

# АНАЛІЗ АРХІТЕКТУРНИХ ШАБЛОНІВ МОНОЛІТНОЇ ТА МІКРОСЕРВІСНОЇ АРХІТЕКТУР В ІНФОРМАЦІЙНИХ СИСТЕМАХ ЕЛЕКТРОННОЇ КОМЕРЦІЇ

Вінницький національний технічний університет

## *Анотація*

*У роботі проведено порівняльний аналіз архітектурних шаблонів монолітної та мікросервісної архітектур в інформаційних системах електронної комерції. Досліджено основні переваги та недоліки кожного підходу з точки зору масштабованості, підтримки, продуктивності та витрат на розробку. Зроблено висновки щодо доцільності використання кожної з архітектур у залежності від потреб та етапу розвитку e-commerce проєктів.*

**Ключові слова:** електронна комерція, архітектура інформаційних систем, монолітна архітектура, мікросервісна архітектура.

## *Abstract*

*This paper presents a comparative analysis of architectural patterns of monolithic and microservice architectures in e-commerce systems. The study examines the main advantages and disadvantages of each approach in terms of scalability, maintainability, performance, and development costs. Conclusions are drawn regarding the feasibility of using each architecture depending on the needs and stage of development of e-commerce projects.*

**Keywords:** e-commerce, information system architecture, monolithic architecture, microservice architecture.

## **Вступ**

Інформаційні системи електронної комерції з кожним роком зростають у масштабах, складності та навантаженні. З цим зростанням підвищується необхідність у виборі ефективної архітектури програмного забезпечення, яка забезпечує стабільну роботу, швидкість розгортання та гнучкість змін.

Актуальність роботи зумовлена зростаючими вимогами до гнучкості, масштабованості та надійності інформаційних систем у сфері електронної комерції. Вибір між монолітною та мікросервісною архітектурою суттєво впливає на ефективність розробки, підтримки та подальшого розвитку таких систем.

Метою даної роботи є аналіз та порівняння архітектурних шаблонів монолітної та мікросервісної архітектур у системах електронної комерції з урахуванням їхніх технічних характеристик, складності розгортання, підтримки, а також ефективності в умовах високих навантажень.

## **Результати роботи**

У рамках дослідження здійснено поглиблений аналіз монолітної та мікросервісної архітектур в контексті їх застосування в інформаційних системах електронної комерції. Основна увага приділена технічним характеристикам, архітектурним шаблонам, організації життєвого циклу систем, а також типових перевагам і обмеженням кожного підходу.

Монолітна архітектура являє собою єдиний блок коду, в якому всі компоненти (UI, логіка бізнес-процесів, доступ до баз даних) інтегровані у спільне виконуваче середовище. Вона добре підходить для невеликих команд і стартапів завдяки швидкому розгортанню, простішому налагодженню та меншій залежності від складної інфраструктури.

Мікросервісна архітектура базується на ідеї розділення функціоналу системи на набір незалежних сервісів, які взаємодіють через мережеві API. Кожен сервіс виконує одну бізнес-функцію та має власну

базу даних або сховище. Цей підхід дозволяє масштабувати систему горизонтально, застосовувати різні технології в різних сервісах і забезпечує вищу стійкість до відмов.

З метою виявлення оптимального архітектурного підходу для інформаційних систем електронної комерції здійснено зіставлення монолітної та мікросервісної архітектур за рядом практично орієнтованих критеріїв. Аналіз охоплює аспекти масштабованості, продуктивності, гнучкості, організації розробки та підтримки, що мають критичне значення в умовах високої динаміки змін і зростаючих вимог до цифрових сервісів. Узагальнені результати наведено в таблиці 1.

| Критерій                   | Монолітна архітектура                                                  | Мікросервісна архітектура                                                  |
|----------------------------|------------------------------------------------------------------------|----------------------------------------------------------------------------|
| Масштабованість            | Лише вертикальна: розгортання копії всієї системи                      | Горизонтальна: окреме масштабування кожного сервісу                        |
| Продуктивність             | Висока на невеликих обсягах завдяки локальній взаємодії модулів        | Може знижуватись через мережеву взаємодію, але добре оптимізується         |
| Гнучкість технологічна     | Обмежена єдиним стеком технологій                                      | Висока: різні мови та фреймворки в різних сервісах                         |
| Швидкість розробки         | Вища на старті: просте середовище, мінімум залежностей                 | Повільніша на початку через потребу в налаштуванні інфраструктури          |
| Надійність                 | Вразлива: помилка в одному модулі впливає на всю систему               | Вища: збій одного сервісу не призводить до падіння всієї системи           |
| Тестування                 | Єдине середовище тестування, менше ізольованих компонентів             | Потрібні окремі модулі для тестування кожного сервісу та інтеграцій        |
| Розгортання та оновлення   | Просте, але вимагає повного перекомпілювання/перезапуску при оновленні | Можливість незалежного оновлення окремих сервісів                          |
| Підтримка командної роботи | Обмежена: складність паралельної розробки в одній кодовій базі         | Команди можуть працювати незалежно над окремими сервісами                  |
| DevOps-вимоги              | Мінімальні: достатньо базового CI/CD                                   | Високі: потрібні засоби оркестрації, автоматичного тестування, моніторингу |
| Витрати на інфраструктуру  | Нижчі на ранніх етапах                                                 | Вищі через складність, але виправдані при масштабуванні                    |

**Таблиця 1.** Порівняльний аналіз монолітної та мікросервісної архітектур

Порівняльний аналіз демонструє, що ефективність застосування тієї чи іншої архітектури безпосередньо залежить від масштабу, складності та бізнес-моделі системи. У випадках із невеликими e-commerce рішеннями із стабільним функціоналом монолітна модель дає змогу швидко реалізувати продукт з мінімальними ресурсами. У той час як мікросервісна архітектура забезпечує гнучкість і високу доступність для великих, динамічних систем, що потребують постійного розвитку, масштабування та незалежної роботи різних команд розробників.

## Висновки

У результаті дослідження проведено порівняльний аналіз монолітної та мікросервісної архітектур, зосереджений на їх застосуванні в інформаційних системах електронної комерції. Розглянуто технічні особливості, архітектурні шаблони, переваги та недоліки обох підходів у контексті сучасних вимог до масштабованості, гнучкості та надійності систем.

Монолітна архітектура виявилася доцільною для проєктів на ранніх етапах розвитку, а також для невеликих платформ зі стабільною логікою бізнес-процесів і обмеженими ресурсами. Її ключовими перевагами є простота впровадження, мінімальні інфраструктурні вимоги та швидке розгортання. Однак зі зростанням складності системи ця модель створює труднощі у масштабуванні, модифікації окремих компонентів і підтримці коду.

Мікросервісна архітектура, своєю чергою, є більш адаптивною до змін і дозволяє будувати масштабовані, розподілені системи з високим рівнем доступності. Вона ефективно підтримує незалежну роботу команд розробників, паралельну розробку й впровадження окремих сервісів. Водночас її впровадження потребує високої технологічної зрілості, розвиненої інфраструктури та значних витрат на початкову розробку й підтримку.

Таким чином, вибір архітектурного підходу має базуватися на специфіці проєкту, очікуваних навантаженнях, ресурсах команди та бізнес-цілях системи. Подальші дослідження можуть бути зосереджені на аналізі гібридних підходів, зокрема переходу від моноліту до мікросервісів, а також на практиках автоматизації розгортання й масштабування у хмарних середовищах.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Newman S. Building Microservices: Designing Fine-Grained Systems. – O'Reilly Media, 2015. – 280 с. ISBN 978-1491950357.
2. Richards M. Software Architecture Patterns. – O'Reilly Media, 2015. – 60 с. ISBN 978-1491924242.
3. Fowler M., Lewis J. Microservices. – URL: <https://martinfowler.com/articles/microservices.html>
4. Thönes J. Microservices // IEEE Software. – 2015. – Vol. 32, No. 1. – P. 116. – DOI: 10.1109/MS.2015.11
5. Dragoni N., Lanese I., Larsen S.T. et al. Microservices: Yesterday, Today, and Tomorrow. – In: Present and Ulterior Software Engineering. – Springer, 2017. – P. 195–216. – DOI: 10.1007/978-3-319-67425-4\_12

*Московко Сергій Геннадійович* — факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: smoskovko@icloud.com.

*Moskovko Serhii G.* — Department of intelligent information technologies and automation, Vinnytsia National Technical University, Vinnytsia, e-mail: smoskovko@icloud.com.