

УДК 004.415:004.2

ДОСЛІДЖЕННЯ АРХІТЕКТУРНИХ ПІДХОДІВ ДО ПРОЕКТУВАННЯ МАЛО-ФУНКЦІОНАЛЬНИХ ВЕБ-ДОДАТКІВ НА ПЛАТФОРМІ .NET

Юдін І. О.

Науковий керівник – проф. Руткас А. Г.

Харківський національний університет радіоелектроніки, каф. ПІ

М. Харків, Україна

e-mail: illia.iudin@nure.ua

This research focuses on analyzing architectural approaches for developing small objects, implementing optimal strategies, among monolithic, microservices, Cloud-Native, and event-driven serverless architectures. Points of strength, productivity and cost-effectiveness often need to be considered, which are important to achieve maximum efficiency for small projects. Select the right architecture that plays a critical role in simplifying operations, speeding up and scaling stalls, while not considering high performance and reducing costs.

Поліпшення роботи веб-браузерів і розширення можливостей веб-технологій, розвиток технологій ШІ, підвищення рівня використання веб-додатків на мобільних пристроях зумовлює актуальність дослідження підходів до проектування веб-додатків. Можливість розміщення веб-додатків в хмарі дозволяє користувачам використовувати веб-додатки на будь-якому пристрої і в будь-якому місці, де є інтернет-з'єднання.

У результаті порівняльного аналізу методів та фундаментальних підходів для визначення оптимальних елементів для невеликих застосунків які не потребують великих інвестиції коштів та часу буде сформовано оптимальний патерн для побудови компактних застосунків.

Важливим завданням є виявлення критеріїв для вибору архітектурних патернів. Архітектура простого, мало-функціонального застосунку повинна відповідати таким критеріям:

- архітектура повинна бути простою і легко зрозумілою для розробників та адміністраторів;
- має бути простим у розгортанні та підтримці без значних зусиль;
- потрібно мати можливість легко масштабувати систему в майбутньому при необхідності;
- потрібно гарантувати, що архітектура відповідає вимогам функціональності застосунку;
- архітектура повинна сприяти швидкій розробці, тестуванню та впровадженню нового функціоналу;
- важливо, щоб архітектура мала низьку загальну вартість власності протягом усього життєвого циклу застосунку.

Реалізація наукового дослідження складається з наступних етапів:

- аналіз предметної області;

- аналіз архітектурних особливостей сучасних архітектурних патернів;

- знаходження фундаментальних закономірностей; порівняльний аналіз та модифікація існуючих рішень.

У дослідженні було проаналізовано різні архітектурні підходи, зокрема Cloud-Native, мікросервісну, монолітну, та Event-driven serverless архітектури. Cloud-Native підхід забезпечує гнучкість, швидке розгортання, ефективне використання ресурсів та відмінну витривалість, але має складності в управлінні та високі вимоги до безпеки та навичок.

Мікросервісна архітектура пропонує незалежне масштабування та спрощене оновлення, але супроводжується проблемами управління та можливими загрозами безпеки.

Монолітна архітектура ідеальна для невеликих проєктів з простим розгортанням, але обмежена в масштабуванні та оновленні. Event-driven serverless архітектура ефективна для відповіді на події, але може мати обмеження залежно від хмарного провайдера[2].

На основі аналізу було визначено що оптимальним рішенням для проєкту буде поєднання монолітної архітектури з Cloud-based architecture. Cloud-based architecture (архітектура, заснована на хмарних технологіях) – це підхід до розробки програмного забезпечення, в якому різні компоненти застосунку розгортаються та працюють у хмарному середовищі.

Замість того, щоб мати власні фізичні сервери та обладнання, розробники використовують послуги хмарних провайдерів, таких як Amazon Web Services (AWS), Microsoft Azure або Google Cloud Platform (GCP).

Це дозволяє розробникам швидко масштабувати застосунки, оптимізувати витрати на інфраструктуру та забезпечувати високу доступність та надійність системи.

Cloud-based architecture також дозволяє легко інтегрувати інші сервіси та засоби хмарної платформи, такі як бази даних, сховища даних, інструменти аналізу даних та багато іншого.

Таке поєднання забезпечує простоту та централізованість монолітного підходу з гнучкістю та масштабованістю хмарних технологій.

Воно дозволяє ефективно використовувати ресурси хмари для масштабування та оптимізації продуктивності, одночасно зберігаючи переваги єдиного кодового базису та спрощеного процесу розробки, характерних для монолітної архітектури.

Приблизний вигляд даної архітектури зображено на рисунку 1.

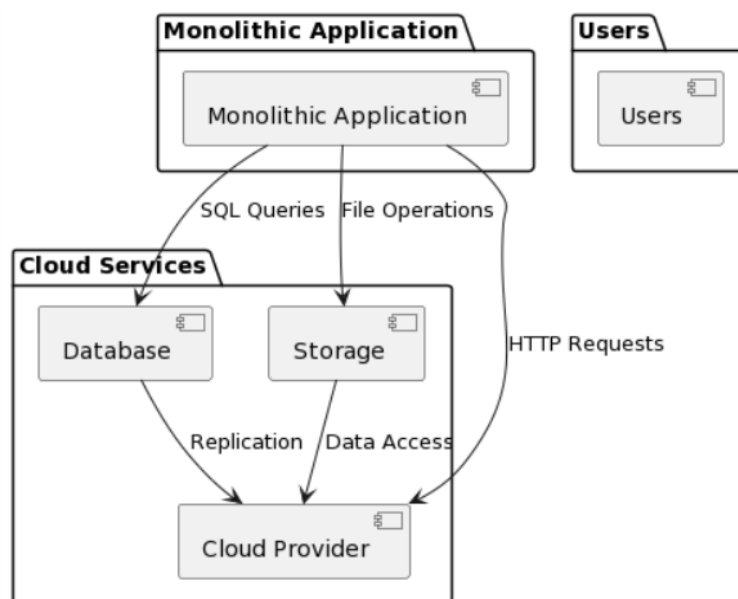


Рисунок 1 – Поєднання монолітної із Cloud-based архітектури

Замість того, щоб самостійно виконувати всі функції, буде можливим використовувати хмарні послуги для певних компонентів додатку. Наприклад, можна використовувати хмарні послуги для автентифікації, кешування, моніторингу тощо. Варто відзначити також додаткові елементи які покращать роботу подібної архітектури:

- CDN (Content Delivery Network) та кешування для покращення продуктивності та швидкості вашого монолітного додатку, особливо для статичного контенту;
- інфраструктурні сервіси хмари, такі як масштабовані сервери, балансувальники навантаження та автоматизація, для оптимізації роботи вашого монолітного додатку в хмарі;
- хмарні інструменти для моніторингу та логування, щоб відстежувати та аналізувати роботу вашого монолітного додатку в реальному часі;
- хмарні інструменти автоматизації для розгортання, масштабування та керування інфраструктурою вашого монолітного додатку.

Список використаних джерел:

1. WhatIsWebArchitecture (WithDefinitionandTips) 2023. Indeed. URL: <https://www.indeed.com/career-advice/finding-a-job/what-is-web-architecture> (дата звернення: 16.05.2023).
2. Архітектура програмної системи підтримки вивчення англійської мови із використанням рукописних текстів, Широкопетлева М. С., ХНУРЕ, 2021.: <https://openarchive.nure.ua/entities/publication/72f90dc3-fc16-4798-9e58-452c0ccbfb6f> (дата звернення: 01.02.2024).