УДК 004.4

В. А. Шатайло

ПОРІВНЯННЯ МІКРОСЕРВІСНОЇ ТА МОНОЛІТНОЇ АРХІТЕКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Вінницький Національний Технічний Університет

Анотація

У даній роботі розглядаються архітектури програмного забезпечення, порівнюються особливості, переваги та недоліки кожної з них.

Ключові слова: Мікросервіси, мікросервісна архітектура, монолітна архітектура, програмне забезпечення.

Abstract

This work examines software architectures, compares the features, advantages and disadvantages of each of them.

Keywords: Microservices, microservice architecture, monolithic architecture, software.

Вступ

Все частіше ми чуємо про мікросервісну архітектуру та їх впровадження в IT індустрії. Мікросервісна архітектура є одним з архітектурних підходів, який використовується при розробці програмного забезпечення. На противагу їй існує монолітна архітектура, яка використовується уже доволі давно та змогла успішно зарекомендувати себе [1]. То в чому ж тоді різниця між ними, та чи є якась причина, чому за останній час все більше компаній схиляються саме до мікросервісів? Давайте більш детально розглянемо ці архітектури і розберемо переваги та недоліки кожної з них.

Огляд та аналіз

Монолітна архітектура є традиційним підходом до розробки програмного забезпечення, яка базується на розробці додатку як одного цілого, тобто усі його компоненти розгортаються, інтегруються та взаємодіють між собою на одному або кількох серверах. Монолітні програми характеризуються простотою на ранніх стадіях розробки [2]. Єдина кодова база містить усі функції та можливості. Однак, зі зростанням програми зростає і її складність. Зміни в одній частині програми можуть ненавмисно вплинути на інші, що ускладнює її підтримку.

Монолітна архітектура добре підходить для невеликих проектів або проектів з відносно низькою складністю. Популярні системи, такі як WordPress і Django, де простота та швидка розробка є головним напрямком, використовують цю архітектуру.

В свою чергу, мікросервісна архітектура розбиває додаток на набір менших сервісів, що розгортаються незалежно один від одного. Кожен сервіс відповідає за певну функціональність і взаємодіє з іншими за допомогою чітко визначених API.

Мікросервіси ідеально підходять для сценаріїв, де масштабованість, гнучкість і швидка адаптація до мінливих вимог є критично важливими. Масштабні додатки, такі як Netflix, Amazon та Uber, використовують підхід мікросервісів для управління своїми складними екосистемами, що постійно розвиваються через невпинний ріст користувачів.

Ключові відмінності

Монолітні додатки мають єдину кодову базу, в той час як мікросервіси складаються з декількох слабко пов'язаних між собою сервісів. Мікросервіси не залежать від платформи та технологічного стеку. Кожен сервіс в архітектурі мікросервісів може бути розроблений з використанням різних мов програмування, баз даних і технологій. Наприклад, можна створити один сервіс за допомогою Node.js з використанням бази даних NoSQL, тоді як інший сервіс може бути розроблений на Java з використанням реляційної бази даних [3]. На противагу цьому, монолітне програмне забезпечення зазвичай використовує один стек технологій для всієї програми. Якщо ви обираєте певний стек технологій для монолітного застосунку, ви зобов'язуєтеся використовувати його для всього застосунку.

Монолітні програми масштабуються шляхом розгортання всієї програми, тоді як мікросервіси дозволяють масштабувати окремі служби. У монолітній архітектурі масштабування зазвичай передбачає розгортання всієї програми. Отже, якщо одна частина програми вимагає більше ресурсів, весь моноліт потрібно масштабувати, що потенційно може призвести до неефективного розподілу ресурсів.

При цьому монолітні програми можуть розроблятися швидше на початковому етапі, але мікросервіси пропонують більшу гнучкість у розробці. Завдяки мікросервісам команди розробників можуть зосередитися на окремих сервісах. Наприклад, одна команда може відповідати за сервіс каталогу продуктів, а інша - за автентифікацію користувачів. Це дозволяє отримати спеціалізовану експертизу та пришвидшити цикл розробки в кожній команді.

Однією з ключових переваг мікросервісів є їхня гнучкість у розгортанні [4, 5]. Кожен сервіс в архітектурі мікросервісів може бути розгорнутий незалежно. Це означає, що коли настає час випустити нову функцію або оновити певний функціонал у програмному забезпеченні, не потрібно буде перерозгортати весь додаток. Натомість, можна розгорнути лише той сервіс, який було змінено.

Висновки

Обидві архітектури мають свої переваги та недоліки. Моноліти можуть стати громіздкими в міру зростання, в той час як мікросервіси ускладнюють управління міжсервісною комунікацією. Найкращі практики, такі як належне планування, проектування та моніторинг, можуть допомогти подолати ці виклики.

Щоб вирішити, яка архітектура підходить для конкретного проєкту, потрібно враховувати такі фактори, як розмір проекту, складність, структуру команди та стек технологій. Невеликі проекти з обмеженою складністю можуть виграти від простоти монолітного підходу, в той час як великі, складні додатки можуть знайти мікросервіси більш підходящими для своєї основи.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Monolith vs Microservice Architecture: A Comparison [Електронний ресурс]. – Режим доступу: https://camunda.com/blog/2023/08/monolith-vs-microservice-architecture-comparison/

2. Мікросервісна архітектура: плюси та мінуси [Електронний ресурс]. – Режим доступу: https://itedu.center/ua/blog/articles/microservices-architecture-advantages-and-disadventages/

3. Монолітна архітектура ПЗ [Електронний ресурс]. – Режим доступу: https://qalight.ua/baza-znaniy/shho-take-monolitna-arhitektura/

4. Monolithic Approach vs. Microservices Approach: Which is Right for Your Application? [Електронний ресурс]. – Режим доступу: https://www.linkedin.com/pulse/monolithic-approach-vs-microservices-which-right-your-majid-sheikh/

5. Monolith Versus Microservices: Weigh the Pros and Cons of Both Configs [Електронний ресурс]. – Режим доступу: https://www.akamai.com/blog/cloud/monolith-versus-microservices-weigh-the-difference

Шатайло В'ячеслав Андрійович — студент групи 2СП-21б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький Національний Технічний Університет, Вінниця, e-mail: viacheslavshatailo@gmail.com

Shatailo Viacheslav Andriyovych — student of group 2SP-21b, faculty of information technologies and computer engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: viacheslavshatailo@gmail.com