

УДК 004.41

Тіменко А.В.¹, Вусу-Гову С.Е.²

¹ асист. НУ «Запорізька політехніка»

² студ. гр. КНТ-519м НУ «Запорізька політехніка»

МІКРОСЕРВІСНА АРХІТЕКТУРА ЯК СПОСІБ РОЗШИРЕННЯ ТА ВДОСКОНАЛЕННЯ МОНОЛІТНОГО ДОДАТКУ ДЛЯ ПИСЬМЕННИКІВ

Термін «Microservice Architecture» набув поширення в останні кілька років як опис способу дизайну додатків у вигляді набору незалежно розгорнутих сервісів. У той час як немає точного опису цього архітектурного стилю, існує якийсь загальний набір характеристик: організація сервісів навколо бізнес-потреб, автоматичне розгортання, перенос логіки від шини повідомлень до приймачів (endpoints) і децентралізований контроль над мовами і даними.

Використання мікросервісної архітектури надає розробнику наступні можливості щодо розширення проекту:

- модулі можна легко замінити в будь-який час: акцент на простоту, незалежність розгортання та оновлення кожного з мікросервісів;
- модулі організовані навколо функцій: мікросервіс по можливості виконує тільки одну досить елементарну функцію;
- модулі можуть бути реалізовані з використанням різних мов програмування, фреймворків, сполучного програмного забезпечення, виконуватися в різних середовищах розробки; виконуватися в різних середовищах контейнеризації, віртуалізації, під керуванням різних операційних систем на різних апаратних платформах
- архітектура симетрична, а не ієрархічна: залежності між мікросервісами однорангові.

Філософія мікросервісів фактично копіює філософію Unix, згідно з якою кожна програма повинна «робити щось одне, і робити це добре» і взаємодіяти з іншими програмами простими засобами.

Для організації мікросервісної архітектури на базі монолітної будуть використані наступні технології та програмне забезпечення, які використовуються для створення клієнт-серверної архітектури на стеку JavaScript технологій: серверна частина складається з Node.js, Koa, MongoDB, Redis; клієнтська була розроблена за допомогою таких технологій як React.js, React-Bootstrap, React Redux, React Router, React Router DOM. Для комунікації мікросервісних модулів було використано open source message broker NATS - система обміну повідомленнями з відкритим вихідним кодом. Сервер NATS написаний на мові програмування Go. На рис.1 представлена

робота та структура системи.

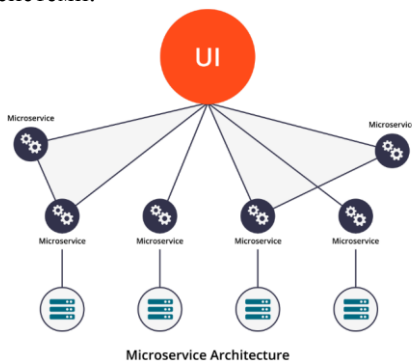


Рисунок 1 – Структура та функціонування мікросервісної системи

Представлена система для розширення монолітної архітектури є кросплатформенною та працює під керуванням операційних систем Windows та Unix/Linux. Робота системи на цих браузерах була перевірена практичним шляхом.

У порівнянні зі старою версією програми мікросервісне вдосконалення надає додатку більш короткий час запуску і можливість розгортання мікросервісів незалежно один від одного, що є дійсно вигідні для CI / CD. Мікросервіси не прив'язані до технології, яка використовується в інших сервісах. Це означає що є можливість використовувати кращі технології підгонки. Старі сервіси можуть бути швидко переписані для використання нових технологій. Саме можливість швидко адаптуватися до нових технологій може ще довго залишати різні проекти актуальними, та розширювати їх потенціал та можливості.

Таким чином була створена мікросервісна система для створення та читання художніх творів, яка складається із серверної та клієнтської частини. На відміну від старої версії додатку, нова характеризується більшою швидкістю обробки даних, має безліч варіантів вдосконалення та є простою в обслуговуванні

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Ньюмен, С. Создание микросервисов [Текст] / Ньюмен С. – СПб. : БХВ-Петербург, 2018. – 304 с.
2. dev.to [Електронний ресурс]. – Режим доступу: <https://dev.to/t/webdev>