



**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Інформаційні системи

Викладач: к.т.н., доц. Саяпіна Інна Олександрівна

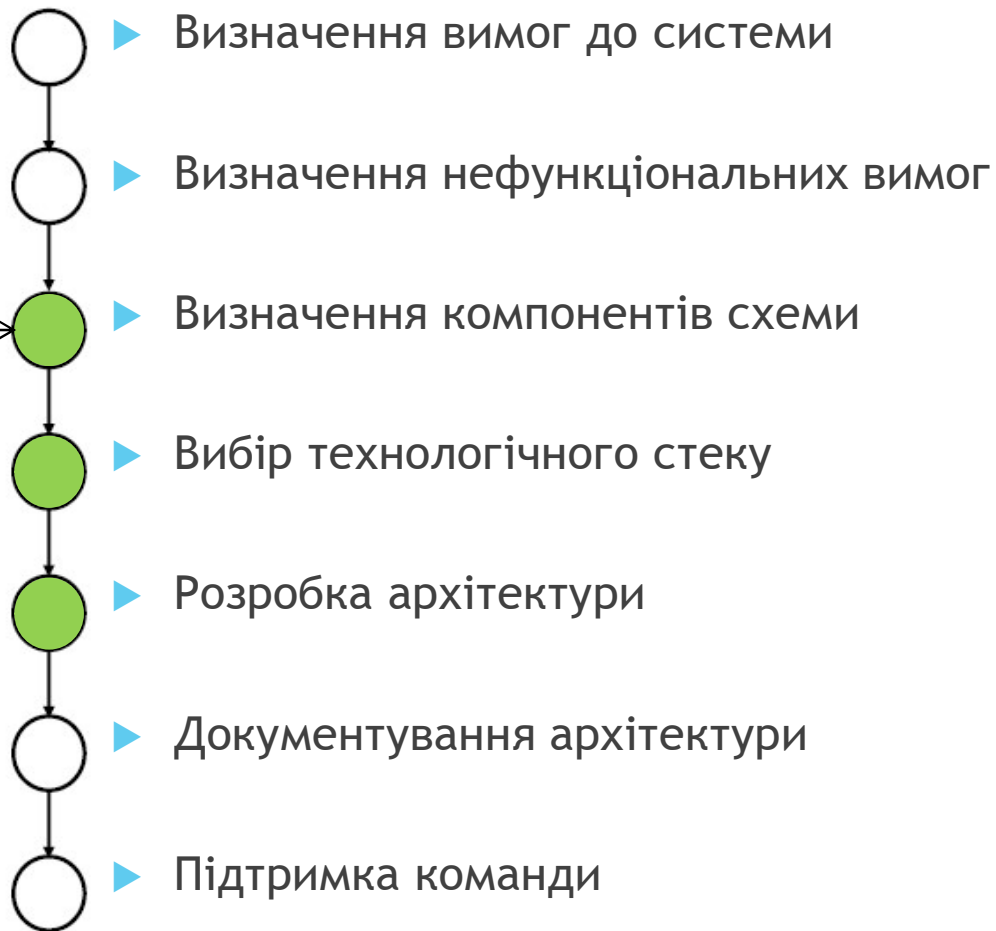
План заняття:

- ▶ Мікросервісна архітектура (Microservice Architecture):
 - ▶ Процес проєктування архітектури
 - ▶ Визначення компонентів
 - ▶ Визначення шаблонів зв'язку



Архітектурний процес

- Визначення компонентів
- Визначення шаблонів зв'язку



Визначення компонентів

- ▶ Визначення різних компонентів системи
- ▶ Пам'ятаймо: компоненти = послуги



Визначення компонентів

- ▶ Визначення компонентів має базуватися на:
 - ▶ Бізнес вимогах
 - ▶ Функціональній автономії
 - ▶ Сутностях даних
 - ▶ Автономності даних



Визначення компонентів

- ▶ Бізнес вимоги:
 - ▶ Збір вимог щодо конкретної бізнес-можливості
 - ▶ Наприклад: управління замовленнями
 - ▶ Додати, видалити, оновити, розрахувати суму
- ▶ Зазвичай бізнес-можливості використовуються як каркас для компонента, а вимоги - це дії, які компонент може виконувати

Визначення компонентів

- ▶ Функціональна автономія:
 - ▶ Це - максимальна функціональність, яка не пов'язана з іншими бізнес-вимогами
 - ▶ Наприклад:
 - ▶ Отримати замовлення, зроблені за останній тиждень
 - ▶ Отримати всі замовлення користувачів у віці 34 - 45 років

Визначення компонентів

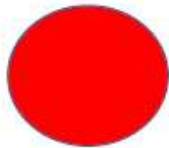
- ▶ Сутності даних:
 - ▶ Сервіс розроблений навколо чітко визначених об'єктів даних
 - ▶ Наприклад: замовлення, товари
 - ▶ Дані можуть бути пов'язані з іншими сутностями, але лише за ідентифікатором
 - ▶ Приклад: замовлення зберігає ідентифікатор клієнта

Визначення компонентів

- ▶ Автономність даних:
- ▶ Базові дані є атомарною одиницею
- ▶ Сервіс не залежить від даних інших сервісів для належного функціонування
- ▶ Наприклад: Сервіс працівників, який покладається на сервіс адрес для повернення даних співробітника

Визначення компонентів. Бізнес-Вимоги

eCommerce System



Inventory



Orders



Customers



Payments

Business Requirements	Manage inventory items	Manage orders	Manage customers	Perform payments

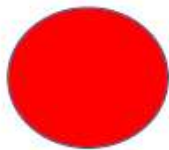
Визначення компонентів. Функціональна автономія



Business Requirements	Manage inventory items	Manage orders	Manage customers	Perform payments
Functional	Add, remove, update, quantity	Add, cancel, calculate sum	Add, update, remove, get account details	Perform payments

Визначення компонентів. Сутності даних

eCommerce System



Inventory



Orders



Customers

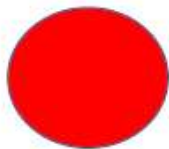


Payments

Business Requirements	Manage inventory items	Manage orders	Manage customers	Perform payments
Functional	Add, remove, update, quantity	Add, cancel, calculate sum	Add, update, remove, get account details	Perform payments
Data Entities	Items	Orders, shipping address	Customer, address, contact details	Payment history

Визначення компонентів. Автономія даних

eCommerce System



Inventory



Orders



Customers



Payments

Business Requirements	Manage inventory items	Manage orders	Manage customers	Perform payments
Functional	Add, remove, update, quantity	Add, cancel, calculate sum	Add, update, remove, get account details	Perform payments
Data Entities	Items	Orders, shipping address	Customer, address, contact details	Payment history
Data Autonomy	None	Related to Items by ID Related to Customer by ID	Related to Orders by ID	None

Визначення компонентів.

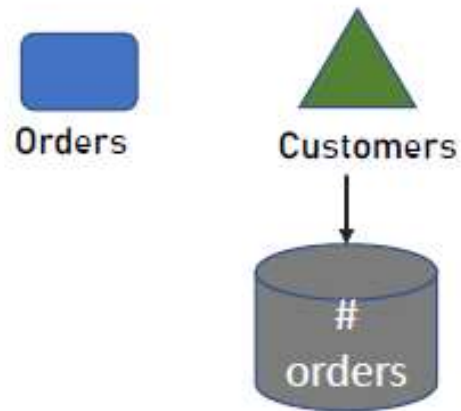
- ▶ Граничний випадок №1:•
- ▶ Визначте всіх клієнтів із Нью-Йорка із загальною кількістю замовлень для кожного клієнта

Customer name	No. of orders
David Smith	16
Diane Rice	23
George Murray	22

Визначення компонентів.

► Три підходи:

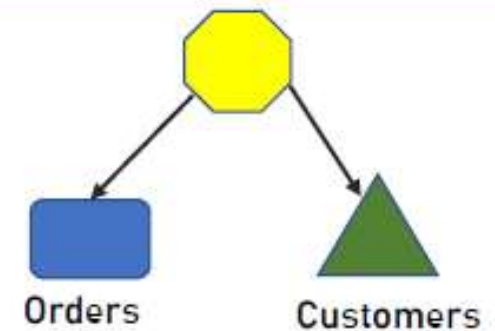
Data Duplication



Service Query



Aggregation Service



Визначення компонентів.

- ▶ Граничний випадок №2:
- ▶ Отримати список усіх замовлень у системі



Визначення компонентів.

- ▶ Сервіси не розроблені для даного сценарію
- ▶ Дізнайтеся, яка мета цього запиту
- ▶ Механізм звітування є кращим механізмом для цього



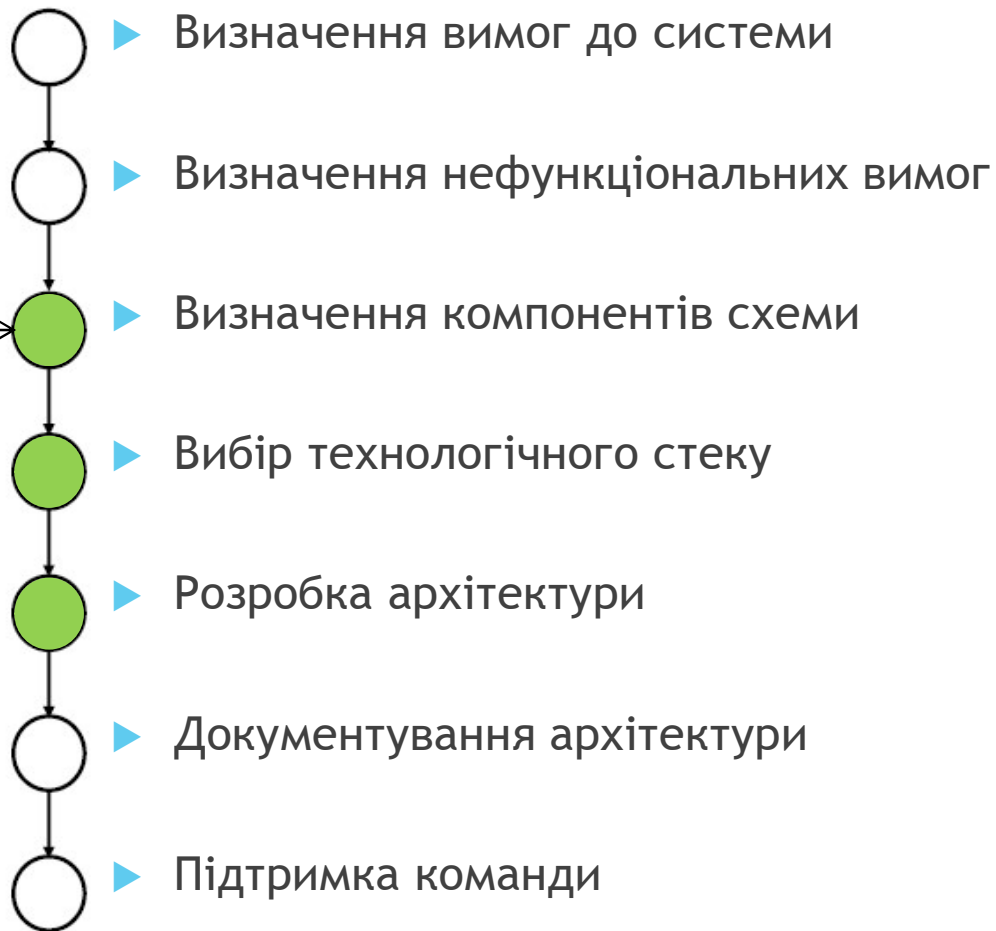
Визначення компонентів.

- ▶ Сервіси, які надають загальносистемні утиліти
- ▶ Загальні приклади:
 - ▶ Логування
 - ▶ Кешування
 - ▶ Керування користувачами
- ▶ **ПОВИННІ** бути відображені у схемі компонентів



Архітектурний процес

- Визначення компонентів
- Визначення шаблонів зв'язку



Визначення шаблонів зв'язку

- ▶ Ефективний зв'язок між службами має вирішальне значення
- ▶ Важливо вибрати правильний шаблон спілкування
 - ▶ Використання неправильного шаблону може призвести до повільної роботи, некерованості системи та поганого розподілу ролей
- ▶ Основні шаблони:
 - ▶ 1-to-1 Sync
 - ▶ 1-to-1 Async
 - ▶ Pub-Sub / Event Driven

1-to-1 Sync

- ▶ Сервіс викликає інший сервіс та чекає на відповідь
- ▶ Використовується в основному, коли першому сервісу потрібна відповідь, щоб продовжити обробку потоку даних.
- ▶ Приклад: Сервіс Orders хотіла б завершити обробку замовлення, але перед цим вона повинна переконатися, що товар у замовленні дійсно є на складі. Для цього Сервіс Orders робить запит до сервісу Inventory



1-to-1 Sync

Переваги

- ▶ Миттєве реагування
- ▶ Обробка помилок
- ▶ Легкість реалізації

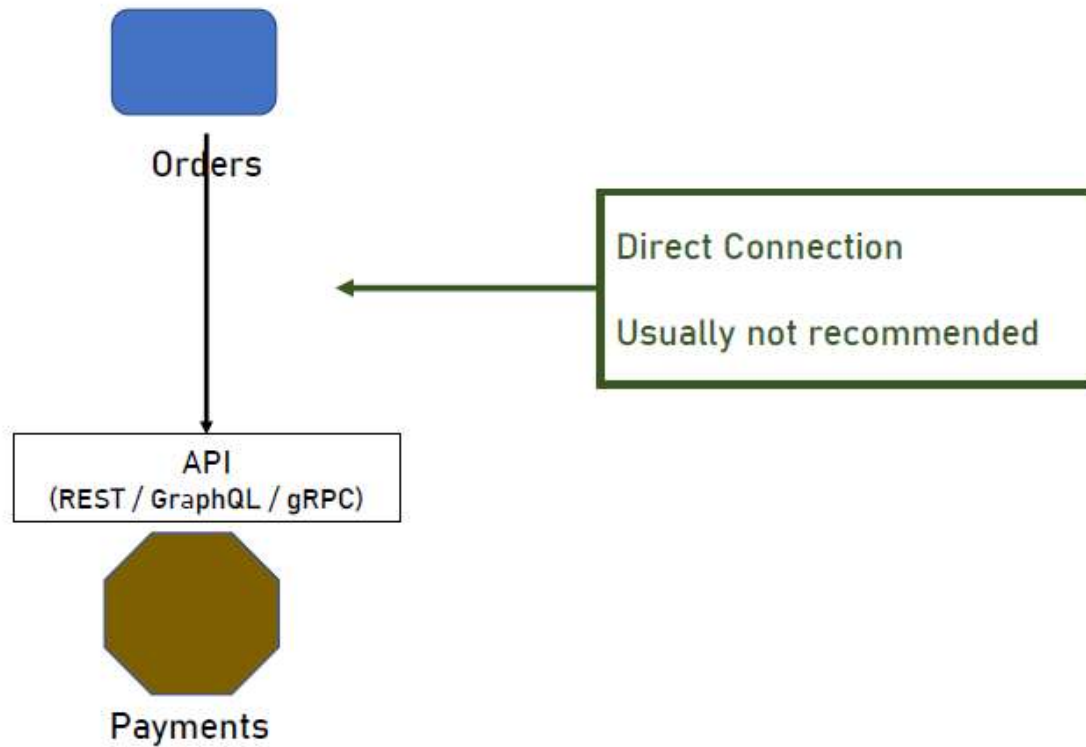
Недоліки

- ▶ **Продуктивність**

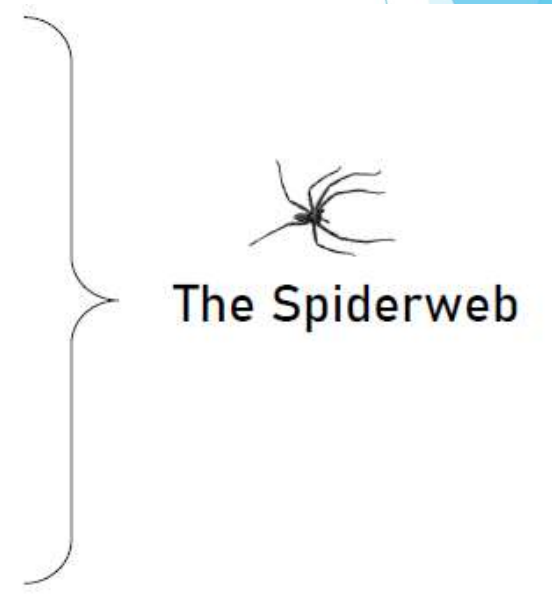
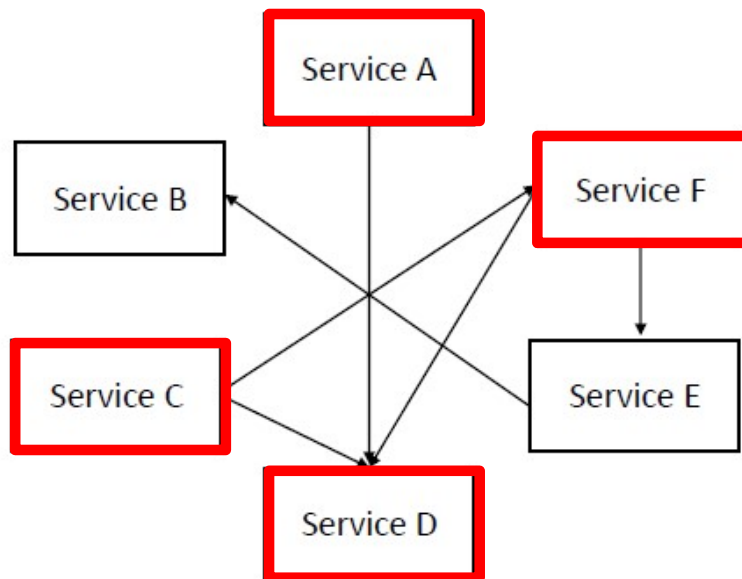


1-to-1 Sync

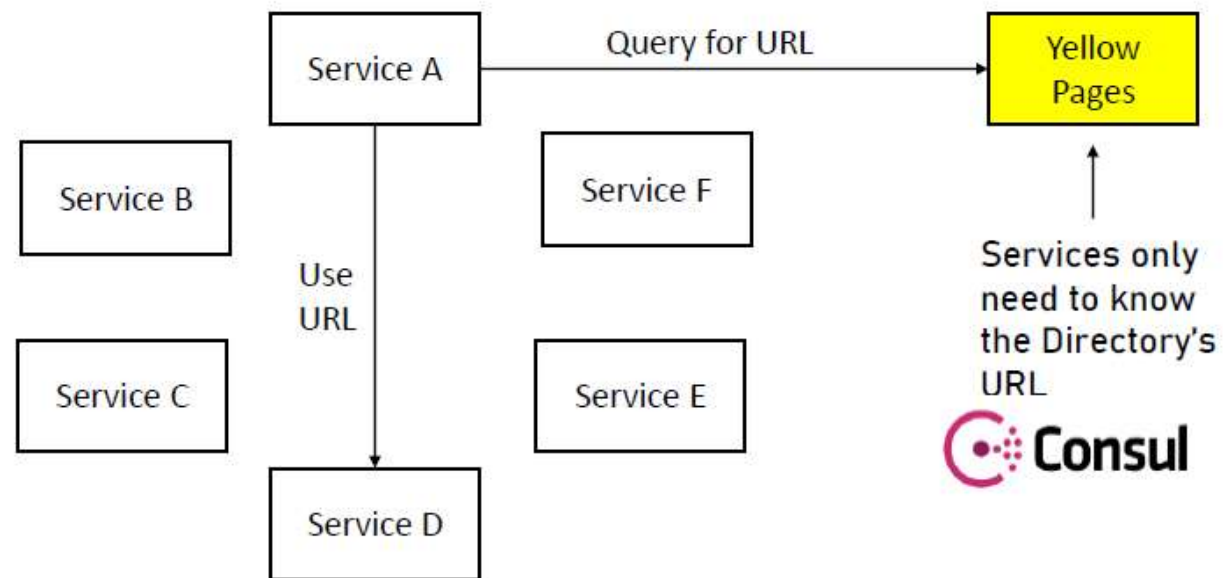
► Реалізація



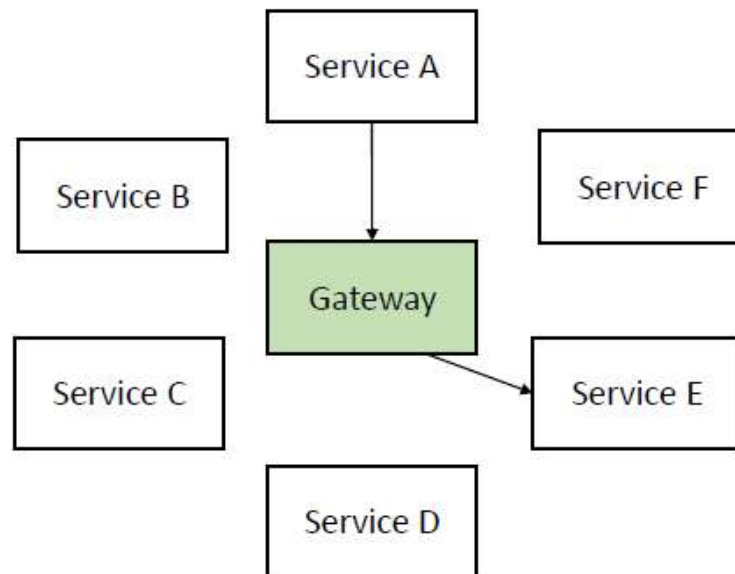
Прямий зв'язок



Виявлення послуг (Service Discovery)



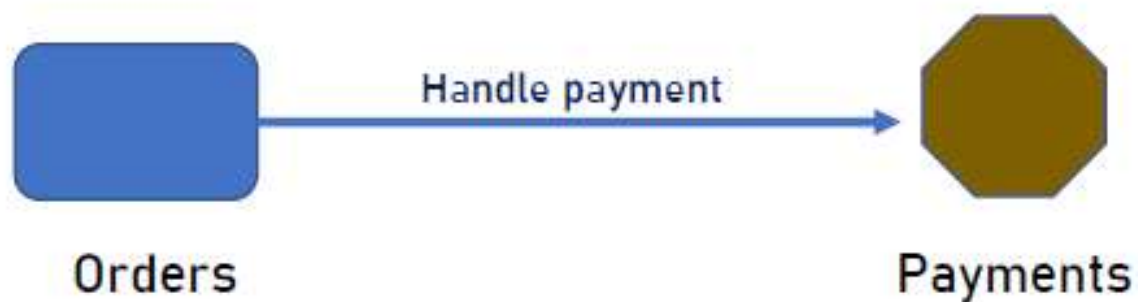
Шлюз (Gateway)



Services only
need to know
the Gateway's
URL

1-to-1 Async

- ▶ Сервіс викликає інший сервіс і продовжує роботу
- ▶ Не чекає на відповідь за принципом «Вистрелив і забув».
- ▶ Використовується в основному, коли перша служба хоче передати повідомлення іншій службі



1-to-1 Async

Переваги

- ▶ Продуктивність

Недоліки

- ▶ Потрібні додаткові налаштування
- ▶ Складна обробка помилок