UDC 621 DOI https://doi.org/10.35546/kntu2078-4481.2024.2.34

O. Y. CHAPLIA
Postgraduate Student at the Department of Specialized Computer Systems
Lviv Polytechnic National University
ORCID: 0009-0005-9298-3538

H. I. KLYM
Doctor of Technical Sciences, Professor,
Professor at the Department of Specialized Computer Systems
Lviv Polytechnic National University
ORCID: 0000-0001-9927-0649

MICROSERVICE ARCHITECTURE FOR CYBER-PHYSICAL SYSTEMS

Cyber-Physical Systems connect the physical and digital worlds. They are composed of hardware connected to the physical world, software, and potentially other types of systems. They are utilized across various industries, including robotics, healthcare, smart cities, automotive, industry, and space. These systems are very complex to design and implement. Cloud computing technologies provide an excellent environment for the Cyber-Physical Systems software and tools for maintaining and scaling the infrastructure. One of the main challenges is how to design cloud systems properly for Cyber-Physical Systems. Currently, microservice architecture is broadly used for software in the cloud. At its core, Microservices provide flexibility, availability, scalability, and independence of modules, as well as agile development and deployment processes. These advantages are well-aligned with the needs of Cyber-Physical Systems.

However, many challenges still exist in implementing a microservice architecture for Cyber-Physical Systems. The challenges include complex distributed system networking, real-time data processing, microservice software architecture, microservice availability, and reliability of the system components. This paper provides a study whose primary goal is to give the main microservice architectural principles and patterns used, summarize the advantages and challenges, and improve the knowledge of the microservice architecture used for Cyber-Physical Systems. At first, a literature review of modern research papers was conducted. Then, each paper was analyzed. A summary of all selected research papers was produced. The results and conclusion sections deliver the summaries and give future research directions.

Key words: *cloud computing, cyber-physical systems, Industry 4.0, Internet of Things, microservices.*

О. Ю. ЧАПЛЯ
аспірант кафедри спеціалізованих комп'ютерних систем
Національний університет «Львівська політехніка»
ORCID: 0009-0005-9298-3538

Г. І. КЛИМ
доктор технічних наук, професор,
професор кафедри спеціалізованих комп'ютерних систем
Національний університет «Львівська політехніка»
ORCID: 0000-0001-9927-0649

МІКРОСЕРВІСНА АРХІТЕКТУРА ДЛЯ КІБЕРФІЗИЧНИХ СИСТЕМ

Кіберфізичні системи з'єднують фізичний і цифровий світи. Вони складаються з апаратного забезпечення, пов'язаного з фізичним світом, програмного забезпечення та, можливо, інших типів систем. Вони використовуються в різних галузях, включаючи робототехніку, охорону здоров'я, розумні міста, автомобілебудування, промисловість і космос. Ці системи дуже складні в проектуванні та реалізації. Технології хмарних обчислень забезпечують чудове середовище для програмного забезпечення та інструментів кіберфізичних систем для підтримки та масштабування інфраструктури. Одна з головних проблем полягає в тому, як правильно проектувати хмарні системи для кіберфізичних систем. В даний час мікросервісна архітектура широко використовується для програмного забезпечення в хмарі. За своєю суттю мікросервіси забезпечують гнучкість, доступність, масштабованість і незалежність модулів, а також гнучкі процеси розробки та розгортання. Ці переваги добре узгоджуються з потребами кіберфізичних систем.

Однак все ще існує багато проблем у впровадженні мікросервісної архітектури для кіберфізичних систем. Виклики включають складну розподілену мережу системи, обробку даних у реальному часі, мікросервісну архітектуру програмного забезпечення, доступність мікросервісів та надійність компонентів системи. У цій статті представлено дослідження, основною метою якого є наведення основних принципів та шаблонів проєктування мікросервісної архітектури, що використовуються, узагальнення переваг та проблем, а також покращення знань про мікросервісну архітектуру, що використовується для кіберфізичних систем. Спочатку був проведений

огляд літератури сучасних наукових робіт. Потім була проаналізована кожна робота. Підготовлено резюме всіх вибраних наукових робіт. Розділи з результатами та висновками містять підсумки та дають подальші напрямки досліджень.

Ключові слова: *хмарні обчислення, кіберфізичні системи, індустрія 4.0, інтернет речей, мікросервіси.*

Introduction

Combining digital technologies with physical processes has created a complex and dynamic domain called cyber-physical systems (CPS) [1]. These systems bring together computation, networking, and physical processes. Within these systems, embedded computers, networks, and cloud computing systems monitor and control the physical processes [1]. Feedback loops enable the physical processes to impact computations and vice versa [1]. CPS includes industries such as autonomous vehicles, healthcare, robotics, industrial automation, green technologies, smart cities, and space technology [1]. Cyber-physical Systems often operate on a scale that demands dynamic resource allocation and reallocation [1].

Cyber-physical systems (CPS) have been extensively researched, focusing on integrating physical and computational processes [1]. However, the combination and use of Microservice Architecture (MSA) for CPS is still being explored [2]. While CPS is significantly explored in the context of embedded devices, physical processes, real-time data processing, and system responsiveness, these insights have not been fully applied to the development and optimization of MSA [2].

Cyber-physical systems (CPS) and Microservice Architecture (MSA) share similar architectural pros and challenges. MSA decomposes applications into more minor, independent services [2][3]. Each service can be developed, deployed, and scaled independently [2][3]. By aligning the principles of CPS and MSA, industries can leverage each other's strengths to develop more efficient systems [1][2]. It's a paradigm shift that promises to effectively address the challenges of building complex, scalable, and adaptable software systems [4]. Adopting DevOps practices is also crucial in microservices and is very usable for building CPS [6]. DevOps practices enable rapid development and deployment cycles, allowing for swift adaptation without extensive downtime or overhauling the entire system [6]. Testing CPS in another exciting direction that provides better validation of the results [7].

Existing MSA architectures, design patterns, and technologies evolve from the design and development of cloud systems, for example, for banking, finance, healthcare, streaming, automotive, and entertainment [2; 3]. However, CPS provides an additional layer of complexity, real-time or near-real time processing, big data, and chaotic physical processes [1]. More analysis, testing, and validation are necessary to adapt the microservice architecture for CPS [2].

This paper aims to analyze modern approaches to microservice architecture that are applied to CPS systems to find challenges, drawbacks, and places for improvements. This paper reviews research papers about CPSs that use microservice architecture. Then, challenges, drawbacks, pros, and cons are defined. A summary is presented based on the microservice architecture analysis of different papers. The summary is the basis for the further study of the microservice patterns and approaches used in CPS systems. After analyzing the results, a plan for further improvements is defined.

Research questions

This chapter outlines the key research questions for this study on Microservice Architecture (MSA) in Cyber-Physical Systems (CPS). The paper delves into the present conditions, challenges, uses, and future possibilities of MSA in CPS, focusing on microservices' design patterns and qualities. A literature search and review provide a foundation for this research. The first research question answers what microservice resilience, reliability, and availability architectural patterns are used in Cyber-Physical Systems (RQ1). The second research question addresses the gaps, challenges, and drawbacks of the presented microservice architectures for CPS by completing a thorough analysis of the selected papers (RQ2). The final result of this paper is a summary of reviewed research papers, as well as defined patterns and drawbacks of MSA for CPS.

Materials and methods

This research paper was conducted to answer the research questions related to integrating Microservice Architecture (MSA) within Cyber-Physical Systems (CPS). The research questions were formulated to provide a structured and comprehensive approach to understanding MSA's current landscape, challenges, applications, and prospects in the context of CPS.

The literature search used electronic databases, including Google Scholar, ACM Digital Library IEEE Xplore, ScienceDirect, and Springer Link. Only available and open-access research papers are selected for this research. The search was limited to articles published between 2022 and 2024 and only included articles in English.

The inclusion criteria for the articles were that they should be peer-reviewed, provide relevant insights into the research questions, and be published in high-quality academic journals or conference proceedings. The exclusion criteria were articles irrelevant to the research questions, articles not peer-reviewed, or articles not published in high-quality academic journals or conference proceedings.

Exclusion criteria were defined to ensure a finite number of results. Excluded items are short papers with less than four pages, papers without available full-text, papers not published in English, published before 2022, not peer-reviewed, and duplicated papers.

The articles that met the criteria were reviewed, and the relevant information was extracted. The extracted data was then analyzed and synthesized to answer the research questions.

The data analysis was conducted using a thematic analysis approach. The extracted information was grouped based on the research questions and analyzed to identify key themes and patterns.

The results of the data analysis were presented in a narrative format, with each research question addressed in a separate section. The literature review also included a discussion of the study's limitations and the implications for future research.

Overall, this literature review research paper provides valuable insights into the current state, challenges, applications, and prospects of Microservice Architecture in the context of Cyber-Physical Systems.

The search terms used are: "Cyber-Physical Systems," "Microservices," "Microservice Architecture," "Resilience," "Reliability," and "Availability." Logical combinations of search terms are defined by "AND" and "OR."

The first 20 research papers provided for each combination of the search keywords are taken into review to have a finite number of literature. When the same research paper appears in another digital library, it is excluded from the literature review to prevent duplicates. Fig. 1 represents the research paper selection and review process algorithm.

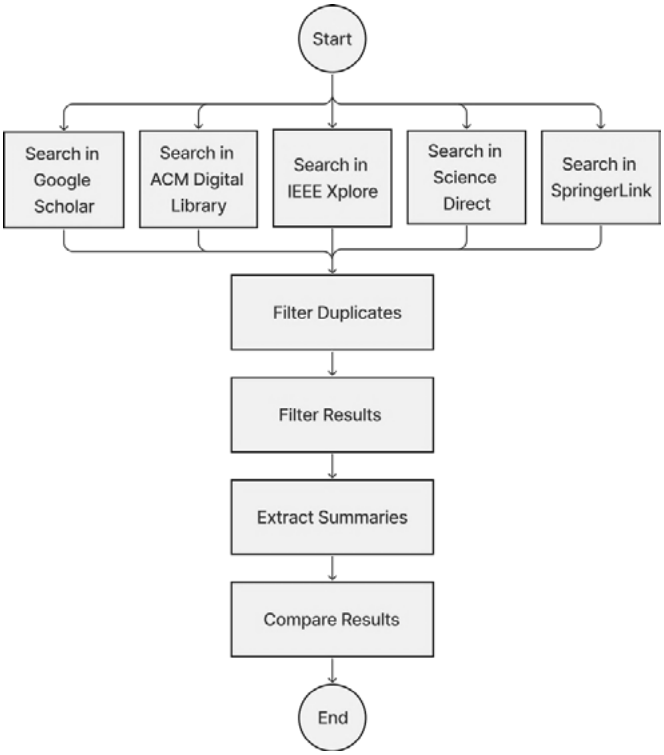

Fig. 1. Literature review algorithm

Literature review

This chapter presents the research paper selection process and paper review based on the methodology described previously.

For this review, 40 unique research papers published from 2022 to 2024 were initially selected based on the search terms provided before. These research papers are about Cyber-Physical Systems, which use microservices as an architectural pattern for implementing improved methods and approaches described by the authors. Then, each research paper was reviewed, and some were filtered according to the relevance criteria. As defined previously, the requirements include the necessity of a microservice architecture for Cyber-Physical Systems and the focus or mention on resilience, reliability, and availability.

The results of the review of selected papers are provided in the following chapters. Because of the similarity of the approaches, a detailed review of other documents is not included. Then, the pros, cons, gaps, and conclusions were defined and provided for this literature review.

The applications of microservice architecture for cyber-physical systems

This chapter provides summaries and results of selected research papers for review according to the defined methodology and answers the first research question (RQ1).

The first paper, titled "A microservice-based framework for multi-level testing of cyber-physical systems" [7] by Aldalur et al., explores the adoption of microservice architectures in the development, maintenance, and testing of Cyber-Physical Systems (CPSs), mainly focusing on their integration within the Internet of Things (IoT) domains.

Aldalur et al. propose a microservice-based testing framework to facilitate multi-level CPS testing (e.g., SiL, HiL, Operation). This framework integrates with a DevOps ecosystem, enabling continuous deployment, monitoring, and validation of CPS. It leverages microservices for validation orchestrators and agents to facilitate continuous deployment, monitoring, and validation, enabling efficient multi-level testing. This setup enhances system resilience and reliability by supporting scalable and adaptable testing processes.

Specific Microservice Architecture (MSA) patterns like dynamic configuration, service discovery, or circuit breakers aren't explicitly mentioned. Instead, the focus is on the operational methodology of microservices within the framework, highlighting their roles in improving testing efficiency, reducing costs, and underscoring the benefits of microservices in enhancing system testing scenarios. Docker was used as the leading technology for deploying microservices. The microservice code uses REST API as the primary communication protocol.

The study "Towards high-availability cyber-physical systems using a microservice architecture" by Mena et al. (2023) [4] examines enhancing cyber-physical systems (CPS) for high availability using microservices, focusing on the Digital Dice framework. The framework utilizes the Web of Things (WoT) to define interaction. It implements resilience, reliability, and availability strategies, including efficient communication with IoT devices, scalable microservice deployment, and robust internal and external communication protocols. The paper compares IoT, WoT, and Digital Dice for reliability properties. The paper states that high fault tolerance is achieved by the implementation of Digital Dice and its isolation. Also, high recoverability is achieved due to saving snapshots. The authors of this paper provided a comprehensive and profound description of the reliability and availability properties of microservices and provided a comparison between different systems and approaches. Docker and Kubernetes were used as container technology. Replication was one of the main approaches to resilience. Also, as described in the paper, Digital Dice uses service mesh Istio and Envoy proxy to solve communication problems. Circuit Breaker and Load Balancer patterns are used to handle possible issues with requests. More patterns may be implemented under the hood of Istio and Envoy.

The paper concludes that the Digital Dice framework significantly enhances the management and efficiency of CPS through improved scalability, resilience, and testing efficiency.

The paper "An Integrated Scalable Framework for Cloud and IoT-Based Green Healthcare System" explores the creation of a scalable, cloud, and IoT-based framework to enhance healthcare systems with a focus on sustainability and efficiency [8]. It introduces an innovative approach to healthcare, where wearable sensors and hierarchical clustering algorithms are utilized for real-time health data collection and analysis. This approach aims to improve patient care and facilitate doctor-patient interaction through an interactive user interface.

The paper emphasizes scalability as the main property of the proposed approach design with an existing foundation in cloud technology. The paper states that disaster recovery, backup, and auto-scaling solutions are implemented without describing internal details. A load balancer, servers, containers, a database and storage layer, and monitoring tools are also included in the primary design of the proposed system. The document does not directly mention specific microservice patterns such as Circuit Breakers, Retries, Limiters, and others. Instead, the paper focuses on the architectural and operational strategies leveraging IoT and cloud technologies.

The paper "Investigating Data Risk Considerations in Emergent Cyber-Physical Production Systems" by Ward and Janczewski (2022) delves into the complexities and risks associated with the integration of Cyber-Physical Production Systems (CPPS) within the Industrial Internet of Things (IIoT) framework [9]. This research highlights the necessity for asset managers to evaluate risks across multiple domains due to the interconnected nature of CPPS, where raw materials, machines, and operations form a tightly integrated network.

The research explicitly addresses connections to microservices in CPPS by examining the role of technologies such as Cloud, Fog, and Mist in facilitating flexible manufacturing automation hierarchies. A key focus is on employing containerized microservices to support CPPS's adaptability and resilience, indicating a move towards a more modular and scalable approach to system architecture.

The paper does not explicitly detail specific MSA patterns implemented within the proposed framework regarding microservice resilience, reliability, and availability. However, it does state the use of containerized microservices for CPPM. This paper presents informative and deep research regarding security issues.

The paper "A Survey on Observability of Distributed Edge & Container-Based Microservices" by Usman et al. (2022) presents a comprehensive review of the state-of-the-art observability for distributed systems, mainly focusing on edge computing and microservices [10]. While this work is not directly connected to CPS, it covers topics tightly related – IoT and IIoT.

Key points related to microservices include exploring how edge computing is a technical enabler for emerging network technologies such as 5G and the Industrial Internet of Things (IIoT).

Topics on microservice resilience, reliability, and availability highlighted in the paper revolve around the observability of distributed edge and container-based microservices using Docker and Kubernetes, among others. The authors describe that observability and monitoring are primary tools to know what happened during or before the outage and then execute root cause analysis. Health checks, metrics, logs, tracing, events, and checking dependencies are critical metrics for monitoring the system. Then, the issue may be fixed. The shift towards observability and monitoring also lies in the cloud infrastructure and the microservice management tools. Therefore, resiliency and reliability are stirred among the cloud infrastructure and the microservices.

The paper "On Evaluating Self-Adaptive and Self-Healing Systems using Chaos Engineering" by Naqvi et al. (2022) proposes CHESS, a systematic approach for evaluating self-adaptive and self-healing systems based on chaos engineering principles [11]. This method addresses the need for systematic evaluation methods for such systems, especially those dealing with unanticipated failures in critical and highly dynamic environments. The approach involves subjecting a system to unexpected conditions to assess its resilience and fault-tolerance capabilities. Crucial aspects of microservices include exploring chaos engineering to build resilient microservice architectures and cyber-physical systems. Self-healing and self-adaptive systems are challenging to implement and manage. Various approaches can provide resilience, including static, reactive, or dynamic solutions, or inspired by control engineering, bio-inspired algorithms, or AI. This paper also provides a profound overview of the chaos engineering approach, self-healing, self-recovery, and failure scenarios in microservices. Containers are also mentioned as a valuable tool to manage microservices. The paper mentions health checks, auto-scaling, and multiple replicas regarding specific patterns.

The paper "Osmotic Cloud-Edge Intelligence for IoT-Based Cyber-Physical Systems" by Loseto et al. (2022) investigates the integration of IoT technologies with cloud-edge computing to enhance the capabilities of the Cyber-Physical Systems (CPS) [12]. The research introduces an "osmotic computing" model that leverages microservices to enable dynamic resource allocation between cloud and edge layers, aiming to optimize CPS performance, resilience, and scalability. This model addresses the challenges associated with data volume, velocity, and the computational demands of IoT devices in CPS by facilitating efficient data processing and decision-making closer to the data sources.

The paper proposes a Cloud-Edge AI microservices approach that includes containerized architecture and microservice encapsulation of each architectural module. Considering the location and distribution context, microservices are automatically adapted to deployment sites. Additionally, the orchestrator binds the runtime of each microservice to its reference location. Dynamic service orchestration based on a feedback loop is achieved.

The orchestrator component is mentioned to manage microservice containers. It schedules migration from Edge to Cloud and vice versa. It can also reassign containers in case of Edge node failures. The prototype implemented in this paper describes the optimistic approach when the infrastructure is available. Scaling the workload is suggested to overcome possible availability issues.

This paper provides a detailed description and summary of its proposed approach based on osmotic computing principles.

The paper "Data Twin-Driven Cyber-Physical Factory for Smart Manufacturing" by Jwo et al. (2022) introduces a novel concept termed "Data Twin" and a deployable service known as the Data Twin Service (DTS) to support simulation in manufacturing, particularly in the aerospace and defense industries [13]. This concept aims to simplify the creation of high-fidelity virtual models in digital twin technologies by adopting machine learning approaches. The main focus is developing a microservice software architecture for a Cyber-Physical Factory (CPF) that simulates the shop floor environment, leveraging the DTS to manage and integrate actual and simulated data for enhanced manufacturing processes.

Regarding microservices, the CPF architecture embodies a microservice approach by facilitating the deployment and interaction of various services (DTSs) within a containerized environment. This paper also describes containerized solutions as a way of organizing and repairing services after failures. Additional information regarding patterns and recovery approaches was not mentioned in this paper.

The paper "Containerized Edge Architecture for Manufacturing Data Analysis in Cyber-Physical Production Systems" by Garcia et al. (2022) focuses on developing a microservice-based containerized edge architecture aimed at simplifying the integration of asynchronous job management for data analysis within manufacturing lines [14]. This approach addresses the complexity barriers posed by the integration of Cyber-Physical Production Systems (CPPS) into real-world manufacturing scenarios, particularly emphasizing the challenges faced by small and medium-sized enterprises (SMEs) lacking in Information Technology (IT) expertise.

The architecture proposed is designed to support a common task in CPPS: the asynchronous management of manufacturing data analysis jobs utilizing a containerized, microservice-based structure. This architecture is validated through a real computer vision quality inspection task, highlighting its practical applicability in the industry for tasks requiring data analysis, such as quality inspections using computer vision technologies.

The paper elaborates on designing and deploying a containerized, micro-service-based edge architecture. It addresses Docker containers as a solution to package and deploy microservices. The proposed system does not mention specific reliability microservice patterns.

The paper "Towards Digital Twin-enabled DevOps for CPS Providing Architecture-Based Service Adaptation & Verification at Runtime" by Dobaj et al. (2022) investigates the application of DevOps principles, traditionally utilized in IT, to the domain of Cyber-Physical Systems (CPS) with a focus on Industrial Product-Service Systems (IPSS) [15]. The main objective is to enhance CPS service delivery and adaptation to evolving needs and environments through Digital Twins (DTs). This aims to reduce design and operational uncertainties, thus ensuring IPSS integrity and availability, particularly for design and service adaptations at runtime.

The DevOps lifecycle practices are applied to reduce provider risks and design uncertainties. The self-adaptive CPS model maps the Information Technology to the concept of the Operational Technology domain of CPS IPSS. The paper describes deployment approaches with downtime and zero-downtime approaches (Blue-Green, Canary, A/B testing, shadow deployment approaches). Load balancers are decoupled from the system and manage the traffic to individual services. The load balancer ensures service availability during deployment (i.e., zero downtime). Shadow deployments allocate and deploy all resources alongside the current release. The A/B testing allocates only resources required for 30 % of the overall user traffic. Then, user requests are split and processed by the current and new service releases. The main focus in the context of microservices and DevOps in this paper is that redundancy and shadow deployments provide no downtime while deploying new components.

Summary of the reviewed papers

This chapter summarizes the review of selected papers described before and answers the first research question (RQ2).

Table 1 on microservice approaches across various papers highlights the diversity and commonalities in implementing microservice architectures for cyber-physical systems (CPS) and related domains.

The commonalities across the papers on microservice approaches for Cyber-Physical Systems and related areas highlight a strong preference for containerization technologies and patterns to enhance scalability, resilience, and deployment efficiency. Containers, prominently featuring Docker, emerge as a foundational element in developing microservice architectures, offering an efficient way to package, deploy, and manage applications across various environments [16]. Kubernetes is frequently mentioned as a critical orchestration tool, facilitating the management of containerized applications at scale, indicating its pivotal role in handling complex deployments and ensuring high availability [16]. Additionally, REST APIs are noted for enabling communication between microservices, underscoring the importance of standardized interfaces for service interaction.

Several papers introduce specific microservice patterns and tools such as Digital Dice, Snapshots, Replication, Istio, Envoy, Circuit Breaker, and Load Balancer, pointing to a broader ecosystem of technologies aimed at enhancing system reliability, monitoring, and network communication. The mention of disaster recovery, backup, auto-scaling, and health checks across various studies further emphasizes the focus on system resilience and the ability to maintain service continuity in the face of failures or demand fluctuations, but no specific implementation details were provided.

Moreover, deployment strategies like Blue-Green, Canary, A/B testing, and shadow deployment are explored in the context of enabling continuous integration and delivery (CI/CD) within the DevOps framework, highlighting the move towards more dynamic and adaptive CPS. These commonalities reflect a collective movement towards leveraging microservice architectures and container technologies to address the challenges of developing, deploying, and managing CPS and IoT systems in a more agile, reliable, and scalable manner.

As a result, a common tendency for all microservice approaches is based on DevOps practices and containers (Docker, Kubernetes). Cloud providers like Amazon AWS, Microsoft Azure, IBM Cloud, and others also provide their own Service-Level Agreement (SLA) that covers the reliability, resilience, and availability of cloud systems, mainly hardware and virtualization software, provided to the user [17]. The more SLAs are available, the more expensive they are [17].

Service Mesh Istio, also mentioned before, is an open-source project that provides a uniform way to connect, manage, and secure microservices [18]. It integrates with Kubernetes but can be used with other environments, too. Istio simplifies the configuration and operation of microservices networks, offering critical features like traffic management, service identity and security, policy enforcement, and observability across your services. By deploying a lightweight proxy alongside your services, Istio enables advanced routing, load balancing, and secure service-to-service communication without requiring changes to the service code.

Envoy Proxy is an open-source edge and service proxy designed for cloud-native applications [19]. It operates as a sidecar to mediate and manage all inbound and outbound traffic for network services. It offers advanced features such as dynamic service discovery, load balancing, TLS termination, HTTP/2 and gRPC support, observability through detailed metrics, and logging, making it versatile for handling microservices communications efficiently. Envoy is designed to be extensible and is used in conjunction with service mesh implementations like Istio to provide a comprehensive networking solution for microservices architectures.

Load balancing and auto-scaling approaches help to achieve some reliability, but their primary purpose is to balance the network load and provide scalability for performance improvements [20].

Deployment strategies like Blue-Green, Canary, A/B testing, and shadow deployment are essential methodologies in the continuous delivery pipeline to manage and mitigate risks associated with releasing new software versions, ensuring high system availability and user satisfaction [21].

Blue-Green Deployment switches traffic between two identical environments after testing the new version in the inactive one, enabling quick rollback and minimal downtime. Canary Deployment slowly introduces changes to a small user group, testing the latest version's stability before a full rollout. A/B Testing compares versions by splitting traffic to determine the better performer based on specific criteria, focusing on user preference. Shadow Deployment duplicates traffic to a new version without affecting users, allowing observation under natural conditions. Each strategy reduces software update risks, maintains service continuity, and facilitates quick rollback if necessary.

Table 1

Summary of the Reviewed Papers

Title	Microservice Approaches
"A microservice-based framework for multi-level testing of cyber-physical systems"	The focus is on the operational methodology of microservices; Docker; REST API;
"Towards high-availability cyber-physical systems using a microservice architecture"	Containers; Docker; Kubernetes; Digital Dice; Snapshots; Replication; Istio; Envoy; Circuit Breaker; Load Balancer;
"An Integrated Scalable Framework for Cloud and IoT-Based Green Healthcare System"	Containers; disaster recovery, backup, and auto-scaling, load balancer, monitoring tools included, but no description provided;
"Investigating Data Risk Considerations in Emergent Cyber-Physical Production Systems"	Containers;
"A Survey on Observability of Distributed Edge & Container-Based Microservices"	Containers; Docker; Kubernetes;
"On Evaluating Self-Adaptive and Self-Healing Systems using Chaos Engineering"	Containers; chaos engineering; health checks, auto-scaling; replication;
"Osmotic Cloud-Edge Intelligence for IoT-Based Cyber-Physical Systems"	Containers; orchestrator;
"Data Twin-Driven Cyber-Physical Factory for Smart Manufacturing"	Containers;
"Containerized Edge Architecture for Manufacturing Data Analysis in Cyber-Physical Production Systems"	Containers; Docker;
"Towards Digital Twin-enabled DevOps for CPS Providing Architecture-Based Service Adaptation & Verification at Runtime"	Containers; Docker; Blue-Green, Canary, A/B testing, shadow deployment;

Conclusions

The combination of CPS and MSA complements each other because of similarities in the design. Although pros exist, some challenges also arise within the system's complexity. System architecture, complex connections between CPS and MSA, development process, maintenance, CI/CD, security, reliability, and availability are some of the main challenges. CPS needs to operate in real-time or with precise timing, which increases the complexity of supportive systems like microservices. Achieving and maintaining high availability is very important in these cases.

This study examined the MSA used for CPS. It reviewed design patterns for the cloud system and their reliability and availability. A literature review summary provided a foundation for a deeper understanding of MSA architecture and patterns. Gaps and challenges in MSA were detected.

Containerization like Docker and Kubernetes are used very often. DevOps practices with CI/CD pipelines support continuous development and deployment. Extensive attention is given to cloud system availability. Reliability and resilience patterns are widely used to overcome failure events. The failure events that can occur within a microservice architecture often. These failures can range from cloud provider and infrastructure failures to specific issues such as OS system-level failures, microservice failures, and dependency problems. The study has demonstrated that although developers may not have complete control over external cloud failures, they can significantly influence the configuration of microservices, the management of cloud resources, and the implementation of resilience patterns that can dramatically improve the system's overall reliability and availability. Patterns like health checks, auto-scaling, replication, retry, response caching, and load balancing can help strengthen the system against potential failures. Health check provides knowledge about service uptime for system monitoring. Auto-scaling improves system performance and availability by adding more service instances managed by a cloud provider but at an additional cost. Retry patterns can eliminate outage time by allowing multiple attempts at executing operations. Response Caching can provide immediate fallback responses to ensure uninterrupted service, even during backend failures. Load Balancing can distribute traffic and computational load evenly across service instances, preventing outages due to resource overutilization and enhancing overall system performance and availability. The summary of reviewed papers emphasizes the importance of adopting a proactive, multi-faceted system design that anticipates and mitigates potential failures at every level of the service stack.

Future directions

Integrating Microservice Architecture (MSA) into Cyber-Physical Systems (CPS) presents a promising yet complex frontier, blending digital innovation with real-world applications. This union notably transforms industries by enabling smarter, faster, and more resilient systems. A prime example is the synergy between MSA and IoT [22], significantly

benefiting sectors like robotics, autonomous driving, and industrial automation [23] by reducing latency and facilitating real-time decision-making.

Further, incorporating AI and Machine Learning (ML) into MSA-equipped CPS paves the way for systems capable of predictive analytics and autonomous operation. This can be seen in smart cities, where traffic management systems learn and adapt to traffic flow patterns in real-time, improving congestion and safety [24].

Security within these systems, especially as they underpin critical infrastructure, is a crucial area of focus [25]. Future developments aim to enhance security through secure service communication and decentralized models like blockchain, which can offer new levels of integrity and trust in distributed environments [26].

In essence, MSA's journey in CPS is about leveraging digital transformation to create systems that are efficient, scalable, and deeply integrated with the physical world, driving innovations across various sectors such as transportation, manufacturing, and urban planning [27]. The path forward involves continuous exploration, adaptation, and collaboration.

References

1. Tyagi, A. K., Sreenath N., (2021). Cyber Physical Systems: Analyses, challenges and possible solutions, Internet of Things and Cyber-Physical Systems, vol. 1, pp. 22–33. https://doi.org/10.1016/j.iotcps.2021.12.002.

2. Serôdio, C., Mestre, P., Cabral, J., Gomes, M., Branco, F., (2024). Software and Architecture Orchestration for Process Control in Industry 4.0 Enabled by Cyber-Physical Systems Technologies, Applied Sciences, vol. 14, no. 5, Art. no. 5. DOI: https://doi.org/10.3390/app14052160.

3. Pontarolli, R. P., Bigheti, J. A., De Sá L. B. R., Godoy, E. P., (2023). Microservice-Oriented Architecture for Industry 4.0, Eng, vol. 4, no. 2, pp. 1179–1197, DOI: https://doi.org/10.3390/eng4020069.

4. Blinowski, G., Ojdowska, A., & Przybylek, A. (2022). Monolithic vs. Microservice Architecture: A performance and scalability evaluation. IEEE Access, 10, 20357–20374. https://doi.org/10.1109/access.2022.3152803.

5. Mena, M., Criado, J., Iribarne, L., Corral, A., Chbeir, R., Manolopoulos, Y., (2023). "Towards high-availability cyber-physical systems using a microservice architecture," Computing, vol. 105, no. 8, pp. 1745–1768. https://doi.org/10.1007/s00607-023-01165-x.

6. Fritzsch, J. et al., (2023). "Adopting microservices and DevOps in the cyber-physical systems domain: A rapid review and case study," Softw Pract Exp, vol. 53, no. 3, pp. 790–810. https://doi.org/10.1002/spe.3169.

7. Aldalur, I., Arrieta, A., Agirre, A., Sagardui, G., Arratibel, M., (2024). "A microservice-based framework for multi-level testing of cyber-physical systems," Software Qual J, vol. 32, no. 1, pp. 193–223. https://doi.org/10.1007/s11219-023-09639-z.

8. Islam, Md. M., Bhuiyan, Z. A., (2023). "An Integrated Scalable Framework for Cloud and IoT Based Green Healthcare System," IEEE Access, vol. 11, pp. 22266–22282, 2023. https://doi.org/10.1109/ACCESS.2023.3250849.

9. Ward, G., Janczewski, L., (2022). "Investigating Data Risk Considerations in Emergent Cyber Physical Production Systems," JSCI, vol. 20, no. 2, pp. 51–62. https://doi.org/10.54808/JSCI.20.02.51.

10. Usman, M., Ferlin, S., Brunstrom, A., Taheri, J., (2022). "A Survey on Observability of Distributed Edge & Container-Based Microservices," IEEE Access, vol. 10, pp. 86904–86919. https://doi.org/10.1109/ACCESS.2022.3193102.

11. Naqvi, M. A., Malik, S., Astekin, M., Moonen, L., (2022). "On Evaluating Self-Adaptive and Self-Healing Systems using Chaos Engineering," in 2022 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS), pp. 1–10. https://doi.org/10.1109/ACSOS55765.2022.00018.

12. Loseto, G. et al., (2022). "Osmotic Cloud-Edge Intelligence for IoT-Based Cyber-Physical Systems," Sensors, vol. 22, no. 6, Art. no. 6. https://doi.org/10.3390/s22062166.

13. Jwo, J.-S., Lee, C.-H., Lin, C.-S., (2022). "Data Twin-Driven Cyber-Physical Factory for Smart Manufacturing," Sensors, vol. 22, no. 8, p. 2821. https://doi.org/10.3390/s22082821.

14. Garcia, A., Franco, J., Sáez, F., Sánchez, J. R., Bruse, J. L., (2022). "Containerized edge architecture for manufacturing data analysis in Cyber-Physical Production Systems," Procedia Computer Science, vol. 204, pp. 378–384. https://doi.org/10.1016/j.procs.2022.08.046.

15. Dobaj, J., Riel, A., Krug, T., Seidl, M., Macher, G., Egretzberger, M., (2022). "Towards digital twin-enabled DevOps for CPS providing architecture-based service adaptation & verification at runtime," in Proceedings of the 17th Symposium on Software Engineering for Adaptive and Self-Managing Systems, Pittsburgh Pennsylvania: ACM, pp. 132–143. https://doi.org/10.1145/3524844.3528057.

16. Muzumdar, P., Bhosale, A., Basyal, G. P., Kurian, G., (2024). "Navigating the Docker Ecosystem: A Comprehensive Taxonomy and Survey," AJRCoS, vol. 17, no. 1, pp. 42–61. https://doi.org/10.9734/ajrcos/2024/v17i1411.

17. Bernal, A., Cambronero, M. E., Núñez, A., Cañizares, P. C., Valero, V., (2022). "Evaluating cloud interactions with costs and SLAs," J Supercomput, vol. 78, no. 6, pp. 7529–7555. https://doi.org/10.1007/s11227-021-04197-2.

18. Elkhatib, Y. Poyato, J. P., (2023). "An Evaluation of Service Mesh Frameworks for Edge Systems," in Proceedings of the 6th International Workshop on Edge Systems, Analytics and Networking, in EdgeSys '23. New York, NY, USA: Association for Computing Machinery, pp. 19–24. https://doi.org/10.1145/3578354.3592867.

19. Aslanpour, M. S., Toosi, A. N., Cheema, M. A., Chhetri, M. B., Salehi, M. A., (2024). "Load balancing for heterogeneous serverless edge computing: A performance-driven and empirical approach," Future Generation Computer Systems, vol. 154, pp. 266–280. https://doi.org/10.1016/j.future.2024.01.020.

20. Boor, M. V., Borst, S. C., Van Leeuwaarden, J. S. H., Mukherjee, D., (2022). "Scalable Load Balancing in Networked Systems: A Survey of Recent Advances," SIAM Rev., vol. 64, no. 3, pp. 554–622. https://doi.org/10.1137/20M1323746.

21. Giamattei, L. et al., (2024). "Monitoring tools for DevOps and microservices: A systematic grey literature review," Journal of Systems and Software, vol. 208, p. 111906. https://doi.org/10.1016/j.jss.2023.111906.

22. Guo, X. et al., (2021). "Towards scalable, secure, and smart mission-critical IoT systems: review and vision," in Proceedings of the 2021 International Conference on Embedded Software, Virtual Event: ACM, pp. 1–10. https://doi.org/10.1145/3477244.3477624.

23. Eze, C., (2024). "Internet of Things Meets Robotics: A Survey of Cloud-based Robots." arXiv, Feb. 20, 2024. https://doi.org/10.48550/arXiv.2306.02586

24. Al-Doghman, F., Moustafa, N., Khalil, I., Sohrabi, N., Tari, Z., Zomaya, A. Y., (2023). "AI-Enabled Secure Microservices in Edge Computing: Opportunities and Challenges," IEEE Trans. Serv. Comput., vol. 16, no. 2, pp. 1485–1504. https://doi.org/10.1109/TSC.2022.3155447.

25. Aldea, C. L., Bocu, R., Vasilescu, A., (2023). "Relevant Cybersecurity Aspects of IoT Microservices Architectures Deployed over Next-Generation Mobile Networks," Sensors, vol. 23, no. 1, 2023. https://doi.org/10.3390/s23010189.

26. Alshudukhi, K. S., Khemakhem, M. A., Eassa, F. E., Jambi, K. M., (2023). "An Interoperable Blockchain Security Frameworks Based on Microservices and Smart Contract in IoT Environment," Electronics (Switzerland), vol. 12, no. 3. https://doi.org/10.3390/electronics12030776.

27. Ali, Z. et al., (2023). "A Generic Internet of Things (IoT) Middleware for Smart City Applications," Sustainability (Switzerland), vol. 15, no. 1. https://doi.org/10.3390/su15010743.