

## АНАЛІЗ ПРИНЦИПІВ СТВОРЕННЯ АРІ У ХМАРНІЙ ІНФАСТРУКТУРІ

Бондар Н.О., Колтун Ю.М.

e-mail: [nataliia.bondar@nure.ua](mailto:nataliia.bondar@nure.ua); e-mail: [yurii.koltun@nure.ua](mailto:yurii.koltun@nure.ua)

Харківський національний університет радіоелектроніки, каф. ІМІ  
м. Харків, Україна

A key solution for ensuring seamless integration and interaction in cloud infrastructure is the API. APIs provide a standardized set of protocols and tools for developing, integrating, and managing software applications, enabling efficient communication between distributed services. This paper analyzes the fundamental principles of API development in cloud ecosystems, focusing on best practices for creating scalable, secure, and reliable services, and explores the advantages of API-driven architectures, including cost optimization, enhanced interoperability, and reduced development complexity. A case study of AWS API Gateway illustrates how managed API services facilitate the design, deployment, and security of cloud-based applications.

Сучасний динамічний розвиток інформаційних технологій, зокрема хмарних сервісів, вимагає чітко визначених принципів побудови, взаємодії та обміну інформацією між учасниками цифрових екосистем. Основною проблемою є забезпечення узгодженого, стандартизованого протоколу комунікації між елементами інфосистем, що є локально встановленими, гібридними або розгорнуті в хмарній інфраструктурі, незалежно від їх кількості, провайдера, типу сервісів та функцій, що ними виконуються. Для такої узгодженої підтримки комунікації між відповідними сервісами, додатками і службами зазначених інфосистем широке застосування отримали інтерфейси АРІ.

Прикладний програмний інтерфейс (Application Programming Interface, АРІ) – це узгоджений набір запитів, методів, протоколів і засобів для створення та інтеграції програмного забезпечення. Він надає розробникам інструменти для швидкої реалізації нових додатків і ефективного використання рішень, що вже існують. Залежно від джерела даних, відповідне програмне забезпечення АРІ виконує їх впорядкування і нормалізацію, що забезпечує уніфіковане використання інформації незалежно від сервісної платформи чи інфраструктури [1].

Зокрема у роботі проведений аналіз існуючих загальноприйнятих практик і принципів розробки, взаємодії та інтеграції АРІ щодо створення надійних, масштабованих та безпечних хмарних застосунків. В процесі досліджень практично було використано сервіс хмарної інфраструктури компанії Amazon AWS API Gateway, який надає АРІ як готовий до застосування сервіс. Amazon API Gateway являє собою керований сервіс, який допомагає створювати, публікувати, підтримувати, моніторити та

забезпечувати безпеку API різних типів відповідно до потреб бізнесу та функціональності [1].

У роботі проаналізовані можливості API щодо реалізації взаємодії різних компонентів програмної екосистеми для забезпечення ефективного обміну даними. Показано, що він не лише виконує функції внутрішньої інтеграції, а й забезпечує зовнішній доступ до сервісів і додатків хмарної інфраструктури. Інтерфейс діє як зв'язуючий компонент між додатками, приховуючи їх внутрішню реалізацію. Такий підхід дозволяє взаємодіяти без глибокого розуміння архітектури кожного окремого сервісу. Основна мета API ґрунтується на принципах інкапсуляції, тобто приховування частини функцій для спрощення архітектури та зменшення ризику помилок під час взаємодії між сервісами і додатками хмарної інфраструктури [2, 3].

Проведений аналіз був спрямований на формування цілісного підходу до використання інтерфейсів для підвищення якості обробки даних, спрощенню архітектури хмарної інфраструктури, зменшенню витрат на розробку та підтримку її сервісів і додатків, підвищенню рівня інтеграції її інформаційних систем та їх відповідності сучасним вимогам до безпеки.

Запропонований у роботі комплексний підхід із дотриманням принципів безпеки, доступності та конфіденційності дозволить забезпечити стабільне виконання інтеграційних процесів у сучасних інформаційних системах. Використання загальноприйнятих практик (методологій проектування, рекомендацій щодо реалізації, прикладів коду) допомагає створювати надійні та масштабовані хмарні сервіси і додатки. Переваги використання API включають економію часу на розробку нових сервісів, забезпечення передбачуваності та стандартизації взаємодії, можливість відкриття API для сторонніх розробників, підтримку незалежності через використання стандартних протоколів, оптимізацію витрат.

Кожен варіант побудови API чи використання керованих сервісів має свої особливості, переваги та недоліки. Оптимальний вибір залежить від потреб хмарної інфраструктури, архітектури та вимог замовника.

#### Список використаних джерел:

1. Best Practices for Designing Amazon API Gateway Private APIs and Private Integration [Електронний ресурс]. – Доступ здійснено 28.02.2025. – Режим доступу: <https://docs.aws.amazon.com/whitepapers/latest/best-practices-api-gateway-private-apis-integration/best-practices-api-gateway-private-apis-integration.html>.
2. Kanjilal J., ASP.NET Web API: Build RESTful web applications and services on the .NET framework, 2013.
3. Joshua S. Ponelat, Lukas L. Rosenstock, Designing APIs with Swagger and OpenAPI, 2022.