Data Domain Servitization for Microservices Architecture

ARTICLE HISTORY

Received 18 September 2024
Accepted 23 October 2024

Patricio Michael Paccha Angamarca
Salesian Polytechnic University
Cuenca, Ecuador
ppaccha@est.ups.edu.ec
ORCID: 0000-0001-6285-7390

Victor Vicente Velepucha Bonett
National Polytechnic School
Quito, Ecuador
victor.velepucha@epn.edu.ec
ORCID: 0000-0002-7335-2571

Data Domain Servitization for Microservices Architecture

Patricio Michael Paccha Angamarca
Salesian Polytechnic University
Cuenca, Ecuador
ppaccha@est.ups.edu.ec
ORCID: 0000-0001-6285-7390

Victor Vicente Velepucha Bonett
National Polytechnic School
Quito, Ecuador
victor.velepucha@epn.edu.ec
ORCID: 0000-0002-7335-2571

Abstract— Microservices have emerged as a software design paradigm where small, autonomous services interact to meet business requirements. However, transitioning from monolithic systems to microservices presents challenges, especially when multiple subdomains share transactional tables to maintain referential integrity across separate databases. Ensuring each microservice handles business data while adhering to ACID properties—namely, atomicity, consistency, isolation, and durability—is crucial. This requires unique, consistent, and low-dependency data from a business domain perspective.

Systematic Literature Review serves as a secondary research method aimed at evaluating the existing body of scientific literature. It helps identify existing work, highlight research gaps, and propose new research directions. In software engineering, SLRs offer a comprehensive overview of studied research areas.

This article reports an empirical study based on a systematic literature review aimed at identifying modeling techniques for segmenting data structures during microservice design. The review found limited methods to address the appropriate level of data granularity per microservice. These findings highlight a need for further research into processes and methodologies that can effectively handle data segmentation and consistency within microservice architectures.

Keywords— *Servitization, Granularity, Data Segmentation, Microservices, Data Architecture, Microservices Architecture*

I. Introduction

Microservices architecture is a software architecture style focused on using small, lightweight services designed to adapt to dynamic business environments [1]. Each microservice results from decomposing monolithic systems or is modeled as an independent component from development to deployment. This approach enables the scalability and evolution of each service with autonomy over its data stores.

A microservices ecosystem manages its governance by delegating business responsibilities to each service, with boundaries defined by subdomains that handle specific parts of the business process. However, defining the subdomain for each microservice introduces several complexities when data entities that are closely related are shared. This approach necessitates maintaining independent data stores for each microservice, leading to various challenges such as ensuring the atomicity, consistency, isolation, and durability (ACID) of data, managing transactions and their dependencies, and defining mapping rules, among others [2].

Microservices are the result of breaking down an application services into smaller components to enhance composability, agility, deployment, and alignment with business objectives [1]. Shahir et al. state that decentralized data management is a crucial design characteristic of microservices. They propose that each microservice should manage its own database, as using a shared database violates the principles of microservices architecture. Therefore, a key element is to ensure that each microservice maintains its own data storage [2].

There is a growing demand for data-driven web applications, such as those used for recommendations, predictions, and segmentation. These applications are often transformed into complex conglomerates of services that operate with challenges in coherence and management within their architecture [3] Microservices can be a potential solution. Unlike traditional services, microservices represent an architectural style focused on decoupling and specialization. The guiding principle is that each microservice should perform a single task in the most efficient way and be easy to understand. This task should have the smallest possible function, but not necessarily the minimal one [1], [4], [5], [6], this leaves software architects with the responsibility of finding a middle ground and determining the appropriate level of function and data granularity based on their experience and judgment. According to authors like Nadareishvili et al., breaking a service into smaller parts requires a method that establishes the criteria for a minimum viable level of granularity to be managed by each microservice [7]. The existence of a minimum viable size is not proposed. Instead, they note that there is no definition in the literature regarding how small or independent microservices should be [8], ultimately leaving it to those designing microservices to establish their own guidelines.

Data modeling initially relies on the abstraction of business entities, evaluated as a unified set with attributes, relationships, and interactions [9]. In microservices design, this modeling must be segmented, with each microservice specializing in a specific section. This involves defining an individual storage strategy, consuming data from other microservices, and mapping to the original data source.

The authors highlight the need for existing or new methods in the literature to guide the segmentation of data structures, ensuring viable microservices implementation. Without such methods, designers may rely on intuition and trial and error, potentially breaking design principles, consuming excessive resources, and blurring distinctions from traditional SOA services.

In a preliminary literature review, the authors did not find a proposed method for segmenting a business data model or determining the appropriate granularity for each microservice. Therefore, a more comprehensive study through a Systematic Literature Mapping is necessary to identify the presence or absence of such methods in the scientific literature. Identifying these methods would help highlight and disseminate them among software architects and researchers, promoting microservices architecture as a viable option. Conversely, if no such methods are found, it would open a new research area focused on this specific topic.

The study aims to conduct an exhaustive literature search to determine if methods exist for segmenting data structures in microservice design. The article explores the challenge of aligning microservice principles with a method to determine the appropriate data granularity (fine or coarse) for each service without physically dividing the business domain while ensuring better efficiency, performance, and balance of the microservice.

This article is structured as follows: Section II describes the Systematic Review procedure, including the method, research questions, inclusion and exclusion criteria, search strategy, selection process, and data extraction. Section III details the techniques found for data modeling in microservices. Section IV discusses the results obtained. Finally, Section V presents the conclusions of the study.

II. Systematic Review

A. Research Framework

According to Kitchenham [11] in several of his collaborations, has established the main approaches to carry out literature reviews rigorously in the field of software engineering such as Systematic Literature Review, SLR; Systematic Mapping Study, SMS [10] and Mapping Review Combined with a Systematic Review. A Systematic Mapping Study (SMS) is defined as “a broad review of primary studies in a specific topic to identify the available evidence in that area”. In this context, primary and secondary studies are distinguished. A primary study is “an empirical study investigating a specific research question”, while a secondary study is “a study that reviews all primary studies related to a specific research question with the aim of integrating/synthesizing the evidence related to that research question”.

A Mapping Review Combined with a Systematic Review provides a structured reporting framework for research, often presenting results through categorization, which frequently offers a visual summary of its findings (the map). The analysis of results is conducted by extracting relevant information and categorizing it to determine the contributions of primary studies within the research area.

In addition to SMS, there are other types of secondary studies, such as Systematic Reviews. According to Kitchenham and Charters [10], a Systematic Literature Review is “a type of secondary study that uses a well-defined methodology to identify, analyze, and interpret all evidence related to a specific research question in an objective and repeatable manner (to some degree)”.

Given the issues outlined in the introduction and the objectives set for this research work, an SMS was chosen as the proposed Research Framework.

B. Methodology

To minimize potential threats to the validity of the research and to provide appropriate answers to the research questions, the authors have chosen to use a Systematic Mapping Study (SMS) following the process below [10], [11]:

1) *Research Question*
   - *a) Problem Statement*
   - *b) Research Questions*

2) *Inclusion and Exclusion Criteria*
   - *a) Selection Criteria*

3) *Search Strategy*
   - *a) Control Group*
   - *b) Search String*
   - *c) Candidate Studies*

4) *Selection Process*
   - *a) Candidate Studies*
   - *b) Study Selection*
   - *c) Primary Studies*

5) *Data Extraction*
   - *a) Feature Extraction*
   - *b) Model Extraction*

Figure 1 presents the sequence of steps performed as part of the Systematic Mapping Study applied to this research project.

C. Research Question

1) Problem Statement

Unlike traditional SOA services, a distinctive characteristic of microservices is that, by design, each one should have its own specialized and independent data model and storage. This approach avoids direct references between services, making them more decoupled and modular, enabling the composition of more complex deployments [2].

During business data modeling, analysts typically abstract all entities, attributes, relationships, and interactions to create a unified representation model [9]. However, in the design of microservices, it is necessary to segment the data to identify, for each microservice, the relevant entities, attributes, relationships, and interactions specific to its limited scope.

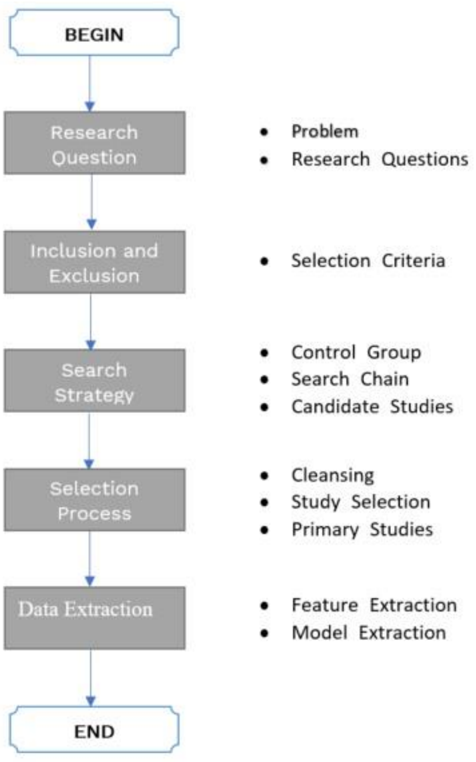

Fig. 1. Systematic Mapping Study Process

This segmentation process is often based on the intuition and judgment of the developer or software architect, lacking a guideline or method grounded in theory that defines the optimal level of granularity and the appropriate boundaries for subdivision.

Thus, there is a need to explore the scientific literature for proposed methods for data structure segmentation. If such methods are found, they could strengthen the theoretical and practical aspects that would encourage broader adoption of microservices architecture according to its theoretical conception. Without a defined method, each practitioner may select the desired level of granularity based on intuition and trial and error. This could lead to violations of certain design principles, excessive resource consumption, and neglect of the necessary differentiation from traditional SOA services, rendering microservices almost unnecessary.

2) Research Question

To ensure a comprehensive research approach to the proposed problem, the following research questions were defined by mutual agreement among the authors:

- **RQ1: What data modeling techniques can be used for business data segmentation in the construction of microservices?**
- **RQ2: How can the granularity of microservices be modeled or established?**

Table 1 presents the aspects that these research questions aim to address.

TABLE I. OBJECTIVES OF THE RESEARCH QUESTIONS

Question ID	Objective
QR1	Determine which data modeling techniques exist for data segmentation in the implementation of microservices. If any are found, detail the characteristics of the proposed data segmentation procedure and discuss its coverage in relation to the proposed problem.
QR2	Identify the approaches used to define and measure the levels of functional and data granularity in the implementation of microservices.

D. Inclusion and Exclusion Criteria

1) Inclusion Criteria:

- The article describes a method aimed at data segmentation in microservices design.
- The article describes a process or data modeling technique focused on data segmentation in microservices design.
- The article describes aspects or criteria used to define the granularity of microservices.
- The article presents examples or case studies demonstrating the application of data segmentation in microservices design.
- The article discusses the challenges of not having or being unable to define a data segmentation method in microservices design.

2) Exclusion Criteria:

- Articles that only define the concepts, benefits, criteria, or implementations of microservices architecture, without addressing the incorporation of a method for data segmentation in microservices design.
- Articles that focus solely on the design or applicability aspects of microservices without addressing data modeling.
- Articles that address data segmentation but do not focus on its applicability to microservices.

3) General Criteria:

- The search will be conducted in recognized scientific databases: Web of Science, IEEE Xplore, Scopus, and Science Direct.
- Only articles with full-text availability will be considered.
- Articles from the last three years will be considered, as this period encompasses the existence of microservices architecture.
- Additional articles will be considered through snowballing or opportunistic searches, as needed.

E. Search Strategy

1) Control Group

Based on the proposed topic, each author independently conducted preliminary searches in scientific databases using the terms identified in the research questions and their intuitive judgment to find related articles. The articles presented by each author are listed in Table II.

TABLE II. INITIALLY IDENTIFIED ARTICLES

Author	Articles
Author 1	[1], [12], [13], [14], [15], [16], [17]
Author 2	[1], [14], [17], [18], [19], [20], [21]

A review meeting was subsequently held to focus on selecting articles most closely aligned with the objective of the study by reviewing titles and abstracts.

The control group was formed with the articles presented in Table III.

TABLE III. ARTICLES FORMING THE CONTROL GROUP

Control Group Articles
[1], [14], [17], [18], [20], [21]

2) Search String

To construct an appropriate search string for this study, the definition of Population, Intervention, Outcomes, and Context (PICO) was used [22]. The terms were derived from the review of the Control Group articles and the Research Questions.

Additionally, synonyms or alternative words for key terms were identified. This broadens the scope of the search and helps to retrieve the maximum number of relevant primary studies on the topic.

The results of this activity are shown in Table IV.

TABLE IV. ELEMENTS IN THE RESEARCH QUESTIONS STRUCTURE

Population	Software architects and analysts, software researchers	Software analysts, software researchers
Intervention	Microservices (alternative: Microservitization)	microservice, microservitization
Outcomes	Data granularity (alternatives: database, data segmentation, data modeling)	granularity, database, data segmentation, data modeling
Context	Software development companies, software development teams, software consultants, software researchers, software product creators, software industry	software companies, software teams, software researchers, software product creators

The search strings were constructed by combining terms from the Intervention and Outcomes sections. The final expression was formed by conjunctions between sub-expressions in each group. A pilot test of preliminary search strings was conducted in the Scopus digital library. The results are presented in Table V.

TABLE V. PILOT RESULTS FOR SEARCH STRING IN SCOPUS

Search String	Count
("microservices" AND "granularity")	14
(("microservices" OR "microservitization") AND ("granularity" OR "database"))	95
(("microservices" OR "microservitization") AND ("granularity" OR "database" OR "data segmentation"))	95
(("microservices" OR "microservitization") AND ("granularity" OR "database" OR "data segmentation" OR "data modeling"))	96

The pilot results show that an adequate number of articles were retrieved using the last proposed search string. The search string includes all articles proposed for the control group. The authors agree that the chosen search string for this study is:

(("microservices" OR "microservitization") AND ("granularity" OR "database" OR "data segmentation" OR "data modeling"))

3) Candidate Studies

Searches were conducted in the following scientific databases: Web of Science, Scopus, IEEE Xplore, and Science Direct, using the search string selected in the previous step in the advanced search option. The results of the articles found are presented in Table VI.

TABLE VI. SEARCH RESULTS

Database	Count
Scopus	96
Web of Science	6
IEEE Xplore	11
Science Direct	86

F. Selection Process

1) Refinement

After querying the scientific databases, a total of 199 candidate articles were identified, establishing the baseline for analysis. However, some of these articles were duplicated across databases, necessitating a refinement process to remove duplicates or articles with identical content. Additionally, any references to entire conferences or workshops, rather than specific articles, were excluded. This resulted in 174 articles proceeding to the next stage.

2) Study Selection

Once the candidate studies were obtained, the selection process proceeds as follows:

- Apply the previously determined inclusion and exclusion criteria by reading the abstracts of the articles (31 articles).
- Obtain the full-text versions of the remaining articles, discarding those for which the full text could not be accessed (25 articles).
- Each author conducted an exploratory reading of each article to assess its relevance and

contribution to the study and the research questions (25 articles reviewed).

- The authors performed a cross-validation through discussion, reaching a consensus on which articles should be selected (16 articles).

At the end of the selection process, 16 articles were chosen to proceed to the next stage.

3) Primary Studies

The selected articles were reviewed, considering the inclusion of additional articles through snowballing or opportunistic search techniques. The authors decided to use only the previously selected articles.

Finally, the authors reached a consensus and approved the list of articles for data extraction. A total of 16 primary articles were selected to proceed to the data extraction stage. The primary articles are as follows: [1], [3], [12], [15], [17], [20], [21], [23], [24], [25], [26], [27], [28], [29], [30], [31].

G. Data Extraction

1) Feature Extraction

The data extraction process involved reading the primary studies and highlighting elements that contribute to answering the research questions. The Atlas.ti tool was used for this purpose, allowing unified coding across all articles, providing traceability, and structuring semantic networks to support the writing of findings. Additionally, information on the publication type, year, and authors was collected and tabulated, enabling the organization of articles as needed. For example, Figure 2 shows the distribution of primary articles by year and publication type.

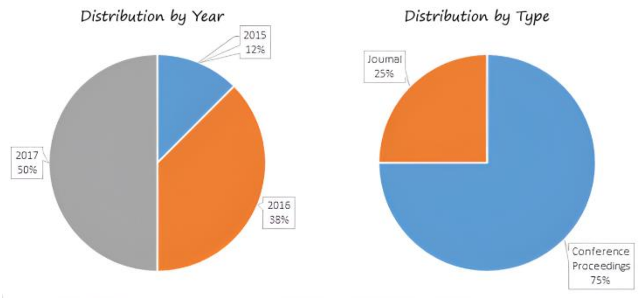

Fig. 2. Characteristics of Primary Studies

2) Model Extraction

Since one of the objectives of the study was to determine if a data modeling method exists for business data segmentation in the construction of microservices, the expectation was to find graphical representations in the primary articles outlining procedures for determining granularity in each case. This section aimed to collect visual schematics to complement the textual coding results. However, upon reviewing the primary articles, no representative graphical elements were found for this purpose. Therefore, the analysis will be based solely on the textual ideas presented in the primary studies.

III. Data Modeling Methods for Microservices

This research identifies architectural approaches for modeling the granularity of microservices to address key research challenges. The findings highlight the use of meta-modeling techniques that define microservice boundaries as adaptable entities, focusing on aspects such as business-driven design, tool heterogeneity, and decentralized governance [1]. These approaches support analysis, evolution, and localization, which are crucial for adapting microservice granularity based on quality attributes [32].

Migration to microservices, or microservitization, enhances autonomy, replaceability, and governance while improving the traceability of software architectures [33]. However, there is still a lack of consensus on the definition, properties, and modeling techniques of microservices. Effective migration involves determining optimal granularity, deployment strategies, and orchestration methods [34].

One of the main challenges is establishing the optimal granularity level, balancing microservice size and number to meet both individual and overall system requirements [1]. Microservitization involves identifying optimal service boundaries to enhance the Quality of Service (QoS) [35].

Recent trends, such as Service-Oriented-Architecture (SOA) and Microservice Architecture (MSA), have emerged as suitable approaches for cloud infrastructures [36]. MSA aims to create flexible, modular applications, but its practical implementation remains a significant research challenge. Modernization efforts involve understanding and transforming large applications into microservices, using model-driven methods to manage complexity and dependencies across business and data layers [37].

IV. Discussion

Currently, more organizations with complex business domains are moving away from monolithic software applications and adopting distributed architectures based on microservices. Microservices architecture aims for agile software development using small services that communicate via APIs, where each service implements complete business functionalities and can run independently. Microservices can be deployed across different machines, using diverse programming languages and data dependencies that the business requires, maximizing scalability and leveraging the strengths of each platform. They are designed as small, simple, and understandable executable units, which makes them easier to modify and maintain.

However, modeling the domain for each microservice with the necessary dependencies within the business context requires software architects to clearly define each service responsibilities and APIs to achieve good cohesion and low structural coupling. The architecture should facilitate parallel development with different teams working on separate microservices, allowing services to be rewritten with minimal effort if necessary. For microservices to be autonomous from development to deployment, architects need strategies for domain modeling, deployment, versioning, monitoring, security, maintenance, and managing business-driven changes.

The challenge posed by the research questions is to define the appropriate data domain modeling for visualizing the structural granularity that each microservice must implement. The research framework focuses on using software archetypes for modeling business data for each microservice, as archetypes are fundamental human mechanisms that organize, summarize, and generalize domain information. This is expected to have applications in software engineering. The framework is built upon a literature review that includes factors demonstrating the application of software archetype principles for business data modeling in software and service engineering. A case study will be conducted in Ecuadorian public and private institutions to model business data and evaluate the advantages and disadvantages of the proposed model.

For technology companies, understanding their business data is crucial, as it represents the backbone of their information systems. Therefore, software engineering employs various methodologies and techniques to address data modeling from different perspectives, from business domains to data storage. The proposed research framework will analyze systems that use data archetypes for microservices.

After conducting the study, the research questions are revisited:

- **RQ1:** What data modeling techniques can be used for business data segmentation in microservices construction?
  - It is proposed that services be stateless (do not manage a database) [24].
  - A centralized data model is proposed [25].
  - The use of decentralized NoSQL databases is proposed

    A method is proposed where only the most critical business elements are subdivided, based on the benefits of microservices [27].

- **RQ2:** How to model or establish microservice granularity?
  - Proposes a type of microservice diagram and microservice invocation diagram
  - Proposes SMART and Entice methods [23].

V. Threats to Validity

Throughout the study, continuous discussions were held regarding the procedures to follow and the potential threats to validity. Efforts were made at every step to maintain the rigor and thoroughness required for the resulting document to contribute to scientific knowledge.

A. Threats in Search String Formation and Primary Study Selection

One challenge was determining the scope of our study since data modeling for microservices is a relatively new and less explored topic in the technological field (emerging over the last three years). Different communities often use varying terminologies for the same concepts. To cover the research questions comprehensively and avoid bias, we searched for terms related to microservices and data modeling across various contexts. While this approach reduces bias, it significantly increased the search effort, necessitating a manageable scope.

B. Threats to Study Selection and Data Extraction Consistency

The formulation of research questions helped in selecting relevant studies. However, two articles that appeared highly relevant based on their abstracts could not be accessed in full-text form and were therefore excluded. Due to time constraints for tabulation and coding, it was not possible to perform a detailed semantic analysis or comprehensive reading. The primary articles were reviewed based on coverage and a focus on relevant section.

VI. Conclusions

Microservices result from breaking down application services into smaller components, with a distinctive feature of having their own database separate from other microservices, enhancing composability and deployment capabilities. Organizations transitioning to a microservices-based-architecture often start with an existing system that already has a unified data model representing the entire business dynamic. Therefore, designing a migration strategy to microservices requires segmenting this model into smaller parts.

A preliminary literature review was conducted to find articles proposing a method for determining the appropriate level of granularity for model division. No references were found on this topic. Performing this task without a theoretical framework may lead to decisions based on intuition or trial and error, increasing resource consumption and questioning the need for microservices compared to the maturity of SOA-based web services.

An empirical study is proposed using the Systematic Mapping Study (SMS) method to conduct a thorough literature review to validate the existence of methods that define reasonable granularity in microservice design.

The literature review procedure followed these steps: Structuring the Research Question, Defining Inclusion and Exclusion Criteria, Formulating the Search Strategy, Executing the Selection Process to identify primary studies, and finally extracting the required data. This process enabled the coding of relevant elements in the primary articles, leading to the synthesis, analysis of results, and answering of the research questions.

Future work involves experimentation with existing model subdivision methods for microservices, so that, once validated across different scenarios, these methods can be considered best practices in implementing this architecture.

Acknowledgment

The authors would like to express their sincere gratitude to the Salesian Polytechnic University for providing the facilities and resources essential for conducting this research. Special thanks are also extended to the National Polytechnic School for its valuable academic support and for fostering an environment that encouraged collaborative learning and innovation throughout this project.

REFERENCES

[1] S. Hassan, N. Ali and R. Bahsoon, "Microservice Ambients: An Architectural Meta-Modelling Approach for Microservice Granularity," p. 1–10, April 2017.

[2] S. D. N. V. Duy, K. Eati, C. M. Ferreira, D. Glozic, V. Gucer, M. Gupta, S. Joshi, V. Lampkin, M. martins, S. Narain and R. Vennam., "Microservices from Theory to Practice Creating Applications in IBM Bluemix Using the Microservices Approach," *Ibm,* p. 170, 2015.

[3] N. a. L. M. a. B. J. a. G. R. a. N. J. Viennot, "Synapse: A microservices architecture for heterogeneous-database web applications," *Proceedings of the 10th European Conference on Computer Systems, EuroSys 2015,* 2015.

[4] L. Marco-Ruiz, D. Moner, J. A. Maldonado, N. Kolstrup and J. G. Bellika, "Archetype-based data warehouse environment to enable the reuse of electronic health record data," *International Journal of Medical Informatics,* vol. 84, p. 702–714, September 2015.

[5] J. Thönes, "Microservices," *IEEE Software,* vol. 32, 2015.

[6] M. Yousif, "Microservices," *IEEE Cloud Computing,* vol. 3, p. 4–5, 2016.

[7] I. Nadareishvili, R. Mitra, M. McLarty and M. Amundsen, "Microservice Architecture: Aligning Principles, Practices, and Culture," 2016.

[8] I. a. H. A. a. M. R. d. S. a. S. F. Salvadori, "Publishing linked data through semantic microservices composition," *Proceedings of the 18th International Conference on Information Integration and Web-based Applications and Services - iiWAS '16,* pp. 443--452, 2016.

[9] C. a. C. S. a. N. S. B. Batini, "Conceptual Database Design: An Entity-Relationship Approach," p. 470, 1992.

[10] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering," vol. 2, p. 1051, 2007.

[11] K. Petersen, R. Feldt, S. Mujtaba and M. Mattsson, "Systematic Mapping Studies in Software Engineering.," vol. 8, p. 68–77, 2008.

[12] F. Rademacher, S. Sachweh and A. Zundorf, "Differences between model-driven development of service-oriented and microservice architecture," *Proceedings - 2017 IEEE International Conference on Software Architecture Workshops, ICSAW 2017: Side Track Proceedings,* p. 38–45, 2017.

[13] B. Mayer and R. Weinreich, "A dashboard for microservice monitoring and management," *Proceedings - 2017 IEEE International Conference on Software Architecture Workshops, ICSAW 2017: Side Track Proceedings,* p. 66–69, 2017.

[14] A. Messina, R. Rizzo, P. Storniolo, M. Tripiciano and A. Urso, "The database-is-the-service pattern for microservice architectures," vol. 9832, p. 223–233, 2016.

[15] V. D. Le, M. M. Neff, R. V. Stewart, R. Kelley, E. Fritzinger, S. M. Dascalu and F. C. Harris, "Microservice-based architecture for the NRDC," *Proceeding - 2015 IEEE International Conference on Industrial Informatics, INDIN 2015,* p. 1659–1664, 2015.

[16] J. Jenkins, G. Shipman, J. Mohd-Yusof, K. Barros, P. Carns and R. Ross, "A Case Study in Computational Caching Microservices for HPC," *2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW),* p. 1309–1316, 2017.

[17] S. Hassan and R. Bahsoon, "Microservices and their design trade-offs: A self-adaptive roadmap," *Proceedings - 2016 IEEE International Conference on Services Computing, SCC 2016,* p. 813–818, 2016.

[18] T. Thiele, T. Sommer, S. Stiehm, S. Jeschke and A. Richert, "Exploring research networks with data science: A data-driven microservice architecture for synergy detection," *Proceedings - 2016 4th International Conference on Future Internet of Things and Cloud Workshops, W-FiCloud 2016,* p. 246–251, 2016.

[19] S. Haselbock and R. Weinreich, "Decision guidance models for microservice monitoring," *Proceedings - 2017 IEEE International Conference on Software Architecture Workshops, ICSAW 2017: Side Track Proceedings,* p. 54–61, 2017.

[20] C. a. T. M. a. I. D. a. I. B. Gadea, "A reference architecture for real-time microservice API consumption," *3rd Workshop on CrossCloud Infrastructures and Platforms, CrossCloud 2016 - Colocated with EuroSys 2016,* 2016.

[21] W. Hasselbring and G. Steinacker, "Microservice architectures for scalability, agility and reliability in e-commerce," *Proceedings - 2017 IEEE International Conference on Software Architecture Workshops, ICSAW 2017: Side Track Proceedings,* p. 243–246, 2017.

[22] B. Kitchenham, "Procedures for performing systematic reviews," *Keele, UK, Keele University,* vol. 33, p. 28, 2004.

[23] P. Di Francesco, "Architecting microservices," *Proceedings - 2017 IEEE International Conference on Software Architecture Workshops, ICSAW 2017: Side Track Proceedings,* pp. 224--229, 2017.

[24] N. H. a. {. D. T. a. {. T. X. a. F. L. a. R. C. Do, "A scalable routing mechanism for stateful microservices," *Proceedings of the 2017 20th Conference on Innovations in Clouds, Internet and Networks, ICIN 2017,* pp. 72--78, 2017.

[25] C. a. C. A. a. C. K.-K. Esposito, "Challenges in Delivering Software in the Cloud as Microservices," *IEEE Cloud Computing,* vol. 3, no. 5, pp. 10--14, 2016.

[26] D. a. C. D. a. A. R. a. C. E. a. G. K. a. P. C. a. C. R. Escobar, "Towards the understanding and evolution of monolithic applications as microservices," *Proceedings of the 2016 42nd Latin American Computing Conference, CLEI 2016,* 2017.

[27] J.-P. a. T. D. Gouigoux, "From monolith to microservices: Lessons learned on an industrial migration to a web oriented architecture," *Proceedings - 2017 IEEE International Conference on Software Architecture Workshops, ICSAW 2017: Side Track Proceedings,* pp. 62--65, 2017.

[28] M. a. K. L. a. G. W. a. Z. O. Gysel, "Service cutter: A systematic approach to service decomposition," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* vol. 9846, pp. 185--200, 2016.

[29] P. a. T. Y. Kookarinrat, "Design and implementation of a decentralized message bus for microservices," *2016 13th International Joint Conference on Computer Science and Software Engineering, JCSSE 2016,* 2016.

[30] J. a. L. L. C. a. H. S. Lin, "Migrating web applications to clouds with microservice architectures," *2016 International Conference on Applied System Innovation, IEEE ICASI 2016,* 2016.

[31] H. a. K. M. a. G. S. Vural, "A systematic literature review on microservices," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* vol. 10409 LNCS, pp. 203--217, 2017.

[32] S. Rochimah, H. I. Rahmani and U. L. Yuhana, "Usability characteristic evaluation on administration module of Academic Information System using ISO/IEC 9126 quality model," 2015.

[33] M. a. J. C. a. S. T. Mikusz, "Software Business," *Lecture Notes in Business Information Processing,* vol. 210, pp. 167--173, 2015.

[34] C. Tantithamthavorn, S. McIntosh, A. E. Hassan and K. Matsumoto, "Comments on Researcher Bias: The Use of Machine Learning in Software Defect Prediction," *IEEE Transactions on Software Engineering,* vol. 42, 2016.

[35] R. Hammad, M. Odeh and Z. Khan, "Towards a Model-Based approach to evaluate the effectiveness of e-Learning," *Proceedings of the European Conference on IS Management and Evaluation, ECIME,* Vols. 2015-Janua, 2015.

[36] G. Prabu, N. Kannan, A. Kovalan and S. Thaddeus, "Software measurement linkage for CMMI implementation," *International Journal of Control Theory and Applications,* vol. 9, 2016.

[37] S. M. Castro, E. Tseytlin, O. Medvedeva, K. Mitchell, S. Visweswaran, T. Bekhuis and R. S. Jacobson, "Automated annotation and classification of BI-RADS assessment from radiology reports," *Journal of Biomedical Informatics,* vol. 69, 2017.

P. Patricio and V. Velepucha,
Data Domain Servitization for Microservices Architecture,
Latin-American Journal of Computing (LAJC), vol. 12, no. 1, 2025.

AUTHORS

Patricio Michael Paccha Angamarca

Patricio Michael Paccha A. (Loja, Ecuador, March 1, 1980). Engineer in Information Systems and Computing, Higher Fourth Level Diploma in Strategic Marketing Management, Master in software engineering and PhD student in Computing.
Skills and experience in managing innovation and software engineering projects, Data y Software Architect, DBA, Business Application Developer, Business Intelligence, Speaker in areas of technological innovation. Currently Professor of Systems Engineering and Computer Engineering at the National Polytechnic School – Ecuador.

Victor Vicente Velepucha Bonett

Víctor Vicente Velepucha Bonett (). Engineer in Information and Computing Systems from the National Polytechnic School, Master in Project Management and Doctor in Computer Science.
Lines of research linked to Computing Applied to Software Engineering, Software Creation and Management and the Organization and Properties of Software. Currently teaching in the Department of Informatics and Computer Sciences of the National Polytechnic School – Ecuador.