

УДК 004.42:004.7

Шутяк Л. - ст. гр. СНМ-61

Тернопільський національний технічний університет імені Івана Пулюя

МІКРОСЕРВІСНА АРХІТЕКТУРА: ВИКЛИКИ ТА ПЕРСПЕКТИВИ РОЗВИТКУ ПРОГРАМНИХ СИСТЕМ

Науковий керівник: д.е.н., професор Матійчук Л.П.

Shutiak L.

Ternopil Ivan Puluj National Technical University

MICROSERVICE ARCHITECTURE: CHALLENGES AND PROSPECTS FOR SOFTWARE SYSTEMS DEVELOPMENT

Supervisor: Doctor of Economic Sciences, Professor Matiichuk L.

Ключові слова: мікросервісна архітектура, модульність, масштабованість.

Key words: microservice architecture, modularity, scalability.

У сучасному світі розвитку інформаційних технологій традиційні монолітні архітектури програмного забезпечення дедалі частіше виступають стримуючим чинником для швидкої модернізації та масштабування систем. Це створює невідповідність між можливостями таких архітектур та зростаючими потребами бізнесу й користувачів. Як відповідь на ці виклики сформувалася мікросервісна архітектура, що передбачає поділ програмного продукту на незалежні сервіси. Вона забезпечує гнучкість та адаптивність, проте її застосування супроводжується новими проблемами, серед яких особливе місце займає складність управління розподіленими системами.

Мікросервісна архітектура виникла як реакція на обмеження традиційних підходів до побудови програмних систем. Її сутність полягає у поділі додатка на низку невеликих, незалежних сервісів, кожен з яких реалізує окрему бізнес - функцію та може розроблятися, розгортатися й масштабуватися автономно. Такий підхід забезпечує командам розробників більшу самостійність, дозволяє застосовувати оптимальні технології для конкретних завдань і значно прискорює впровадження нових функціональних можливостей [1].

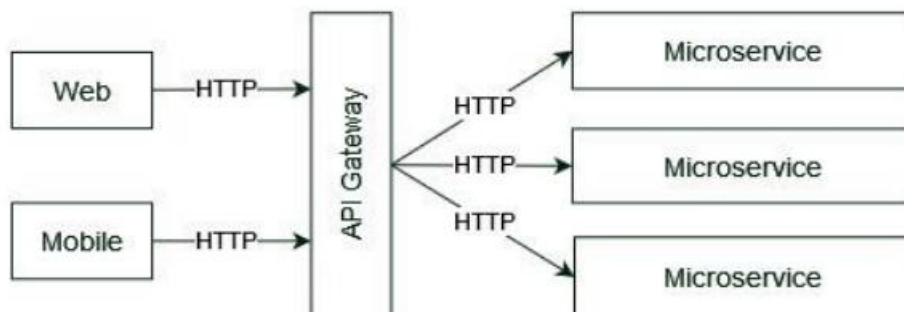


Рис. 1. Схема мікросервісної архітектури [1]

Ефективність мікросервісної моделі підтверджена практикою провідних компаній – Netflix, Amazon та Spotify, які завдяки її використанню досягли істотного зростання продуктивності та масштабованості своїх систем. Водночас перехід до

мікросервісів супроводжується низкою викликів, серед яких управління розподіленою інфраструктурою, забезпечення безпеки, організація моніторингу та тестування [2]. Крім того, не всі організації готові до повної трансформації архітектури. У таких випадках застосовується концепція модульної архітектури, яка поєднує переваги модульності мікросервісів із простотою монолітного розгортання. Архітектура модульності забезпечує можливість організувати програмний додаток у вигляді окремих модулів із чітко визначеними інтерфейсами, що значно полегшує підтримку та розвиток системи без необхідності істотних змін інфраструктури [3].

Мікросервісна архітектура, у свою чергу, є сучасним підходом до створення програмного забезпечення, який передбачає поділ додатка на набір невеликих, автономних сервісів. Кожен із них реалізує конкретну бізнес-функцію та може бути незалежно розроблений, розгорнутий і масштабований. Основними рисами мікросервісної моделі є автономність сервісів, простота їхньої взаємодії, децентралізоване управління даними та незалежність процесів розгортання. Кожен мікросервіс відповідає за окрему бізнес-можливість або функцію, що забезпечує командам розробників більшу самостійність, сприяє прискоренню процесів розробки та впровадження нових функціональних можливостей.

Сервіси взаємодіють між собою за допомогою простих механізмів комунікації, найчастіше через протоколи HTTP/REST, що забезпечує легку інтеграцію та знижує рівень залежності між компонентами системи [4]. Кожен мікросервіс має власну базу даних або модель даних, що дозволяє уникати конфліктів при доступі до спільних ресурсів і сприяє підвищенню продуктивності [5]. Завдяки можливості незалежного розгортання мікросервісів забезпечується оперативне впровадження змін та нових функцій без негативного впливу на роботу всієї системи. Мікросервісна архітектура є результатом поступової еволюції програмних архітектур, спрямованої на створення більш гнучких, масштабованих і стійких рішень. Для ґрунтовного розуміння її сутності необхідно розглянути попередні архітектурні моделі та технологічні тенденції, що стали основою її формування.

1. Киселевич, В. В., Усата, О. Ю., Сікора, Я. Б., Вербівський, Д. С., & Іванов, Д. Є. (2024). Мікросервісна архітектура: переваги та недоліки її практичного застосування. *Information Technology: Computer Science, Software Engineering and Cyber Security*, (2), 50-59.
2. S. Newman, *Building Microservices: Designing Fine-Grained Systems*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2021.
3. I. Nadareishvili, R. Mitra, M. McLarty, and M. Amundsen, *Microservice Architecture: Aligning Principles, Practices, and Culture*. Sebastopol, CA, USA: O'Reilly Media, 2016.
4. M. Beckers, "Understanding the potential of Modulith architecture," *Worldline Tech Blog*, Jan. 2024. Accessed Oct. 2, 2023. [Online]. URL: <https://blog.worldline.tech/2024/01/23/modulith.html>
5. C. Richardson, *Microservices Patterns: With Examples in Java*. Shelter Island, NY, USA: Manning Publications, 2018.