

С. Хом'як // Матеріали XII Міжнародної науково-практичної конференції молодих учених та студентів „Актуальні задачі сучасних технологій“, 6-7 грудня 2023 року. Тернопіль : ФОП Паляниця В. А., 2023. С. 408–409.

УДК 004.415

П.М. Семчишин, студент гр. СНм-61

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

АРХІТЕКТУРНІ РІШЕННЯ ДЛЯ РОЗРОБКИ ВЕБ-ЗАСТОСУНКІВ

P.M. Semchyshyn

ARCHITECTURAL SOLUTIONS FOR WEB APPLICATION DEVELOPMENT

Коли йдеться про проектування та розробку веб-застосунків, при виборі архітектурного рішення є два основні варіанти – монолітна або мікросервісна архітектури.

Монолітна архітектура в веб-застосунках є підходом, при якому весь функціонал програми виконується в одному цілому, зазвичай у вигляді єдиного виконуваного файлу або програми [1]. Основні компоненти програми, такі як інтерфейс користувача, бізнес-логіка та доступ до даних, містяться всередині однієї програми.

Перевагами такого виду архітектури є:

- простота розгортання - оскільки всі компоненти знаходяться в одному місці, розгортання програми, зазвичай, відбувається швидше та простіше;
- простота розробки - відсутність складної інфраструктури полегшує процес розробки та налагодження програми;
- зручність масштабування на початку - при невеликому обсязі трафіку користувача монолітний застосунок може забезпечити достатню продуктивність без необхідності поділу на окремі сервіси.

Недоліки монолітної архітектури:

- складність підтримки - при збільшенні розміру програми та додаванні нових функцій може виникнути складність у підтримці та зміні коду через його єдиний монолітний характер;
- обмежена масштабованість - при досягненні межі продуктивності або необхідності масштабування окремих компонентів програми можуть виникнути труднощі через їх взаємозв'язок усередині моноліту.
- обмежена гнучкість - через єдиний характер застосування зміни в одній його частині можуть торкнутися інші компоненти, що ускладнює зміни та впровадження нових технологій.

Монолітна архітектура може бути хорошим вибором для простих застосунків з невисоким навантаженням та невеликими вимогами до масштабованості та гнучкості [1].

Мікросервісна архітектура у веб-застосунках є підходом, у якому застосунок розбивається на невеликі автономні сервіси, кожен із яких відповідає окремому функціоналу. Кожен сервіс, зазвичай, розгортається та масштабується незалежно [2].

Переваги такої архітектури:

- гнучкість та масштабованість - завдяки поділу програми на незалежні сервіси, кожен з них можна масштабувати та оновлювати незалежно від інших компонентів, що покращує гнучкість та масштабованість всієї програми;

- легкість підтримки та розробки - кожен сервіс меншого розміру та має чітко визначені межі відповідальності, що полегшує підтримку та розробку програми;
- технологічна різноманітність - різні сервіси можуть бути написані різними мовами програмування або використовувати різні технології, що дозволяє вибирати найбільш потрібні інструменти для кожного конкретного завдання.

Проте є і недоліки:

- складність конфігурації та управління - управління великою кількістю незалежних сервісів вимагає наявності складної інфраструктури та механізмів управління, що може спричинити додаткові складності та витрати;
- мережева затримка - оскільки кожен сервіс взаємодіє з іншими через мережу, виникає затримка, яка може вплинути на загальну продуктивність програми;
- складність тестування - тестування мікросервісів може бути складним через необхідність керування їх взаємодією та залежностями [2].

Таким чином, мікросервісна архітектура часто застосовується у великих та складних застосунках, де потрібна висока гнучкість, масштабованість та можливість використання різних технологій.

Якщо підбити підсумок щодо архітектурних рішень, то можна виділити те, що головний плюс мікросервісної архітектури - це велика гнучкість і масштабованість. Розбиття програми на окремі сервіси дозволяє розробляти і впроваджувати новий функціонал швидше і ефективніше, оскільки зміни в одному сервісі не торкаються інших. Це особливо важливо для сучасних веб-застосунків, які передбачають високу динамічність та швидке впровадження нових можливостей.

Використання мікросервісної архітектури також може допомогти прискорити розробку, оскільки робота над кожним сервісом ведеться незалежно і може виконуватися різними людьми/командами в різні терміни.

Кожен мікросервіс може бути масштабований незалежно від його навантаження, що забезпечує значно ефективніше застосування обчислювальних ресурсів і збільшує надійність усієї системи. Але також дана архітектура має багато нюансів, які можуть викликати проблеми. Наприклад, складність побудови інфраструктури та складність управління всім, можуть (але не обов'язково) виникнути проблеми з безпекою, так як доводиться часто передавати дані між сервісами, відповідно зростає потенційний ризик витоків інформації, ну і той факт, що дані розділені за різними процесами всієї інформації складніше.

Що ж до монолітної архітектури, то всіх цих мінусів можна уникнути, але така архітектура зручна рівно до певних розмірів застосунка, далі його стає досить проблематично підтримувати, є такий термін “Big Ball of Mud” (позначає ситуацію, коли застосунок стає надто складним і важким для підтримки та розвитку, їх складно розділити на більш дрібні та незалежні частини). Що стосується масштабованості, то там і зовсім є межа.

Способи взаємодії мікросервісів є ключовим аспектом при розробці веб-застосунків на основі мікросервісної архітектури. Якщо за монолітної архітектури інформація може легко передаватися між різними класами всередині застосунку, то у мікросервісах її треба відправляти до інших систем. Для забезпечення ефективної роботи системи необхідно вибрати відповідні методи взаємодії, які відповідатимуть вимогам проекту та забезпечуватимуть високу продуктивність, надійність та масштабованість

Література

1. Що краще моноліт чи мікросервіси? Як обрати архітектуру проекту?
URL: <https://iampm.club/ua/blog/shho-krashhe-monolit-chi-mikroservisi-yak-obrati>

[arhitekturu-projektu/](#) (дата звертання: 03.11.2025).

2. Що таке мікросервісна архітектура: шлях до гнучкого та масштабованого середовища розробки. URL: <https://blog.colobridge.net/uk/2024/01/what-is-microservices-architecture-ua/> (дата звертання: 04.11.2025).

УДК 621.865.8

С.О.Синишен-студент групи TP-102

Науковий керівник: Недошитко Л.М., викладач-методист

(Відокремлений структурний підрозділ “Тернопільський фаховий коледж”

Тернопільський національний технічний коледж імені Івана Пулюя, Україна)

МІКРО- ТА НАНОРОБОТИ: ТЕХНІЧНІ ВІДМІННОСТІ, ПЕРЕВАГИ ТА НЕДОЛІКИ MPI I GDI

S.O.Synyshen-student of TP-102 group

Scientific supervisor: Nedoshytko L.M. , teacher-methologist

MICRO- AND NANOROBOTS: TECHNICAL DIFFERENCES, ADVANTAGES AND DISADVANTAGES OF MPI AND GDI TECHNOLOGIES

Ідея створення мікро- та нанороботів з’явилася ще у 1950-х роках, коли вчені почали уявляти машини, здатні працювати на рівні клітин або молекул.

Першим цю ідею популяризував Річард Фейнман у лекції «Там унизу ще багато місця» (1959), де він передбачив технології, які дозволять створювати мікромашини, що зможуть працювати всередині організму.

У 2000-х роках, завдяки розвитку нанотехнологій, 3D-друку, біоінженерії та молекулярної фізики, вчені почали створювати перші діючі прототипи нанороботів. Сьогодні ці технології активно розвиваються в медицині, хімії, електроніці та промисловості.

Мікро- та нанороботи (MNRs) — це високотехнологічні пристрої розміром від кількох мікрометрів до нанометрів, здатні виконувати цільові завдання в живих організмах або матеріалах. Вони працюють на основі хімічних, магнітних, ультразвукових чи біогібридних принципів руху.

MPI (Magnetic Particle Imaging) — це сучасний метод візуалізації, який дозволяє відстежувати мікро- та нанороботи всередині організму з дуже високою точністю. На відміну від МРТ, MPI «бачить» лише магнітні наночастинки, тому дає чітке зображення їхнього руху без шумів. Цю технологію застосовують у біоробототехніці, щоб точно контролювати переміщення нанороботів у кровоносних судинах або тканинах.

Таблиця 1. Переваги і недоліки MPI

MPI	Переваги	Недоліки
MPI (Magnetic Particle Imaging) — це сучасний метод візуалізації, який дозволяє відстежувати мікро- та нанороботи всередині організму з дуже високою точністю.	Висока точність у доставленні препаратів до уражених клітин. Мінімальна інвазивність процедур. Можливість автономного керування за допомогою штучного інтелекту. Використання в	Висока вартість розробки та виробництва. Труднощі контролю в складних біологічних середовищах. Потенційна токсичність наноматеріалів.