International Journal of Computing and Digital Systems
2025, VOL. 17, NO.1, 1-16

http://dx.doi.org/10.12785/ijcds/1571016597

Modulith Architecture: Adoption Patterns, Challenges, and Emerging Trends

Mehdi AIT SAID[1], Lahcen BELOUADDANE[1], Soukaina MIHI[1] and Abdellah EZZATI[1]

[1]*Hassan first University, Settat, Morocco*

Received 5 April 2024, Revised 29 November 2024, Accepted 5 December 2024

Abstract: Over the past year, the software architecture field has been dominated by the contrast between microservices and monolithic architectures, driven by the demand for scalable solutions for modern applications. Microservices, with their focus on modularity and independence, have become popular for large-scale systems, offering benefits like enhanced scalability and simplified maintenance. Conversely, monolithic architectures, known for their cohesive design, have been a traditional choice, favored for their simplicity in development. However, they may struggle with scalability as applications grow in complexity. Amidst this, Modulith Architecture (MDA) has emerged in recent years as a solution to the complexities of microservices and the limitations of traditional monolithic architectures. Combining the structural integrity of monolithic systems with the modularity of microservices, MDA offers a holistic approach to software design and development. This study investigates the adoption of MDA through a comprehensive analysis of 32 practitioners' insights. Our objective is to explore the motivations, challenges, and trends surrounding MDA adoption. Employing a qualitative approach through in-depth interviews, we uncover nuanced adoption patterns and identify key factors influencing practitioners' choices. Results indicate a varied adoption spectrum, with motivations ranging from simplicity to cost-effectiveness advantages. Technical challenges, including module dependencies and communication overhead, highlight the intricacies of MDA integration. Emerging trends, such as dynamic module loading, underscore the evolving practices within the field. This study contributes to a deeper understanding of MDA adoption dynamics, offering insights for both researchers and practitioners.

Keywords: Modulith, Microservices, Modular Monolith, Software Architecture, Qualitative Study, Industrial Inquiry

1. Introduction

In software architecture and engineering, the pursuit of optimal frameworks has been a perpetual quest to meet the evolving needs of modern technology. The foundational decisions underlying software architecture play a pivotal role in determining an application's scalability, maintainability, and adaptability over time [1][2]. As the software development field continues to evolve, the dichotomy between traditional Monolithic Architecture (MA) and the more contemporary Microservice Architecture (MSA) has come to the forefront of architectural discourse [3].

Historically, MA has been the cornerstone of software design, embodying a cohesive approach where all components of an application are tightly integrated into a singular unit [4][5]. This traditional methodology, while offering simplicity and ease of initial development, encounters challenges as applications scale in complexity [6]. The inherent interdependencies within monoliths can impede agility and hinder efficient resource utilization [7]. In contrast, MSA represents a paradigm shift towards modular, decentralized systems [8]. MSA advocates for breaking down applications into small, independent services, each capable of autonomous deployment and communication through well-defined APIs [9]. This approach not only addresses the shortcomings of MA but also fosters flexibility, scalability, and rapid development [10]. The granular nature of microservices allows teams to work independently on specific services, facilitating continuous delivery and reducing bottlenecks [11][12]. In recent years, MSA has emerged as the dominant force in software architecture, steering the industry towards a more agile and scalable future. The trend towards MSA has been fueled by its alignment with agile development practices and adaptability to dynamic business requirements [13][14]. Organizations across various industries have transitioned towards microservices as the architectural framework of choice, underscoring its prevalence as the architectural trend of recent years [15].

In March 2023, the tech industry was abuzz with news of Amazon's paradigm-shifting decision to transition from a distributed MSA to a monolithic application. This strategic move garnered attention for yielding substantial benefits, including increased scale, enhanced resilience, and an astonishing 90% cost reduction in Prime Video's audio/video monitoring [16]. While this shift demonstrated

E-mail address: m.aitsaid@uhp.ac.ma, lahcen.belouaddane@uhp.ac.ma , soukaina.mihi@uhp.ac.ma, abdellah.ezzati@uhp.ac.ma

the adaptability of software architecture choices, it also underscored the persistent challenges associated with MSA, particularly in terms of cost and complexity. Despite the merits of MSA, organizations have grappled with challenges related to the overheads of managing a multitude of services, intricate inter-service communication, and the associated operational expenses [17]. In response to these challenges emerges a novel architectural paradigm — the Modular Monolith Architecture or Modulith Architecture (MDA) (The term "Modulith" derives from the fusion of "Modular" and "Monolith."). Situated between the extremes of MSA and MA, MDA seeks to strike a harmonious balance, providing a pragmatic approach to architectural design. MDA embodies a synthesis of the strengths of both MSA and MA, offering a middle ground that mitigates the drawbacks of each. This architectural model advocates for modularization, where applications are divided into cohesive, independently deployable modules, each encapsulating a specific business capability. These modules communicate through well-defined interfaces, fostering ease of maintenance, scalability, and flexibility.

The prominence of MDA as a prevailing trend in the software industry is noteworthy, underscored by its increasing presence in various industry expos, conferences, and workshops throughout the past 3 years. Such as Devoxx in 2023 [18] contributed to its visibility. However, despite the growing industry traction, academic discourse on MDA remains sparse. Our primary **objective** is to bridge this gap by providing a comprehensive overview of the state of the art and practices in MDA, transcending from industry insights to academic understanding. With MDA gaining traction, we aim to present a nuanced exploration of its adoption, challenges faced by practitioners, and the emerging trends shaping its trajectory.

To achieve our objectives, we have employed an industrial inquiry **methodology**, conducting in-depth interviews with 32 practitioners actively involved in MDA across diverse organizational scales—from startups to multinational corporations. The **qualitative** nature of our study allows us to delve into the intricate details of MDA adoption, revealing insights beyond quantitative metrics. Through semi-structured interviews, we navigate the experiences, challenges, and success stories of practitioners, capturing the essence of their real-world implementations.

In our previous study investigating the adoption of MSA [19][8], the MDE emerged as a strangler strategy to adopt MSA incrementally. This study serves as a sequel to our previous research, shifting the focus to MDA as a novel architectural methodology. This research makes substantial **contributions** to both academia and industry within the software architecture domain. The identified challenges and trends from practitioners' experiences serve as invaluable insights for academics, offering a nuanced understanding of the dynamics surrounding MDA adoption. Simultaneously, practitioners can leverage these findings as practical guidance, informing strategic decisions and design considerations. Specifically, our study identifies 10 challenges and delineates 5 trends within the MDA domain. These findings are not just retrospective; they provide a forward-looking perspective that can guide future endeavors in this evolving field. For academics, these challenges represent avenues for further research, while practitioners can use them as benchmarks to refine and enhance their MDA implementations. The trends, on the other hand, offer a glimpse into the potential evolution of MDA practices, providing foresight for both academic exploration and industry adaptation.

The remainder of this paper is organized as follows: Section 2 provides background information on MDA. In Section 3, we review related work to contextualize our research. Section 4 outlines the methodology employed in this study, including the qualitative approach and data collection methods utilized. In Section 5, we present our findings and engage in a comprehensive discussion, highlighting the adoption patterns, challenges, and emerging trends surrounding MDA. Section 6 explores the implications of our research findings for both researchers and practitioners. Section 7 addresses potential threats to the validity of our study. Finally, in Section 8, we offer concluding remarks summarizing the key insights gleaned from our investigation and suggesting avenues for future research.

2. Background

MDA embodies a nuanced synthesis, strategically positioned between the traditional MA and the more contemporary MSA. This architectural paradigm introduces a unique approach to application design, offering a middle ground that combines the benefits of both MA and MSA while mitigating their respective drawbacks. As shown in Figure 1, in the MA paradigm, where the UI, business logic, and data access harmoniously coexist within a singular artifact, the simplicity of development is notable [2]. However, as the application scales in complexity, challenges emerge. The interconnected nature of components within the monolith can lead to potential bottlenecks, hindering the agility and scalability desired in larger and more intricate systems [20][21]. Furthermore, the uniform deployment environment and shared process may pose challenges in terms of resource utilization and efficient scaling, especially as the application undergoes dynamic changes over time. Conversely, in MSA, the deliberate deconstruction of the application into modular components introduces a new set of challenges and opportunities [22]. While the UI can be decomposed into micro frontends, providing a flexible user interface, the core business logic, and data access functionalities reside within individual microservices [10][8]. Each microservice operates independently, fostering scalability and flexibility [9][14]. Nevertheless, challenges arise in managing the complexity of distributed systems, defining effective communication strategies between microservices, and ensuring cohesive development across diverse technologies [23][24].

MDA navigates a middle path between MA and MSA, offering a distinctive blend of modularity and integration. In MDA, the application is logically decomposed into microservices, which, unlike MSA, are constructed as modules within the same codebase. These modules are deployed in a shared environment and executed within a unified process. Communication can occur through in-memory calls or APIs, providing flexibility akin to MA. Noteworthy is the post-deployment adaptability in MDA; stakeholders monitor the application, triggering actions such as scalability that may lead to the transformation of a module into its own microservice. This distinctive positioning of MDA allows organizations to strike a balance between the simplicity of MA and the modularity of MSA, offering a pragmatic approach to architectural design that caters to the evolving needs of modern software development[21][8].

The roots of MDA trace back to the early years of this century. While the principle of modular applications has ancient origins, the modern concept of a modular monolith gained prominence in 2018 at the GOTO conference [25]. Simon Brown's presentation, "Modular Monoliths" challenged prevailing notions around microservices, suggesting that a well-structured monolith shares common design thinking with a good MSA. This presentation, questioning the rush towards microservices, has shed light on the potential benefits of a modular monolith.

In 2019, Oliver Drotbohm made a significant contribution to the discussion on modular monoliths with his presentation at the SpringOne Platform [26]. Drotbohm underscored the practicality of a modular, monolithic application as a feasible alternative to microservices. He tackled the common problems associated with unstructured monoliths and offered solutions, focusing on package design, component structure, transactions, and the use of events for bounded context interaction. This presentation was a turning point, as the term "Modulith" began to gain recognition in the industry.

The momentum propelling MDA forward gained significant traction with notable support from influential players in the industry. Google, recognizing the potential of this approach, joined the movement by developing the Service Weaver[1]. Furthermore, frameworks like Spring[2] responded to the escalating demand for tools tailored to support this innovative architectural paradigm. The unveiling of Spring Modulith[3] not only demonstrated a commitment to empowering developers with the requisite tools for proficiently implementing and maintaining modular monoliths but also signaled a broader industry acknowledgment of the merits of MDA. This evolution underscores a dynamic response to the ever-evolving field of software architecture, offering a pragmatic middle ground that wholeheartedly embraces modularity while effectively navigating the challenges associated with both traditional monoliths and microservices.

[1]https://serviceweaver.dev
[2]https://spring.io/
[3]https://spring.io/projects/spring-modulith

3. Related Work

MDA has emerged as a noteworthy trend in software architecture, garnering increased attention from practitioners. However, there remains a notable gap between industry practices and academic exploration, with only a limited number of studies delving into the intricacies of MDA.

In a study conducted by Ruoyu and Xiaozhou in early 2024 [27], the authors set out to comprehend the definition of MDA in the industry. Their investigation focused on frameworks and cases that implement MDA employing a systematic grey literature review, the study revealed that MDA combines the advantages of both traditional monoliths and microservices. The authors identified three frameworks and four cases exemplifying the adoption of MDA. Their findings suggest that MDA serves as an alternative to microservices, offering a potential precursor before systems transition into a fully microservices-oriented architecture.

In the study published by Martin Skalický at the beginning of 2024 [28], the author delves into the prevalent architectural paradigms of MA, MSA, and MDA. Their work critically evaluates these design approaches, with a particular focus on highlighting the often-overlooked negative consequences of microservices in favor of MDA. The author draws on personal experience working on microservices projects, discussing practical challenges faced with this architectural choice. To substantiate the analysis, the author develops a proof-of-concept application for each architecture type, subjecting them to thorough performance and latency assessments. The study concludes by proposing a methodology for adopting the MDA approach in new projects, emphasizing its potential evolution throughout the application lifecycle.

Our work focused on MDA, stands out as a practitioner-centric exploration, aiming to close the gap between industry practices and academic discourse. In contrast to the first study by Ruoyu and Xiaozhou, which primarily concentrates on the definition and implementation of MDA by employing a systematic grey literature review, our work extends beyond a comparative framework. Similarly, the second study by Martin Skalický explores various architectural paradigms, emphasizing the drawbacks of microservices and advocating for MDA. Our research complements these studies by offering a real-world perspective through in-depth interviews with 32 MDA practitioners. By delving into their experiences, challenges, and emerging trends, our study not only validates the existing knowledge on MDA but significantly augments it. The qualitative methodology employed in our research contributes to a nuanced understanding of the dynamics surrounding MDA adoption in the contemporary software field, providing valuable insights for both academics and practitioners.

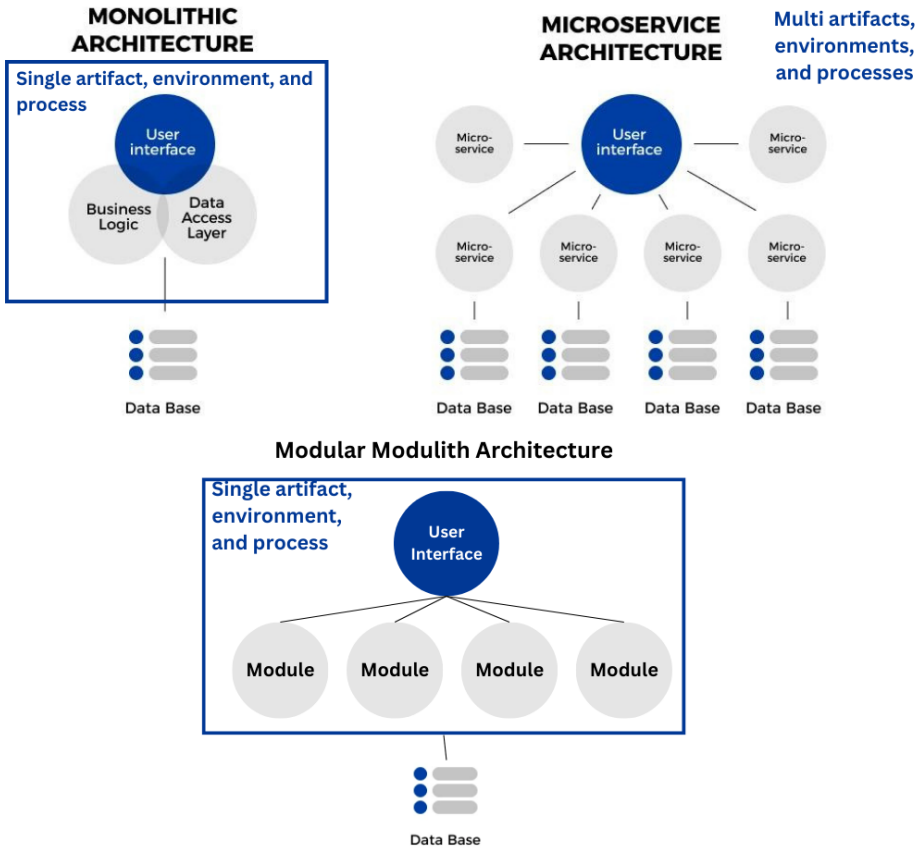

Figure 1. MA VS MSA vs MDA

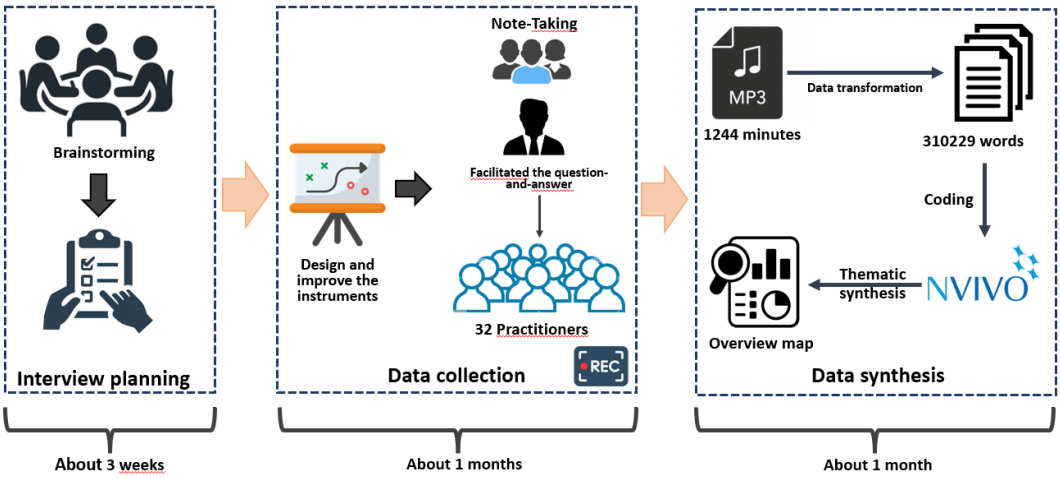

Figure 2. The procedure of the interview study

4. Methodology

A. Research design

Our study employs an industrial inquiry and qualitative research approach [29] to delve into the complexities of MDA adoption. We conducted in-depth interviews with 32 practitioners deeply involved in the MDA domain, starting in January 2024 and spanning 2 months and 3 weeks. Following a qualitative methodology outlined by Brinkmann & Kvale [30], our research methodology involved collaborative efforts among three Ph.D. student researchers and their supervisor. Together, we designed the interview framework, conducted interviews with practitioners, and performed data analysis to ensure comprehensive reporting.

This qualitative approach enabled a comprehensive examination of participants' perspectives and experiences through structured interviews [31]. The use of this methodology was intentional, recognizing its effectiveness in eliciting detailed and nuanced insights. By engaging practitioners who are actively involved in the implementation and development of MDA, the approach ensured that the data collected was rich and highly relevant. Structured interviews allowed for consistency in the questions posed while providing the flexibility needed to explore individual experiences and viewpoints in depth. This combination of structure and depth facilitated a thorough understanding of the practical realities and challenges faced by MDA practitioners.

B. Research Procedure

As illustrated in Figure 2, the research procedure for this study was meticulously crafted to ensure a systematic and thorough exploration of MDA adoption. Interviews were conducted with a strong emphasis on fostering in-depth discussions between the interviewees and interviewers. This approach involved a mix of open-ended questions designed to extract both expected and unexpected insights throughout the interview process. The procedure comprised five stages, each aimed at gathering insights from seasoned practitioners in the field.

1) **Brainstorming and Interview Plan Development:** The initial stage involved collaborative brainstorming among the four researchers to identify key areas of interest concerning MDA adoption based on the insights from our previous study [19]. This collective effort allowed for the delineation of relevant topics and themes to guide the subsequent interview process, ensuring the exploration of diverse aspects of MDA adoption and enhancing the study's comprehensiveness.
2) **Design of Interview Instrument:** Building upon the outcomes of the brainstorming sessions and drawing inspiration from the findings of our previous study [19], the researchers crafted the interview instrument. This instrument was meticulously designed to align with the identified areas of interest, aiming to extract in-depth insights from practitioners. The questions within the instrument were thoughtfully formulated to facilitate a rich and nuanced exploration of MDA adoption experiences.
3) **Practitioner Selection and Invitation:** Concurrently, the researchers identified suitable practitioners to participate in the interviews. To ensure a diverse pool of experienced practitioners, participants were selected from Linkedin, international software engineering conferences, and tech expos held in USA, Morocco, and France. These events attracted experts in the field, making them ideal platforms for practitioner recruitment.
4) **Criteria-Based Selection:** Selected practitioners possessed a substantial background in software architecture, with over five years of industry experience in software engineering especially in MSA, including at least 1 year of practical engagement with MDA. This rigorous criterion aimed to ensure that participating practitioners had ample insights and firsthand experience to contribute to the study's objectives.
5) **Interview Initiation:** The final stage commenced with the initiation of interviews with practitioners who accepted the invitation. Researchers engaged in open and exploratory conversations with selected practitioners, utilizing the interview instrument as a guide. These interviews aimed to uncover practitioners' perspectives, experiences, and insights related to MDA adoption.

C. Research Questions

Our study on MDA is driven by a set of clear objectives. Firstly, we aim to provide a comprehensive overview of the current state of the-art and practices in MDA, bridging the gap between industry applications and academic understanding. Secondly, our focus is on exploring the nuanced aspects of MDA adoption, shedding light on challenges faced by practitioners. Additionally, we strive to identify and elucidate emerging trends within the field, encompassing new tools, frameworks, and practices. To achieve this, we formulated four research questions:

1) **RQ1**: (What is the current level of adoption of MDA in the industry, and how has it evolved over time?) This question aims to provide a comprehensive understanding of the current industry adoption trends of MDA. By exploring the evolution over time, we seek to identify patterns, shifts, and any noteworthy changes in the adoption field. This information is crucial for establishing a baseline understanding of the prevalence and dynamics of MDA in the software industry.
2) **RQ2**: (In which specific use cases and scenarios are organizations choosing to implement MDA, and what types of projects or applications are most suited for this approach?) The goal here is to uncover the practical applications and scenarios where MDA is being implemented. By identifying specific use

cases and understanding the types of projects or applications that find MDA most suitable, we aim to provide insights into the contextual appropriateness of MDA. This information is valuable for practitioners considering or currently implementing MDA.

3) **RQ3**: (What are the primary technical and organizational challenges that organizations face when adopting MDA?) This question focuses on the challenges encountered during the adoption of MDA. By delving into technical and organizational aspects, the objective is to provide a nuanced understanding of the hurdles organizations confront. This information is essential for practitioners, offering insights into potential obstacles and facilitating better preparedness for successful MDA adoption.
4) **RQ4**: (Are there emerging trends or innovations within the field of MDA, such as new tools, frameworks, or practices that organizations should be aware of?) The objective here is to identify and highlight emerging trends and innovations within the MDA landscape. By exploring new tools, frameworks, or practices, we aim to inform organizations about the latest advancements in MDA. This knowledge is crucial for staying abreast of industry developments and making informed decisions when adopting or evolving MDA practices.

D. Data Collection

To gather comprehensive and contextualized insights, semi-structured interviews were employed as the primary data collection method. An interview guide was meticulously developed, comprising open-ended questions explicitly aligned with our research objectives. Each practitioner underwent an interview guided by a script consisting of 12 questions, as outlined in Table I. These questions were formulated based on the research objectives and underwent refinement through pilot testing. The interviews, conducted either in-person or remotely, were tailored to the preferences and availability of the participants. Each session lasted approximately 30-40 minutes, with audio recording conducted with explicit consent. Detailed notes were also diligently taken during the interviews to capture non-verbal cues and additional contextual information. Language preferences were accommodated, with interviews conducted in French, Arabic, or English, depending on the proficiency and preferences of the participants.

E. Data Analysis

The data synthesis process was meticulously executed to ensure a comprehensive analysis of the interview data. Initially, one researcher transcribed the interview recordings meticulously, followed by a rigorous dual verification conducted by other two researchers to maintain accuracy. To mitigate potential researcher bias, an independent coding strategy was employed. Each interview transcript was coded individually by the four researchers, contributing to a diverse and unbiased perspective in the analysis. Following Welsh's guidance [32], the data synthesis procedure utilized a combination of manual and computer-assisted techniques. The qualitative data synthesis tool, NVivo[4], was utilized to facilitate efficient pattern recognition within the extensive textual data, enhancing the precision of the synthesis. Subsequently, the data underwent thematic synthesis, following the method outlined by Patton [33]. This involved a progressive encoding process applied to the data, with researchers conducting a systematic manual analysis to identify prominent themes, concerns, and practices associated with various phases of MDA adoption.

F. Demographic Characteristics

This section provides an overview of the demographic characteristics of the interviewees, shedding light on the profiles of the participants contributing to the study. Table II summarizes the demographic attributes of the interviewees, including their domains, locations, and organization sizes. These insights enhance the robustness and relevance of the study's findings, capturing a diverse group of professionals involved in MDA adoption across various organizational contexts.

- **Professional Roles:** The interviewees encompass a range of roles within the MDA environment, including developers (13/32), project managers (4/32), software architects (9/32), and tech leads (6/32). This diverse mix reflects the broad engagement of professionals with varying levels of responsibility and expertise in the MDA field.
- **Experience and Adoption:** Each interviewed practitioner brings substantial experience to the study, with a minimum of five years in software engineering. Moreover, every participant has dedicated at least one year to adopting MDA, indicating a depth of familiarity with MDA principles, practices, and challenges.
- **Domain, Location, and Organization Size:** The demographic diversity extends to the domains of the interviewees' organizations, spanning Education (ED), Healthcare (HC), Banking and Finance (BF), E-commerce (E), Technology and Software Development (S), and Business Management (BM). Participants are located in various cities, including Casablanca, Paris, New York, and Rabat, across the USA, Morocco, and France. Additionally, the represented organizations vary in size, from local businesses with fewer than 100 employees to multinational corporations with over 10,000 personnel like IBM and Oracle, reflecting the broad spectrum of enterprises engaging with MDA.

[4]https://lumivero.com/products/nvivo/

TABLE I. The interview questions

No.	RQ	Question
Q1	RQ1	Can you provide insights into the current adoption of MDA within your organization?
Q2	RQ1	How long has your organization been utilizing MDA?
Q3	RQ1	Can you describe the history and evolution of MDA adoption within your organization?
Q4	RQ1	What were the initial reasons for adopting MDA, and have those reasons evolved over time?
Q5	RQ2	In which specific projects or applications has MDA been most effective for your organization?
Q6	RQ2	Are there particular scenarios or use cases where MDA has demonstrated clear advantages over other architectural approaches?
Q7	RQ2	Can you provide examples of projects where MDA might not be the most suitable choice, and the reasons behind those decisions?
Q8	RQ3	What technical challenges did your organization encounter during the adoption of MDA?
Q9	RQ3	How did the organizational structure and culture impact the adoption and ongoing use of MDA?
Q10	RQ3	What strategies or best practices did your organization employ to address the identified challenges during MDA adoption?
Q11	RQ4	Are there any new tools or frameworks that your organization has adopted to enhance its MDA practices?
Q12	RQ4	In your opinion, are there areas within MDA that are currently undergoing significant advancements or changes that organizations should be aware of?

TABLE II. The detailed information of the 30 practitioners

No.	Domain	Company Location	Company Size
HC1	HC	Rabat/Morocco	100-1000
ED1	ED	Casablanca/Morocco	100-1000
ED2	ED	Casablanca/Morocco	1000-5000
BF1	BF	Casablanca/Morocco	100-1000
BF2	BF	Casablanca/Morocco	<100
BF3	BF	Casablanca/Morocco	100-1000
BF4	BF	New York/USA	>10 000
E1	E	Tangier/Morocco	<100
E2	E	Casablanca/Morocco	100-1000
E3	E	Casablanca/Morocco	5000-10 000
E4	E	Casablanca/Morocco	1000-5000
E5	E	Casablanca/Morocco	<100
S1	S	Casablanca/Morocco	100-1000
S2	S	Phoenix/USA	>10 000
S3	S	Casablanca/Morocco	<100
S4	S	Casablanca/Morocco	<100
S5	S	Casablanca/Morocco	100-1000
S6	S	Lyon/France	1000-5000
S7	S	Rabat/Morocco	100-1000
S8	S	Tangier/Morocco	<100
B1	B	Casablanca/Morocco	100-1000
B2	B	Casablanca/Morocco	100-1000
B3	B	Rabat/Morocco	<100
B4	B	Casablanca/Morocco	1000-5000
B5	B	Rabat/Morocco	1000-5000
B6	B	Rabat/Morocco	100-1000
B7	B	Paris/France	1000-5000
B8	B	Paris/France	1000-5000
B9	B	Rabat/Morocco	<100
B10	B	Casablanca/Morocco	<100
B11	B	Lyon/France	<100
B12	B	New York/USA	100-1000

5. Findings and discussion

In this section, we unveil key findings from interviews with 32 MDA practitioners, shedding light on adoption patterns, practitioner challenges, and emerging trends. This concise analysis offers insights into the current state-of-the-art practices in MDA, paving the way for a focused discussion on their implications for both researchers and practitioners.

A. Adoption Over Time (RQ1)

The findings from the interview questions Q1, Q2, and Q3, addressing the current adoption, temporal utilization, and historical evolution of MDA, are succinctly visualized in Figure 3. The graphical representation provides a comprehensive overview of the nuanced perspectives shared by the 32 practitioners regarding MDA adoption within their respective organizations. Additionally, the results stemming from Q4, which delves into the initial motivations and their evolution over time, are encapsulated in Figure 4. This figure encapsulates the multifaceted factors influencing the

practitioners' decisions to adopt MDA, offering insights into the dynamic nature of these motivations.

1) Current level of MDA adoption (Q1)

The responses to the first interview question provide a nuanced perspective on the current adoption of MDA within diverse organizational contexts. Of the 32 practitioners interviewed, three practitioners cited single-time use, indicating specific project applications. Notably, the fully adoption category, highlighted by nine practitioners, was observed exclusively within startup organizations, showcasing MDA as a foundational approach for these dynamic entities. The majority, 20 practitioners, noted partial adoption, highlighting a phased integration across projects or departments. These findings illustrate the spectrum of MDA adoption, offering a comprehensive view of its varied and evolving role within organizational architectures.

2) Duration of MDA adoption (Q2)

In response to the second interview question regarding the duration of MDA utilization within organizations, diverse timelines emerged among the 32 practitioners. four organizations reported a robust adoption of more than three years, indicating a well-established use of MDA. Notably, a growing trend was observed among 11 practitioners who implemented MDA within the 2 to 3 years timeframe. The majority, encompassing 17 practitioners, reflected a surge in recent adoption within the 1 to 2 years period. This temporal distribution suggests a noteworthy increase in MDA adoption over the years.

3) Organizational adoption history (Q3)

Examining the history and evolution of MDA adoption within organizations, insights from 32 practitioners revealed diverse trajectories. three practitioners reported a single-time use, emphasizing MDA's adoption for a substantial project with over 18 months of development. A substantial majority, consisting of 28 practitioners, highlighted an increased adoption of MDA, indicating a positive trajectory over time. Notably, one practitioner reported a decreased adoption. These findings provide a nuanced understanding of the varied historical journeys of MDA adoption, with a prevailing trend of increased integration across organizations.

4) Initial motivations (Q4)

The multifaceted analysis of the initial motivations and their evolution over time provides a nuanced understanding of the factors influencing MDA adoption among the 32 practitioners. Initially, factors such as simplicity and rapid development, cost-effectiveness, easier debugging and testing, team expertise, faster time to market, resource efficiency, and the absence of evident scaling challenges played pivotal roles. As presented in Figure 4, simplicity and rapid development were noted by 13 practitioners, cost-effectiveness was universally acknowledged by all 31, and easier debugging and testing were highlighted by 15. Leverage of team expertise was crucial for 25 practitioners while achieving a faster time to market was cited by eight. Resource efficiency and the absence of evident scaling challenges influenced 18 and 29 practitioners, respectively. Regarding the evolution of these reasons over time, 21 practitioners indicated that motivations have evolved, five noted a consistent set of reasons, and six observed nuanced changes in specific projects. These findings illuminate the dynamic nature of MDA adoption motivations, reflecting a nuanced interplay of factors that shape its continued relevance and evolution within organizational contexts.

The aspect of cost-effectiveness emerged prominently across the majority of practitioners in response to this question. 31/32 practitioners mentioned cost-effectiveness as one of the initial reasons for adopting MDA. For instance, a practitioner highlighted, "Well, you see, the main key reason we jumped on board with modulith was the fantastic cost-effectiveness it promised. It's like hitting two birds with one stone – we get a modular approach that suits our needs, and it doesn't break the bank. Cost savings and a robust architecture – who wouldn't love that combo? Plus, the beauty is that we don't need to spin up more containers or VPS like we did with microservices. With modulith, we make a move when we spot a scalability issue in a specific module of the app – it's like having a scalable solution without unnecessary overhead. It's been a game-changer for us" (E1).

5) Summarize to answer RQ1

The investigation into the adoption of MDA offers insights into its current prevalence and temporal evolution within organizational contexts. The varying levels of MDA adoption depict a diverse scenario. While some organizations have utilized MDA in isolated instances (Single-time Use), others have fully integrated it across their architectural field. Notably, the fully adopted stance was predominantly noted among startup organizations. The majority, however, indicated a partial adoption, suggesting a phased approach tailored to specific projects or departments. Examining the duration of MDA adoption highlights a growing trend. Organizations adopting MDA for more than three years signify an established usage, while a considerable number reported adoption within the last two to three years, indicating recent traction. The surge in adoption within the last one to two years emphasizes the increasing popularity of MDA. The historical trajectories of MDA adoption depict an interesting pattern. Instances of single-time use were reported, indicating specific project-based adoptions. The predominant trend, however, is an increase in MDA adoption over time, with only one reported instance of decreased adoption. This points towards a sustained or growing interest in MDA within organizational practices. The motivations behind MDA adoption are diverse, ranging from simplicity and rapid development to cost-effectiveness, easier debugging, team expertise, faster time to market, resource efficiency, and the absence of evident scaling challenges. The evolution of these motivations over time is dynamic. While some organizations experience a shift

in motivations, others maintain consistency, reflecting the nuanced nature of decision-making in MDA adoption.

B. Effectiveness in projects (RQ2)

1) Effective MDA Implementations (Q5)

The examination of responses to Q5 offers valuable insights into the adoption effectiveness of MDA across various domains. When practitioners were queried about the specific projects or applications where MDA proved most effective for their organizations, a diverse array of domains emerged. As shown in Figure 5, E-Commerce stood out with 19 mentions, followed by CRM with 14 mentions, ERP with 16 mentions, Logistics with 7 mentions, Booking with 10 mentions, Healthcare with 5 mentions, and Monitoring with 21 mentions. These findings underscore the versatility and efficacy of MDA across a broad spectrum of applications and industries. From managing customer relationships to streamlining logistical operations and optimizing monitoring systems, MDA has demonstrated its adaptability and effectiveness in addressing diverse organizational needs.

2) Key scenarios and use cases (Q6)

The responses to Q6 shed light on four specific scenarios and use cases where MDA demonstrates clear advantages over other architectural approaches. Among the identified advantages:

- **Scaling Challenges Not Evident (23 practitioners)**: In scenarios where the scalability needs of a specific application module are not clearly defined during the initial stages of development, MDA proves advantageous. The modular approach allows for the development of modules without committing to an MSA until scalability requirements become evident to move a specific module to an independent microservice as mentioned by a practitioner: "...in scenarios with uncertain scalability needs during initial development... Modulith allows module development without an immediate engagement with MSA. The modular approach enables a smooth transition to an independent microservice when scalability requirements become clearer..." (S3)

- **Microservices Identification not Confirmed (12 practitioners)**: MDA offers an advantage when decomposing a system into microservices is not confirmed due to uncertainties or changes in project requirements. This flexibility allows for the postponement of microservices identification until a clearer understanding emerges, preventing premature decisions.

- **High Flexibility with Low Deployment Cost (32 practitioners)**: For projects that experience frequent changes requiring high flexibility but do not necessarily demand extensive scalability such as local business management systems that are used only by the employees of the organization, MDA excels. It provides the needed flexibility without incurring the higher deployment costs associated with a fully distributed MSA. One practitioner mentions this: "In projects where flexibility is crucial, Modulith shines. It's like having a tool that adapts seamlessly to evolving requirements without the hefty price tag of a fully distributed microservices setup. Modulith gave us the flexibility we needed, keeping costs in check..." (B1).

- **Dynamic Workloads (18 practitioners)**: MDA demonstrates advantages in scenarios with dynamic or fluctuating workloads. Its modular structure allows for adaptability to varying demands, ensuring efficient performance even in environments with unpredictable workloads.

3) Inappropriate application instances and reasons (Q7)

The responses to Q7 shed light on scenarios where MDA might not be the most suitable choice, along with the underlying reasons. Among the 32 practitioners interviewed, four considerations were highlighted:

- **Highly Distributed Systems (26 Mentions)**: A predominant concern among practitioners was the applicability of MDA in highly distributed systems. These scenarios, often characterized by a significant geographical span or intricate network structures, may pose challenges to the effective implementation of MDA.

- **Elastic Scalability Requirements (11 Mentions)**: Some practitioners raised concerns about the compatibility of MDA with projects having elastic scalability requirements. In scenarios where dynamic and rapid scalability is paramount, MDA may face limitations compared to other architectural approaches.

- **Parallel Development Needs (7 Mentions)**: The requirement for parallel development processes, where multiple teams work concurrently on different components or modules, was cited as a factor influencing the choice of architecture. In cases demanding extensive parallel development, practitioners suggested that alternative architectural approaches might be more suitable.

- **Isolation of Failures (5 Mentions)**: The need for a high degree of failure isolation, particularly in mission-critical systems, was mentioned as a consideration where MDA might not be the optimal choice. In such cases, alternative architectures offering enhanced failure isolation mechanisms were deemed more appropriate.

4) Summarize to answer RQ2

The findings from Q5, Q6, and Q7 collectively provide comprehensive insights into the suitability and effectiveness of MDA across various scenarios. Q5 reveals the diverse domains where MDA has proven most effective, including

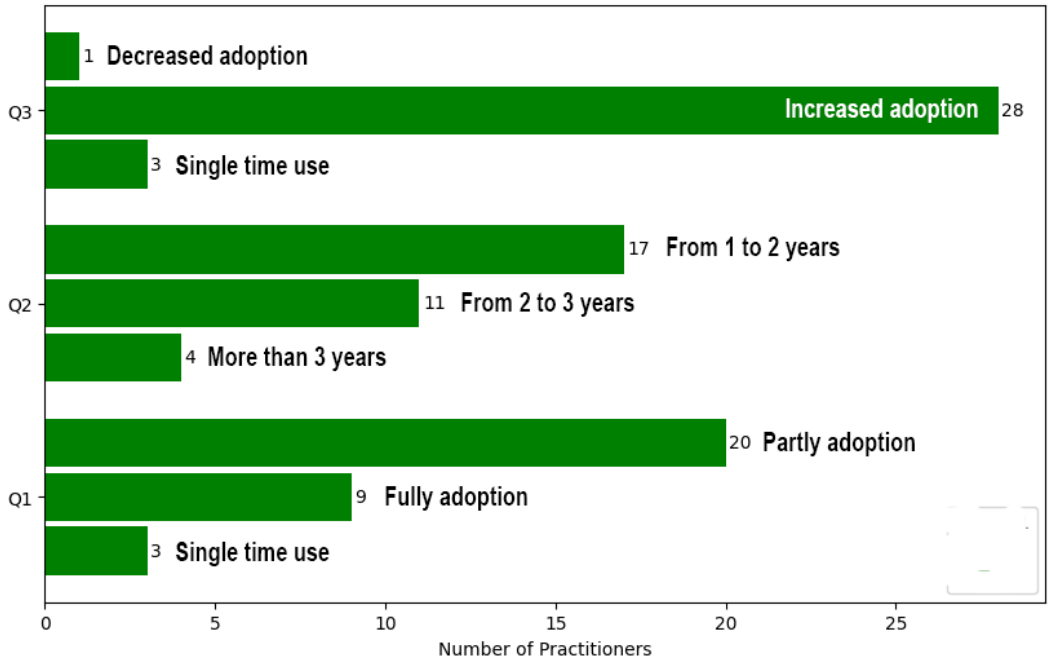

Figure 3. Adoption Levels Over Questions (Q1, Q2, Q3)

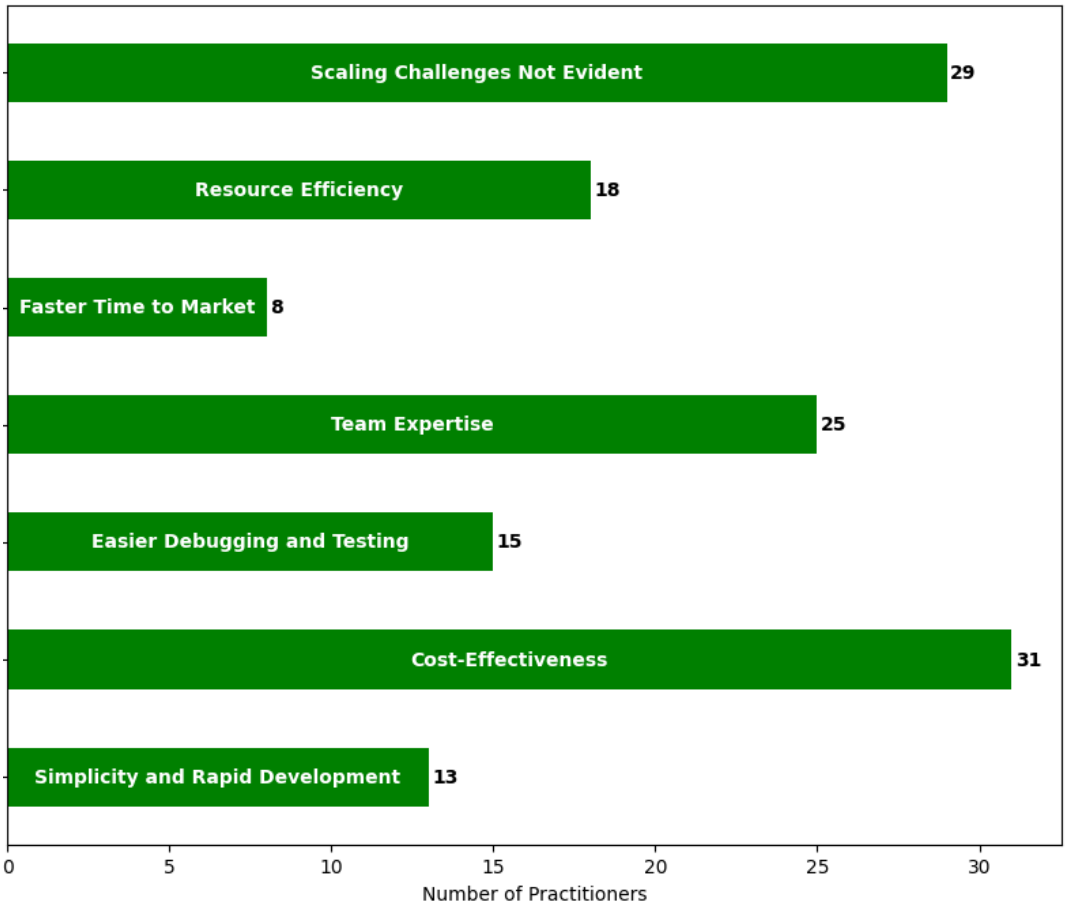

Figure 4. Motivations of MDA Adoption

E-Commerce, CRM, ERP, Logistics, Booking, Healthcare, Monitoring, and other applications. These domains exemplify the adaptability and efficacy of MDA in addressing diverse organizational needs. Q6 further delves into specific scenarios where MDA demonstrates clear advantages over other architectural approaches. The identified advantages include managing scenarios where scalability needs are uncertain, delaying microservices identification in evolving projects, providing high flexibility with low deployment costs, and accommodating dynamic workloads. Conversely, Q7 explores scenarios where MDA might not be the most suitable choice, considering factors like highly distributed systems, elastic scalability requirements, parallel development needs, and the isolation of failures. These considerations highlight the nuanced decision-making process involved in selecting the appropriate architectural approach based on project-specific requirements and constraints. The collective findings contribute to a more nuanced understanding of the diverse contexts in which MDA excels and the scenarios where alternative architectures may be more suitable. These findings suggest that MDA is particularly effective in projects with evolving needs, dynamic workloads, and scenarios where microservices identification is not confirmed. Its versatility makes it applicable across various domains, offering a modular and adaptive solution. However, organizations should carefully evaluate project-specific factors, considering the potential limitations of MDA in highly distributed systems, scenarios requiring specialized team expertise, elastic scalability needs, parallel development requirements, and those demanding a high degree of failure isolation. Overall, the study provides valuable insights into the nuanced application of MDA, offering guidance for organizations seeking an architectural approach that aligns with their specific project contexts and objectives.

C. Challenges (RQ3)

1) Technical challenges (Q8)

The examination of technical challenges encountered during the adoption of MDA, as explored in responses to Q8, reveals multifaceted issues. As presented in Figure 5, the challenges encompassed 10 facets: Module Dependencies (18 Mentions) and Module Communication Overhead (29 Mentions) were recurrent concerns, emphasizing the intricacies of managing dependencies and optimizing communication between modules. Consistency in Module Interfaces (13 Mentions) emerged as a focus on standardizing interfaces for cohesive interactions, while Module Interdependencies (9 Mentions) and Module Ownership (11 Mentions) underscored the complexities of interrelationships and ownership structures. Module Deployment Coordination (12 Mentions) highlighted challenges in deploying modules seamlessly, while Monitoring and Debugging (14 Mentions) emphasized the importance of robust tools for effective oversight. Resource Sharing and Isolation (28 Mentions) raised questions about balancing resource utilization and isolation, and DevOps Integration (27 Mentions) illuminated challenges in integrating MDA with DevOps practices. Dynamic Module Loading (28 Mentions) surfaced as a substantial challenge, demanding careful implementation. These findings collectively depict a nuanced field of technical challenges, offering valuable insights for organizations navigating the adoption of MDA. Addressing these challenges proactively is crucial for a successful and streamlined integration of modular architecture into organizational practices.

2) Organizational challenges (Q9)

The examination of responses to Q9 unveils insights into how organizational structure and culture influenced the adoption and continuous utilization of MDA. The identified aspects include:

- **Historical Resistance**: A notable number of practitioners, accounting for 14 mentions, highlighted historical resistance within their organizations as a significant factor affecting the adoption of MDA. The resistance stemmed from established practices, existing architectures, or past experiences that created reluctance among stakeholders to embrace a new architectural paradigm.
- **Hierarchy and Decision-Making**: The organizational structure and decision-making processes played a role in MDA adoption, as noted by 17 practitioners. Hierarchical structures and decision-making mechanisms influenced the speed and ease with which MDA was adopted, emphasizing the importance of aligning organizational structures with the requirements of adopting new architectural approaches.
- **Leadership Support**: The impact of leadership support on MDA adoption was mentioned by 11 practitioners. Having supportive leadership proved crucial in facilitating a smoother transition to MDA, providing the necessary resources, and fostering a culture that values innovation and experimentation.

3) Addressing the challenges (Q10)

In addressing the challenges encountered during MDA adoption, organizations have implemented a range of effective strategies and best practices. Dynamic Module Loading Strategies, acknowledged by 26 practitioners, exemplify a proactive approach, enabling on-demand loading of modules at runtime, and tools such as OSGi[5] were mentioned several times. This dynamic flexibility aids in adapting to evolving requirements and optimizing resource utilization. Resource Sharing and Isolation Guidelines, recognized by 12 practitioners, underscore the importance of clear protocols for balancing resource sharing and isolation, ensuring efficient module interactions. The adoption of Robust Monitoring and Debugging Tools, mentioned by 21 practitioners, reflects a commitment to real-time oversight and issue resolution, enhancing the overall stability and performance of

[5]https://www.osgi.org

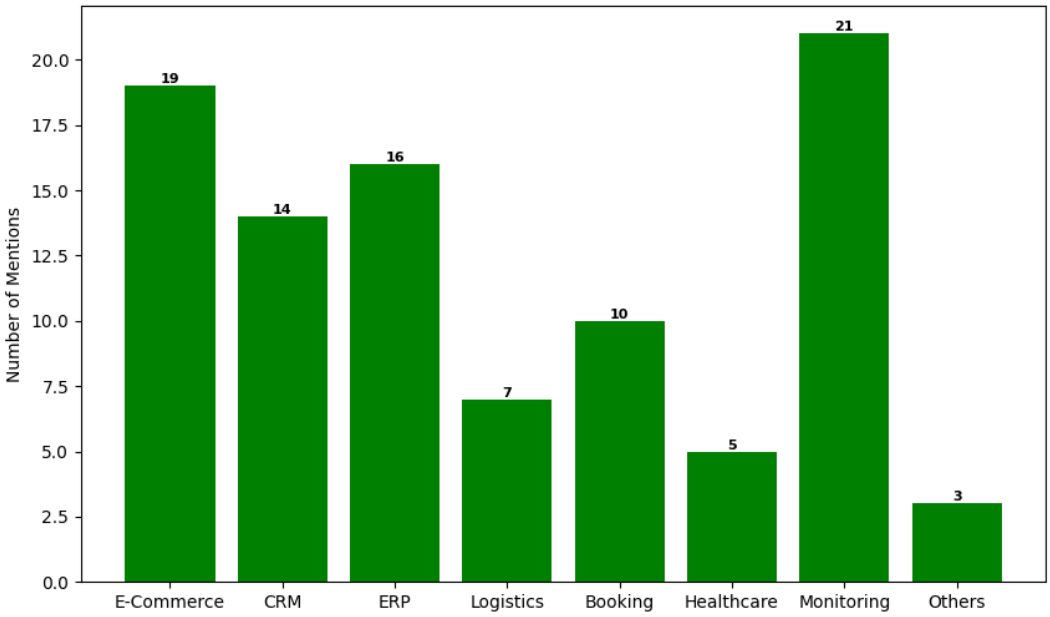

Figure 5. Effectiveness of MDA Across Different Domains

the modular system, ELK Stack (Elasticsearch[6], Logstash[7], Kibana[8]) was mentioned 15 times as monitoring framework. Deployment Coordination Protocols, highlighted by 10 practitioners, demonstrate the implementation of streamlined and collaborative deployment processes, ensuring synchronized module releases. The establishment of a Modular Ownership Framework, acknowledged by 11 practitioners, emphasizes the importance of delineating clear roles and responsibilities for each module, fostering a sense of accountability. Effective Communication Channels, noted by 13 practitioners, play a pivotal role in facilitating transparent communication among teams, enabling efficient collaboration, and problem-solving. These findings collectively illustrate the multifaceted strategies organizations employ to navigate challenges and ensure the successful adoption and implementation of MDA.

4) Summarize to answer RQ3

The adoption of MDA presents a spectrum of challenges spanning technical, organizational, and cultural dimensions. Technically, organizations grapple with intricate challenges such as managing Module Dependencies and minimizing Communication Overhead to ensure efficient module communication. The standardization of Module Interfaces poses consistency challenges while navigating Module Interdependencies and Ownership complexities requires careful consideration. Deployment Coordination demands seamless processes, and effective Monitoring and Debugging tools are essential. Striking a balance between Resource Sharing and Isolation poses a technical dilemma, and integrating MDA with DevOps practices requires meticulous attention. The dynamic nature of Module Loading introduces additional technical complexities. Organizational challenges manifest in historical resistance stemming from established practices and architectures. Decision-making structures and hierarchy influence the pace of adoption, and the crucial role of leadership support becomes apparent. The organizational structure significantly impacts the adoption and sustained use of MDA, highlighting the need for alignment with architectural shifts.

D. Emerging innovations and trends (RQ4)

1) Tools and Frameworks (Q11)

The exploration of tools and frameworks adopted to enhance MDA practices, as captured in responses to Q11, illuminates key trends among the 32 practitioners interviewed. The findings revealed a proactive embrace of new tools across four categories:

- **Micro Frontends Frameworks (22 Mentions)**: A notable trend emerged with practitioners incorporating Micro Frontends Frameworks into their MDA practices. This signifies a strategic focus on frontend development methodologies that align with the modular architecture, enabling the seamless integration of frontend components within the broader modular ecosystem.
- **Modular Development Platforms (29 Mentions)**: A significant emphasis on Modular Development Platforms was evident, with tools such as Spring

[6]https://www.elastic.co
[7]https://www.elastic.co/logstash
[8]https://www.elastic.co/kibana

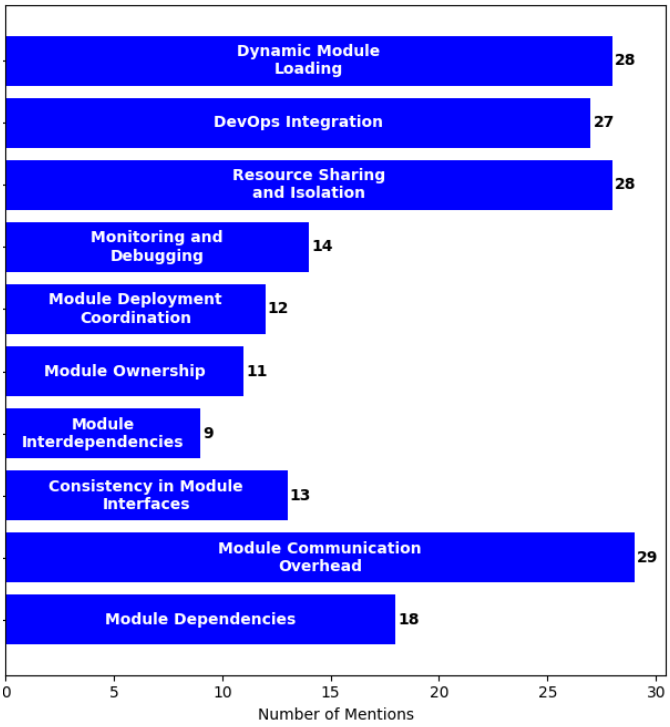

Figure 6. Technical Challenges During MDA Adoption

Modulith[9], Service Weaver[10], and Eclipse OSGi[11] being prominently mentioned. This trend underscores the practitioners' commitment to leveraging comprehensive platforms designed explicitly for modular development. These platforms facilitate the streamlined creation, deployment, and management of modular components within the MDA framework.

- **GraphQL[12] for Module Communication (27 Mentions)**: The adoption of GraphQL for Module Communication emerged as a prominent trend. The practitioners recognized the flexibility of GraphQL, especially its ease in transitioning to MSA by utilizing consistent interfaces. This indicates a strategic move to ensure compatibility and adaptability as organizational needs evolve. One practitioner mentioned this by: "... embracing GraphQL for module communication has been a real win for us in the world of MDA. when we decide to transition a module into its own microservice, we don't have to rewrite interfaces, thanks to GraphQL's awesome flexibility. It's all about keeping things smooth and future-ready in our modular playground ..." (S2).
- **Monitoring and Observability Tools (32 Mentions)**: A universal trend across all respondents was the adoption of robust Monitoring and Observability Tools such as ELK Stack[13] and Prometheus[14]. Monitoring plays a pivotal role in the decision-making process when contemplating the migration of a module to a microservice. By closely monitoring various metrics such as performance, resource utilization, and system behavior, organizations can gain valuable insights into the health and efficiency of their software modules. These metrics provide vital information on factors like scalability, fault tolerance, and overall system stability. With comprehensive monitoring in place, organizations can make informed decisions regarding the optimal timing and strategy for transitioning a module to a microservice, ensuring seamless integration and maximizing the benefits of the new architecture.

2) *Emerging trends (Q12)*

The insights garnered from practitioners in response to Q12 reveals key areas within MDA that are witnessing notable advancements and changes (Figure 7). Module Loading, cited by 25 practitioners, signifies a dynamic evolution, emphasizing the continuous development of strategies for loading modules at runtime to enhance adaptability and flexibility. Declarative Module Configuration, acknowledged by 10 practitioners, indicates a focus on simplifying and automating module configuration processes, reflecting ongoing efforts to streamline and enhance module setup. Security Measures for Modules, mentioned by 15 practitioners, highlight a heightened emphasis on bolstering security protocols within modular architectures, showcasing a response to the evolving threat landscape and the importance of securing individual modules. Cross-language modularization, recognized by 18 practitioners, suggests a significant trend toward facilitating seamless integration and interaction between modules developed in different programming languages. Finally, Automated Module Lifecycle Management, noted by 21 practitioners, underscores the increasing automation of processes related to module lifecycle management, indicating a drive toward efficiency and minimizing manual intervention. These emerging trends collectively underscore the dynamic nature of MDA, with advancements in dynamic loading, configuration, security, cross-language compatibility, and lifecycle management, emphasizing areas where organizations should stay informed and adapt to evolving best practices.

[9]https://spring.io/projects/spring-modulith
[10]https://serviceweaver.dev
[11]https://www.osgi.org
[12]https://graphql.org
[13]https://www.elastic.co/elastic-stack
[14]https://prometheus.io

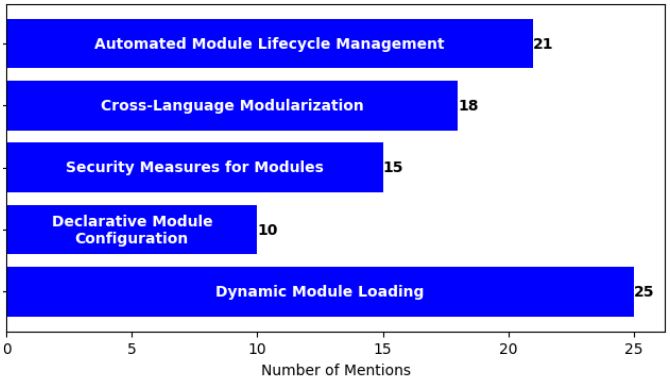

Figure 7. Adoption Trends in Tools and Frameworks for MDA Practices

3) Summarize to answer RQ4

The analysis of responses to Q11 and Q12 reveals compelling insights into emerging trends and innovations within MDA. Organizations are strategically adopting Micro Frontends Frameworks to modularize front-end development, fostering flexibility and maintainability. Concurrently, the emphasis on robust Modular Development Platforms, including tools like Spring Modulith[15], Service Weaver[16], and Eclipse OSGi[17], underscores the growing importance of comprehensive support structures for modular architecture. GraphQL[18] for Module Communication has gained prominence, showcasing a trend toward standardized and flexible interfaces, facilitating a seamless transition to MSA. Moreover, the widespread adoption of Monitoring and Observability Tools signals a commitment to real-time insights and proactive issue resolution. In terms of emerging practices, dynamic Module Loading is a notable trend, offering organizations on-demand loading capabilities at runtime for enhanced flexibility and resource efficiency. The shift toward Declarative Module Configuration reflects a desire for clear and expressive configuration methodologies, reducing complexity in specifying module configurations. Security Measures for Modules are increasingly prioritized, highlighting the critical role of security in modular architectures. Cross-language modularization practices indicate a move toward language-agnostic module development, promoting interoperability and flexibility. Finally, Automated Module Lifecycle Management is gaining traction, streamlining deployment, updates, and retirement processes to reduce manual effort and minimize the risk of errors. These emerging trends collectively depict a dynamic field, showcasing how organizations are adapting their practices and adopting innovative tools to optimize modular architectures. Staying abreast of these trends is crucial for organizations seeking to enhance their modular development processes and navigate the evolving field of software architecture effectively.

[15]https://spring.io/projects/spring-modulith
[16]https://serviceweaver.dev
[17]https://www.osgi.org
[18]https://graphql.org/

6. Implications

A. Implications to researchers

The findings from this study hold several implications to researchers exploring MDA. The identified trends in MDA, including Dynamic Module Loading, Declarative Module Configuration, Security Measures for Modules, Cross-Language Modularization, and Automated Module Lifecycle Management, present fertile ground for further investigation. Researchers should delve deeper into these emerging areas, exploring the intricacies, benefits, and potential challenges associated with each trend. Examining the impact of these trends on system performance, scalability, and maintainability will contribute valuable insights to the evolving landscape of modular architectures. Moreover, the study uncovered a range of technical challenges encountered during MDA adoption, such as Module Dependencies, Module Communication Overhead, Consistency in Module Interfaces, and more. Researchers are encouraged to focus on addressing these challenges and proposing innovative solutions. Investigating the root causes of these technical hurdles, developing best practices to mitigate them, and assessing their impact on overall system performance will enrich the scholarly discourse on modular architectures. In addition, organizational challenges like Historical Resistance, Hierarchy and Decision-Making, and Leadership Support point to the crucial intersection of technology and organizational culture. Researchers should explore methodologies to navigate organizational challenges during MDA adoption, considering factors like leadership strategies and cultural alignment. Understanding how these organizational dynamics influence the success of modular architecture implementation will contribute valuable insights to the field. As researchers explore both the trends and challenges identified in this study, a comprehensive understanding of MDA's current state and potential future directions will emerge. By addressing these challenges and reporting on trends, researchers can contribute to the refinement and advancement of modular architecture practices, providing a solid foundation for the broader community of practitioners and organizations.

1) Implications for practitioners

For practitioners contemplating the adoption of MDA, understanding the relevant practices and challenges is crucial for informed decision-making and successful implementation. The identified practices shed light on key strategies that organizations might find useful in adopting MDA, while the challenges highlight potential obstacles that practitioners may encounter along the way. Dynamic Module Loading Strategies offer a flexible approach to module management, enabling modules to be loaded dynamically at runtime, thereby enhancing system adaptability and scalability. Resource Sharing and Isolation Guidelines ensure efficient resource allocation and prevent resource conflicts among modules, promoting optimal system performance and stability. Robust Monitoring and Debugging Tools are essential for the migration of modules to microservices. Deployment Coordination Protocols streamline the deployment

process by establishing clear guidelines and procedures, reducing the risk of deployment errors and ensuring system reliability. A Modular Ownership Framework clarifies module responsibilities and ownership, promoting accountability and fostering a clear understanding of module interactions within the organization. Effective Communication Channels facilitate transparent communication and collaboration among teams, enhancing coordination and alignment across modules and stakeholders. However, practitioners should also be aware of the technical and organizational challenges associated with MDA adoption. Technical challenges such as Module Dependencies, Module Communication Overhead, Consistency in Module Interfaces, and Module Interdependencies may pose hurdles to seamless integration and operation of modular architectures. Additionally, challenges related to Module Ownership, Module Deployment Coordination, and Monitoring and Debugging require careful consideration and proactive management to mitigate risks and ensure smooth operation. Organizational challenges, including Historical Resistance, Hierarchy and Decision-Making, and Leadership Support, can impact the adoption process and organizational readiness for MDA. Addressing historical resistance by fostering a culture of openness to change, aligning decision-making structures with modular principles, and securing leadership support are critical steps in overcoming organizational barriers to MDA adoption.

7. Validity Threats

A. Trustworthiness and Validity

To enhance the trustworthiness and validity of the study, multiple strategies were implemented. Member checking was conducted by sharing the preliminary findings with a subset of participants, validating the accuracy of the interpretation of their responses and ensuring that their perspectives were faithfully represented. Furthermore, the research team actively participated in continuous discussions and engaged in reflexive memo writing, facilitating reflection on individual biases and preconceptions. This approach aimed to enhance transparency and rigor throughout the analysis process. The involvement of multiple researchers and consistent team meetings played a crucial role in upholding the credibility and dependability of the study's findings.

B. Ethical Considerations

The study adhered to ethical guidelines at every stage. All participants provided informed consent, with an assurance of the confidentiality and anonymity of their responses. Ethical standards for research involving human participants were strictly followed, ensuring that no personal or sensitive information was disclosed without explicit consent.

C. Limitations

It is crucial to recognize specific limitations inherent in the study. The findings derive from a relatively modest sample size of 32 MDA practitioners, potentially constraining the generalizability of the results. Nevertheless, the emphasis was placed on acquiring comprehensive and detailed insights from participants with diverse backgrounds and experiences. Additionally, as is characteristic of any qualitative study, there exists the potential for researcher bias and subjectivity in data interpretation. To address this concern, the research team implemented reflexivity and engaged in peer debriefing, fostering critical reflection on their assumptions and perspectives throughout the research process.

8. Conclusion

This study provides a comprehensive overview of MDA adoption trends and challenges. The findings showcase diverse adoption patterns, emphasizing the adaptability of MDA across various organizational contexts. Motivations for adoption, such as simplicity and cost-effectiveness, evolve over time, reflecting the dynamic nature of organizational needs. Effectiveness in specific domains, challenges faced, and emerging trends, including tools like GraphQL and dynamic module loading, are identified. The study highlights the evolving landscape of MDA practices and its constant alignment with industry needs. While our study offers valuable insights, future research can delve deeper into sector-specific challenges, long-term impacts, and the interplay between MDA adoption and emerging technologies. Investigating the evolving trends and the interaction with technologies like containerization could uncover new possibilities. A longitudinal study and exploration of organizational culture's role in MDA success present avenues for further investigation, contributing to the continuous evolution of modular architecture practices.

References

[1] J. Christian, Steven, A. Kurniawan, and M. S. Anggreainy, "Analyzing microservices and monolithic systems: key factors in architecture, development, and operations," 9 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10331155

[2] K. Gos and W. Zabierowski, "The comparison of microservice and monolithic architecture," pp. 150–153, 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9109514

[3] B. B. De Andrade, S. L. D. Santos, and A. R. Silva, *A comparison of static and dynamic analysis to identify microservices in monolith systems*, 1 2023.

[4] N. C. Mendonça, C. Box, C. Manolache, and L. Ryan, "The monolith strikes back: Why istio migrated from microservices to a monolithic architecture," pp. 17–22, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9520758

[5] D. Gravanis, G. Kakarontzas, and V. C. Gerogiannis, "You don't need a Microservices Architecture (yet)," *European Symposium on Software Engineering*, 11 2021. [Online]. Available: https://doi.org/10.1145/3501774.3501780

[6] G. Blinowski, A. Ojdowska, and A. Przybyłek, "Monolithic vs. microservice architecture: A performance and scalability evaluation," pp. 20 357–20 374, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9717259

[7] A. Goli, O. Hajihassani, H. Khazaei, O. Ardakanian, M. Rashidi, and T. Dauphinee, "Migrating from Monolithic to Serverless: A

FinTech Case Study," *Companion of the ACM/SPEC International Conference on Performance Engineering*, 4 2020. [Online]. Available: https://doi.org/10.1145/3375555.3384380

[8] M. Ait Said, A. Ezzati, and S. Arezki, "Microservice-Specific language, a step to the Low-Code platforms," pp. 817–828, 1 2023. [Online]. Available: https://doi.org/10.1007/978-3-031-26384-2_72

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

[10] M. Ait Said, A. Ezzati, and S. Arezki, "Microservices, a Step from the Low-Code to the No-Code," pp. 779–788, 11 2022. [Online]. Available: https://doi.org/10.1007/978-3-031-20601-6_64

[11] S. Newman, *Building Microservices: Designing Fine-Grained Systems*, 2 2015, iSBN: 1491950358, URL: https://www.oreilly.com/library/view/building-microservices/9781491950340.

[12] L. Qian, J. Li, X. He, R. Gu, J. Shao, and Y. Lu, "Microservice extraction using graph deep clustering based on dual view fusion," *Information Software Technology*, vol. 158, p. 107171, 6 2023. [Online]. Available: https://doi.org/10.1016/j.infsof.2023.107171

[13] X. Zhou, S. Li, L. Cao, H. Zhang, Z. Jia, C. Zhong, Z. Shan, and M. A. Babar, "Revisiting the practices and pains of microservice architecture in reality: An industrial inquiry," *Journal of Systems and Software*, vol. 195, p. 111521, 1 2023. [Online]. Available: https://doi.org/10.1016/j.jss.2022.111521

[14] M. Ait Said, A. Ezzati, S. Mihi, and L. Belouaddane, "Enhancing Reusability in Microservice Architecture," *4th International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, pp. 1–7, 5 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10549318

[15] R. M. Munaf, J. Ahmed, F. Khakwani, and T. Rana, "Microservices Architecture: Challenges and Proposed Conceptual Design," *International Conference on Communication Technologies*, 3 2019. [Online]. Available: https://doi.org/10.1109/comtech.2019.8737831

[16] M. Kolny, "Scaling up the Prime Video audio/video monitoring service and reducing costs by 90

[17] S. Li, H. Zhang, Z. Jia, C. Zhong, C. Zhang, Z. Shan, J. Shen, and M. A. Babar, "Understanding and addressing quality attributes of microservices architecture: A Systematic literature review," *Information Software Technology*, vol. 131, p. 106449, 3 2021. [Online]. Available: https://doi.org/10.1016/j.infsof.2020.106449

[18] O. Drotbohm, "Devoxx Talk: Spring Modulith – Spring for the Architecturally Curious Developer from Devoxx — Class Central," 2 2024. [Online]. Available: http://bit.ly/49ewlLH

[19] M. AIT SAID, A. EZZATI, S. MIHI, and L. BELOUADDANE, "Microservices Adoption: An Industrial Inquiry into Factors Influencing Decisions and Implementation Strategies," *International Journal of Computing and Digital Systems*, vol. 15, pp. 1417–1432, 3 2024. [Online]. Available: http://dx.doi.org/10.12785/ijcds/1501100

[20] R. Laigner, M. Kalinowski, P. Diniz, L. Barros, C. Cassino, M. Lemos, D. Arruda, S. Lifschitz, and Y. Zhou, "From a monolithic big data system to a microservices event-driven architecture," pp. 213–220, 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9226286

[21] L. Belouaddane, M. Ait Said, A. Marzouk, and A. Benmakhlouf, "Microservice Architecture DevOps Integration Challenges: A Qualitative Study," *Lecture notes in networks and systems*, pp. 96–108, 1 2024. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-031-70924-1_8

[22] G. Lv, B. Huang, Z. Liang, M. Qin, H. Zou, and Z. Li, "Microservices: architecture, container, and challenges," *IEEE 20th International Conference on Software Quality, Reliability and Security Companion*, 12 2020. [Online]. Available: https://doi.org/10.1109/qrs-c51114.2020.00107

[23] P. Di Francesco, I. Malavolta, and P. Lago, "Research on Architecting Microservices: Trends, Focus, and Potential for Industrial Adoption," *IEEE International Conference on Software Architecture*, 4 2017. [Online]. Available: https://doi.org/10.1109/icsa.2017.24

[24] P. Di Francesco, P. Lago, and I. Malavolta, "Architecting with microservices: A systematic mapping study," *The Journal of systems and software*, vol. 150, pp. 77–97, 4 2019. [Online]. Available: https://doi.org/10.1016/j.jss.2019.01.001

[25] S. Brown, "Modular monoliths," 11 2018. [Online]. Available: https://gotober.com/2018/sessions/515/modular-monoliths

[26] O. Drotbohm, "Building Better Monoliths: Implementing Modulithic Applications with Spring," 1 2019. [Online]. Available: https://www.infoq.com/presentations/monoliths-modules/

[27] R. Su and X. Li, "Modular monolith: Is this the trend in software architecture?" 1 2024. [Online]. Available: https://arxiv.org/abs/2401.11867

[28] V. Martin, Skalický Tomáš and M. Jiří, "Analysis and comparison of application architecture: monolith, microservices and modular approach," 2 2024. [Online]. Available: https://dspace.cvut.cz/handle/10467/113769

[29] J. W. Creswell and C. Poth, *Qualitative Inquiry and research design : choosing among five approaches*, 4 2017, iSBN: 9781506361178, URL: https://openlibrary.org/books/OL28633749M/Qualitative_Inquiry_and_Research_Design.

[30] S. Brinkmann and S. Kvale, *Doing interviews*, 1 2018, iSBN: 9781529716665, DOI: https://doi.org/10.4135/9781529716665, URL: https://www.sagepub.com/hi/nam/doing-interviews/book259212.

[31] D. Silverman, *Doing qualitative research*. Sage Publications Limited, 12 2021, iSBN: 9781529769012, URL: https://uk.sagepub.com/en-gb/eur/doing-qualitative-research/book276570.

[32] E. Welsh, "Dealing with Data: Using NVivo in the Qualitative Data Analysis Process," *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, vol. 3, no. 2, 5 2002. [Online]. Available: https://www.mendeley.com/catalogue/bab4fb29-4c2a-31a2-bf74-52f708ea0cbc/

[33] M. Q. Patton, "Qualitative evaluation and research methods, 2nd ed." *Sage Publications, Inc*, 1 1990. [Online]. Available: https://psycnet.apa.org/record/1990-97369-000