

# Розробка фреймворку для масштабованої ETL-системи

Виконав: Копійка В. Г., МП ІПЗ-2

Кервіник: к.ф.-м.н., доцент Гороховський С.С.

## Мета роботи

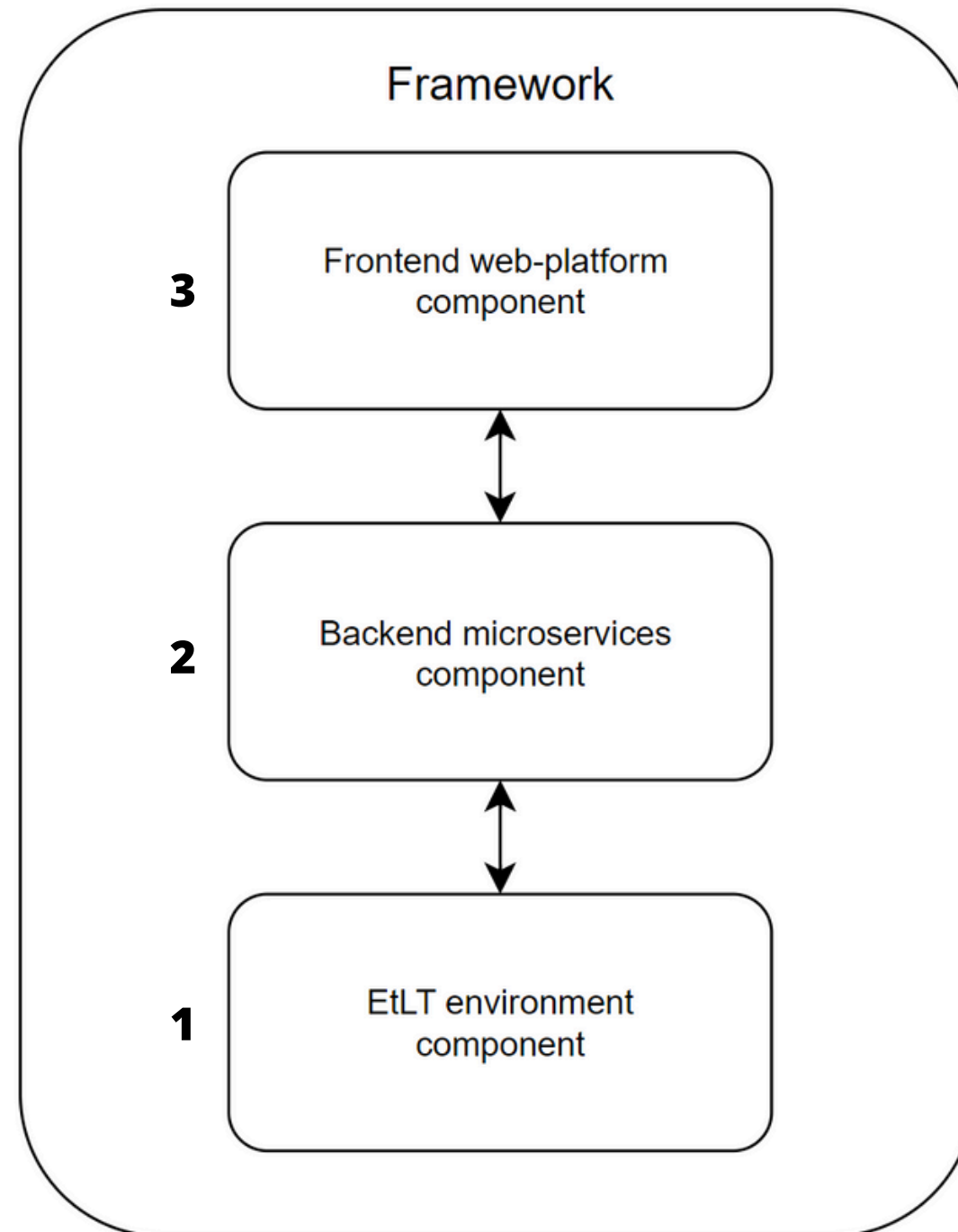
**Розробити фреймворк, який би:**

- враховував сучасні практики та технології в інтеграції даних
- був здатний масштабуватися у відповідності до використання ресурсів сервісами
- став у пригоді користувачам під час побудови процесу ETL з мінімальним втручанням в код

## Постановка задачі

- Проаналізувати виклики, з якими стикалися системи інтеграції даних ETL, та їхню еволюцію
- Розглянути сучасні практики та технології в побудові процесу інтеграції даних
- Реалізувати фреймворк для масштабованої ETL-системи з врахуванням аспектів вище

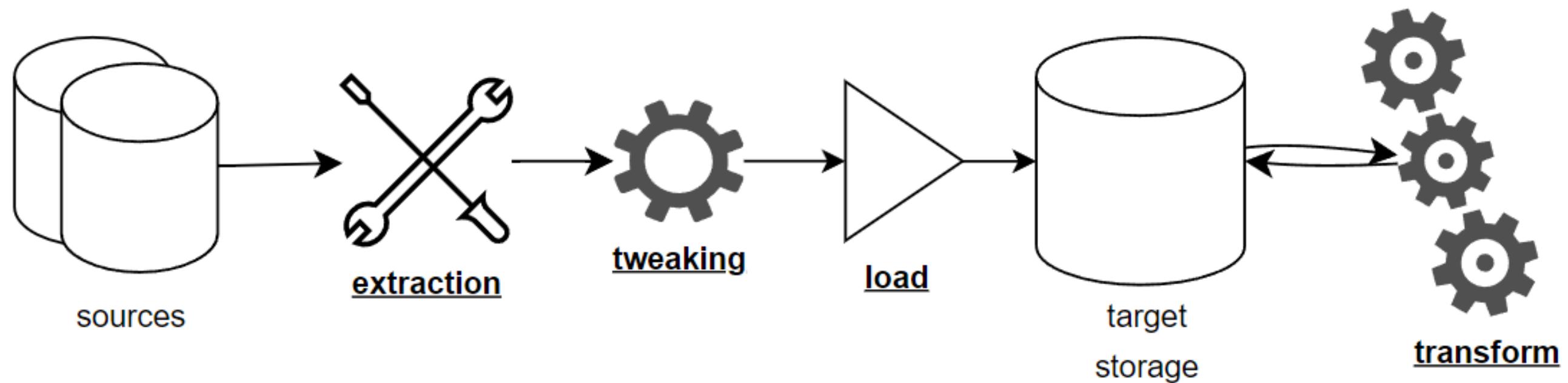
# Реалізація. Компоненти фреймворку



# Реалізація.

## 1. Компонента середовища інтеграції даних.

### 1.1 Архітектура

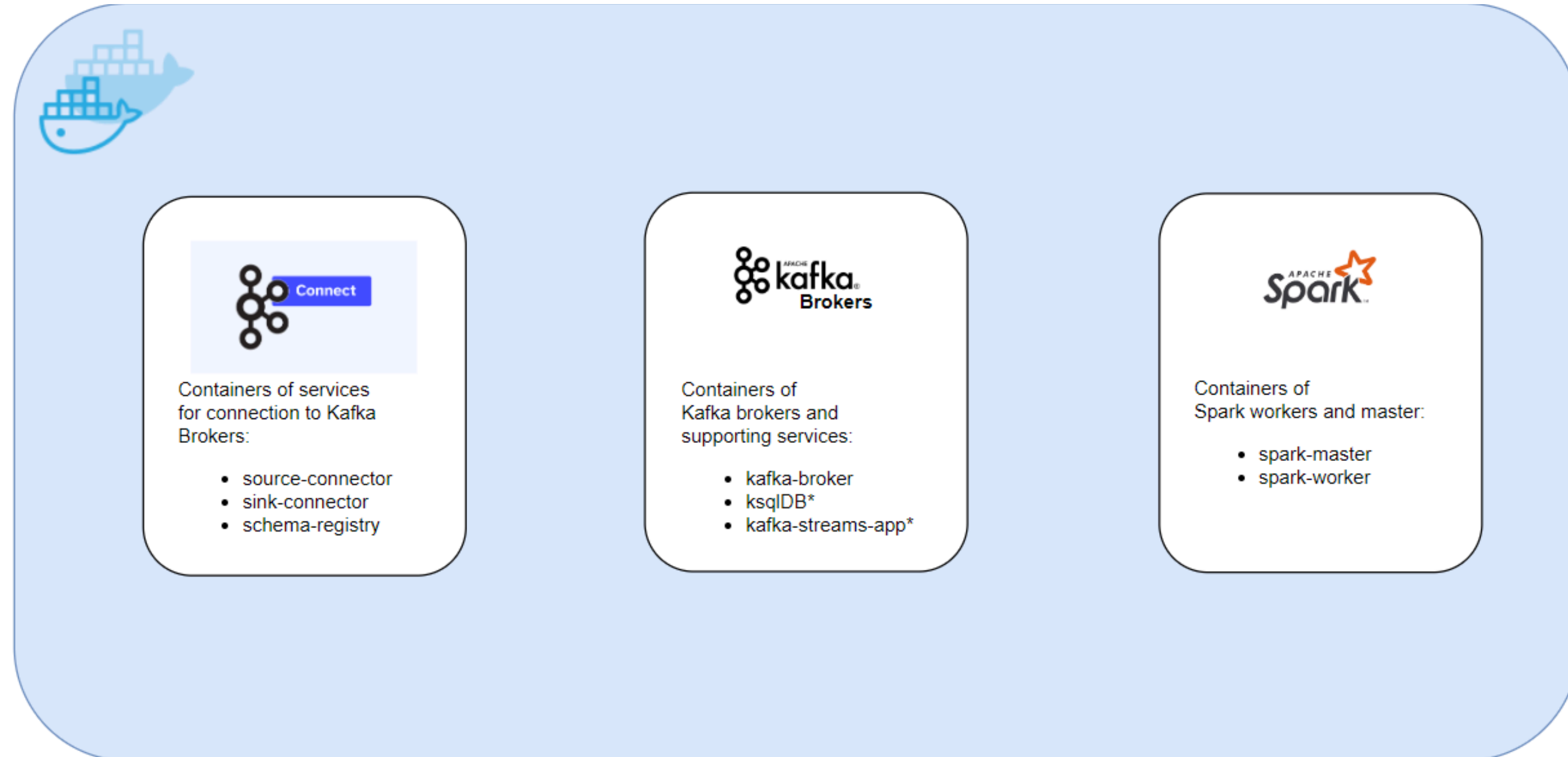


Процес інтеграції даних відбувається у відповідності до архітектури EtLT

# Реалізація.

## 1. Компонента середовища інтеграції даних.

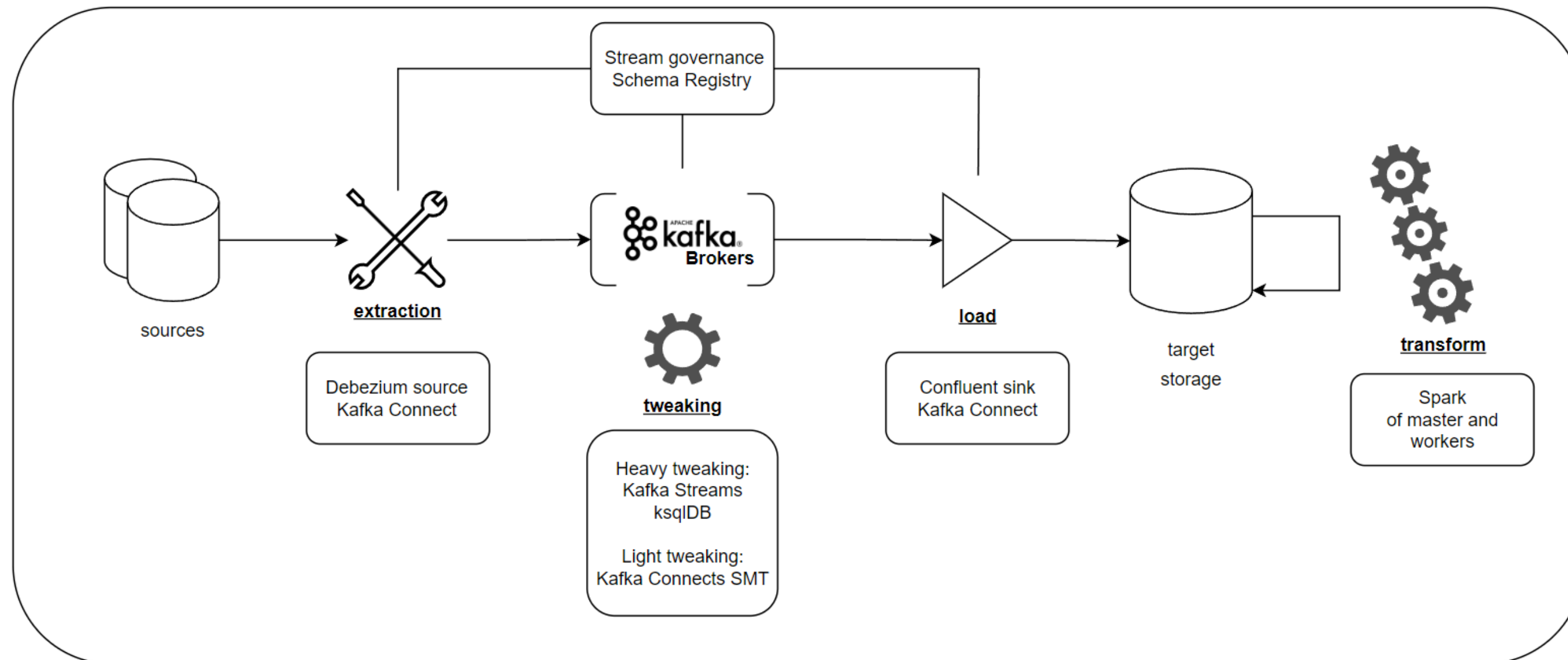
### 1.2 Головні набори сервісів



# Реалізація.

## 1. Компонента середовища інтеграції даних.

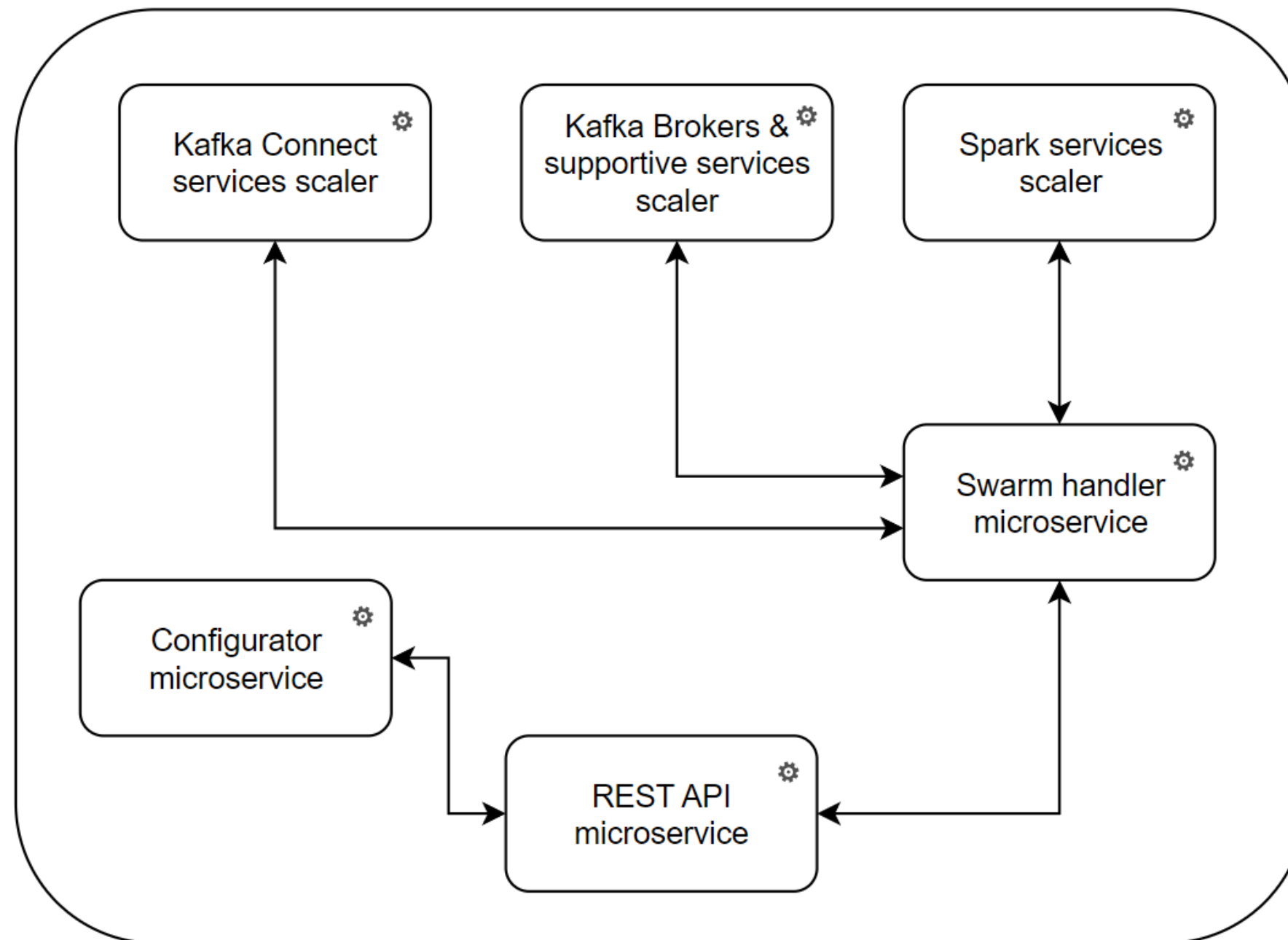
### 1.3 Сервіси в архітектурі



# Реалізація.

## 2. Компонента мікросервісів керування.

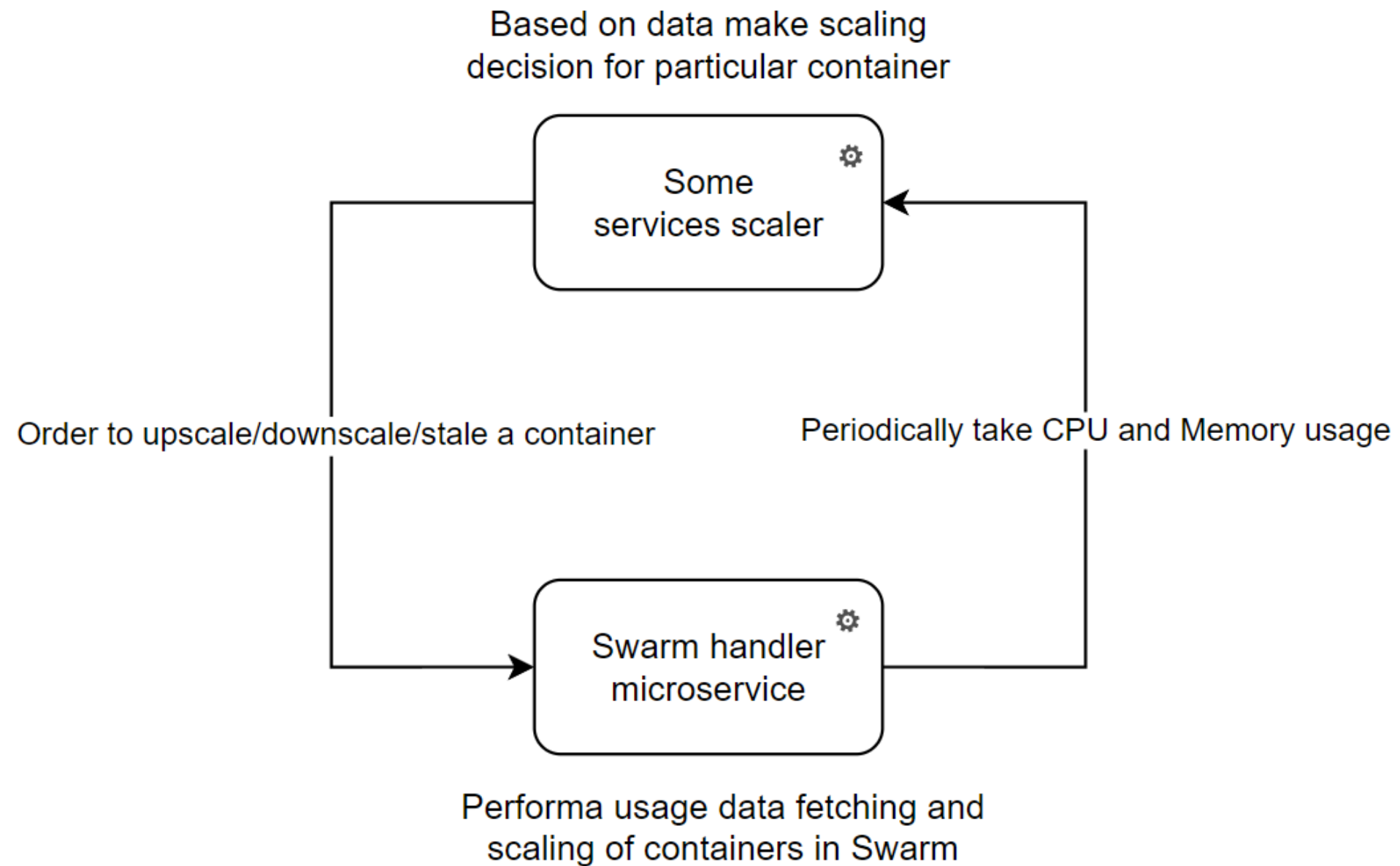
### 2.1 Взаємодія мікросервісів



# Реалізація.

## 2. Компонента мікросервісів керування.

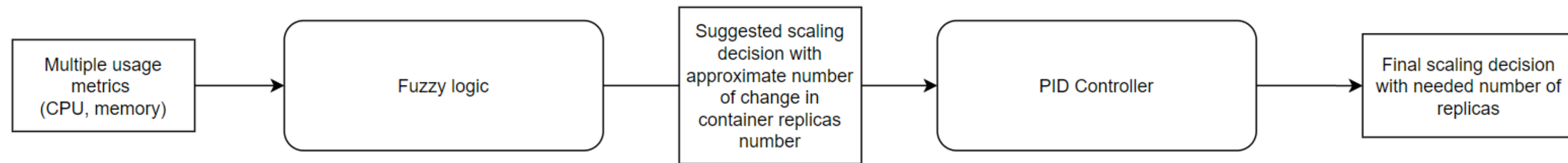
### 2.2 Процес взаємодії для масштабування



# Реалізація.

## 2. Компонента мікросервісів керування.

### 2.2.1 Алгоритм масштабування на мікросервісах-масштабувальниках

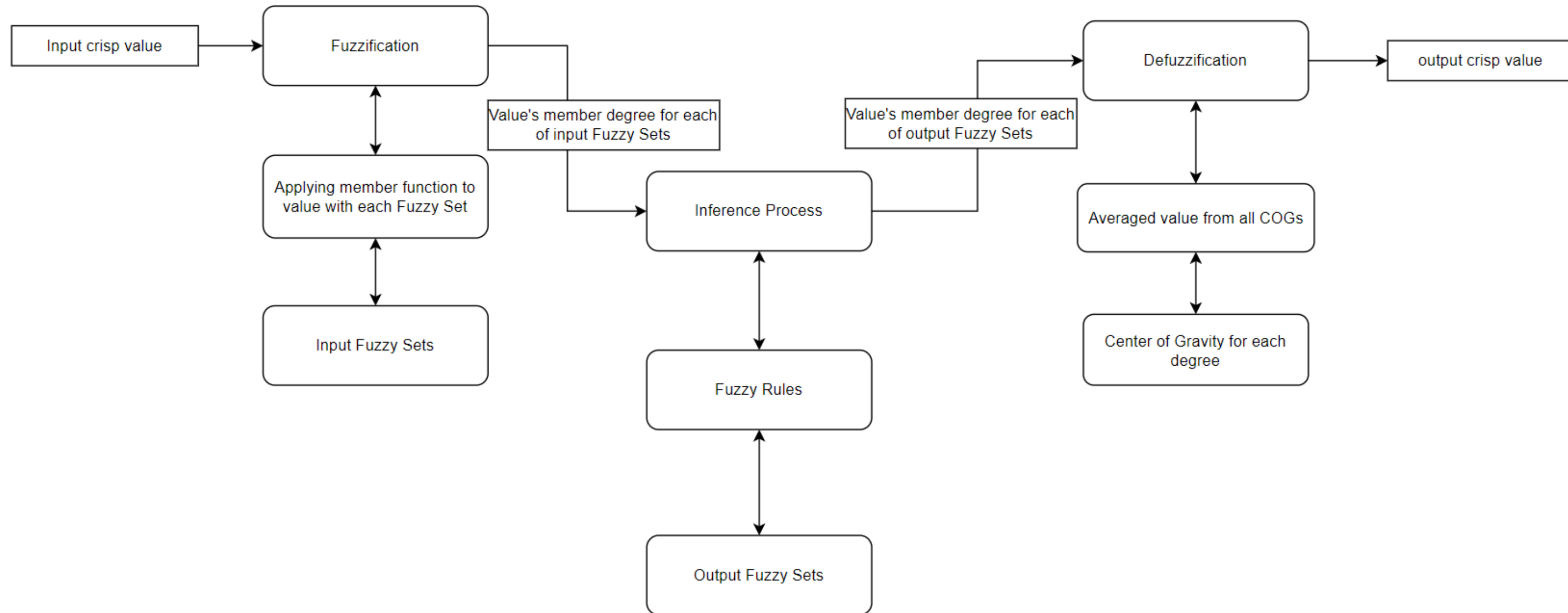


**Гібридна техніка масштабування контейнерів на основі Fuzzy Logic та PID-controller**

# Реалізація.

## 2. Компонента мікросервісів керування.

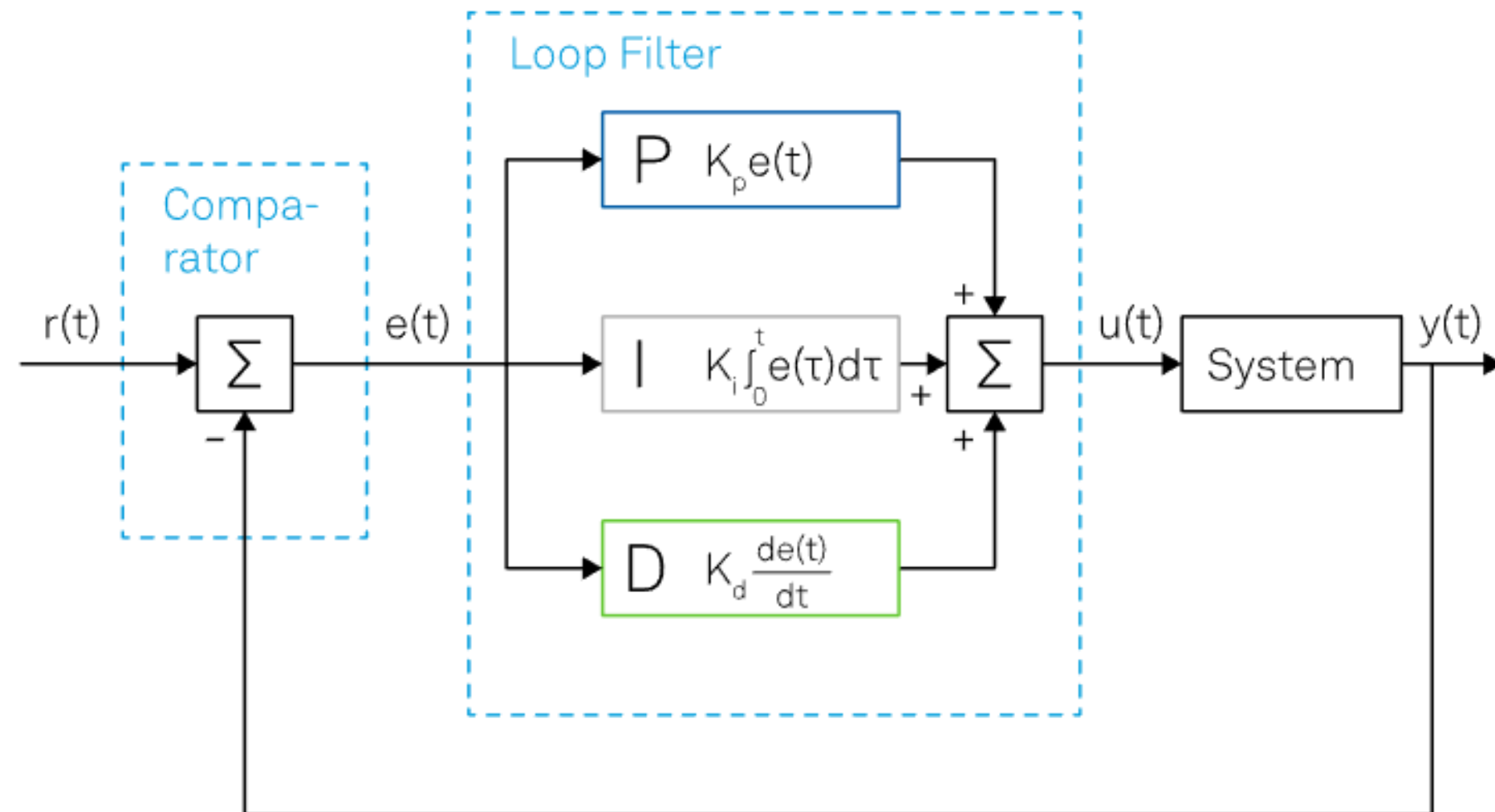
### 2.2.2 Fuzzy logic



# Реалізація.

## 2. Компонента мікросервісів керування.

### 2.2.3 PID-controller



$r(t)$  = System Setpoint

$e(t)$  = Error Signal

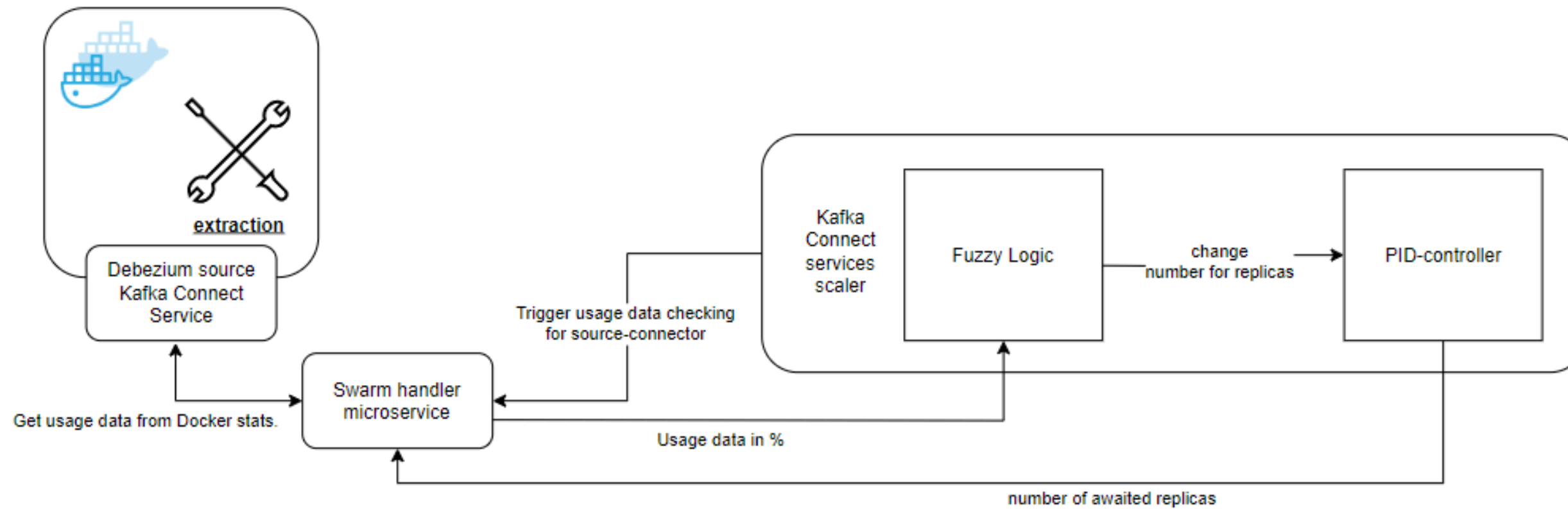
$u(t)$  = Control Signal

$y(t)$  = System Output

# Реалізація.

## 2. Компонента мікросервісів керування.

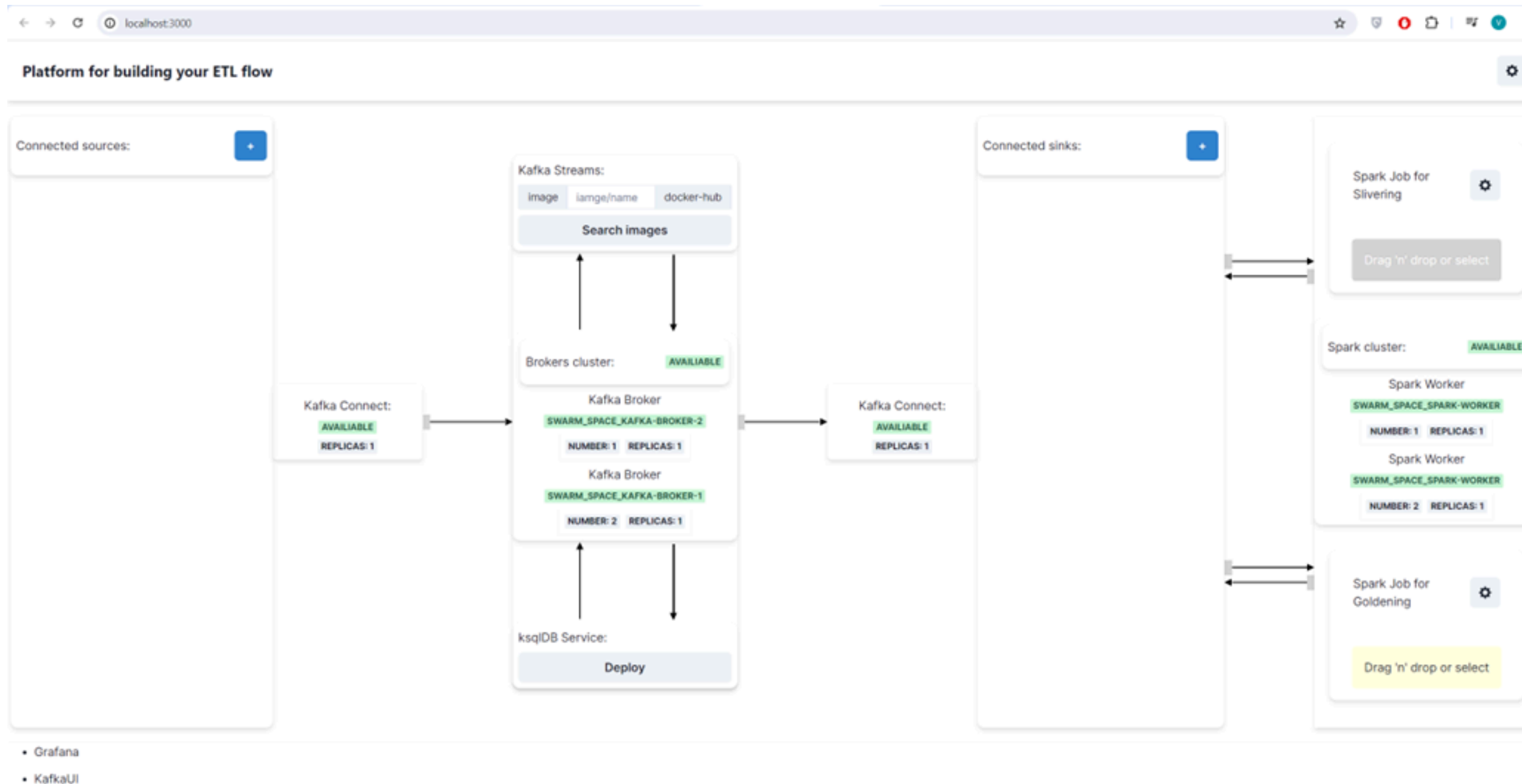
### 2.3 Приклад роботи алгоритму



# Реалізація.

## 3. Компонента графічного інтерфейсу.

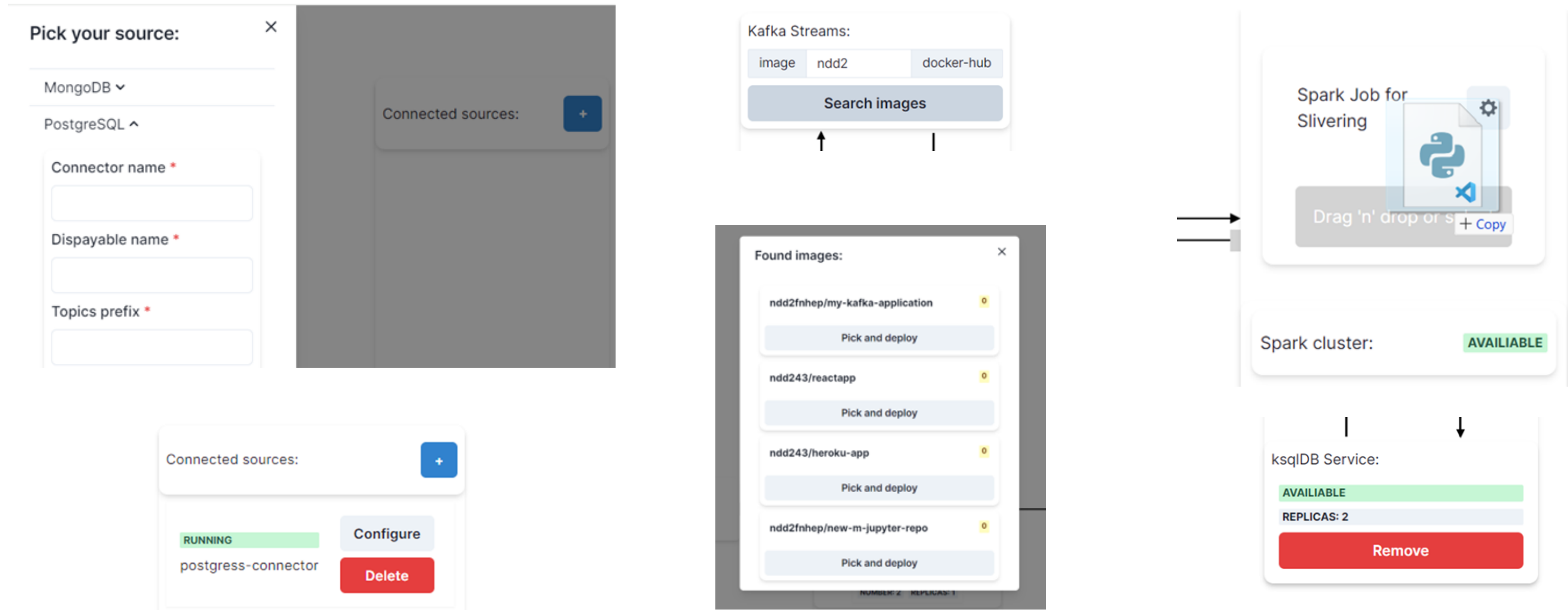
### 3.1 Приклад панелі UI



# Реалізація.

## 3. Компонента графічного інтерфейсу.

### 3.2 Деякі з можливостей



## Висновки

- Здійснено обґрунтований вибір технологій та практик для побудови системи інтеграції даних
- На основі вибору був розроблений фреймворк, який відповідає сучасній архітектурі EtLT та масштабується у відповідності до використання ресурсів сервісами його складових
- Завдяки мікросервісності головного компонента керування системою, складові за потреби можна замінювати, незважаючи на мови та особливості реалізації
- Розроблений фреймворк можна розгорнути локально і має графічний інтерфейс, що мінімізує втручання користувача в код

## Напрямки покращень

- Інтегрування редактору запитів для ksqlDB, Spark
- Залучення машинного навчання для масштабування
- Можливість розгортання середовища Swarm у хмарних постачальниках

**Дякую за увагу!**