

Course: Digital Financial Platforms and FinTech Systems

Lecture 2. Architecture of Digital Financial Platforms

Lecturer: Polishchuk Inna, PhD.

Курс: Цифрові фінансові платформи та FinTech-системи

Лекція 2. Архітектура цифрових фінансових платформ

Лектор: Поліщук Інна, PhD.

Зміст

- Концепція архітектури цифрових фінансових платформ
- Багаторівнева архітектура FinTech-платформ
- Мікросервісна архітектура у цифрових фінансових системах
- API-орієнтована архітектура та інтеграційні механізми
- Інфраструктурні технології цифрових фінансових платформ
- Архітектура управління даними у фінансових платформах
- Архітектура безпеки та управління доступом

01

Концепція архітектури цифрових фінансових платформ



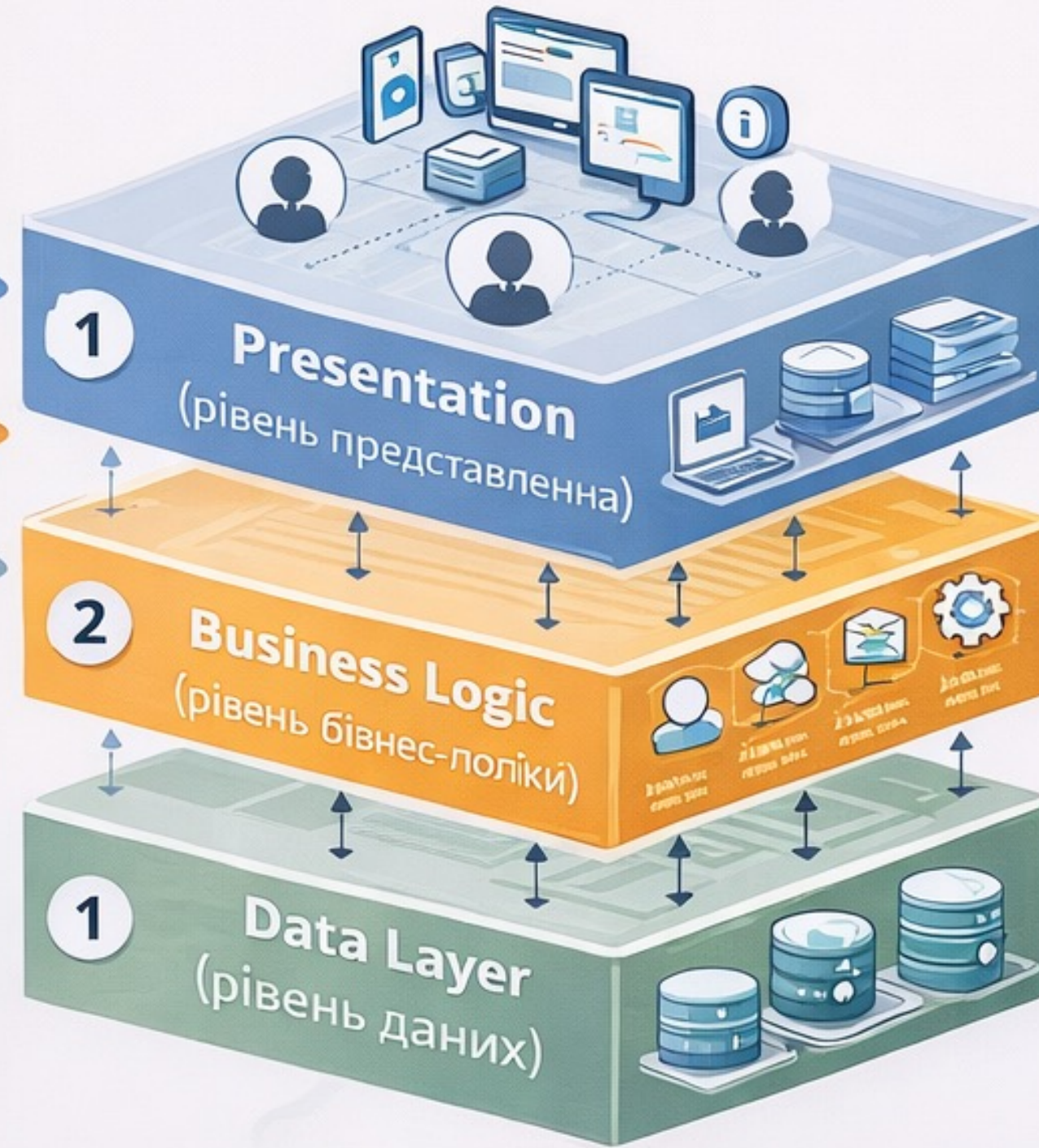
У межах цієї теми переходимо до розгляду одного з базових питань сучасних цифрових фінансів, а саме – **концепції архітектури цифрових фінансових платформ**. Для правильного розуміння функціонування FinTech-систем недостатньо знати лише, які послуги вони надають. Значно важливіше усвідомити, як саме ці системи побудовані, з яких елементів вони складаються, яким чином взаємодіють між собою їхні компоненти та за рахунок чого забезпечуються **надійність, безпека і масштабованість** фінансових сервісів [1,2,3].

Концепція архітектури цифрових фінансових платформ

Тривимірний підхід



- 1 Структурна складова
- 2 Функціональна складова
- 1 Інфраструктурна складова



Інфраструктура

Ключові принципи архітектури цифрових фінансових платформ

- Модульність
- Стандартизована взаємодія компонентів
- Багаторівнева структура
- Забезпечення безпеки
- Високонадійність і відмовостійкість
- Масштабованість і оптимізація

Ключові принципи архітектури цифрових фінансових платформ

- Модульність
- Стандартизована взаємодія компонентів
- Багаторівнева структура

Source: Розроблено автором

Архітектура цифрової фінансової платформи – це впорядкована сукупність її компонентів, зв'язків між ними, принципів організації та правил взаємодії, які забезпечують реалізацію фінансових послуг у цифровому середовищі. Іншими словами, архітектура визначає не стільки перелік функцій платформи, скільки внутрішню логіку її побудови. Саме архітектура дає відповідь на запитання, яким чином у системі реалізовано платежі, обробку даних, доступ користувачів, інтеграцію із зовнішніми сервісами та механізми захисту інформації.

Одним із базових принципів побудови цифрових фінансових платформ є **модульність**. Під модульністю розуміють поділ складної системи на окремі функціональні блоки або модулі, кожен із яких виконує визначене завдання. Наприклад, один модуль може відповідати за **ідентифікацію користувача**, інший – за **платіжну обробку**, ще один – за **аналіз ризиків**, а окремий компонент – за **формування звітності**. Такий підхід дозволяє зробити систему більш зрозумілою, керованою та гнучкою. Якщо виникає потреба змінити або оновити один модуль, це не повинно руйнувати роботу всієї платформи.

Наступним важливим положенням є багаторівнева організація архітектури. У найзагальнішому вигляді цифрову фінансову платформу можна подати як систему, що складається з кількох взаємопов'язаних рівнів.

- **Перший** – це рівень представлення, або presentation layer, який забезпечує взаємодію користувача із системою через вебінтерфейс, мобільний застосунок або інший клієнтський канал.
- **Другий** – це рівень бізнес-логіки, або business logic layer, де реалізуються правила виконання фінансових операцій, перевірки, розрахунки та алгоритми прийняття рішень.
- **Третій** – це рівень даних, або data layer, який відповідає за збереження, оновлення, пошук і захист інформації. Такий поділ дозволяє логічно структурувати платформу та спрощує її аналіз і проєктування.

Не менш важливим є поняття **безпеки архітектури**. Цифрові фінансові платформи працюють із конфіденційними даними: персональною інформацією користувачів, банківськими реквізитами, історією транзакцій, цифровими ідентифікаторами. Відповідно, архітектура такої системи повинна будуватися з урахуванням принципу *security by design*, тобто безпека має бути інтегрована в систему з самого початку, а не додаватися після її створення. Це передбачає використання аутентифікації, авторизації, криптографічного захисту, багатофакторної перевірки доступу та механізмів моніторингу потенційних загроз. Для фінансового сектору це не додаткова перевага, а обов'язкова вимога.

У навчальному контексті доцільно розглядати архітектуру цифрових фінансових платформ як поєднання трьох взаємозалежних складових. Перша – **структурна складова**, яка відповідає на запитання, з яких елементів побудована система. Друга – **функціональна складова**, яка пояснює, які процеси й сервіси реалізуються в межах платформи. Третя – **інфраструктурна складова**, яка визначає, на яких технологічних ресурсах функціонує система: серверному середовищі, хмарних платформах, базах даних, мережевих сервісах. Такий підхід є важливим для формування системного мислення, оскільки дозволяє майбутньому фахівцеві бачити платформу цілісно, а не лише як набір окремих функцій.

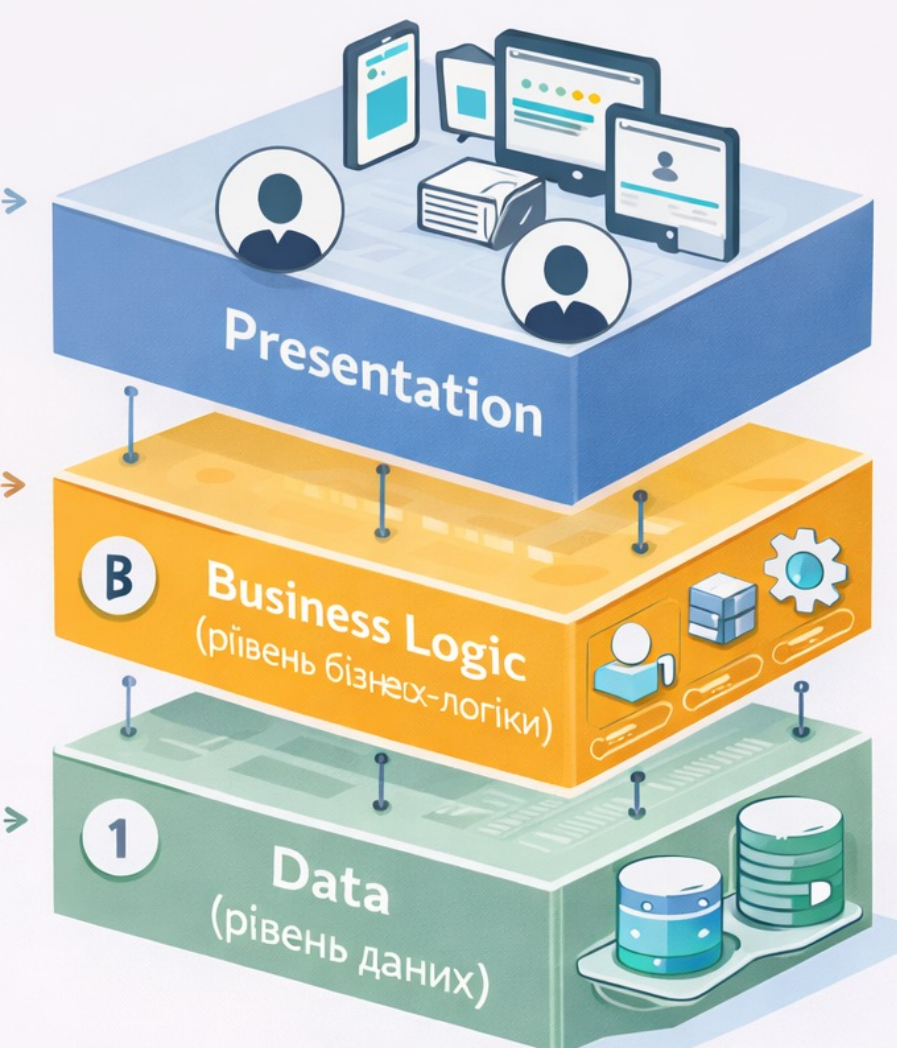
02

Багаторівнева архітектура FinTech- платформ

Presentation
(рівень представлення)

Business Logic
(рівень бізнес-логіки)

Data Layer
(рівень даних)



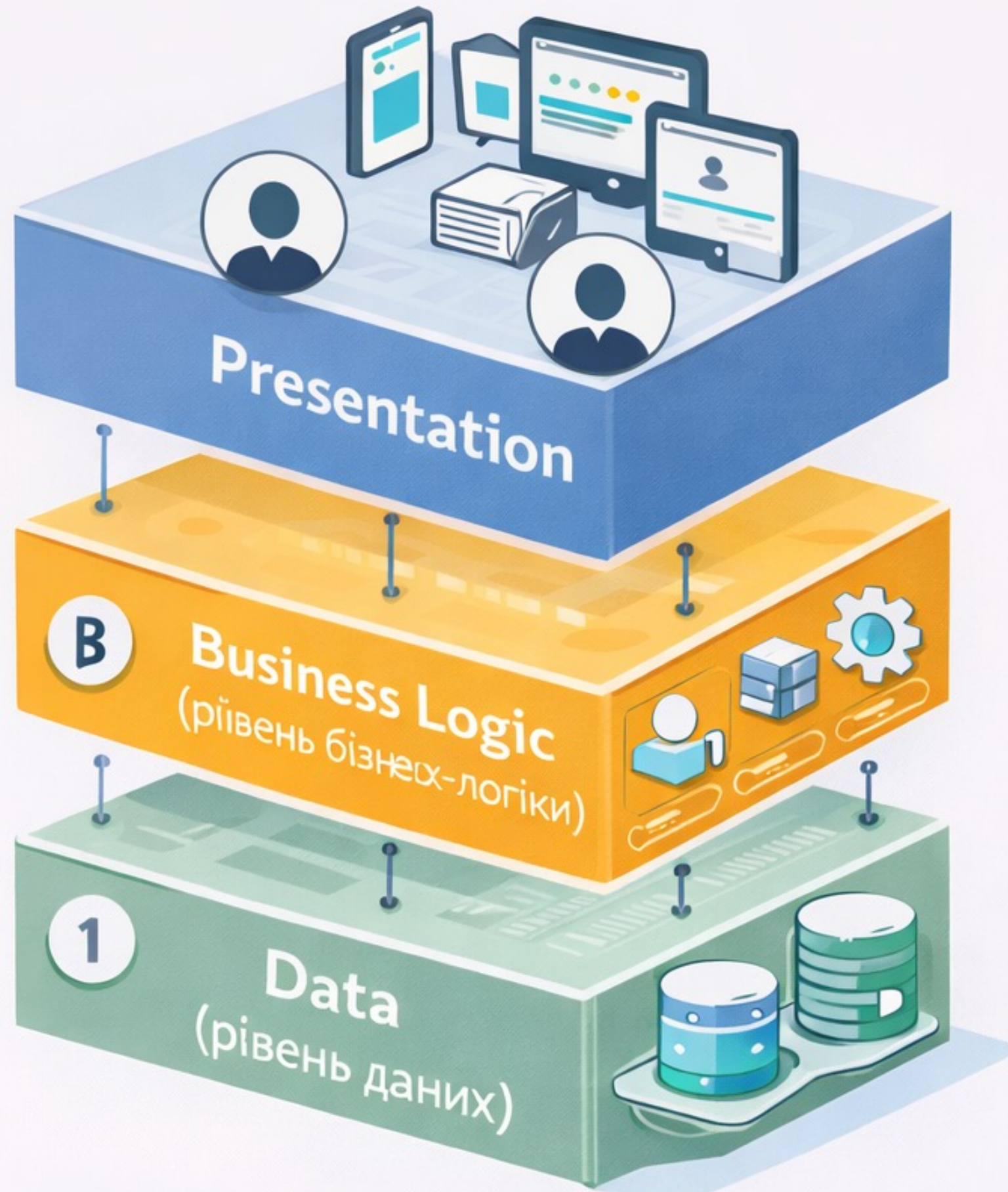
Source: Розроблено автором

Розглянемо **багаторівневу архітектуру FinTech-платформ** як одну з базових концепцій проектування сучасних цифрових фінансових систем. Для правильного розуміння роботи FinTech-рішень важливо не лише знати, які послуги вони надають, а й усвідомлювати, як структурно організовано систему, яким чином розподіляються її функції та як забезпечується узгоджена робота всіх компонентів. Саме багаторівневий підхід дозволяє впорядкувати складну платформу, зробити її зрозумілішою для розроблення, супроводу та масштабування [1,3].

Presentation
(рівень представлення) ----->

Business Logic
(рівень бізнес-логіки) ----->

Data Layer
(рівень даних) ----->



Source: Розроблено автором

Між зазначеними рівнями існує послідовна логіка взаємодії. Користувач через рівень представлення надсилає певний запит до системи. Далі цей запит надходить на рівень бізнес-логіки, де інтерпретується, перевіряється та обробляється відповідно до правил платформи. Після цього система звертається до рівня даних для отримання або зміни необхідної інформації. Результат повертається у зворотному напрямі: від даних до бізнес-логіки, а потім до інтерфейсу користувача. Така організація є важливою не лише з технічної точки зору, а й із методичної, оскільки вона допомагає студентам зрозуміти повний цикл проходження фінансової операції в цифровій системі.

У сучасних FinTech-платформах базова трирівнева модель часто доповнюється **інтеграційним рівнем**. Його призначення полягає у взаємодії із зовнішніми системами: банківськими API платіжними шлюзами, сервісами верифікації особи, кредитними бюро, аналітичними платформами та регуляторними інформаційними системами. Завдяки такому рівню платформа може не лише виконувати внутрішні операції, а й бути частиною ширшої цифрової екосистеми. Для FinTech-сфери це особливо актуально, адже сучасні фінансові послуги майже завжди базуються на обміні даними між кількома сервісами та організаціями.

вивчення багаторівневої архітектури FinTech-платформ є важливим етапом професійної підготовки студентів. Воно формує системне мислення, дозволяє перейти від сприйняття платформи як «єдиного застосунку» до розуміння її як багатокомпонентної структури з чітко розмежованими функціями. Такий підхід є особливо корисним для майбутніх розробників, системних аналітиків, архітекторів програмного забезпечення та фахівців з цифрових фінансів, оскільки саме він створює основу для подальшого вивчення **мікросервісної архітектури, API-орієнтованих моделей, хмарної інфраструктури та інструментів DevOps.**

03

Мікросервісна архітектура у цифрових фінансових системах



Source: developed by the author

У цьому параграфі ми детальніше розглянемо **мікросервісну архітектуру** як одну з ключових моделей побудови сучасних цифрових фінансових систем. Якщо раніше ми зосереджувалися на загальній логіці архітектурної трансформації FinTech-платформ, то тепер наше завдання полягає в тому, щоб зрозуміти, яким чином **мікросервісний підхід** реалізується на практиці, які можливості він відкриває та які вимоги ставить до проєктування цифрової платформи [4].

Під **мікросервісною архітектурою** розуміють підхід до побудови програмної системи, за якого вона складається з набору невеликих, автономних і функціонально спеціалізованих сервісів. Кожен із таких сервісів реалізує окрему бізнес-функцію і може розроблятися, тестуватися, розгортатися та оновлюватися незалежно від інших компонентів. У контексті цифрових фінансів це означає, що окремими сервісами можуть бути модуль платежів, сервіс керування рахунками, модуль автентифікації, система сповіщень, сервіс оцінювання ризику або аналітичний модуль.

Однією з головних переваг мікросервісної архітектури є **гнучкість розроблення**. Оскільки окремі сервіси не залежать повністю один від одного, команди розробників можуть працювати паралельно над різними частинами системи. У результаті платформа може швидше адаптуватися до нових вимог ринку, змін у регуляторному середовищі або потреб користувачів. Для фінансової сфери це має особливе значення, адже цифрові сервіси часто потребують регулярних оновлень, інтеграції нових функцій і швидкого реагування на зміни в нормативних вимогах.

Водночас **мікросервісна архітектура** ускладнює загальне керування системою. Якщо у платформі працюють десятки окремих сервісів, необхідно контролювати їхню взаємодію, стан, версії API, маршрутизацію запитів, журналювання подій та виявлення збоїв. Це означає, що мікросервісний підхід вимагає не меншої, а більшої архітектурної дисципліни. Складність переноситься з внутрішньої структури одного застосунку на рівень узгодження великої кількості розподілених компонентів.

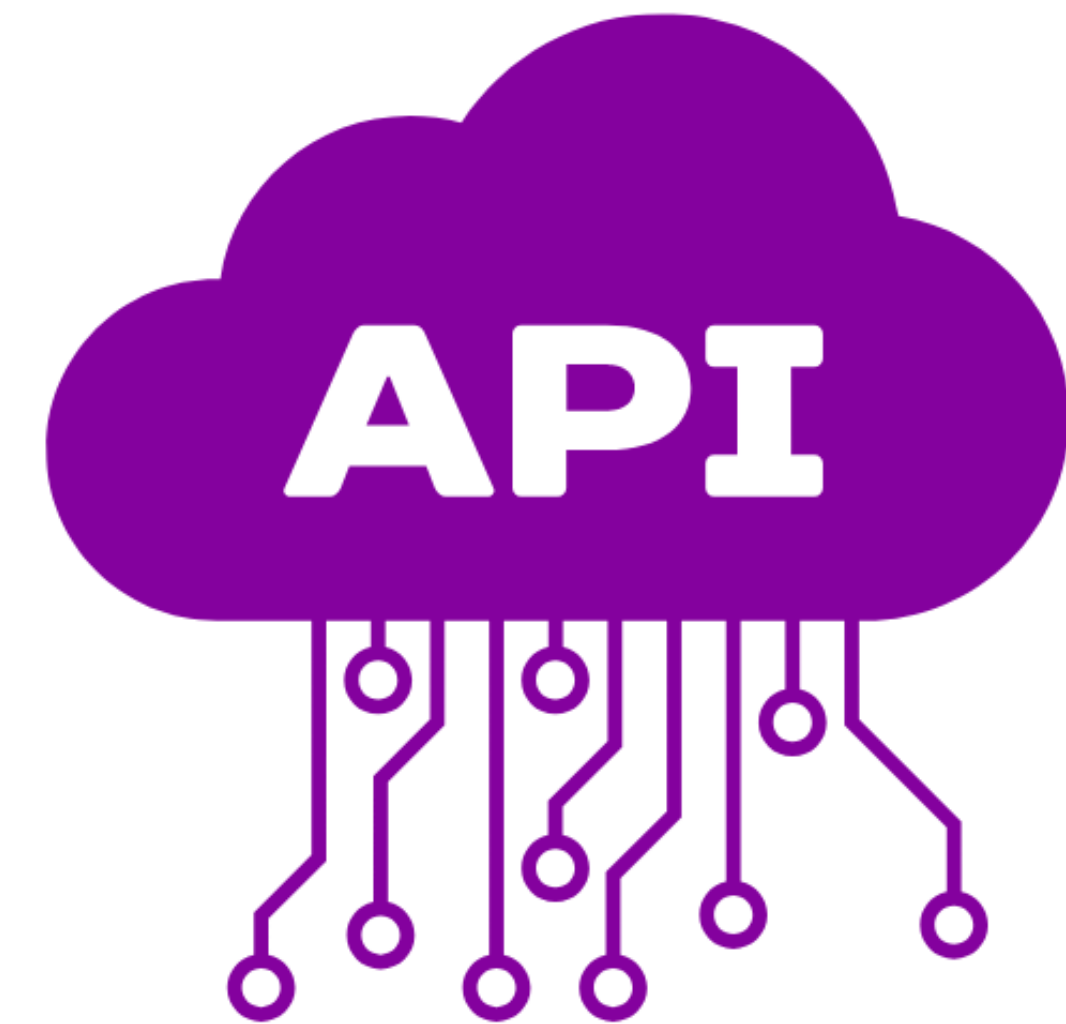
Практична реалізація мікросервісної архітектури зазвичай пов'язана з використанням **контейнеризації, оркестрації та DevOps-підходів**. Кожен сервіс може розгортатися у власному контейнері, а автоматизовані інструменти дозволяють координувати запуск, оновлення та відновлення великої кількості компонентів. У навчальному аспекті важливо розуміти, що мікросервіси – це не лише схема поділу системи на модулі, а й ціла технологічна модель експлуатації сучасної цифрової платформи, яка поєднує архітектурні, інфраструктурні та організаційні рішення.

Критерій	Переваги мікросервісної архітектури	Недоліки мікросервісної архітектури
Структура системи	Поділ системи на автономні функціональні сервіси	Підвищена складність загальної архітектури
Розроблення	Можливість паралельної роботи кількох команд	Потреба в узгодженні стандартів між командами
Оновлення	Незалежне розгортання окремих сервісів	Ризик несумісності версій API
Масштабованість	Масштабування лише навантажених компонентів	Необхідність складнішого керування ресурсами
Відмовостійкість	Локалізація збоїв в окремих сервісах	Можливі каскадні помилки при неправильній інтеграції
Інтеграція	Зручна взаємодія через API та зовнішні сервіси	Більша кількість точок обміну даними
Безпека	Можливість гнучкого розмежування доступу	Розширена поверхня потенційних атак
Дані	Автономність роботи окремих сервісів із даними	Проблеми узгодженості та синхронізації даних
Експлуатація	Гнучкість у розвитку платформи	Потреба в моніторингу, логуванні та оркестрації
Доцільність використання	Ефективна для великих і динамічних FinTech-платформ	Надмірно складна для малих і простих систем

Source: developed by the author based on [1-5].

04

API-орієнтована архітектура та інтеграційні механізми



Source: developed by the author

Розглянемо API-орієнтовану архітектуру як одну з ключових основ побудови сучасних цифрових фінансових платформ. У попередніх темах ми вже з'ясували, що FinTech-системи складаються з багатьох функціональних компонентів, які повинні працювати узгоджено, швидко та безпечно. Однак сама наявність окремих модулів ще не гарантує ефективної роботи платформи. Вирішальним стає питання, яким чином ці компоненти взаємодіють між собою, як вони обмінюються даними та як інтегруються із зовнішнім цифровим середовищем. Саме тут центральну роль відіграє API-орієнтований підхід [1,2,3].

Почнемо з базового означення.

API або Application Programming Interface – це стандартизований інтерфейс, який дозволяє одній програмній системі звертатися до функцій, даних або сервісів іншої системи.

Для цифрових фінансових платформ API-орієнтований підхід має особливе значення, оскільки фінансові сервіси майже ніколи не функціонують ізольовано. Наприклад, мобільний банкінг повинен взаємодіяти з модулем автентифікації, платіжною підсистемою, сервісом перевірки балансу, аналітичним модулем і системою сповіщень. Окрім цього, платформа часто інтегрується із зовнішніми платіжними шлюзами, кредитними бюро, сервісами електронної ідентифікації, державними реєстрами та партнерськими екосистемами. Саме тому API стає не просто технічним інструментом, а ключовим механізмом функціонування всієї фінансової інфраструктури.

У сучасних цифрових фінансових системах найпоширенішими є **REST API, webhooks, механізми асинхронного обміну повідомленнями, а також API gateway**. REST API використовується для стандартних запитів на отримання або зміну даних. Асинхронні повідомлення використовуються тоді, коли не обов'язково отримувати відповідь негайно, але важливо гарантувати доставку події чи команди. API gateway, своєю чергою, виступає єдиною точкою входу для зовнішніх і внутрішніх запитів, маршрутизує їх до відповідних сервісів і забезпечує додатковий контроль доступу, моніторинг та балансування навантаження.

Елемент	Сутність	Роль у FinTech-платформі
API	Стандартизований інтерфейс взаємодії між системами	Забезпечує доступ до функцій і даних сервісів
REST API	Архітектурний стиль обміну запитами та відповідями	Використовується для отримання, створення та оновлення даних
API Gateway	Єдина точка входу для API-запитів	Маршрутизація запитів, контроль доступу, моніторинг
Webhooks	Механізм автоматичного сповіщення про події	Передавання повідомлень про зміни стану транзакцій або операцій
Асинхронний обмін повідомленнями	Взаємодія без обов'язкової миттєвої відповіді	Підтримка подій, черг повідомлень, фонових процесів
API management	Управління життєвим циклом API	Версіонування, документація, аналітика, контроль використання
Аутентифікація та авторизація	Перевірка особи та прав доступу	Захист фінансових даних і сервісів
Інтеграційні механізми	Сукупність засобів обміну даними між системами	Забезпечення зв'язку між внутрішніми та зовнішніми сервісами

Source: developed by the author based on [1-5].

05

Інфраструктурні технології цифрових фінансових платформ



Source: developed by the author

Розглянемо **інфраструктурні технології цифрових фінансових платформ** як матеріально-технологічну основу функціонування сучасних FinTech-систем. Якщо в попередніх темах увага зосереджувалася на архітектурних підходах, рівнях системи, мікросервісах та API-взаємодії, то тепер важливо зрозуміти, на яких саме технологічних засобах ці рішення реально працюють. Іншими словами, інфраструктура відповідає на запитання не лише як організована система, а й де, у якому середовищі та за допомогою яких інструментів вона функціонує [5].

Інфраструктурні технології цифрових фінансових платформ



Під інфраструктурними технологіями доцільно розуміти сукупність апаратних, програмних, мережових і хмарних засобів, які забезпечують розгортання, підтримку, виконання та масштабування цифрової фінансової платформи. До цієї групи входять **серверні середовища, хмарні обчислення, віртуалізація, контейнеризація, системи зберігання даних, мережеві сервіси, засоби моніторингу, інструменти автоматизації розгортання та платформи для забезпечення безперервної роботи сервісів.**

Особливе місце в сучасних цифрових фінансах посідають **хмарні технології або cloud computing**. Під ними розуміють модель надання обчислювальних ресурсів, сховищ, мережевих сервісів і платформ через віддалену інфраструктуру постачальника. Для FinTech-платформ хмара є особливо важливою, оскільки вона дозволяє швидко масштабувати ресурси, гнучко керувати навантаженням і скорочувати час розгортання нових сервісів. Наприклад, якщо цифрова платформа різко отримує зростання кількості користувачів або транзакцій, хмарне середовище дає можливість оперативно розширити ресурси без фізичного оновлення серверного обладнання.

У сучасних FinTech-системах дедалі більшого значення набуває **контейнеризація**. Якщо віртуалізація створює окремі віртуальні машини, то контейнери дозволяють запускати окремі сервіси в легших ізольованих середовищах. Найчастіше у цьому контексті згадують **Docker** як інструмент пакування застосунків і їхніх залежностей у стандартизовані контейнери. Для цифрових фінансових платформ це має суттєві переваги: кожен сервіс можна швидко розгорнути, перенести між середовищами, оновити або відновити. Саме тому контейнеризація тісно пов'язана з мікросервісною архітектурою та сучасними підходами до експлуатації цифрових систем.

Окремим інфраструктурним напрямом є **технології зберігання даних**. Цифрові фінансові платформи працюють із різними типами інформації: транзакційними записами, профілями користувачів, журналами подій, аналітичними даними, налаштуваннями сервісів та резервними копіями. Відповідно, інфраструктура повинна включати бази даних, системи резервного копіювання, розподілені сховища та механізми відновлення інформації. Особливу увагу тут приділяють надійності збереження, цілісності даних та швидкому доступу до них. У FinTech-середовищі втрата або пошкодження даних є критичним ризиком, тому інфраструктурні рішення в цій сфері мають бути особливо стійкими.

Важливу роль у сучасній інфраструктурі відіграють **DevOps-підходи та автоматизація розгортання**. У цифрових фінансових системах недостатньо просто створити програмний код – необхідно забезпечити його стабільне, контрольоване та повторюване впровадження у виробниче середовище. Для цього застосовуються CI/CD-процеси, тобто безперервна інтеграція та безперервне розгортання. Вони дозволяють автоматизувати тестування, перевірку, збірку і доставку нових версій сервісів. У FinTech-сфері це особливо важливо, тому що кожне оновлення системи повинно бути не лише швидким, а й безпечним, перевіреним та сумісним із регуляторними вимогами.

06

Архітектура управління даними у фінансових платформах



Source: developed by the author

Розглянемо **архітектуру управління даними** як одну з ключових основ функціонування сучасних фінансових платформ. Саме дані забезпечують виконання транзакцій, аналітику, оцінювання ризиків, регуляторну звітність і персоналізацію сервісів. Тому архітектура управління даними визначає, як інформація збирається, зберігається, обробляється, передається та захищається у межах цифрової системи [1,2,3].

Одним із базових принципів є **розмежування операційної та аналітичної обробки даних**. **OLTP-середовище** орієнтоване на швидке виконання транзакцій у режимі реального часу, тоді як OLAP-середовище використовується для аналізу великих масивів інформації, формування звітів та підтримки управлінських рішень. Для фінансової платформи важливо, щоб ці два режими були узгодженими, але не перевантажували один одного.

Важливим напрямом є **потокова обробка даних**. Вона дає змогу аналізувати події не після їх накопичення, а безпосередньо в момент надходження. У фінансових системах це необхідно для оперативного підтвердження транзакцій, виявлення підозрілої активності та миттєвого реагування на ризикові операції.

Не менш важливим є **управління якістю даних**. Дані повинні бути повними, актуальними, узгодженими та достовірними. Для цього використовують валідацію, очищення, нормалізацію та усунення дублювання. У фінансовій сфері низька якість даних може призвести до помилок у розрахунках, звітності та прийнятті рішень.

07

Архітектура безпеки та управління доступом



Source: developed by the author

Розглянемо архітектуру безпеки та управління доступом як одну з ключових складових сучасних цифрових фінансових платформ. Для фінансової сфери це має особливе значення, оскільки платформа повинна не лише виконувати операції, а й забезпечувати захист даних, контроль доступу та стійкість до загроз.

Під архітектурою безпеки розуміють цілісну систему принципів, механізмів і технологій, що забезпечують захист платформи на всіх рівнях її функціонування. Вона охоплює користувачів, сервіси, дані, мережеву взаємодію та адміністративні інструменти. У фінансових системах безпека має бути частиною загальної архітектури, а не окремим додатком після завершення розроблення.

Базовим механізмом є **аутентифікація**, тобто перевірка особи користувача або сервісу. У сучасних фінансових платформах цього недостатньо здійснювати лише за логіном і паролем, тому застосовуються **багатофакторна аутентифікація**, одноразові коди, підтвердження через мобільний застосунок або біометричні засоби. Аутентифікація відповідає на запитання: **хто саме входить у систему.**

Для практичної реалізації авторизації часто застосовується **модель керування доступом на основі ролей** – керування доступом на основі ролей. У такій моделі права доступу закріплюються не окремо за кожною особою, а за певними ролями: клієнт, оператор, аналітик, адміністратор. Це спрощує керування доступом і зменшує ризик помилок у наданні прав.

Окремої уваги потребує захист даних. У фінансовій платформі дані повинні бути захищені як у процесі передавання, так і під час зберігання. Для цього застосовуються шифрування, захист сесій, токени доступу та безпечне зберігання облікових даних. Це забезпечує **конфіденційність, цілісність і надійність інформації.**

Підсумок лекції

Отже, у межах лекції було розглянуто, що архітектура цифрових фінансових платформ є системною основою функціонування сучасних FinTech-рішень і визначає принципи побудови, взаємодії та розвитку цифрових фінансових сервісів. Було з'ясовано, що ефективна платформа повинна мати багаторівневу структуру, у якій узгоджено функціонують рівень представлення, рівень бізнес-логіки та рівень даних. Такий підхід забезпечує чіткий розподіл функцій, спрощує супровід системи та створює умови для її подальшого масштабування.

Підсумок лекції

Таким чином, архітектура цифрових фінансових платформ повинна розглядатися як комплексна модель, що поєднує структурні, технологічні, дані-орієнтовані та безпекові аспекти функціонування системи. Розуміння цих принципів є необхідною умовою підготовки фахівців у сфері FinTech, оскільки саме архітектурне мислення дозволяє не лише аналізувати вже існуючі цифрові рішення, а й проєктувати нові фінансові платформи, здатні ефективно працювати в умовах сучасної цифрової економіки [1-5].

Пропонована література

[1]. Hensen J., & Kötting B. — From open banking to embedded finance: The essential factors for a successful digital transformation (Journal of Digital Banking).

Це наукова стаття, яка розглядає розвиток і взаємозв'язок Open Banking та embedded finance в контексті цифрової трансформації фінансового сектору.

https://www.researchgate.net/publication/382570638_From_open_banking_to_embedded_finance_The_essential_factors_for_a_successful_digital_transformation

[2]. Modesti P. et al. – Security analysis of the open banking account and transaction API protocol (International Journal of Critical Infrastructure Protection).

У статті подано формальний аналіз безпеки Open Banking API зокрема для доступу до рахунків і транзакцій. Це особливо корисне джерело для підтеми «Архітектура безпеки та управління доступом», оскільки воно показує практичні аспекти захисту фінансових API та стандартизованого обміну даними.

<https://www.sciencedirect.com/science/article/pii/S2772918425000141>

Пропонована література

[3]. Exploring and evaluating the development of an open API architecture for FinTech applications (Business Process Management Journal).

Це наукова публікація, присвячена розвитку Open API architecture у сфері FinTech.

Джерело є доречним для висвітлення тем API-орієнтованої архітектури, інтеграційних механізмів і побудови відкритих цифрових фінансових сервісів.

<https://www.emerald.com/bpmj/article/30/5/1564/1235823/Exploring-and-evaluating-the-development-of-an>

[4]. Khan A. et al. – Open banking, APIs and digital financial transformation (University of Central Lancashire / institutional PDF).

У матеріалі узагальнюється роль Open Banking, API та регуляторних підходів у розвитку сучасних цифрових фінансів. Це джерело добре підходить для лекційного пояснення того, як API стають технічною основою інтеграції фінансових платформ і сервісів.

<https://knowledge.lancashire.ac.uk/id/eprint/52899/1/52899%20Arun%20et%20al.%20VOR.pdf>

Пропонована література

[5]. Singh P. – Modernizing Legacy FinTech Systems through Cloud-Native Microservices Architecture (IJSAT, PDF).

Стаття присвячена модернізації застарілих FinTech-систем за допомогою cloud-native microservices architecture. Її доцільно використати для тем мікросервісної архітектури, інфраструктурних технологій цифрових фінансових платформ та побудови масштабованих фінансових сервісів.

<https://www.ijسات.org/papers/2024/1/6070.pdf>

Дякую за
увагу!