

СУЧАСНІ ПІДХОДИ ПРИ РОЗРОБЦІ АРХІТЕКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

¹Вінницький національний технічний університет

Анотація

Дана робота присвячена аналізу та огляду сучасних підходів у розробці архітектури програмного забезпечення. Розглянуто тенденції, методології та інструменти, які використовуються в сучасній практиці для створення високоефективних, масштабованих та надійних програмних систем. Зосереджено увагу на перевагах та недоліках мікросервісної та трирівневої архітектур.

Ключові слова: архітектура програмного забезпечення, відмовостійкості, масштабованість, REST API, взаємодія.

Abstract

This work is dedicated to the analysis and review of modern approaches in the development of software architecture. It examines trends, methodologies, and tools used in contemporary practice to create high-performance, scalable, and reliable software systems. The focus is on the advantages and disadvantages of microservices architecture and three-tier architecture.

Keywords: software architecture, fault tolerance, scalability, REST API, interaction.

Вступ

Сучасний швидкий розвиток інформаційних технологій суттєво визначає вимоги до якості та ефективності програмного забезпечення. Розробка архітектури програмних продуктів стає ключовим аспектом успішної інженерії програмного забезпечення, оскільки вона визначає основні принципи організації системи та впливає на її майбутню масштабованість, надійність та ефективність.

В даній роботі розглядаються сучасні підходи до створення архітектури програмного забезпечення з урахуванням стрімкого розвитку технологій, великої конкуренції та зростаючих вимог користувачів. Розуміння та впровадження передових стратегій стають критичними факторами для досягнення успіху в цій області.

Мета роботи полягає в аналізі підходів для розробки архітектури програмного забезпечення, зокрема ціллю є створення продуктивної та масштабованої системи. Робота розглядає сучасні методології та інструменти, які сприяють створенню програмних продуктів, що відповідають сучасним вимогам до інформаційних систем.

Проектування архітектури програмного забезпечення

Проектування архітектури програмного забезпечення (ПЗ) – це процес визначення основних структурних компонентів, їх взаємодії та організації системи для досягнення певних цілей. Цей процес включає в себе визначення вимог до системи, розробку архітектурних концепцій, вибір підходів до реалізації та побудову загальної структури програмного продукту.

Трирівнева архітектура (рисунк 1) – це тип архітектури програмного забезпечення, при якому система складається з трьох рівнів: рівень презентації: відповідає за взаємодію з користувачем; рівень бізнес-логіки: відповідає за обробку бізнес-логіки системи; рівень даних: відповідає за зберігання даних [1].

Переваги такої архітектури: простота у визначенні та розробці; зручність розробки та обслуговування; безпека: система з такою архітектурою більш безпечна, оскільки кожен рівень має обмежений доступ до даних

Недоліки такої архітектури: складніше масштабування окремих компонентів; обмежені можливості у відновленні окремих служб без впливу на інші

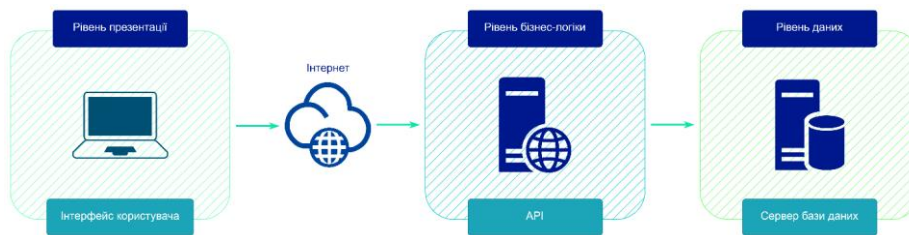


Рисунок 1 – Трирівнева архітектура програмного забезпечення

Мікросервісна архітектура (рисунок 2) – це тип архітектури програмного забезпечення, при якому система складається з набору невеликих, незалежних сервісів. Кожний сервіс відповідає за виконання однієї або кількох функцій системи, які можна розгорнути та масштабувати незалежно. Кожна служба також забезпечує чіткі межі модулів, дозволяючи писати різні служби на різних мовах програмування [2].

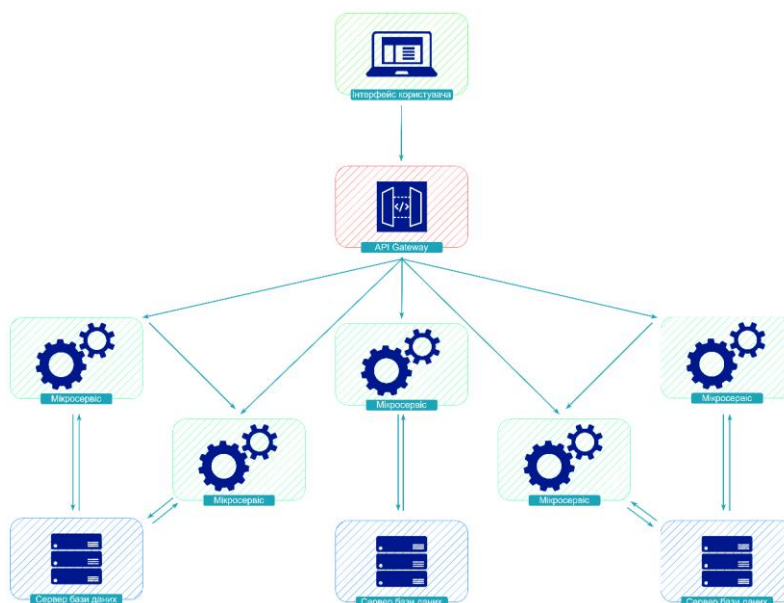


Рисунок 2 – Мікросервісна архітектура програмного забезпечення

Переваги такої архітектури: масштабованість: система легко масштабується, оскільки кожен сервіс може бути розгорнутий на декількох серверах. зручність розробки та обслуговування: система легко поділяється на невеликі, незалежні сервіси, що полегшує розробку та обслуговування кожного з них; еластичність: система більш еластична, оскільки сервіси можуть бути легко додані або видалені без впливу на інші сервіси.

Недоліки такої архітектури: збільшення складності: система стає більш складною у обслуговуванні, оскільки потрібно враховувати взаємодію між багатьма сервісами; збільшення витрат: система вимагає більших витрат на інфраструктуру, оскільки кожен сервіс повинен бути розгорнутий на окремому сервері.

Окрім архітектури програмного забезпечення, важливою складовою програмного продукту є взаємодія окремих компонентів системи. Для реалізації цієї взаємодії є різні інструменти. Розглянемо один з найпопулярніших – REST (рисунок 3).

Це архітектурний стиль взаємодії компонентів розподіленої системи у комп'ютерній мережі. Простіше кажучи, REST визначає стиль взаємодії між різними компонентами системи, кожна з яких може розташовуватися фізично в різних місцях [3].

Переваги, які дає REST: продуктивність (за рахунок використання кешу); масштабованість; прозорість системи взаємодії; простота інтерфейсів; портативність компонентів; легкість внесення змін; здатність еволюціонувати, пристосовуючись нових вимог [3].

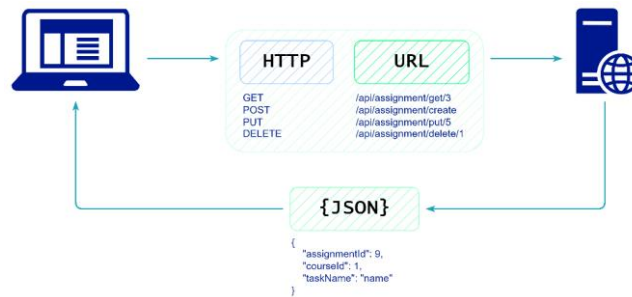


Рисунок 3 – Взаємодія компонентів системи при використанні REST

Висновки

У результаті аналізу архітектури програмного забезпечення та порівняння трирівневої та мікросервісної архітектур вдалося зробити кілька важливих висновків, що визначають ефективність та придатність обох підходів для розробки програмних продуктів.

У великих проектах мікросервісна архітектура має значні переваги, такі як масштабованість та гнучкість, суттєво впливають на результативність розробки та підтримки програмного продукту. Незважаючи на визначені недоліки, правильно налаштована мікросервісна архітектура стає ефективним засобом досягнення високої ефективності та забезпечує можливість до майбутніх змін у програмних рішеннях.

В свою чергу, трирівнева архітектура пропонує більш традиційний підхід до розробки програмного забезпечення. Зокрема, простота у визначенні та розробці є важливою перевагою трирівневої архітектури. Кожен рівень відповідає за конкретний аспект функціональності, що спрощує взаємодію та обслуговування системи.

Важливим аспектом розробки програмного забезпечення є взаємодія окремих компонентів системи. Для цього можна використовувати різні архітектурні стилі, наприклад, REST. Використання REST як підходу до комунікації дозволяє ефективно вирішувати завдання взаємодії між компонентами, використовуючи HTTP протокол.

Отже, вибір між мікросервісною та трирівневою архітектурою повинен ґрунтуватися на конкретних вимогах та особливостях проекту. Мікросервісна архітектура підходить для великих, розподілених проектів, де ключовою є масштабованість та гнучкість, в той час як трирівнева архітектура може бути ефективною для менших проектів або тих, де простота розробки та обслуговування має велике значення.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Three-tier client server architecture [Електронний ресурс] – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/three-tier-client-server-architecture-in-distributed-system/>
2. Microservice architecture [Електронний ресурс] – Режим доступу до ресурсу: <https://martinfowler.com/articles/microservices.html>
3. Огляд REST [Електронний ресурс] – Режим доступу до ресурсу: <https://javarush.com/ua/groups/posts/uk.2486.ogljad-rest-chastina-1-jsho-take>

Соболев Данило Олександрович – студент групи КН-22мс, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, Вінниця, e-mail: 01-22-295.stud@vntu.vn.ua

Федишен Богдан Вікторович – студент групи КН-22мс, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, Вінниця, e-mail: 01-22-298.stud@vntu.vn.ua

Денисов Ігор Костянтинович – асистент кафедри КН, Вінницький національний технічний університет, Вінниця, e-mail: idenysov@vntu.edu.ua

Sobolev Danylo O. – student of the CS-22JS group, faculty of intellectual information technologies and automation, Vinnytsa national technical university, Vinnytsa, e-mail: 01-22-295.stud@vntu.vn.ua

Fedyshen Bohdan V. – student of the CS-22JS group, faculty of intellectual information technologies and automation, Vinnytsa national technical university, Vinnytsa, e-mail: 01-22-298.stud@vntu.vn.ua

Denisov Ihor K. – assistant of the department of computer sciences, Vinnytsa national technical university, Vinnytsa, e-mail: idenysov@vntu.edu.ua