УДК 004.4

Бабюк Н. П.
Свіца О. С.

ВИБІР АРХІТЕКТУРИ ОНЛАЙН ПЛАТФОРМИ ДЛЯ КОМУНІКАЦІЇ ПРОГРАМІСТІВ

Вінницький національний технічний університет

Анотація

У статті проаналізовано вплив архітектурних рішень на розробку програмного забезпечення. Обґрунтовано вибір мікросервісної архітектури онлайн платформи для комунікації користувачів. Наведено базові принципи, що забезпечують коректну реалізацію обраного архітектурного рішення.

Ключові слова: архітектура, модуль, мікросервіс, програма

Abstract

The article analyzes the impact of architectural decisions on software development. The choice of a microservices architecture for an online communication platform is justified. The basic principles ensuring the correct implementation of the chosen architectural solution are outlined.

Keywords: architecture, module, microservice, program

Вибір та реалізація архітектури є важливим етапом проектування будь-якої системи. Коректно вибрана архітектура може значно спростити процес розробки та впровадження програми, забезпечуючи зрозумілість, стійкість до помилок та можливість до перевикористання коду, в той час як помилка у цьому аспекті скоріше за все матиме наслідком довгі години роботи та рефакторингу.

Суть будь-якої архітектури зводиться до розбиття коду на модулі та налагодження зв'язку між ними. Відмінність полягає лише в принципах за якими відбувається його групування. Модуль це логічне групування пов'язаного коду, що може бути групою класів в об'єктно-орієнтованій мові програмування або функцій в структурній чи функціональній [1]. При цьому код всередині модуля має мати мінімальну кількість зв'язків з іншими модулями та велику кількість зв'язків з кодом цього модуля. Це дозволяє максимально абстрагувати модуль від системи та інкапсулювати його стан. Саме такий підхід знижує залежність всієї системи від конкретного модуля, а отже зміни в модулі матимуть мінімальний вплив на код всієї програми. Крім того, його код стає більш зрозумілим, оскільки можливість змінювати стан модуля поза його межами зведена до мінімуму.

Безперечно немає однієї «правильної» архітектури, яка з успіхом могла б бути використана на більшості проєктах. Рішення, що чудово підходять для одних програм можуть виявитися абсолютно помилковими для інших. Вирішальну роль у цьому питанні відіграє сам проєкт, його складність, вимоги, функціонал, який він реалізує.

Найпопулярніші архітектурні рішення, які довели свою ефективність при правильному застосуванні включають:

- MVC.
- Чисту архітектуру.
- Мікросервісну архітектуру.
- Архітектуру, що базується на подіях.

Для того, щоб обрати архітектурне рішення необхідно зрозуміти, вимоги до створюваної програми. Платформа для комунікації програмістів має витримувати великі навантаження, бути доступною 24/7, зменшувати вплив будь-якої помилки в системі та максимально швидко після неї відновлюватися. Мікросервісна архітектура найкраще задовольняє ці вимоги.

Існує декілька підходів розбиття системи на модулі відповідно до мікросервісної архітектури. Найпрактичнішим є групування коду навколо об’єктів, які входять до предметної області системи. В такому випадку кожен мікросервіс взаємодіятиме з іншим, якщо буде потребувати функціональності, яку той забезпечує [2]. Наприклад основними об’єктами онлайн платформи для комунікації користувачів є пост, коментар, повідомлення, користувач, чат. Такий підхід дозволяє глибше зануритися в предметну область, встановити єдині терміни, що будуть зрозумілими для всіх членів команди, забезпечити гнучкість системи. Не обов’язково реалізовувати мікросервіс для кожного об’єкта. Наприклад, такі об’єкти як пост, коментар та повідомлення є спорідненими, тому реалізовують схожий функціонал. Їх розбиття призвело б до утворення численних зв’язків між результуючими модулями, тому варто їх залишити в межах одного мікросервісу. Водночас, об’єкти користувач та пост не мають нічого спільного тому можуть реалізовані в різних мікросервісах.

Отже для забезпечення стабільної та надійної роботи онлайн платформи для комунікації програмістів обрано мікросервісну архітектуру. Групування коду відбувається навколо об’єктів, що представляють предметну область системи. Наслідком такого підходу є створення модулів, які легко зрозуміти та змінювати.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Mark Richards. Fundamentals of Software Architecture / Mark Richards, Neal Ford – Boston : O’reilly, 2020. – 400 p.
2. Sam Newman. Building microservices / Sam Newman – Boston : O’reilly, 2021. – 586 p.

Свіца Олександр Сергійович
студент, Вінницький національний технічний університет, м. Вінниця, svicasasha@gmail.com

Науковий керівник: Бабюк Наталя Петрівна, доцент, Вінницький національний технічний університет, м. Вінниця

Svitsa Oleksandr
Faculty of Information Technology and Computer Engineering

Babiuk Natalia Petrivna
Associate Professor, Vinnytsia National Technical University, Vinnytsia