature review process and categorized into four categories: (1) Organization, (2) Process, (3) Tools & System, and (4) Knowledge, Skills & Behaviors. To perform the evaluation process of the priority level of each factor and category, the Analytical Hierarchy Process (AHP) was selected as the decision-making methodology based on survey data conducted among project participants and practitioners.

The conclusions from the results of this study, as depicted in Table VI, indicate that from a categorical perspective, Organization ranks first as the most important. The top three sub-factors of critical success for the implementation of Microservices Architecture (MSA) are: (1) Top Management Support, (2) Clear Vision, and (3) Adequate Resources. When examining the priority of each category, based on local weights, it is evident that in the Organization category, the primary priority factor is Top Management Support (F1.1). For the second category, Process, the main priority is factor Project Management (F2.1). In the category Tools & System (F3.1), the factor API Management & Standardization is the priority, and for the last category, Knowledge, Skills & Behaviors, Microservices Experience (F4.5) ranks as the top factor.

To ensure the successful implementation of Microservices Architecture (MSA) in the organization, it is crucial to enhance leadership engagement, strategic clarity, and resource allocation. Strengthening top management support is vital, as their active involvement and oversight are essential for maintaining momentum and addressing challenges effectively. Additionally, articulating a clear vision and developing a robust communication strategy will ensure that teams are aligned and working towards common objectives, thereby minimizing misalignment and inefficiencies.

Furthermore, the company should prioritize the allocation of sufficient resources, both financial and human, to support the MSA initiative. This includes investing in the recruitment and upskilling of personnel to develop the necessary expertise in microservices, as well as ensuring access to essential tools and systems. By addressing these critical areas, the company can leverage the identified critical success factors to mitigate potential risks, optimize the execution of the project, and fully realize the benefits of MSA, including enhanced scalability, flexibility, and operational efficiency.

For future work, it is recommended that further research be conducted to explore the long-term impacts of the identified critical success factors (CSFs) on the sustainability and scalability of Microservices Architecture (MSA) implementations. Additionally, conducting comparative studies across different industries and organizational sizes would be valuable to determine whether the CSFs identified in this study are universally applicable or if they vary based on context.

References

[1] Smith, D., Jones, M., Williams, R., & Johnson, L. (2020). Embracing Microservices: A Comprehensive Overview of Microservice Architecture and Its Impact on Software Development. Journal of Software Engineering and Applications, 13(4), 123-139.

[2] Gupta, S. (2016). Microservices. Procedia Computer Science, 78, 278-283.

[3] Johnson, R., & Ho, A. (2019). Adopting microservices at Netflix: Lessons for architectural design. IEEE Software, 36(1), 39-45.

[4] Lewis, J., & Fowler, M. (2014). Microservices. online: https://martinfowler.com/articles/microservices.html (Accessed 20 May 2023)

[5] Newman, S. (2015). Building Microservices: Designing Fine-Grained Systems. O'Reilly Media, Inc, 551.

[6] Dragoni, N., Giallorenzo, S., Lafuente, A. L., Mazzara, M., Montesi, F., Mustafin, R., & Safina, L. (2017). Microservices: yesterday, today, and tomorrow. Present and Ulterior Software Engineering, 195-216.

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

[8] Loukides, Mike. & SwoyerLewis, Steve. (2020). Microservices Adoption in 2020. online: https://www.oreilly.com/radar/microservices-adoption-in-2020/ (Accessed 20 May 2023)

[9] Jacopo Soldani, Damian Andrew Tamburri, Willem-Jan Van Den Heuvel. (2018). The pains and gains of microservices: A Systematic

gray literature review. The Journal of Systems and Software 146 (2018) 215-232.

[10] Xin Zhou, Shanshan Li a, Lingli Cao, He Zhang, Zijia Jia, Chenxing Zhong, Zhihao Shan, Muhammad Ali Babar. (2023). Revisiting the practices and pains of microservice architecture in reality: An industrial inquiry. The Journal of Systems & Software 195 (2023) 111521

[11] Jamshidi, P., Pahl, C., Mendonca, N.C., Lewis, J., Tilkov, S., 2018. Microservices: the journey so far and challenges ahead. IEEE Softw. 35 (3), 24-35. doi:10.1109/MS.2018.2141039.

[12] Strategy& (2014). What is a capability?. online: https://www.strategyand.pwc.com/gx/en/about/media/videos/2015-and-older/what-is-a-capability.html (Accessed 20 May 2023)

[13] Saaty TL. What is the analytic hierarchy process? In:Mathematical Models for Decision Support. Berlin, Heidelberg: Springer; 1988:109-121.

[14] Saaty TL. The analytic hierarchy process. New York: McGraw Hill; 1980.

[15] Saaty TL. Decision making for leaders. Pittsburgh: RWS Publications; 1990.

[16] Cheng, E.W. and Li, H. (2001),"Analytic hierarchy process: an approach to determine measures for business performance",Measuring Business Excellence, Vol. 5 No. 3, pp. 30-37.

[17] Shi, H., Peng, S.Z., Liu, Y. and Zhong, P. (2008),"Barriers to the implementation of cleaner production in chinese SMEs: government, industry and expert stakeholders perspectives",Journal of CleanerProduction, Vol. 16 No. 7, pp. 842-852.

[18] Saaty, T.L., 1986. Axiomatic foundation of the analytic hierarchy process. Manag. Sci. 32 (7), 841-855.

[19] Khan, A. A., & Shameem, M. (2020). Multicriteria decision-making taxonomy for DevOps challenging factors using analytical hierarchy process. Journal of Software: Evolution and Process, 32(10). https://doi.org/10.1002/smr.2263

[20] Akbar, M. A., Naveed, W., Mahmood, S., Alsanad, A. A., Alsanad, A., Gumaei, A., & Mateen, A. (2020). Prioritization Based Taxonomy of DevOps Challenges Using Fuzzy AHP Analysis. IEEE Access, 8, 202487-202507. https://doi.org/10.1109/ACCESS.2020.3035880

[21] Sambasivan, M., & Fei, N. Y. (2008). Evaluation of critical success factors of implementation of ISO 14001 using analytic hierarchy process (AHP): a case study from Malaysia. Journal of Cleaner Production, 16(13), 1424-1433. https://doi.org/10.1016/j.jclepro.2007.08.003

[22] Lenarduzzi, V., Lomio, F., Saarim√§ki, N., & Taibi, D. (2020). Does migrating a monolithic system to microservices decrease the technical debt? In Journal of Systems and Software (Vol. 169). Elsevier Inc. https://doi.org/10.1016/j.jss.2020.110710

[23] Auer, F., Lenarduzzi, V., Felderer, M., & Taibi, D. (2021). From monolithic systems to Microservices: An assessment framework. Information and Software Technology, 137. https://doi.org/10.1016/j.infsof.2021.106600

[24] Taibi, D., Lenarduzzi, V., Pahl, C., & Janes, A. (2017). Microservices in agile software development: A workshop-based study into issues, advantages, and disadvantages. ACM International Conference Proceeding Series, Part F129907. https://doi.org/10.1145/3120459.3120483

[25] Stefanou, C. J. (n.d.). Association for Information Systems Association for Information Systems AIS Electronic Library (AISeL) AIS Electronic Library (AISeL) The Selection Process of Enterprise Resource Planning (ERP) The Selection Process of Enterprise Resource Planning (ERP) Systems Systems The Selection Process of Enterprise Resource Planning (ERP) Systems. https://aisel.aisnet.org/amcis2000/418

[26] Karabey Aksakalli, I., √áelik, T., Can, A. B., & TekiÃánerdoƒüan, 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

[27] Nordli, E. T., Haugeland, S. G., Nguyen, P. H., Song, H., & Chauvel, F. (2023). Migrating monoliths to cloud-native microservices for customizable SaaS. Information and Software Technology, 160. https://doi.org/10.1016/j.infsof.2023.107230

[28] Waseem, M., Liang, P., & Shahin, M. (2020). A Systematic Mapping Study on Microservices Architecture in DevOps. Journal of Systems and Software, 170. https://doi.org/10.1016/j.jss.2020.110798

[29] Hasan, M. H., Osman, M. H., Admodisastro, N. I., & Muhammad, M. S. (2023). Legacy systems to cloud migration: A review from the architectural perspective. Journal of Systems and Software, 202. https://doi.org/10.1016/j.jss.2023.111702

[30] Waseem, M., Liang, P., Shahin, M., di Salle, A., & M√° rquez, G. (2021). Design, monitoring, and testing of microservices systems: The practitioners' perspective. Journal of Systems and Software, 182. https://doi.org/10.1016/j.jss.2021.111061

[31] Merhi, M. I. (2021). Evaluating the critical success factors of data intelligence implementation in the public sector using analytical hierarchy process. Technological Forecasting and Social Change, 173. https://doi.org/10.1016/j.techfore.2021.121180

[32] Merhi, M. I., & Leighton, J. (2015). A process model leading to successful implementation of electronic health record systems. In Int. J. Electronic Healthcare (Vol. 8).

[33] Young, R., & Jordan, E. (2008). Top management support: Mantra or necessity? International Journal of Project Management, 26(7), 713-725. https://doi.org/10.1016/j.ijproman.2008.06.001

[34] Gaardboe, R., Nyvang, T., & Sandalgaard, N. (2017). Business Intelligence Success applied to Healthcare Information Systems. Procedia Computer Science, 121, 483-490. https://doi.org/10.1016/j.procs.2017.11.065

[35] Warrick, D. D. (2017). What leaders need to know about organizational culture. Business Horizons, 60(3), 395-404. https://doi.org/10.1016/j.bushor.2017.01.011

[36] Waseem, M., Liang, P., Ahmad, A., Shahin, M., Khan, A. A., & M√° rquez, G. (2022). Decision models for selecting patterns and strategies in microservices systems and their evaluation by practitioners. 135-144. https://doi.org/10.1145/3510457.3513079

[37] Poniszewska-Maranda, A., MacIoch, J., Borowska, B., & Maranda, W. (2021). Mechanisms for Transition from Monolithic to Distributed Architecture in Software Development Process. Proceedings - IEEE Computer Society's Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, MASCOTS. https://doi.org/10.1109/MASCOTS53633.2021.9614287

[38] Nino-Martinez, V. M., Octavio Ocharan-Hernandez, J., Limon, X., & Perez-Arriaga, J. C. (2021). Microservices Deployment: A Systematic Mapping Study. Proceedings - 2021 9th International Conference in Software Engineering Research and Innovation, CONISOFT 2021, 24-33. https://doi.org/10.1109/CONISOFT52520.2021.00016

[39] Waseem, M., Liang, P., Marquez, G., & Salle, A. di. (2020). Testing microservices architecture-based applications: A systematic mapping study. Proceedings - Asia-Pacific Software Engineering Conference, APSEC, 2020-December, 119-128. https://doi.org/10.1109/APSEC51365.2020.00020

[40] Premarathna, D., & Pathirana, A. (2021). Theoretical framework to address the challenges in Microservice Architecture. Proceedings - International Research Conference on Smart Computing and Systems Engineering, SCSE 2021, 195-202. https://doi.org/10.1109/SCSE53661.2021.9568346

[41] Sultan, S., Ahmad, I., & Dimitriou, T. (2019). Container security: Issues, challenges, and the road ahead. IEEE Access, 7, 52976-52996. https://doi.org/10.1109/ACCESS.2019.2911732

[42] Zhong, C., Li, S., Huang, H., Liu, X., Chen, Z., Zhang, Y., & Zhang, H. (2024). Domain-Driven Design for Microservices: An Evidence-Based Investigation. IEEE Transactions on Software Engineering, 50(6), 1425–1449. https://doi.org/10.1109/TSE.2024.3385835

[43] Ayas, H. M., Hebig, R., & Leitner, P. (2024). The Roles, Responsibilities, and Skills of Engineers in the Era of Microservices-Based Architectures. 2024 IEEE/ACM 17th International Conference on Cooperative and Human Aspects of Software Engineering (CHASE), 13–23.

[44] Purfallah Mazraemolla, Z., & Rasoolzadegan, A. (2024). An effective failure detection method for microservice-based systems using distributed tracing data. Engineering Applications of Artificial

Intelligence, 133(PF), 108558. https://doi.org/10.1016/j.engappai.2024.108558

[45] Michael Ayas, H., Hebig, R., & Leitner, P. (2024). An empirical investigation on the competences and roles of practitioners in Microservices-based Architectures. Journal of Systems and Software, 213(March), 112055. https://doi.org/10.1016/j.jss.2024.112055

[46] Lercher, A., Glock, J., Macho, C., & Pinzger, M. (2024). Microservice API Evolution in Practice: A Study on Strategies and Challenges. Journal of Systems and Software, 215(May), 112110. https://doi.org/10.1016/j.jss.2024.112110

[47] Ünlü, H., Kennouche, D. E., Soylu, G. K., & Demirörs, O. (2024). Microservice-based projects in agile world: A structured interview. Information and Software Technology, 165(September 2023). https://doi.org/10.1016/j.infsof.2023.107334

[48] Giamattei, L., Guerriero, A., Pietrantuono, R., Russo, S., Malavolta, I., Islam, T., Dînga, M., Koziolek, A., Singh, S., Armbruster, M., Gutierrez-Martinez, J. M., Caro-Alvaro, S., Rodriguez, D., Weber, S., Henss, J., Vogelin, E. F., & Panojo, F. S. (2024). Monitoring tools for DevOps and microservices: A systematic grey literature review. Journal of Systems and Software, 208(November 2023), 111906. https://doi.org/10.1016/j.jss.2023.111906