

ОГЛЯД ПІДХОДІВ ПОБУДОВИ ТА ПРИКЛАД РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ

Інформаційно-аналітичні системи (ІАС) у сучасних умовах виконують ключову роль у підтримці управлінських рішень, особливо в контекстах кризових подій, де потрібна оперативна обробка великого обсягу даних. Протягом останнього десятиліття в архітектурі ІАС спостерігається перехід від монолітних централізованих рішень до модульних, децентралізованих платформ з відкритими інтерфейсами обміну даними [1]. Найпоширенішими архітектурними підходами є тришарова архітектура, мікросервісна архітектура, а також гібридні моделі. Тришарова архітектура передбачає поділ системи на три логічно незалежні рівні: презентаційний, бізнес-логіки, доступу до даних. Перевагами тришарової архітектури є чітке розділення відповідальності між рівнями. Однак цей підхід має обмеження у масштабованості та гнучкості в умовах зростаючих обсягів даних і підвищених вимог до швидкості їх обробки. Мікросервісна архітектура базується на ідеї поділу системи на набір незалежних сервісів, кожен з яких виконує одну конкретну бізнес-функцію і взаємодіє з іншими через чітко визначені інтерфейси (зазвичай REST або GraphQL). Цей підхід є більш гнучким і краще масштабується, оскільки є можливість проводити розробку і оновлення модулів незалежно один від одного. Однак через таку розподілену структуру значно зростає складність впровадження та підтримки такої системи, оскільки потрібно налаштування взаємодії між усіма сервісами та слідкувати за їх роботою окремо [2]. Гібридні архітектури об'єднують кращі риси попередніх підходів, використовуючи мікросервіси для певних функцій, де потрібно забезпечити високу масштабованість та незалежність, але зберігаючи централізовану модель для інших частин системи. До особливостей гібридних архітектур також належить підтримка потокової (streaming) аналітики - тобто обробки даних у режимі реального часу або з мінімальною затримкою, одразу в момент їх надходження [3, 4].

На основі аналізу сучасних архітектурних підходів була реалізована власна інформаційно-аналітична система для збору, обробки та візуалізації даних щодо руйнувань цивільної інфраструктури та об'єктів культурної спадщини внаслідок повномасштабної війни в Україні. Зважаючи на вимоги до масштабованості, швидкої обробки великого обсягу даних, зручної аналітики та інтерактивності, було прийнято рішення про реалізацію гібридної архітектури, що поєднує елементи тришарової моделі, REST-

інтерфейсів та окремих модулів для аналітики в реальному часі [5]. Основу системи складає класична тришарова структура: клієнтський інтерфейс (frontend), серверна логіка (backend) та рівень зберігання даних (база даних). Клієнт реалізований у вигляді SPA (Single Page Application) із використанням фреймворку Vue.js, що забезпечує динамічну взаємодію користувача з даними, зокрема - з інтерактивною мапою пошкоджень. Серверна частина реалізована на PHP-фреймворку Laravel, що виконує роль API-шлюзу, відповідає за авторизацію, маршрутизацію запитів, обробку бізнес-логіки та формування відповіді у форматі JSON.

Для забезпечення аналітичної складової в систему було інтегровано OLAP-аналітику, реалізовану на рівні backend за допомогою спеціалізованих запитів до агрегованих таблиць. Аналітика підтримує фільтрацію за кількома вимірами - час, категорія об'єкта, тип, регіон, ступінь пошкодження, та дозволяє оперативно отримати ключові метрики: кількість зруйнованих об'єктів і вартість їх відновлення. Дані для аналізу виводяться у вигляді динамічної таблиці з можливістю зміни виміру та додаткової фільтрації. Окрім цього, архітектура системи передбачає потенційне розширення за рахунок модулів потокової аналітики, що дозволить у перспективі автоматично обробляти вхідні дані (наприклад, нові повідомлення про пошкодження) у режимі реального часу без необхідності попереднього збереження в базу. Такий підхід дозволяє гнучко адаптуватися до нових сценаріїв використання, де потрібна оперативна реакція на події - зокрема для оновлення дашбордів, інформування відповідальних осіб, або запуску автоматичних сценаріїв реагування.

Завдяки поєднанню переваг різних архітектурних підходів, розроблена система досягає високої гнучкості, масштабованості та зручності для користувача. Водночас, вона залишається доступною для підтримки та подальшого розвитку, оскільки побудована на популярних фреймворках із відкритим кодом.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Анісімов В. Перехід від монолітної до мікросервісної архітектури: методологія та досвід впровадження / В. Г. Анісімов, Н. Е. Кунанець // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. - 2023. - № 45. - С. 50–64.
2. The rise of “big data” on cloud computing: Review and open research issues / I. Hashem [et al.] // Information Systems. - 2015. - Vol. 47. - P. 98-115. - DOI: <https://doi.org/10.1016/j.is.2014.07.006>
3. Нефьодов Д. А. Мікросервісна архітектура системи потокової обробки великих даних / Д. А. Нефьодов, С. Г. Удовенко, Л. Е. Чала // АСУ та прилади автоматики : всеукраїнський міжвідомчий науково-технічний

збірник. – Харків : ХНУРЕ. – 2022. – Вип. 178. – С. 50–64. - DOI: <https://doi.org/10.30837/0135-1710.2022.178.050>

4. Подійно-орієнтована архітектура системи збору та обробки даних / В. О. Кузьмініх [та ін.] // Телекомунікації та інформаційні технології. – 2023. - № 1(2). - С. 45–59. - DOI: <https://doi.org/10.31673/2412-9070.2023.012126>

5. Analytics module for the system for recording destruction due to russian aggression / A. Bakurova [et al.] // XVII International Scientific Conference "Monitoring of Geological Processes and Ecological Condition of the Environment", 7 - 10 November 2023. - Kyiv, 2023, - P. 1 – 5 - DOI: <https://doi.org/10.3997/2214-4609.2023520232>