

Розробка лабораторного  
практикуму з дисципліни  
«Основи сервіс-орієнтованих  
обчислень і архітектур»

Виконала:

студентка групи ДА-51с

Акінфієва Анастасія

Науковий керівник:

Харченко Костянтин Васильович

## Цілі

- Дослідити використання мікросервісної архітектури при створенні додатків.
- Провести порівняння з монолітною архітектурою
- Провести порівняння відомих фреймворків для створення мікросервісної архітектури
- Обрати фреймворк і на основі нього створити лабораторний практикум з короткими методичними вказівками до нього

## Предмет дослідження

- Доречність застосування мікросервісної архітектури при створенні додатків

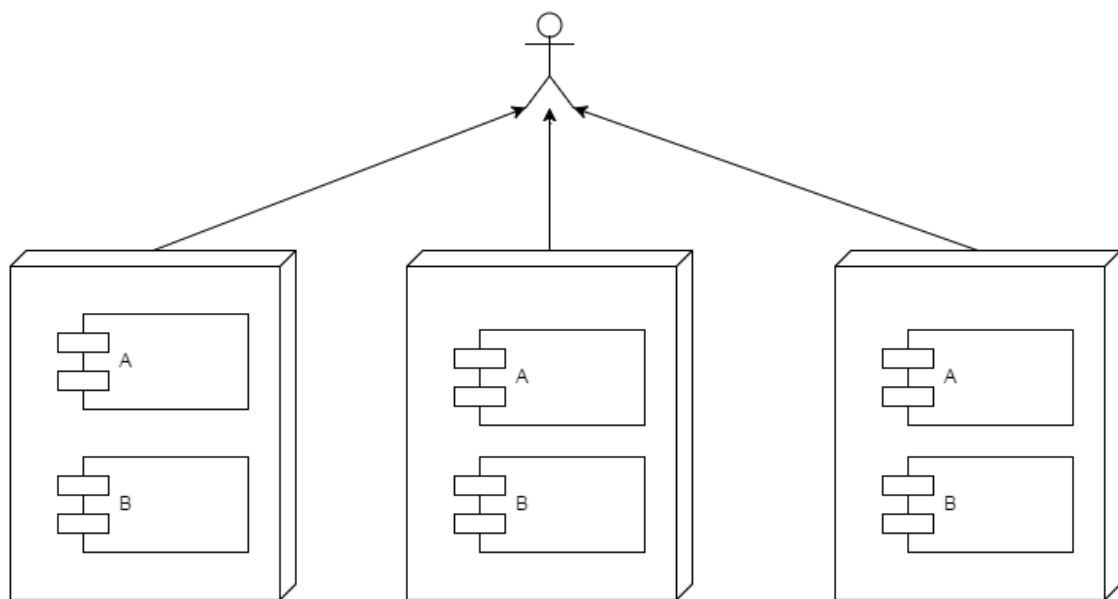
## Актуальність задачі

- Якомога детальніше розглянути мікросервісну архітектуру та підтвердити потрібність застосування її при тих чи інших обставинах.
- Збір та систематизація наявної інформації по даній темі для формування методичних матеріалів, які можуть бути застосовані студентами для набуття практичних навичок для роботи з даним підходом.

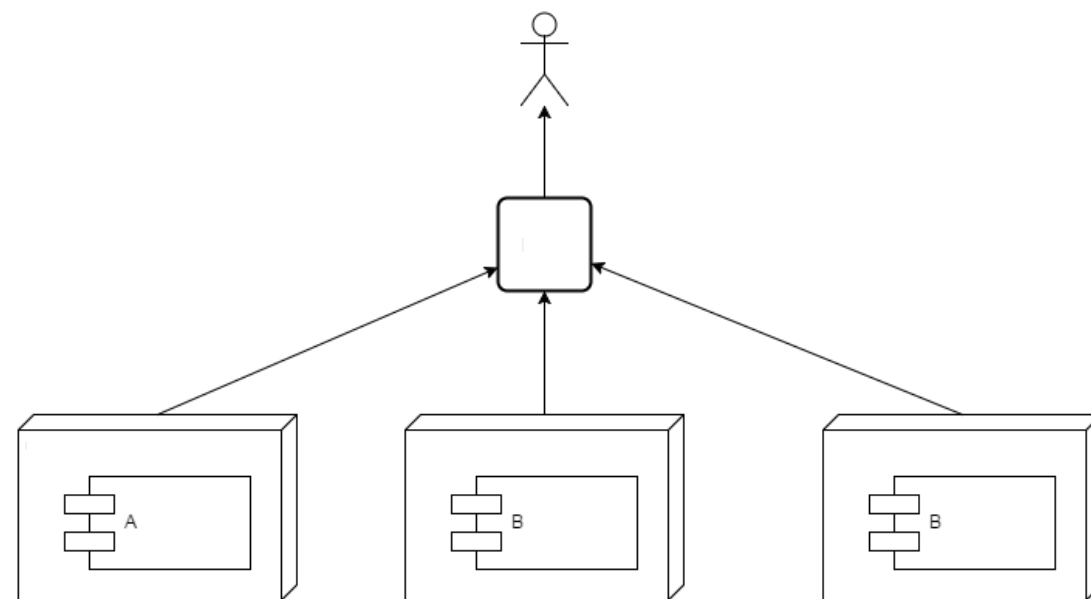
# Які методи досліджуються

- Створення простого мікросервісного додатку
- Розгортання мікросервісних додатків
- Комплексне порівняння відомих фреймворків з вибором найкращого для надання студентам в рамках лабораторного практикуму

# Моноліт і мікросервіси



Моноліт – кілька модулів в одному процесі



Мікросервіси – модулі запуснені в різних процесах

# Чому саме мікросервіси?

- Легші для розвитку, розуміння та підтримки
- Швидший запуск, в порівнянні з монолітом
- Легке розгортання локальних змін
- Незалежне масштабування
- Краща ізоляція збоїв
- Немає прив'язки до однієї технології

# Основні складові мікросервісної архітектури

- API Gateway
- Міжпроцесні комунікації (Inter-Process Communication)
- Виявлення сервісів (Service discovery)
- Управління даними

# Як реалізовані дані компоненти?

- API Gateway. Кожен мікросервіс в прикладі координує всі свої дії з Eureka API, який надається разом із Spring. Тобто, всі шляхи кластера мікросервісів знаходяться в одному місці. Це рішення дозволяє кожному сервісу отримати збалансоване навантаження та єдину точку входу.
- Service Discovery. Даний проект містить дві служби виявлення сервісів – Netflix Eureka та Consul. Вони обидва також поставляються разом із Spring та даний підхід дозволяє використовувати один (Consul) в якості DNS для мікросервісів, а інший (Eureka) в якості проксі-подібного API Gateway.

# Реалізація компонентів

- Inter-Process Communication. Міжпроцесна взаємодія в прикладі проекту відбувається із застосуванням стандартних методів REST API.
- Управління даними. В прикладі використовується схема розподілених БД (кожен сервіс має своє примітивне сховище) для кожного мікросервісу. Для того, щоб не ускладнювати його, мікросервіси не взаємодіють з БД один одного за допомогою EDD.



# Чому саме Spring?

- Чудова документація, яка допоможе студентам при вивченні курсу
- Застосування в більшості корпоративних додатків
- Простота реалізації мікросервісів в лабораторних умовах
- Простота реалізації REST API для взаємодії мікросервісів
- Захищеність всіх операцій та простота налаштування безпеки
- Найкращі інструменти для логування та відслідковування стану додатку

# Результати

- порівняно мікросервісну архітектуру із монолітною. Описано недоліки та переваги кожної.
- детально розглянуто основні компоненти мікросервісної архітектури
- порівняно можливі рішення та фреймворки для створення мікросервісів. Вибрано найоптимальніший, який можна надати студентам для ознайомлення.
- розгорнуто мікросервісний додаток та в ході цього створено короткі методичні вказівки для проведення лабораторного практикуму по темі мікросервісної архітектури

# Висновки

- Було проведено комплексне дослідження мікросервісної архітектури та її застосування при створенні додатків. Доведено актуальність застосування цього підходу.
- Розгорнуто мікросервісну систему в ході чого здобуто необхідні навички для подальшого створення та розгортання більш складних систем із застосуванням даної архітектури.
- На основі отриманого досвіду створено короткі методичні вказівки для лабораторного практикуму по даній темі. В подальшому робота може бути продовжена і на основі зібраних матеріалів можна побудувати розгорнуті методичні матеріали.