Скаковський В. О., здобувач,
Савіцький Р. С., аспірант, ст. викладач,
Державний університет "Житомирська політехніка"

ПРОЄКТУВАННЯ АРХІТЕКТУРИ СИСТЕМИ УПРАВЛІННЯ СПІЛЬНОТАМИ ТА ПОДІЯМИ

З точки зору архітектури застосунку, існує вибір між монолітною та архітектурою мікросервісів.

Монолітна архітектура – найбільш поширений варіант архітектури, яка побудована як система з єдиною кодовою базою, яка інкапсулює усю бізнес-логіку. Тому, якщо потрібно внести деякі зміни в додаток, потрібно перебудовувати і деплоїти цілу систему. Проте, моноліти є зручними на ранній стадії життя проекту. Приклад монолітної архітектури зображено на рис. 1.

Переваги монолітної архітектури:

- простота розробки, деплоймента, тестування, відлагодження;
- продуктивність.

Рис. 1. Монолітна архітектура

Мікросервісна архітектура включає в себе серію сервісів, які можуть бути незалежно задеплоєні. Кожен сервіс має свою ціль, кодову базу, бізнес-логіку та сховище даних [1]. Можливості мікросервісної архітектури реалізують незалежну зміну, перебудову та деплоймент окремого сервісу, без впливу на інші, що ускладнює розробку проекту. Переваги мікросервісної архітектури:

- гнучке масштабування;
- незалежний деплоймент;
- гнучкість в розробці.

Приклад мікросервісної архітектури зображено на рис. 2.

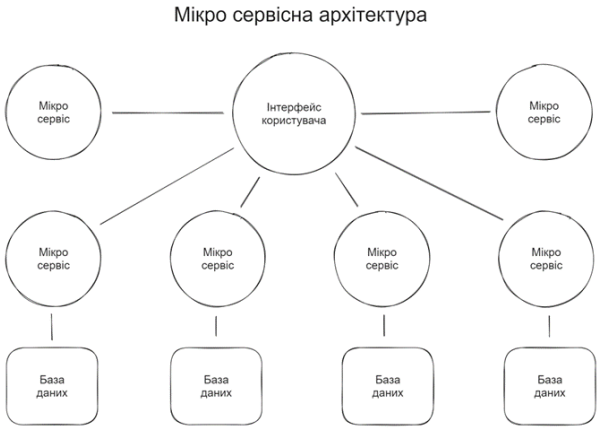

Рис. 2. Мікросервісна архітектура

Чудовою практикою є розробка додатку з використанням монолітної архітектури за замовчуванням і, при необхідності, виокремлення окремих сервісів для отримання деяких переваг мікросервісів. Наприклад, масштабування одного сервісу, якщо його продуктивність у межах моноліту є недостатньою. Раннє впровадження мікросервісної архітектури може призвести до непотрібного ускладнення системи та уповільнення розробки.

Спочатку мікросервісна архітектура уповільнюватиме розробку, так як важко визначити, які сервіси необхідно адаптовувати. Наперед не відомо, яка частина додатку матиме найбільше навантаження та буде вузьким місцем у системі, тому і неможливо виділити код заздалегіть в окремий сервіс для майбутнього масштабування.

Одні з найбільш складних рішень (і дорогих, якщо зроблено неправильно) – це те, як декомпозувати додаток в набір мікросервісів, які взаємодіють між собою. Якщо це зроблено неправильно, з'являться проблеми з повільним внесенням змін, поганою продуктивністю та неузгодженістю даних. Це великий ризик.

Зважаючи на плюси кожної з архітектур, для розробки системи управління спільнотами та подіями краще розпочинати з монолітної архітектури.

Список використаних джерел

1. Larsson M. Microserviceswith Spring Boot 3 and Spring Cloud: Buildresilientandscalablemicroservicesusing Spring Cloud, Istio, andKubernetes / MagnusLarsson., 2023. – 706 с.