

<https://doi.org/10.31891/2307-5732-2026-363-81>

УДК 004.75

KRAVCHUK OLHA

Khmelnitsky National University

<https://orcid.org/0000-0001-6937-5001>

e-mail: kravchukoa2@gmail.com

REVIEW AND ANALYSIS OF DATA PROCESSING INFORMATION TECHNOLOGIES IN THE COURSE OF MODERN COMPUTER SCIENCE

The article provides a comprehensive analysis of modern architectural approaches to software development using cloud technologies in the context of growing demands for scalability, fault tolerance, and security of information systems. It substantiates the relevance of the transition from traditional local deployment models to cloud environments that provide flexible resource management, elastic scaling, and cost optimization. It considers the evolution of cloud service delivery models (IaaS, PaaS, SaaS) and determines their impact on the formation of architectural solutions. Key architectural paradigms are analyzed, including monolithic, microservice, containerized, and serverless architectures. They are compared in terms of scalability, performance, fault tolerance, management complexity, and information security. Particular attention is paid to the role of containerization, resource orchestration, deployment automation, and the implementation of DevOps practices as important components of the modern cloud ecosystem.

The paper proposes a generalized model for evaluating architectural approaches, which allows comparing technical characteristics with practical requirements for software systems of various scales and purposes. The conditions for the feasibility of each approach are determined depending on the specifics of the project, the predicted load, and resource constraints. The results of the study can be used in the design, modernization, and optimization of software systems in a cloud environment, as well as in scientific research aimed at developing architectural design methodologies in the context of digital transformation. Further development of scientific research in the field of cloud architectures will contribute to improving the efficiency, reliability, and adaptability of modern software systems in the context of global digital transformation.

Keywords: cloud technologies, software, microservices, containerization, DevOps, scalability, serverless.

КРАВЧУК ОЛЬГА

Хмельницький національний університет

ОГЛЯД ТА АНАЛІЗ АРХІТЕКТУРНИХ ПІДХОДІВ ЩОДО РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З ВИКОРИСТАННЯМ ХМАРНИХ ТЕХНОЛОГІЙ

У статті здійснено комплексний аналіз сучасних архітектурних підходів до розробки програмного забезпечення з використанням хмарних технологій в умовах зростання вимог до масштабованості, відмовостійкості та безпеки інформаційних систем. Обґрунтовано актуальність переходу від традиційних локальних моделей розгортання до хмарних середовищ, що забезпечують гнучке управління ресурсами, еластичне масштабування та економічну оптимізацію витрат. Розглянуто еволюцію моделей надання хмарних сервісів (IaaS, PaaS, SaaS) та визначено їх вплив на формування архітектурних рішень. Проаналізовано ключові архітектурні парадигми, зокрема монолітну, мікросервісну, контейнеризовану та serverless-архітектуру. Проведено їх порівняльну характеристику за критеріями масштабованості, продуктивності, відмовостійкості, складності управління та забезпечення інформаційної безпеки. Особливу увагу приділено ролі контейнеризації, оркестрації ресурсів, автоматизації розгортання та впровадженню практик DevOps як важливих складових сучасної хмарної екосистеми.

У роботі запропоновано узагальнену модель оцінювання архітектурних підходів, що дозволяє зіставити технічні характеристики з практичними вимогами до програмних систем різного масштабу та призначення. Визначено умови доцільності застосування кожного підходу залежно від специфіки проєкту, прогнозованого навантаження та ресурсних обмежень. Результати дослідження можуть бути використані під час проєктування, модернізації та оптимізації програмних систем у хмарному середовищі, а також у наукових дослідженнях, спрямованих на розвиток методології архітектурного проєктування в умовах цифрової трансформації.

Ключові слова: хмарні технології, програмне забезпечення, мікросервіси, контейнеризація, DevOps, масштабованість, serverless.

Стаття надійшла до редакції / Received 16.01.2026

Прийнята до друку / Accepted 13.02.2026

Опубліковано / Published 26.03.2026



This is an Open Access article distributed under the terms of the [Creative Commons CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)

© Кравчук Ольга

Statement of the problem

The rapid digitization of society, the growth of data volumes, and the increasing complexity of software systems necessitate a transition to flexible, scalable, and fault-tolerant architectural solutions. The use of cloud technologies, which are being actively implemented by leading providers such as Amazon Web Services, Microsoft Azure, and Google Cloud, opens up vast opportunities for optimizing infrastructure and improving software development efficiency. At the same time, the diversity of service delivery models (IaaS, PaaS, SaaS) and architectural paradigms (monolithic, microservices, containerized, serverless) makes it difficult to make an informed choice of the optimal solution for a particular system. The problem lies in the lack of a universal approach to software design in a cloud environment that would simultaneously ensure scalability, security, economic feasibility, performance, and ease of maintenance. The mismatch between the architectural choice and the functional and non-functional requirements of the system can lead to resource overruns, reduced reliability, complicated integration and maintenance processes, and increased information security risks.

In a scientific context, the problem is related to the need to develop a generalized model for evaluating architectural solutions in a cloud environment that would take into account technical, economic, and operational criteria. This involves the development of methods for analyzing distributed systems, researching orchestration mechanisms,

infrastructure automation (Infrastructure as Code), and implementing DevOps practices as an integrated model of development and operation interaction.

The practical significance of the problem lies in the need to ensure the effective digital transformation of enterprises, government agencies, and scientific organizations. The choice of an adequate architecture directly affects the competitiveness of an IT product, the speed of its market launch, and its ability to adapt to changes in load and user requirements. In addition, modern cyber threats and regulatory requirements for data protection reinforce the relevance of researching secure architectural models in a cloud environment.

Thus, solving the outlined problem will contribute to increasing the scientific validity of the choice of architectural approaches and ensuring the practical effectiveness of software development using cloud technologies.

Analysis of recent research and publications

The issue of architectural approaches to software development using cloud technologies is actively researched by domestic and foreign scientists. A significant number of publications are devoted to the evolution of cloud service models (IaaS, PaaS, SaaS), as well as the transformation of architectural styles in the context of distributed systems [1].

The theoretical foundations of cloud computing and service-oriented models are discussed in works devoted to the development of the concept of cloud computing and the economics of computing resource utilization. Researchers focus on issues of virtualization, elastic scaling, multi-user environments, and cost optimization. A separate area of research analyzes the impact of cloud infrastructure on the transformation of business processes and the digital transformation of organizations [1,2].

In the field of architectural paradigms, considerable attention is paid to the transition from monolithic systems to microservice architecture. The practical aspects of implementing microservices are described in detail in the works of Martin Fowler and Sam Newman, which justify the feasibility of decomposing a system into independent services with autonomous life cycles. Scientific articles also explore issues of inter-service interaction, configuration management, monitoring, and fault tolerance [3,4].

A significant contribution to the development of containerization has been made by the spread of Docker technology and the Kubernetes orchestration system. Recent publications analyze mechanisms for automatic scaling, load balancing, and container cluster management, which provide flexibility and mobility for deploying applications in a cloud environment [5,6].

A separate layer of research is devoted to serverless architecture as a new stage in the evolution of cloud solutions. Recent works examine function-as-a-service (FaaS) models, event-driven approaches, and the cost-effectiveness of serverless computing. Researchers analyze the limitations of serverless solutions, including cold start delays, debugging complexity, and dependence on cloud service providers [7,8].

An important place in current research is occupied by issues of integrating DevOps practices, infrastructure automation (Infrastructure as Code), and continuous integration and delivery (CI/CD). Publications show that combining architectural approaches with process methodologies significantly improves the quality of software products and shortens their development cycle [9].

At the same time, analysis of scientific sources shows that most studies focus on individual aspects – technical, organizational, or economic – without forming a single integrated model for evaluating architectural solutions. The criteria for comparing different architectures within a comprehensive approach that takes into account both functional and non-functional requirements for the system remain insufficiently systematized.

Thus, the analysis confirms the relevance of further research aimed at generalizing existing approaches and developing methodological recommendations for the rational choice of software architecture in a cloud environment.

Formulation of the article's objectives

The purpose of the article is to conduct a comprehensive analysis of modern architectural approaches to software development using cloud technologies and to systematize their key characteristics. The study aims to compare the main architectural paradigms in terms of scalability, fault tolerance, security, and performance. The result of the work is the formation of practical recommendations for a reasonable choice of architectural solution depending on the requirements for the software system.

Presentation of the main material

The development of cloud technologies has significantly transformed approaches to software design and deployment. The traditional model of local infrastructure, which involved the use of proprietary servers and limited computing resources, has gradually given way to cloud solutions with flexible scaling, automated management, and cost optimization. At the heart of this transformation are service delivery models – IaaS, PaaS, and SaaS – each of which defines the level of responsibility of the developer and infrastructure provider.

Infrastructure as a Service (IaaS) provides users with access to virtualized computing resources, network components, and data storage systems. Within this model, the organization gains a high level of control over the configuration of the environment, but is responsible for administering operating systems and application software. Platform as a Service (PaaS) provides a ready-made environment for developing and deploying applications, reducing the complexity of infrastructure management. Software as a Service (SaaS) involves the use of ready-made software solutions via the network without the need for local installation. Leading cloud providers, including Amazon Web Services, Microsoft Azure, and Google Cloud, implement all of these models, forming comprehensive service ecosystems [7,8,10].

Architectural approaches to building software systems in a cloud environment have evolved in line with scalability and flexibility requirements. Monolithic architecture, which involves a single code base and joint deployment of all system components, is relatively simple to implement and test in the early stages of development. However, as functionality grows, monoliths make it difficult to make changes, scale individual modules, and support the system as a whole. An alternative is microservice architecture, which involves decomposing the system into independent services with their own lifecycles. This approach provides scalability flexibility, the ability to use different technology stacks, and increased fault tolerance, since the failure of one service does not necessarily lead to the shutdown of the entire system. At the same time, microservices require advanced mechanisms for inter-service communication, centralized monitoring, and automated configuration management, which increases the complexity of the architecture.

The next step in development was containerization, which allows applications and their dependencies to be isolated in standardized execution environments. Using the Docker platform ensures portability and predictability of deployment, while the Kubernetes orchestration system provides tools for automatic scaling, load balancing, and container cluster management. Containerized architecture combines the advantages of a microservice approach with resource efficiency [5,6].

Special attention should be paid to serverless architecture, which is based on a model of executing functions in response to events. In this case, the developer focuses exclusively on business logic, while infrastructure management is entirely entrusted to the cloud provider. This approach allows for cost optimization by paying only for resources actually used and provides automatic scaling. However, there are limitations related to cold start delays, complexity of debugging, and dependence on a specific platform [13].

An important component of modern cloud architecture is the implementation of DevOps practices that integrate development and operation processes. Automated deployment, continuous integration and delivery (CI/CD), and the use of the Infrastructure as Code approach help reduce time to market and increase system stability. Resource orchestration, centralized monitoring, and logging are becoming prerequisites for effective management of distributed systems [9,11,12].

Table 1

Comparative analysis of architectural approaches (schematic representation)

Criterion	Monolithic architecture	Microservice architecture	Containerized architecture	Serverless architecture
System structure	Single code base, shared deployment	Set of independent services	Services in containers	Separate functions executed on events
Scalability	Limited (entire system scales)	Flexible scaling of individual services	Automatic scaling through orchestration	Automatic at the function level
Fault tolerance	Low (failure affects failures) High (failure localization) High due to container isolation High, managed by provider	Fault tolerance Low (failure affects failures) High (failure localization) High due to container isolation High, managed by provider	Fault tolerance Low (failure affects failures) High (failure localization) High due to container isolation High, managed by provider	Fault tolerance Low (failure affects failures) High (failure localization) High due to container isolation High, managed by provider
Complexity of management	Low at the beginning	High (network interactions, DevOps)	Medium-high (requires orchestration)	Low for the developer, high dependence on the provider
Productivity	High at low loads	Depends on the optimization of interactions	High, efficient use of resources	May experience “cold start” delays
Flexibility of development	Limited	High	High	High
Expenses	Lower at the start	Higher due to infrastructure complexity	Optimized for scaling	Payment for actual usage

A comparative analysis of architectural approaches shows that the choice of the optimal solution depends on the scale of the project, the projected load, security requirements, and budget. Monolithic architecture may be appropriate for small or prototype systems. Microservice and containerized architectures are effective for complex enterprise solutions with high scalability and fault tolerance requirements. The serverless approach is optimal for event-driven systems with uneven loads.

Thus, modern cloud technologies form a multidimensional environment for architectural design, combining technical, economic, and organizational factors. A rational choice of architectural model requires a comprehensive analysis of system requirements and consideration of the long-term aspects of its development and maintenance.

Let's look at an example of architectural approaches in a specific company, Inten, which develops an online e-commerce platform. At the initial stage, the team is tasked with quickly creating a minimum viable product (MVP) to test the business model. The expected load is moderate—up to several tens of thousands of users per day—and the main

requirements are speed of development, minimization of costs, and ease of maintenance. In such conditions, the use of monolithic architecture becomes an appropriate solution. All system functionality – product catalog, shopping cart, ordering system, payment processing, user profile management – is implemented within a single code base and deployed as a single application. To simplify infrastructure management, a PaaS model using a cloud environment, such as Microsoft Azure, is chosen. This allows you to focus on developing business logic, delegating server administration issues to the cloud service provider. This approach ensures a quick product launch and is economically justified at the initial stage. As the platform grows in popularity, the load increases significantly. The number of users can reach hundreds of thousands per day, with peak periods during marketing campaigns and seasonal sales. Monolithic architecture is beginning to show its limitations: scaling requires duplicating the entire application, even if only the order module is overloaded; making changes to the code is complicated due to the high interdependence of components; there is a growing risk that a failure in one module will cause the entire system to fail. In these conditions, the company decides to gradually transform the system into a microservice architecture. The application is decomposed into separate services: user management, product catalog, orders, payments, analytics. Each service has its own database and is deployed independently. This allows scaling only those components that experience increased load, such as the order processing service during promotional periods. To ensure deployment flexibility, Docker-based containerization is used, which guarantees uniformity of development, testing, and operating environments. Container clusters are managed using Kubernetes, which provides automatic scaling, load balancing, and service recovery in case of failures. The infrastructure is deployed in the Amazon Web Services environment using automatic scaling mechanisms. As a result of the transformation, the system becomes more fault-tolerant: the failure of one service does not paralyze the entire platform. Updates can be implemented separately for each component, which shortens the release cycle. Resource optimization allows for more efficient use of cloud infrastructure and reduces costs during uneven loads. Additionally, a serverless approach is implemented for event-driven analytics. For example, functions for processing user events (viewing a product, adding it to the cart, placing an order) are performed only when the corresponding event occurs. Infrastructure management is fully handled by the cloud provider, and payment is made only for the actual time spent performing the functions. This allows for flexible response to peak loads without maintaining constantly active server resources.

Thus, the example above demonstrates the evolution of the architectural approach depending on the stage of system development. At the initial stage, it is advisable to use a monolith to minimize complexity and costs. As the scale grows and the requirements for performance and fault tolerance increase, the transition to a microservice and containerized architecture becomes justified. Serverless solutions complement the system in terms of event processing and resource optimization. This phased approach allows for a balance between technical efficiency, economic feasibility, and strategic project development goals.

Conclusions from this study and prospects for further research in this area

The study found that the choice of architectural approach to software development using cloud technologies should be determined by a set of functional and non-functional requirements for the system, including requirements for scalability, fault tolerance, security, performance, and cost-effectiveness. The analyzed models of cloud service provision (IaaS, PaaS, SaaS) form different levels of responsibility and infrastructure management, which directly affects architectural decisions. It has been established that monolithic architecture remains appropriate for small or initial projects, while microservice, containerized, and serverless approaches are more effective for complex distributed systems with dynamic loads.

A comparative analysis has shown that the implementation of DevOps practices, deployment automation, and resource orchestration are necessary conditions for ensuring the stability and manageability of modern cloud systems. Systematizing architectural approaches within a single analytical evaluation model allows for more informed technical decisions and reduces the risks associated with operating software products in a cloud environment.

Prospects for further research are related to the development of formalized methods for quantitative assessment of the effectiveness of architectural solutions, taking into account the economic indicators of the total life cycle of the system (Total Cost of Ownership), as well as studying the impact of artificial intelligence on the automation of cloud infrastructure management. Another relevant area of research is the study of approaches to ensuring cybersecurity in multi-cloud and hybrid environments and the improvement of cross-platform compatibility mechanisms.

Thus, further development of scientific research in the field of cloud architectures will contribute to improving the efficiency, reliability, and adaptability of modern software systems in the context of global digital transformation.

References

1. V.V. Koval. Theoretical basis of cloud migration - Visnyk KNTU No. 2(93), Part 2, 2025- P.165-169 DOI <https://doi.org/10.35546/kntu2078-4481.2025.2.2.19>
2. Tan, K. L., Chi, C.-H., Lam, K.-Y. Survey on Digital Sovereignty and Identity: From Digitization to Digitalization. ACM Computing Surveys. 2023. Vol. 56(3), Article 61. DOI: <https://doi.org/10.1145/3616400>
3. Ranjan A. (2023) From on-premise to cloud: Evolving IT infrastructure for the AI age. World Journal of Advanced Research and Reviews, vol. 20(3), pp. 1898–1934. <https://doi.org/10.30574/wjarr.2023.20.3.1590>
4. Transition to microservices [Electronic resource]. – Access to the resource: <https://dou.ua/forums/topic/47724/>

5. Morabito, R. (2017) Virtualization on Internet of Things edge devices with Container technologies: a performance evaluation documentation [Electronic resource]. – Access to the resource: <https://doi.org/10.1109/ACCESS.2017.2704444>.
6. Makovieva V. K. Containerization and orchestration: Docker and Kubernetes. <https://openarchive.nure.ua/server/api/core/bitstreams/d7d33a4f-c139-43d4-8afa-87bba4eca2c8/content>
7. Serverless architecture and function-as-a-service (FaaS) platforms [Electronic resource]. – Access to the resource: <https://www.hostragons.com/uk>
8. Gurzhiy A.M., Povoroznyuk N.I., Samsonov V.V. Informatics and Information Technologies - Kharkiv, 2017-352 p.
9. Kravchuk O.A. Model of CI/CD implementation for IT project management optimization / Olga Kravchuk / International scientific and technical journal “Measuring and computing devices in technological processes.” - Khmelnytskyi, Khmelnytskyi National University, No. 3 (2023). - P. 73–82 DOI: <https://doi.org/10.31891/2219-9365-2023-75-8>
10. Kravchuk, O.A. Artificial intelligence in programming: how AI is changing the approach to code development and automation / Olga Kravchuk // Bulletin of Khmelnytsky National University. Series: Technical Sciences – Khmelnytskyi: Khmelnytskyi National University, 2024, No. 6. Pp. 238–242 <https://heraldts.khmnu.edu.ua/index.php/heraldts/article/view/1621/1624>
11. Kravchuk, O. (2025). Integration of Agile and Waterfall methodologies into hybrid IT project management models: challenges and advantages. Measuring and computing devices in technological processes, (3), 99–104. <https://doi.org/10.31891/2219-9365-2025-83-13>
12. Kravchuk, O., & Kachurynets, D. (2025). Information and communication technologies as a catalyst for digital transformation in the educational environment: technical analysis of cloud platform use. Herald of Khmelnytskyi National University. Technical Sciences, 353(3.2), 447-453. <https://doi.org/10.31891/2307-5732-2025-353-61>
13. Kravchuk, O. (2025). Cost optimization in IT projects: models and approaches to cost management. Measuring and computing devices in technological processes, 82(2), 198–203. <https://doi.org/10.31891/2219-9365-2025-82-27>