

# Дослідження рішення **Red Hat OpenShift Service Mesh** для реалізації мікросервісної архітектури

**Виконала:**

студентка IV курсу, групи ДА-61

Фецун Анна Володимирівна

**Керівник:**

Доцент, к.т.н.

Гіоргізова-Гай Вікторія Шалвівна

## **МЕТА РОБОТИ:**

Дослідити рішення Red Hat OpenShift Service Mesh для подолання основних проблем, які виникають при створенні додатків на основі мікросервісної архітектури.

---

## **ЗАВДАННЯ:**

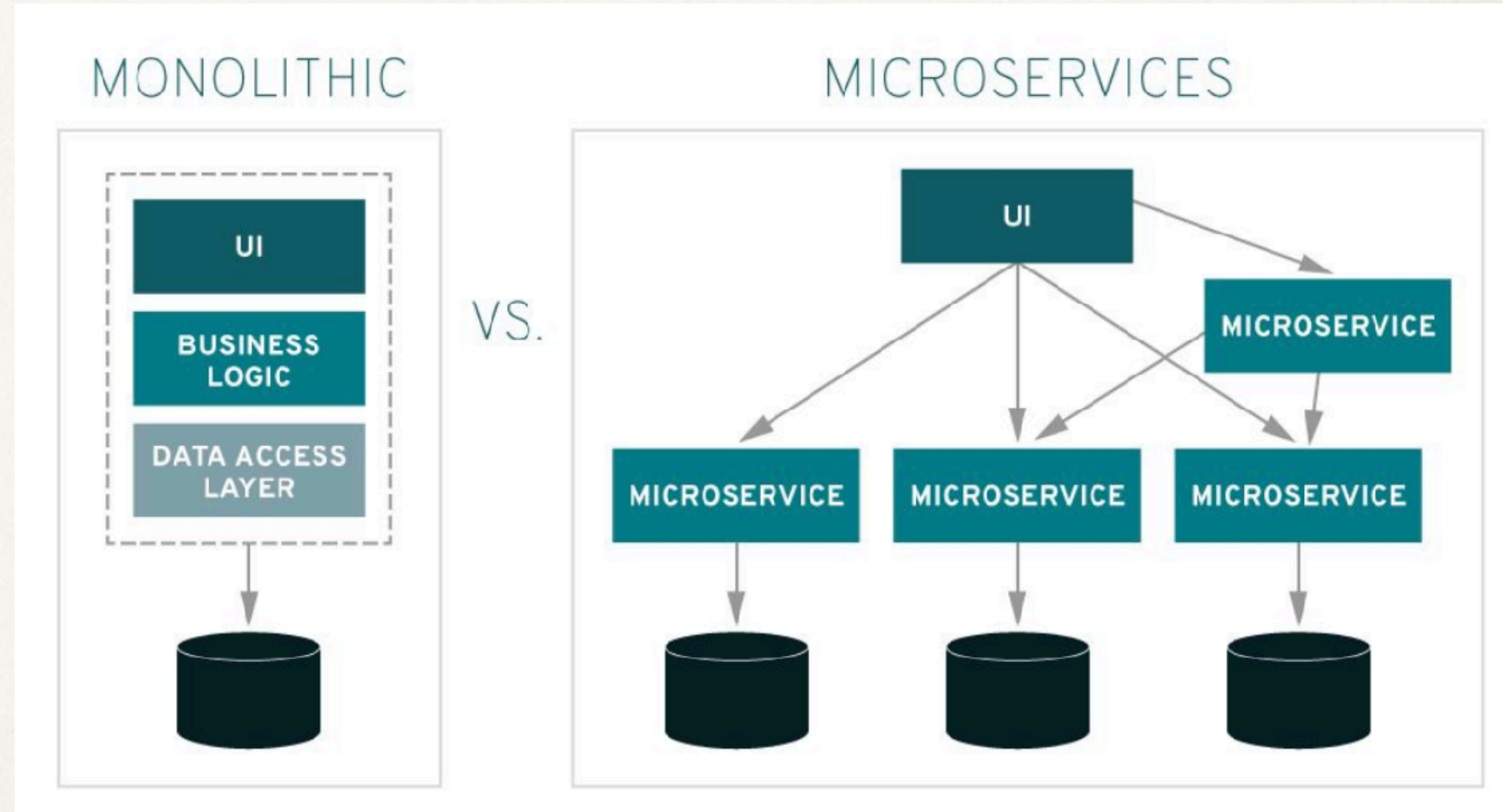
1. Аналіз мікросервісної архітектури, її переваги та недоліки.
2. Дослідження засобів вирішення проблем мікросервісних додатків.
3. Дослідити засоби технології OpenShift Service Mesh.
4. Продемонструвати на практичному прикладі переваги застосування OpenShift Service Mesh.

# Мікросервісна архітектура

**Мікросервісна архітектура** - це техніка розробки програмного забезпечення.

**Ключова характеристика мікросервісів**

— є можливість оновлювати сервери незалежно одне від одного.



Порівняння архітектур монолітного та мікросервісного додатків

# Переваги мікросервісної архітектури

---

- ◆ Запуск сервісу виконується швидше, що дає розробникам можливість бути продуктивнішими та прискорює розгортання;
- ◆ Висока масштабованість;
- ◆ Кожен сервіс – незалежний;
- ◆ Баг в одному мікросервісі не повинен підірвати роботу системи та зламати додаток.
- ◆ При розробці нового сервісу вибирається стек технологій, які найкраще підходять для мікросервісу.
- ◆ Мікросервісна архітектура, як правило, краще структурує та організовує модулі;
- ◆ Відокремлені сервіси легше перекомпонувати і переналаштувати.

**Чимало організацій успішно використовують мікросервісний підхід - Netflix, Amazon, eBay, The Guardian**

# Недоліки мікросервісної архітектури

- Складним спостереження взаємодії між сервісами;
- Управління трафіком;
- Складність забезпечення зв'язку за допомогою шифрування потоків даних;
- Управління тайм-аутами та збоями в комунікації між службами може призвести до каскадних збоїв.

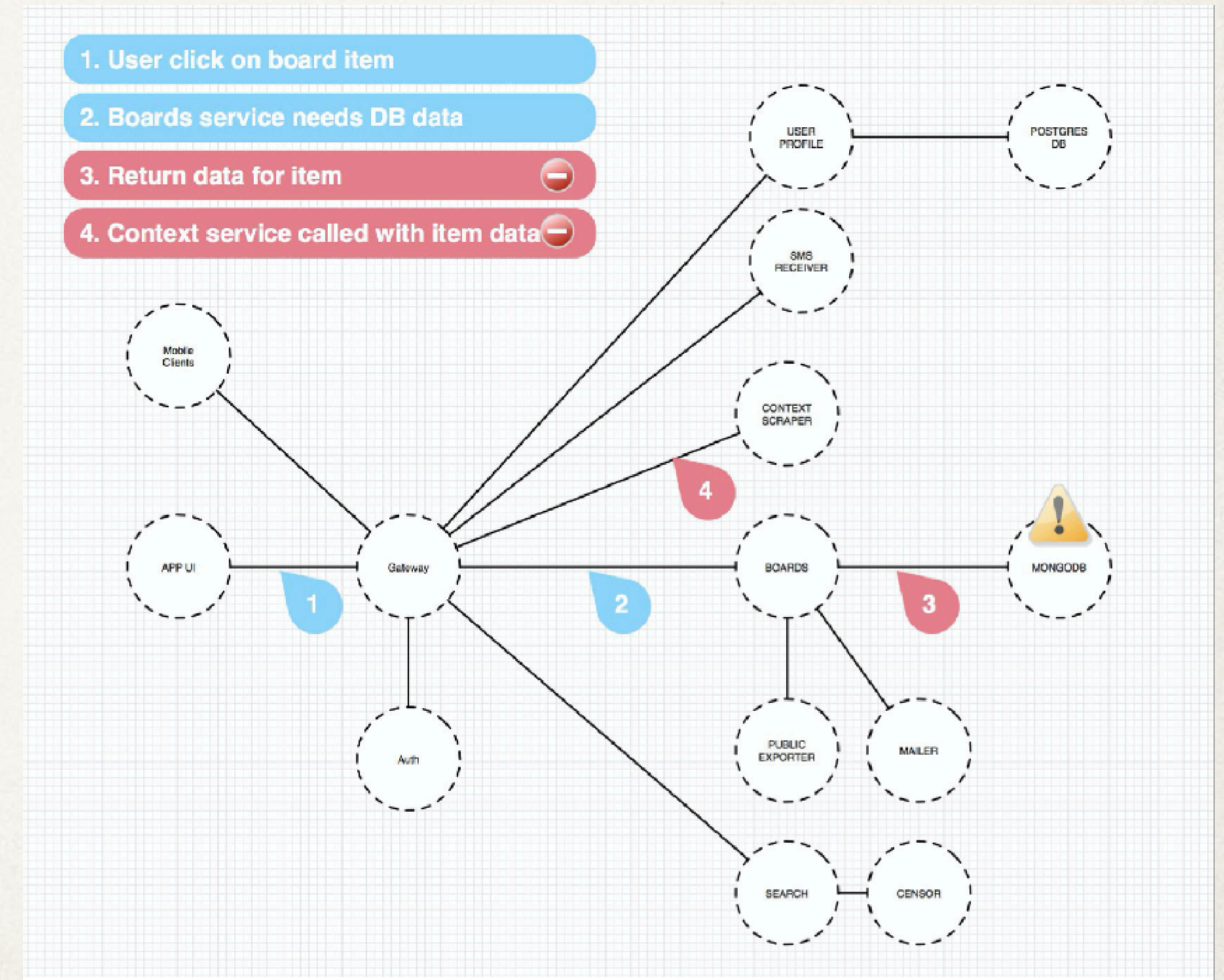


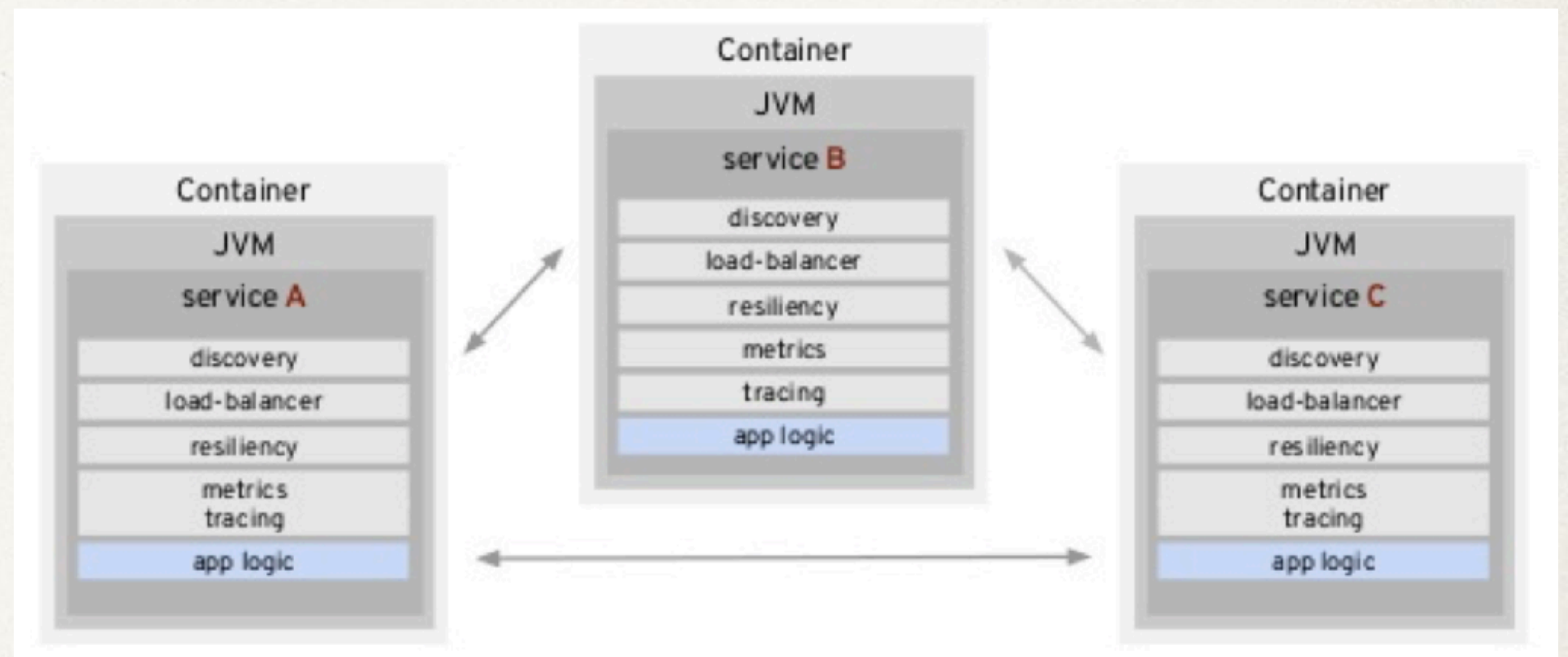
Схема каскадного збою мікросервісного додатка

# Сервісна сітка

**Сервісна сітка** - це абстрактний шар інфраструктури.

**Надає наступні основні можливості:**

- ◆ Формування трафіку (Traffic shaping);
- ◆ Підтримка стійкості для службового зв'язку;
- ◆ Спостереження трафіку між службами;
- ◆ Простеження комунікаційних потоків;
- ◆ Безпечна комунікація між службами;



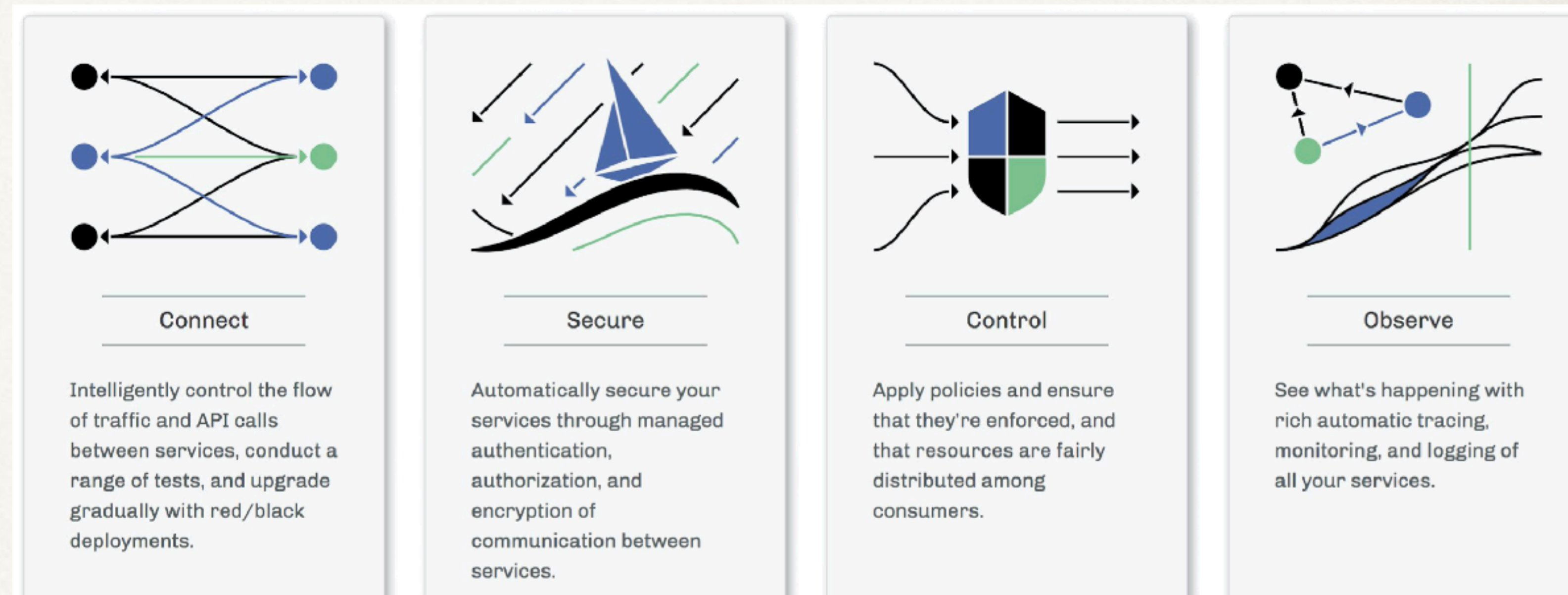
Мікросервіси без сервісної сітки

# Istio Mesh

Сучасний спосіб керування складними мікросервісними додатками

## Можливості Istio:

- ◆ моніторинг,
- ◆ трасування,
- ◆ автоматичні вимикачі,
- ◆ маршрутизація,
- ◆ балансування навантаження,
- ◆ інжекція несправностей,
- ◆ повтори, паузи,
- ◆ зазеркалення,
- ◆ контролювання доступу,
- ◆ обмеження швидкості
- ◆ інших можливостей.



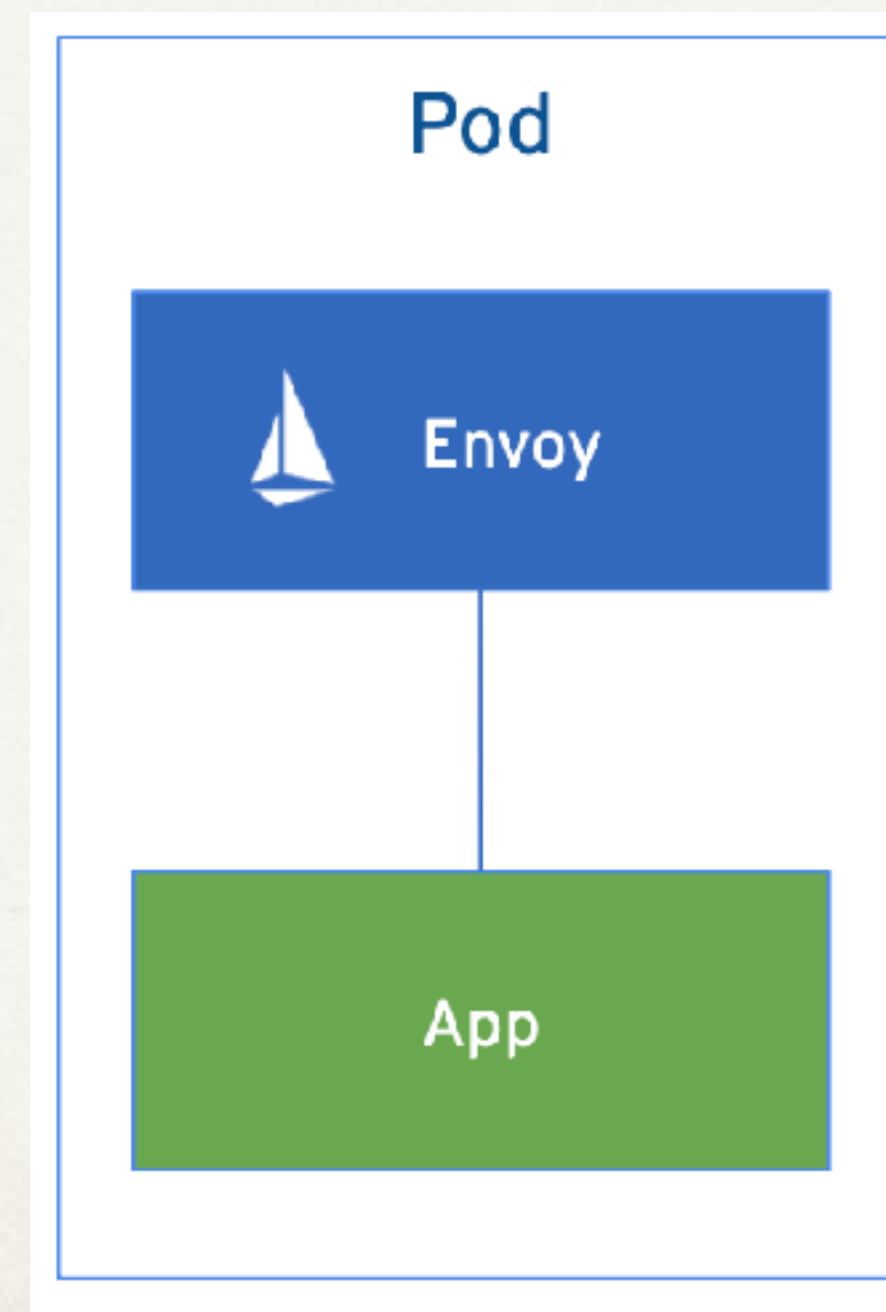
Основні можливості Istio

# Envoy

## НАДАЄ МОЖЛИВОСТІ:

- ◆ HTTP Proxy
- ◆ HTTP / 2, gRPC
- ◆ Завершення TLS
- ◆ Перевірки стану сервера
- ◆ Служби виявлення
- ◆ Моніторинг
- ◆ Балансування навантаження
- ◆ Інжекція несправностей
- ◆ Інші

Envoy проху перехоплює весь вхідний та вихідний трафік для всіх сервісів сервісної мережі. Envoy розгортається як sidecar до відповідної сервіса у тому самому pod'і.



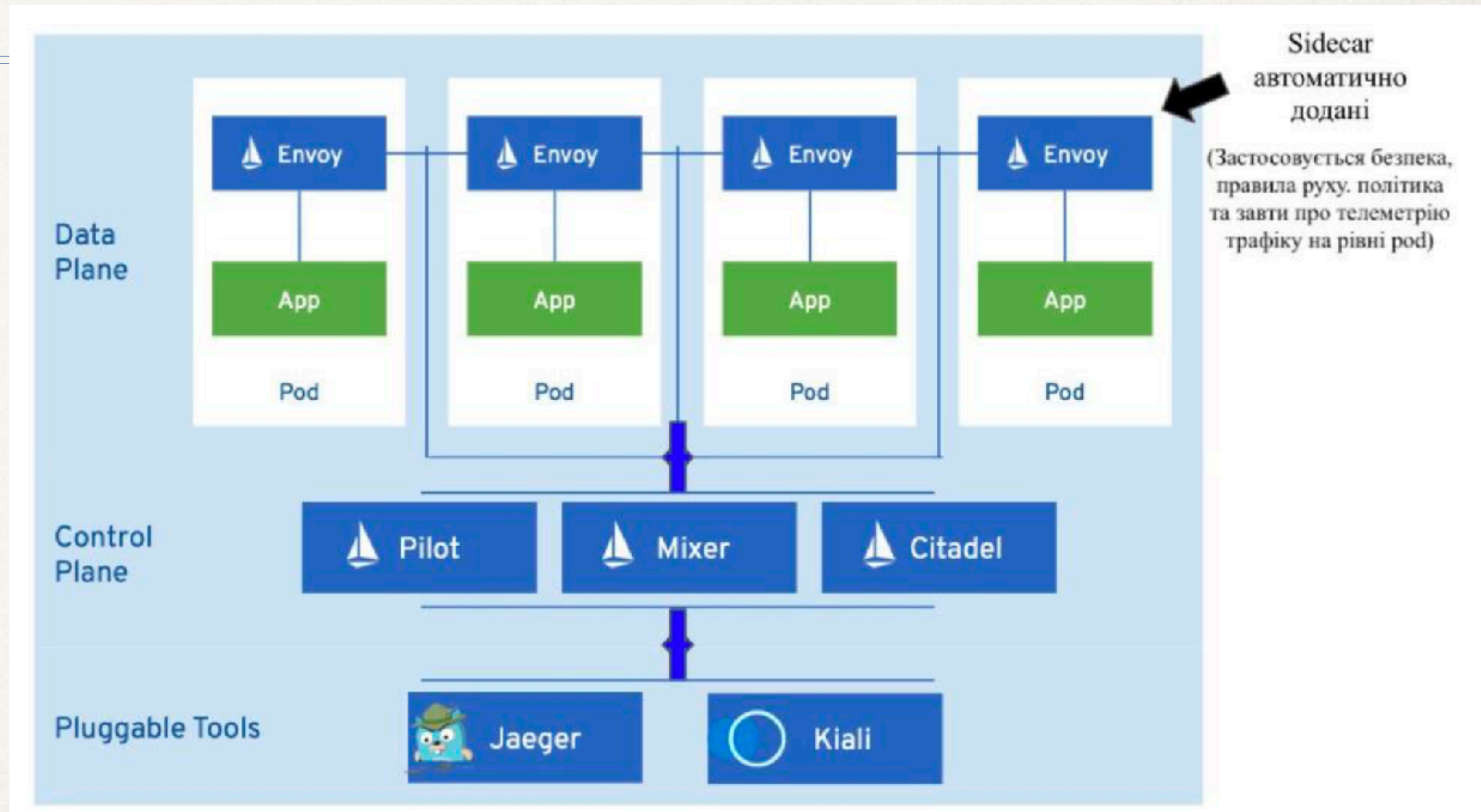
Архітектура sidecars



**Data plane** - це набір інтелектуальних proxies, розгорнутих як sidecars.

**Control plane** керує та налаштовує проксі-сервери.

Ключові компоненти control plane - Pilot, Citadel, Mixer і Galley.

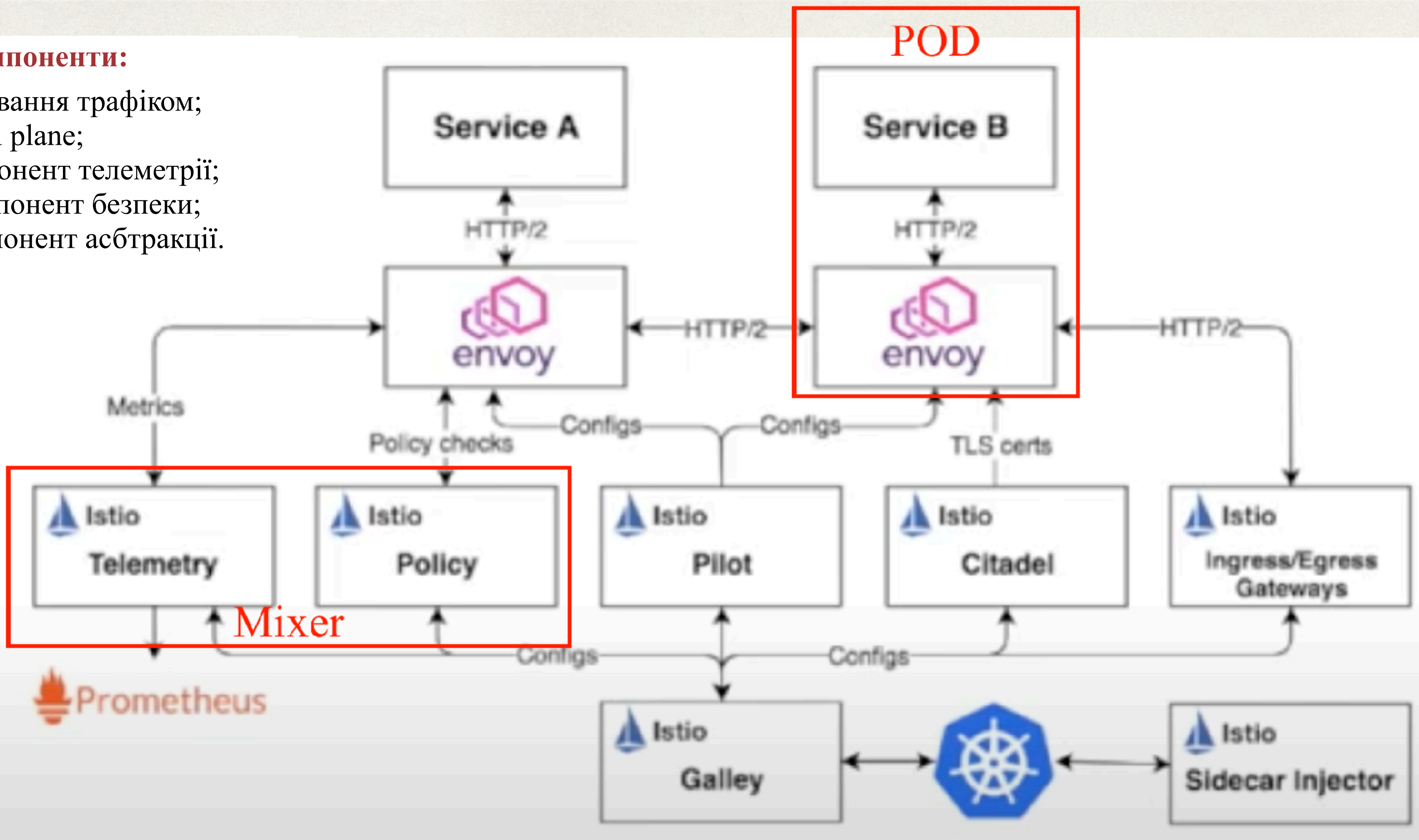


Структура сервісної сітки Istio

# Архітектура Istio

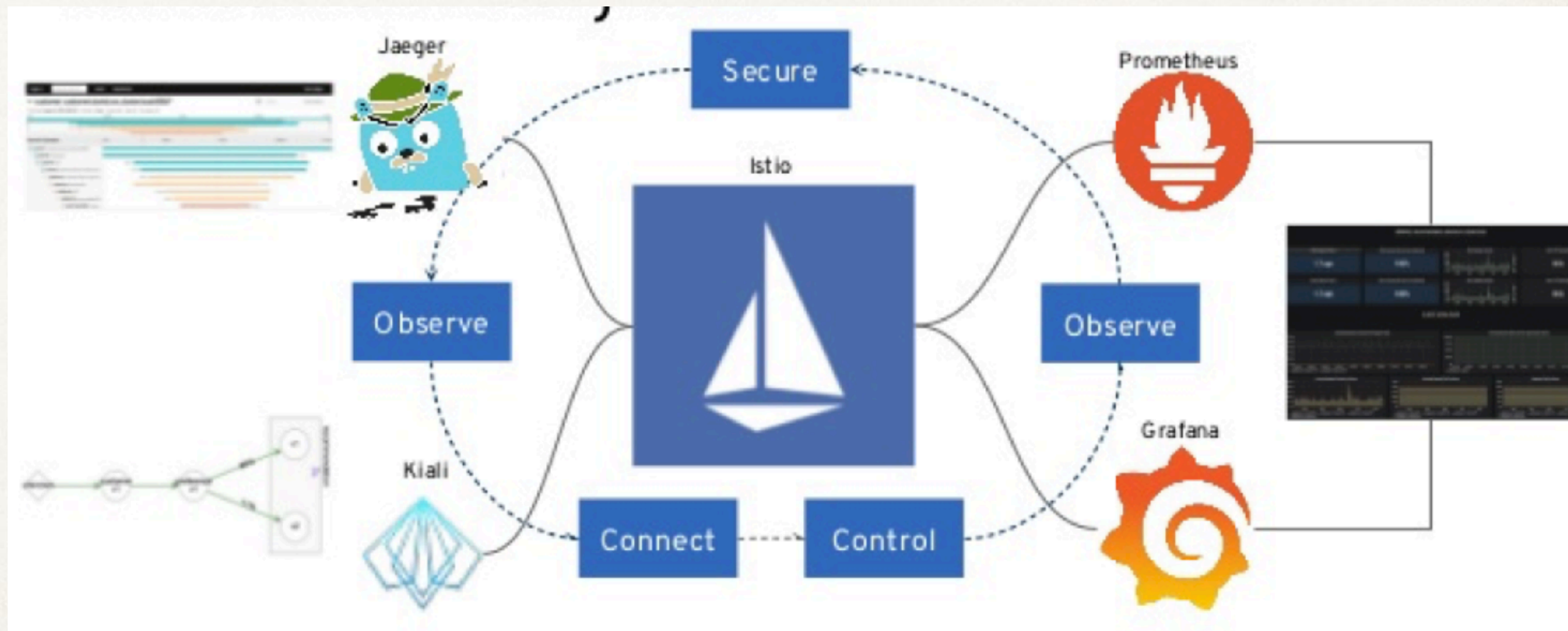
## Основні компоненти:

- Envoy - керування трафіком;
- Pilot - control plane;
- Mixer - компонент телеметрії;
- Citadel - компонент безпеки;
- Galley - компонент абстракції.



Структура сервісної сітки Istio

# Екосистема Red Hat OpenShift Service Mesh



Екосистема

# Переваги Red Hat OpenShift Service Mesh

<b>OpenShift Container Platform надає можливості:</b>	<b>Переваги Red Hat OpenShift Service Mesh:</b>
◆ Вбудований Jenkins	◆ Трасування і моніторинг (Jaeger)
◆ BuildConfig objects	◆ Візуалізація (Kiali)
◆ ImageStreams	◆ Kubernetes-оператор Service Mesh
◆ Правила безпеки	◆ Підтримка декількох мережових інтерфейсів (multus)
◆ Моніторинг	◆ Інтеграція з Red Hat 3scale API Management
◆ Вбудована підтримка OAuth	

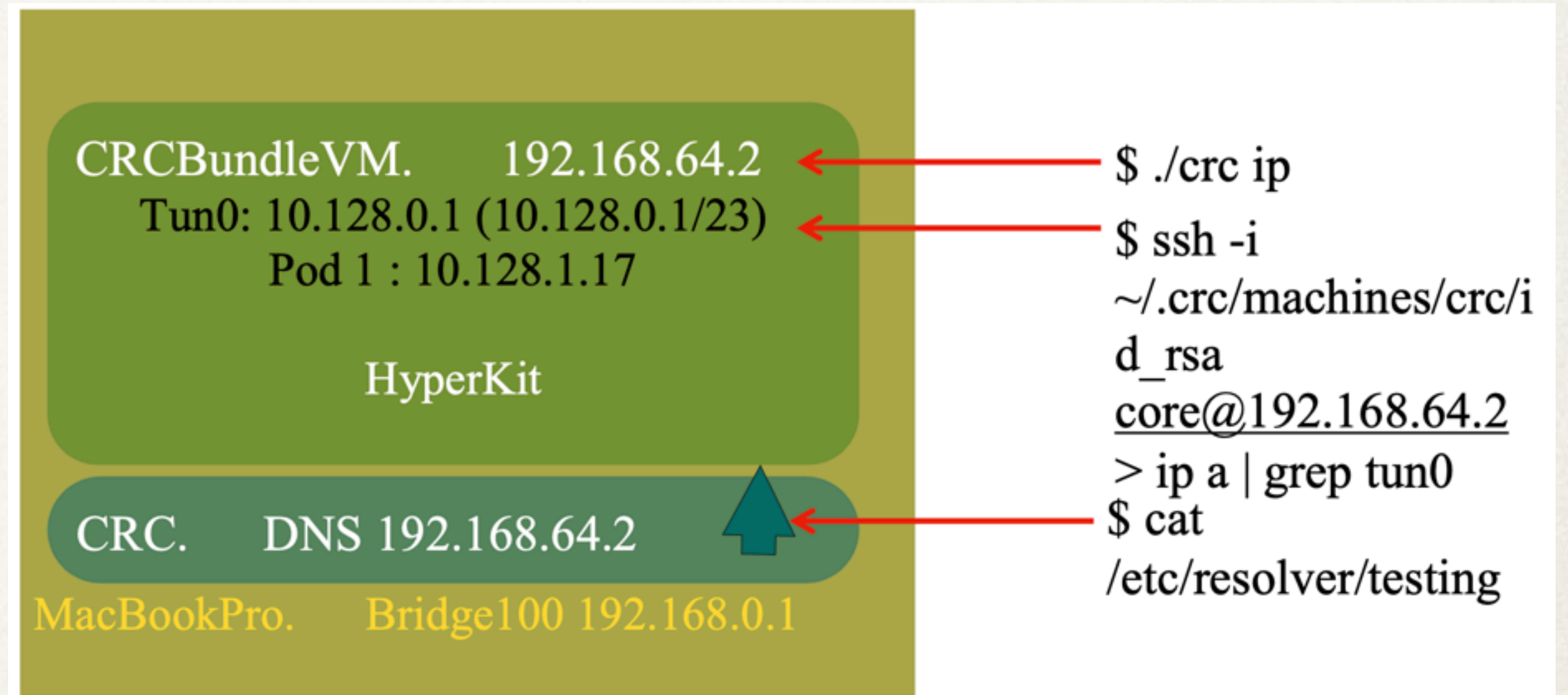
# CodeReady Containers

ДОЗВОЛЯЄ ЗАПУСТИТИ КЛАСТЕР OpenShift 4.1+ ЯК КОНФІГУРАЦІЮ З ОДИМ ВУЗЛОМ НА ПЕРСОНАЛЬНОМУ КОМП'ЮТЕРІ. Red Hat CodeReady Containers ВИКОРИСТОВУЄ ЄДИНУ NODE.

```
[popcorn@MacBook cloud-native-starter % oc get nodes
NAME                STATUS    ROLES    AGE   VERSION
crc-dv9sm-master-0  Ready    master,worker  17d   v1.17.1
```

- CodeReady Containers Version # command on macOS: `./crc version`  
crc version: 1.10.0+9025021  
OpenShift version: 4.4.3 (embedded in binary)
- CodeReady Containers Status # command on macOS: `./crc status`  
CRC VM: Running  
OpenShift: Running  
Disk Usage: 15.71GB of 32.72GB (Inside the CRC VM)  
Cache Usage: 12.05GB  
Cache Directory: `/Users/popcorn/.crc/cache`

Загальна характеристика контейнера на ОС macOS X



Встановленні компоненти на ПК

# Створення сервісної сітки

The screenshot shows the Red Hat OpenShift Container Platform console. The left sidebar contains navigation options: Administrator, Home, Overview, Projects, Search, Explore, Events, Operators (selected), OperatorHub, Installed Operators (highlighted), Workloads, Networking, Storage, Builds, Monitoring, Compute, and User Management. The main content area displays the 'Installed Operators' page for the 'istio-system' project. A table lists the installed operators, with the 'Status' column highlighted by a red box. The operators listed are:

Name	Namespace	Status	Deployment	Provided APIs
Elasticsearch Operator 4.4.0-202004261927 provided by Red Hat, Inc	istio-system	Succeeded Up to date	elasticsearch-operator	Elasticsearch
Red Hat OpenShift Jaeger 1.17.2 provided by Red Hat	istio-system	Succeeded Up to date	jaeger-operator	Jaeger
Kiali Operator 1.12.11 provided by Red Hat	istio-system	Succeeded Up to date	kiali-operator	Kiali Monitoring Dashboard
Red Hat OpenShift Service Mesh 1.11 provided by Red Hat, Inc.	istio-system	Succeeded Up to date	istio-operator	Istio Service Mesh Control Plane Istio Service Mesh Member Istio Service Mesh Member Roll

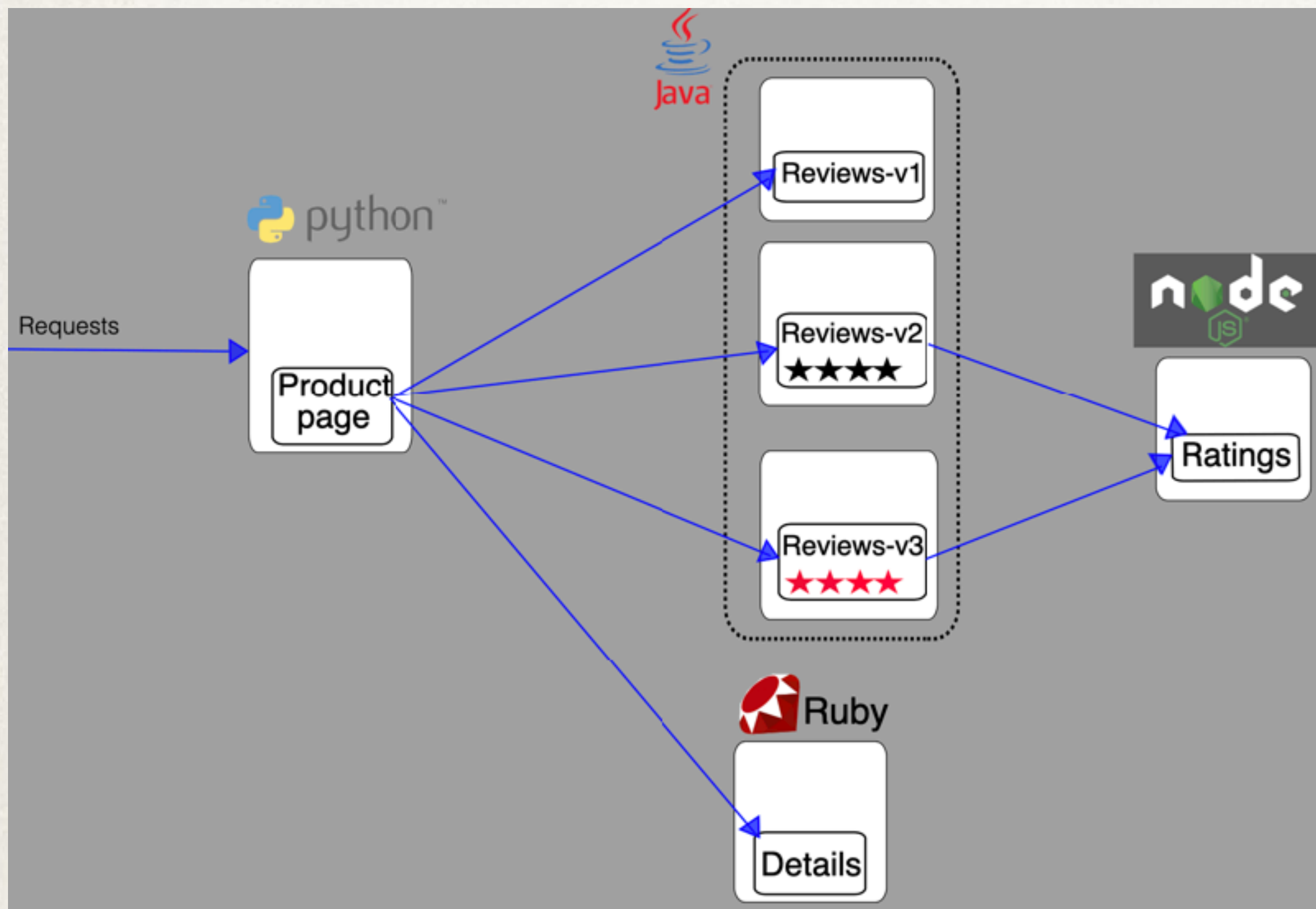
Встановленні оператори для створення сервісної сітки

The screenshot shows the 'Create ServiceMeshControlPlane' form in the console. The form title is 'Create ServiceMeshControlPlane' and the subtitle is 'Create by manually entering YAML or JSON definitions, or by dragging and dropping a file into the editor'. The form contains a text area with the following YAML configuration:

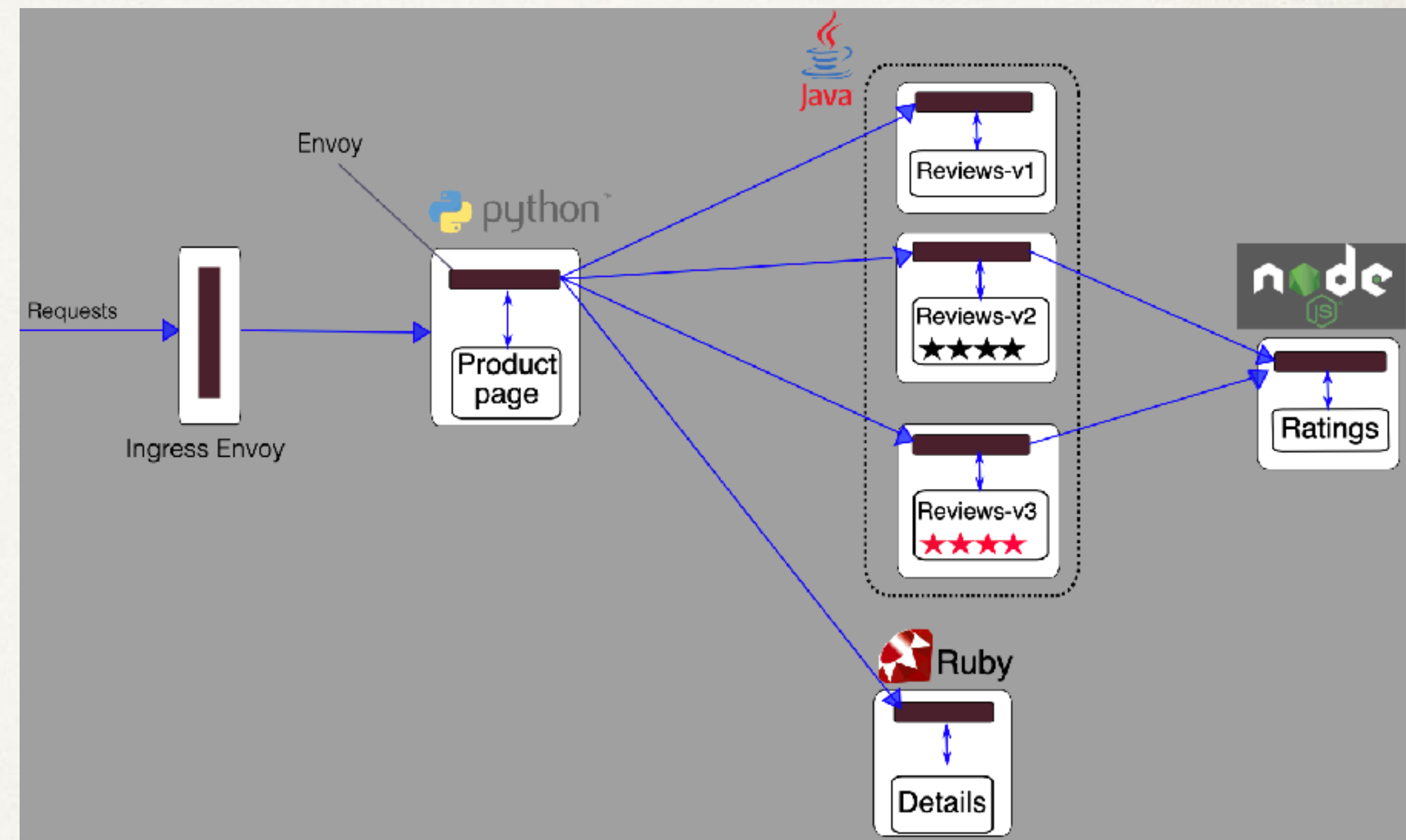
```
1  apiVersion: maistra.io/v1
2  kind: ServiceMeshControlPlane
3  metadata:
4    name: basic-install
5    namespace: istio-example
6  spec:
7    version: v1.1
8    istio:
9      gateways:
10       istio-egressgateway:
11         autoscaleEnabled: false
12       istio-ingressgateway:
13         autoscaleEnabled: false
14         ior_enabled: false
15      mixer:
16       policy:
17         autoscaleEnabled: false
18       telemetry:
19         autoscaleEnabled: false
20      pilot:
21       autoscaleEnabled: false
22       traceSampling: 100
23      kiali:
24       enabled: true
25      grafana:
26       enabled: true
```

Створення ServiceMeshControlPlane в проєкті istio-system

# Архітектура прикладу



Архітектура додатка



Архітектура додатка при використанні сервісної сітки

# Можливості платформи

The screenshot shows the 'Pods' page in the 'bookinfo' project. It features a 'Create Pod' button and a filter input. A summary bar indicates 6 Running pods, 0 Pending, 0 Terminating, 0 CrashLoopBackOff, 0 Completed, 0 Failed, and 0 Unknown. Below is a table of pods:

Name	Namespace	Owner	Node	Status	Readiness
details-v1-d7db4d55b-zcgt4	NS bookinfo	RS details-v1-d7db4d55b	N crc-bq5fv-master-0	Running	Ready
productpage-v1-5f598fbbf4-t47cs	NS bookinfo	RS productpage-v1-5f598fbbf4	N crc-bq5fv-master-0	Running	Ready
ratings-v1-85957d89d8-qvfbq	NS bookinfo	RS ratings-v1-85957d89d8	N crc-bq5fv-master-0	Running	Ready
reviews-v1-67d9b4bcc-5n476	NS bookinfo	RS reviews-v1-67d9b4bcc	N crc-bq5fv-master-0	Running	Ready
reviews-v2-67b465c497-cskk9	NS bookinfo	RS reviews-v2-67b465c497	N crc-bq5fv-master-0	Running	Ready
reviews-v3-7bd659b757-76bdh	NS bookinfo	RS reviews-v3-7bd659b757	N crc-bq5fv-master-0	Running	Ready

Інформація за допомогою вебконсолі про стан pod та можливі дії над ними у тестовому проекті

The screenshot shows the 'Projects' page in the web console. It features a 'Create Project' button and a filter input. A table lists the following projects:

Name	Display Name	Status	Requester	Memory	CPU
PR cloud-native-starter	No display name	Active	kube:admin	-	-
PR default	No display name	Active	No requester	-	-
PR kafka	No display name	Active	kube:admin	-	-
PR kube-node-lease	No display name	Active	No requester	-	-
PR kube-public	No display name	Active	No requester	-	-
PR kube-system	No display name	Active	No requester	-	-
PR openshift	No display name	Active	No requester	-	-
PR openshift-apiserver	No display name	Active	No requester	-	-
PR openshift-apiserver-operator	No display name	Active	No requester	-	-
PR openshift-...	No display name	Active	No requester	-	-

Інформація за допомогою вебконсолі про стан існуючих проектів у кластері та можливі дії над ними

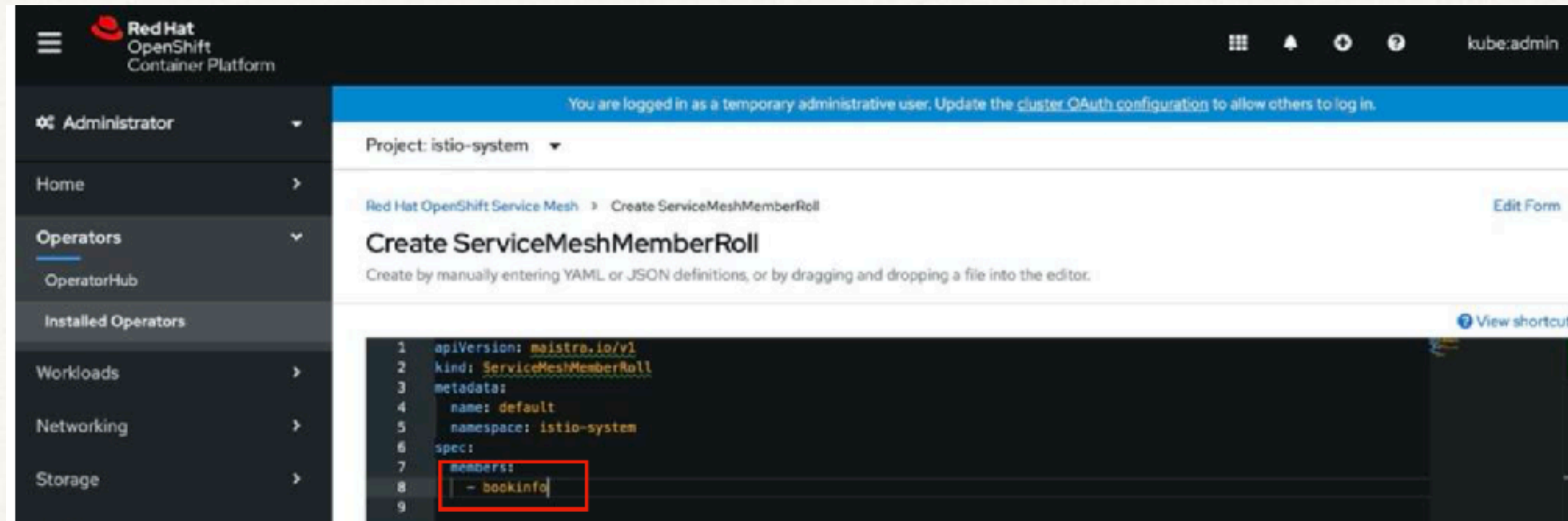
The screenshot shows the 'Machines' page in the web console. It features a 'Create Machine' button and a filter input. A table lists the following machine:

Name	Namespace	Node	Phase	Provider State	Region	Availability Zone
M crc-dv9sm-master-0	NS openshift-machine-api	N crc-dv9sm-master-0	-	Running	-	-

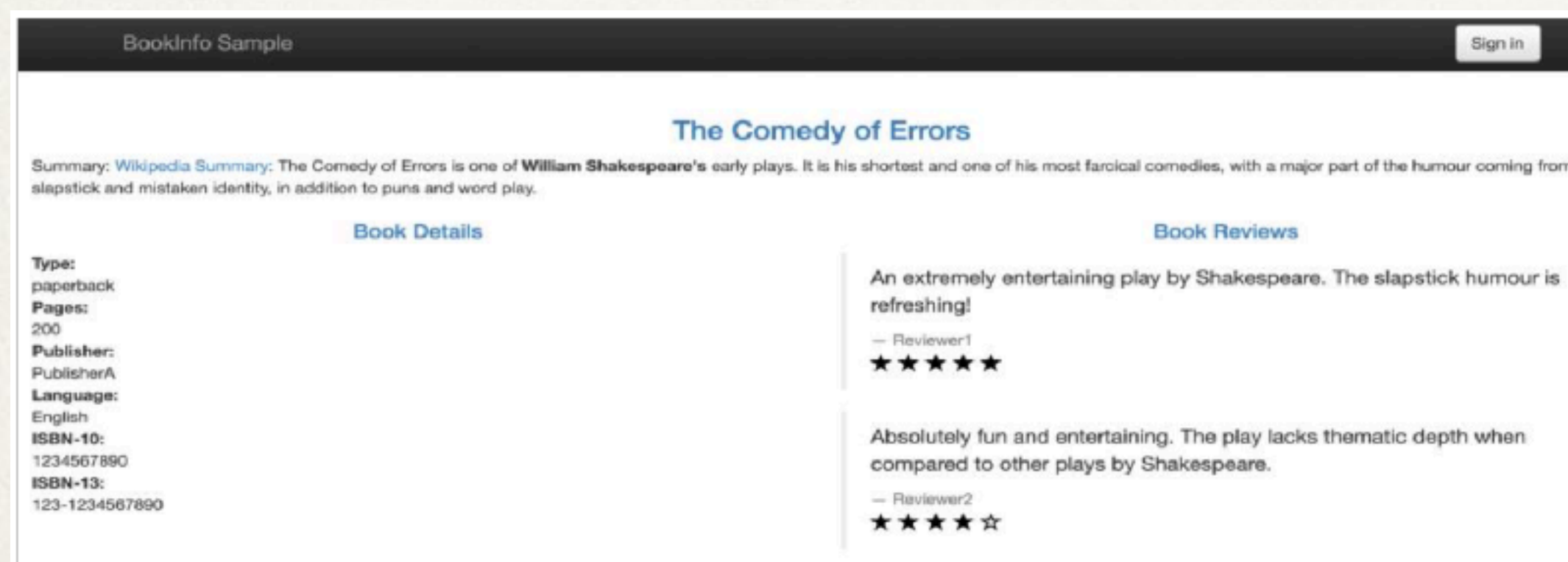
Інформація за допомогою вебконсолі про стан та можливі дії з machines



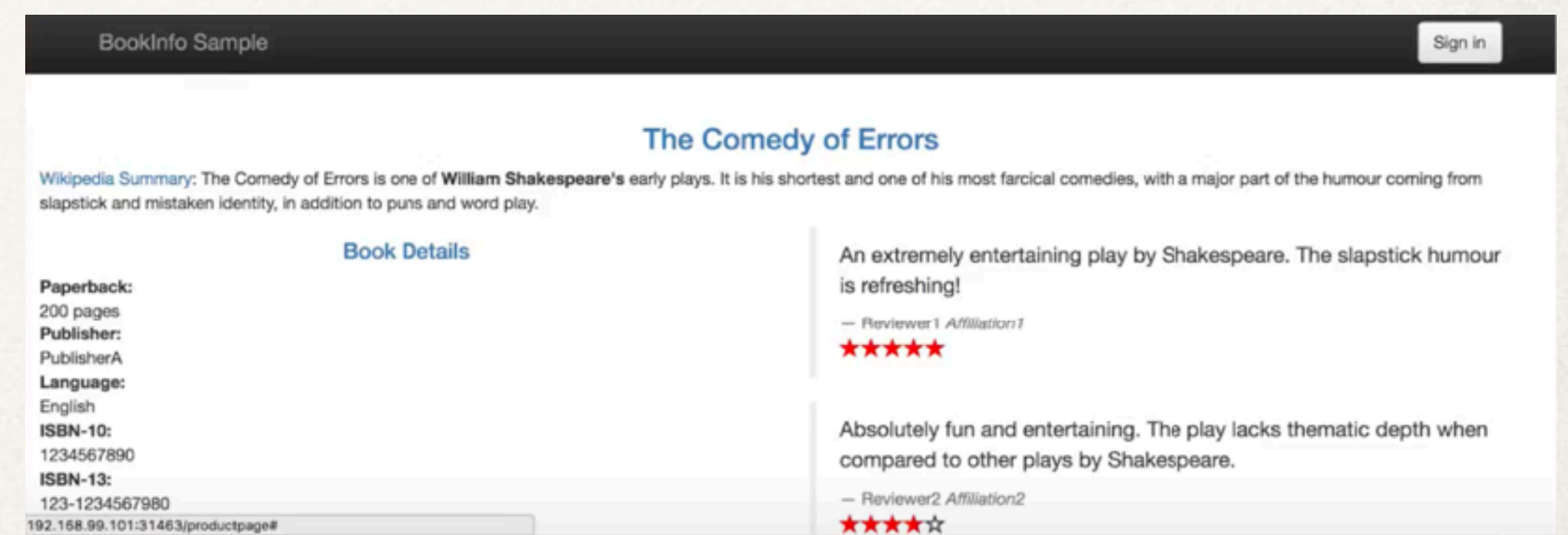
# Приклад “Bookinfo”



Створення ServiceMeshMemberRoll в проєкті istio-system

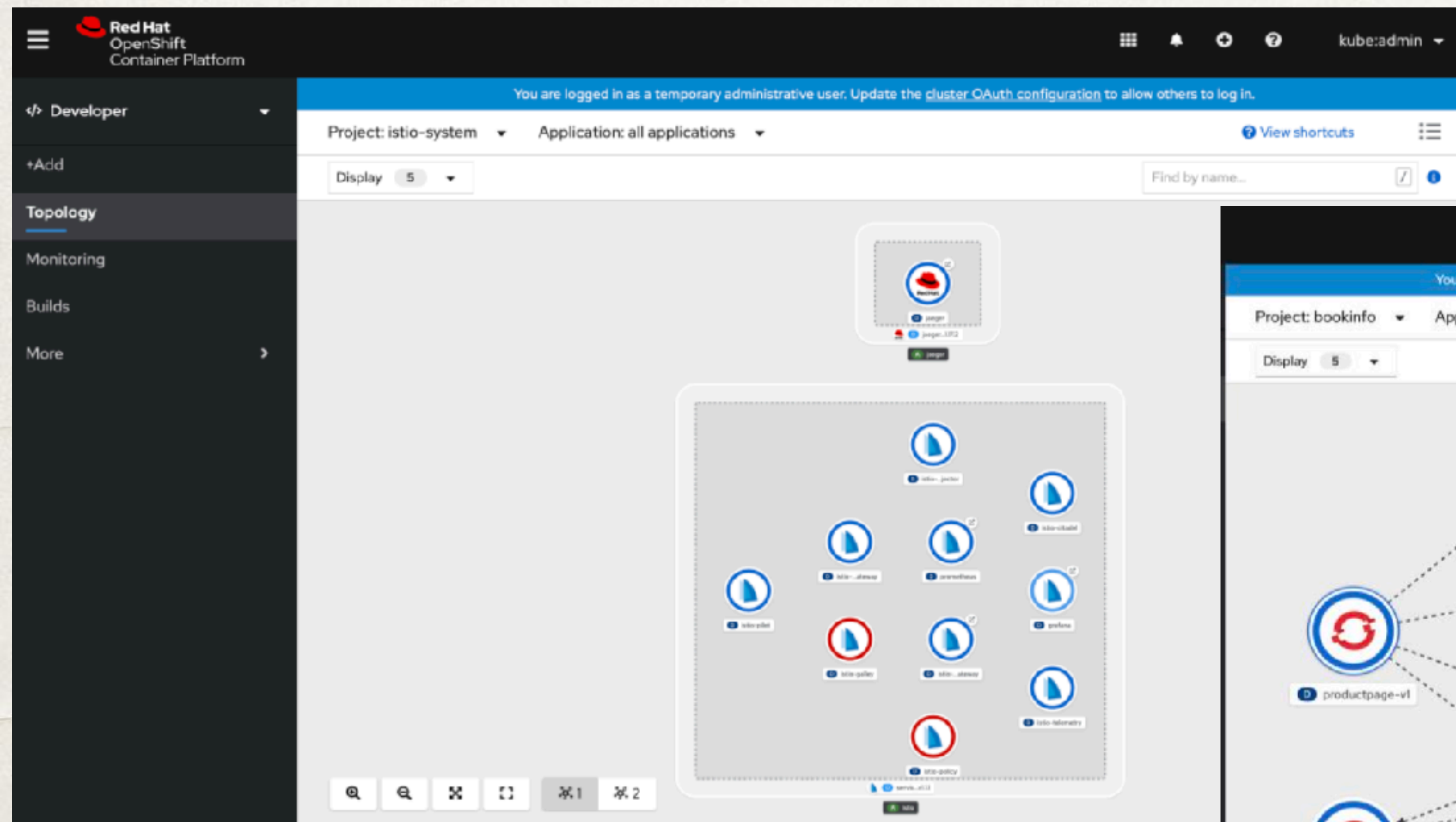


UI тестового додатку

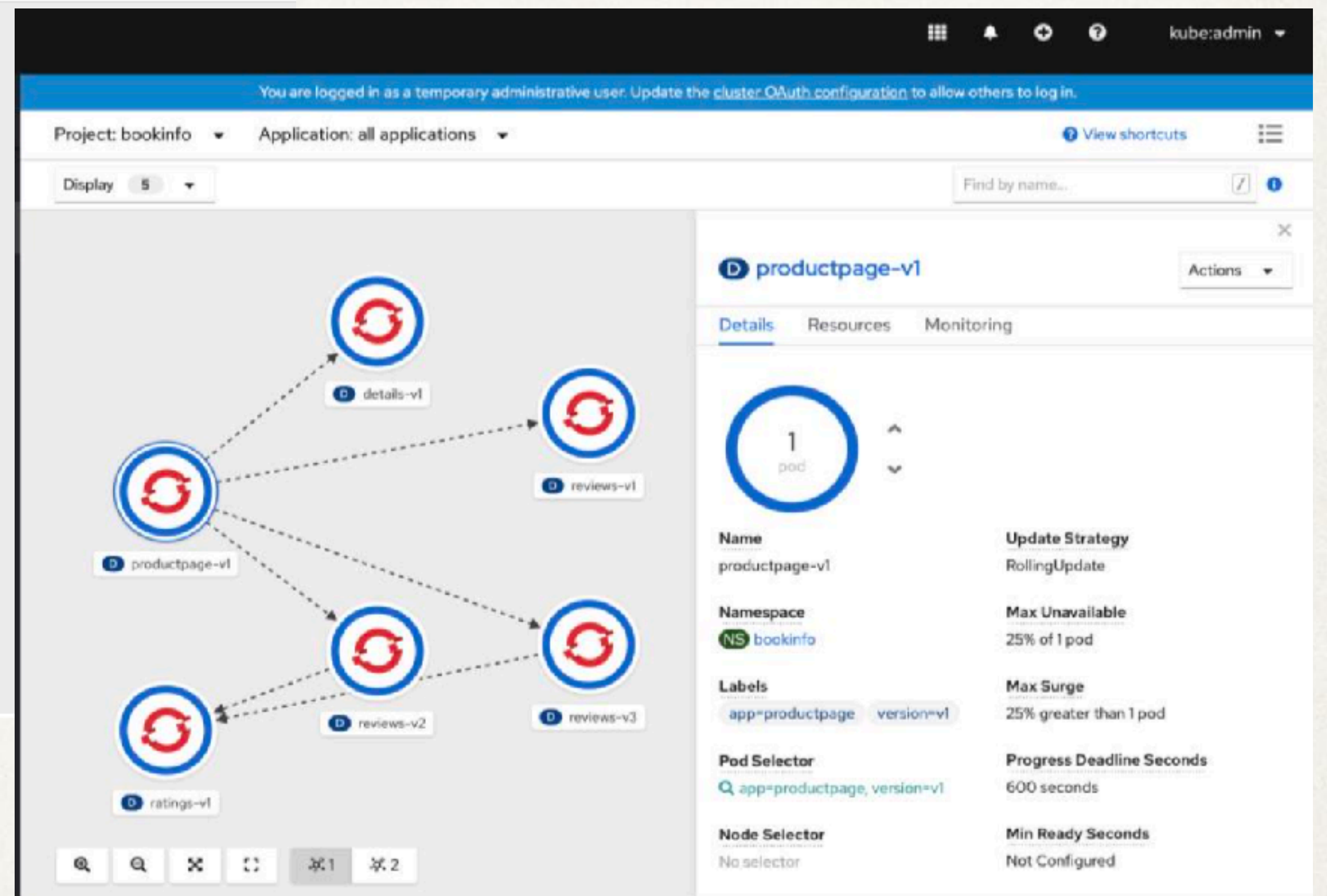


UI тестового додатку

# Моніторинг додатку “Bookinfo”



Візуалізація додатку з панелі Developer  
Топологія проекту istio-system

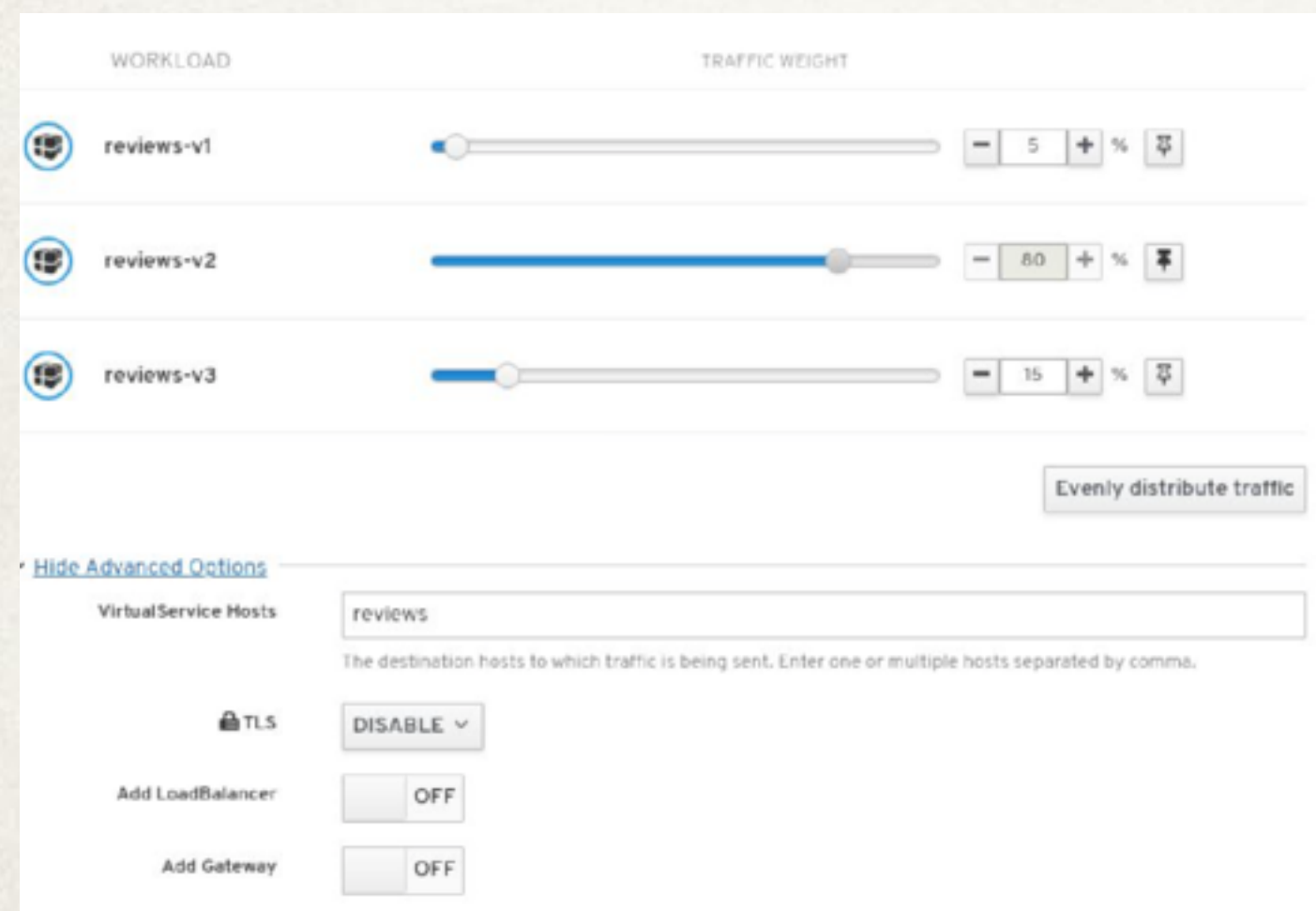


Візуалізація додатку з панелі Developer  
Топологія проекту bookinfo

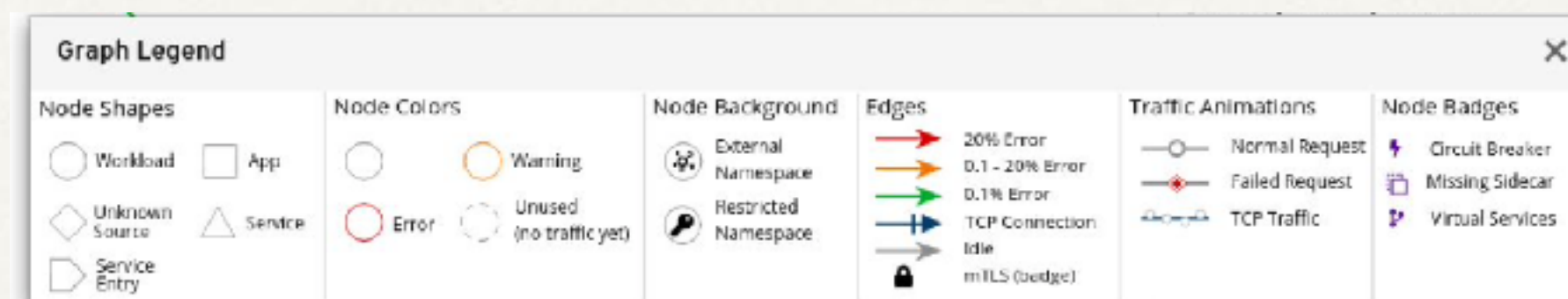
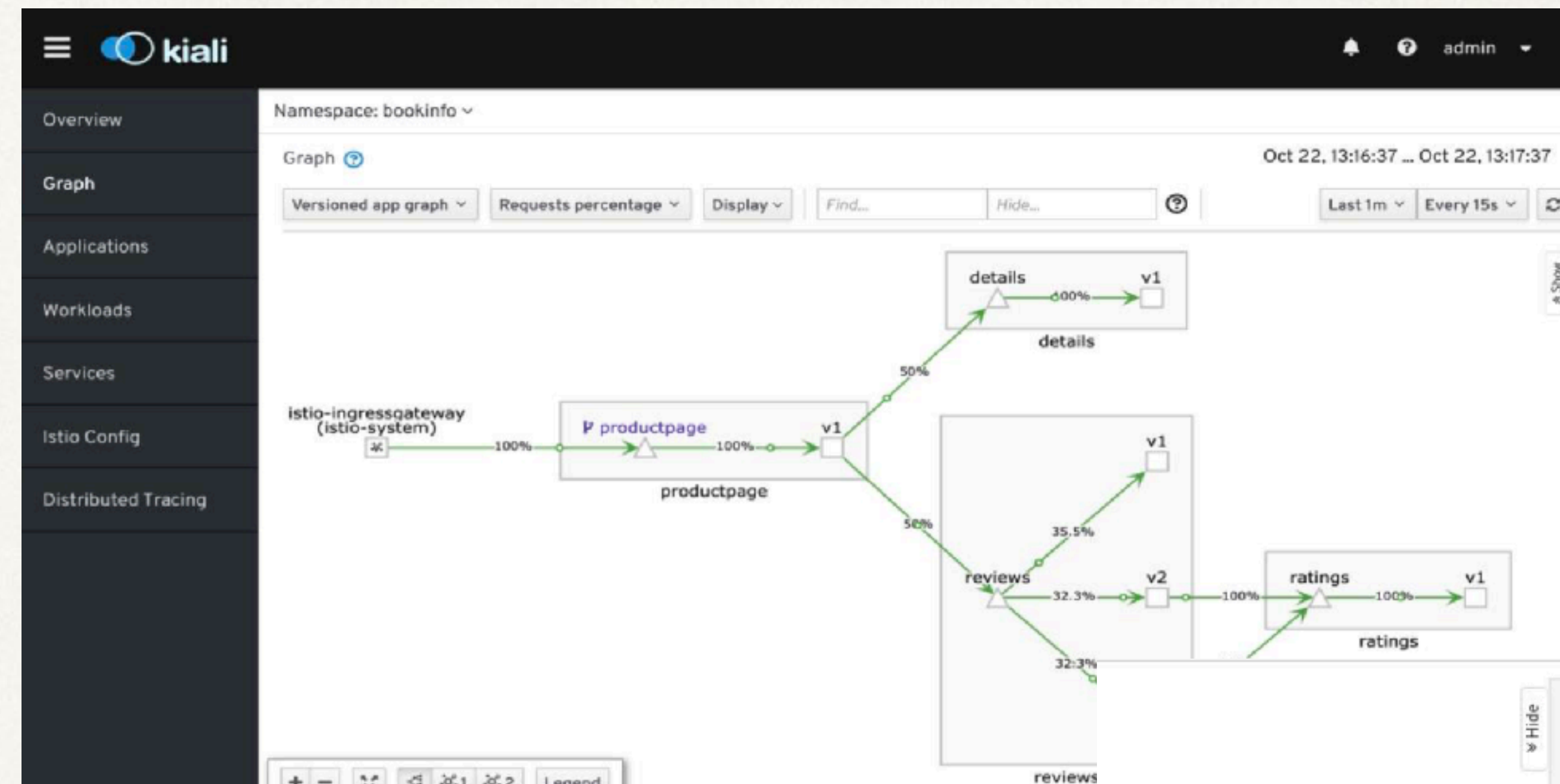
# Моніторинг додатку “Bookinfo”

## Візуалізація за допомогою Kiali:

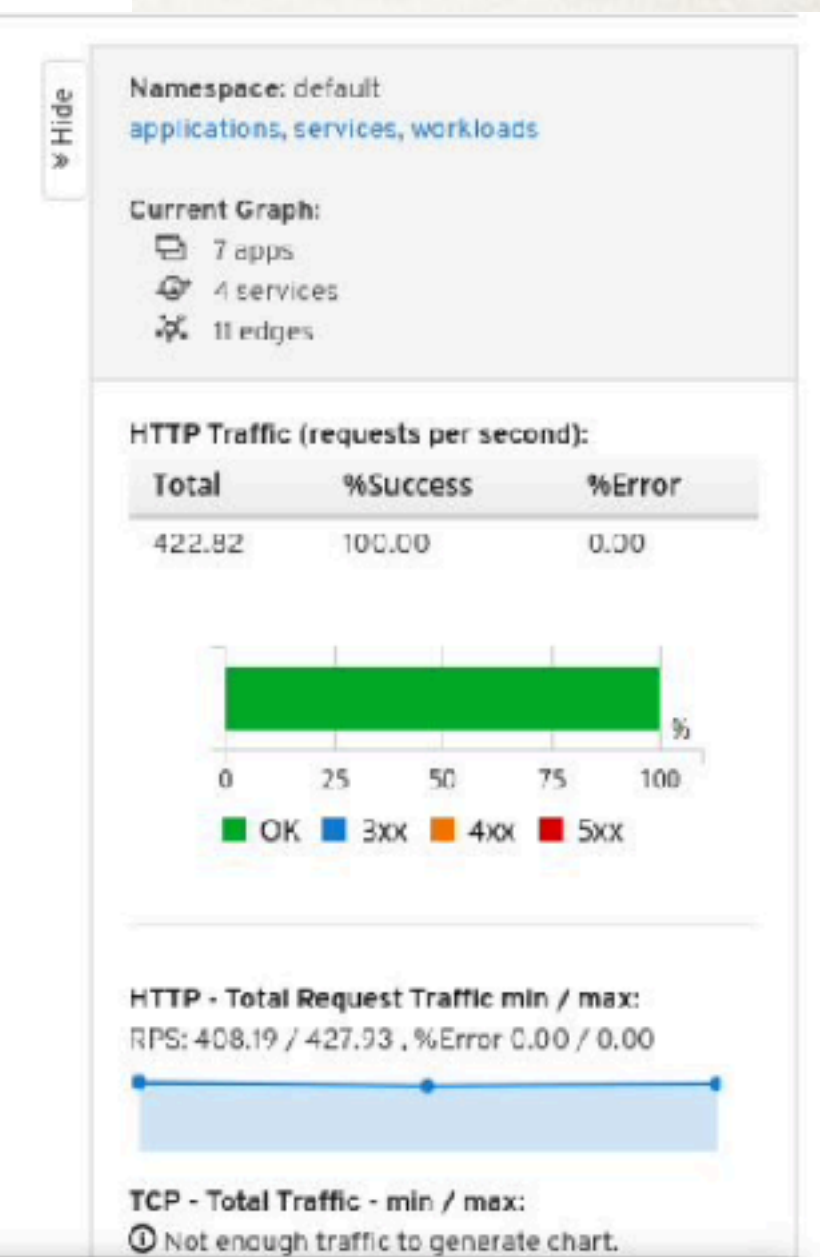
- Топологія додатка
- Пропускна здатність трафіку
- Коефіцієнт помилок
- Затримка сервіса
- Візуалізація сервіса



Навантаження трафіка



Позначення легенд у Kiali



Аналіз специфіки протокола та деталізація за допомогою Kiali

# Висновки

## Основні переваги Red Hat OpenShift Service Mesh:

---

- ◆ Вбудована підтримка таких проектів як Kiali, Jaeger, Istio та чимало інших;
- ◆ Уніфіковані процеси створення, розгортання та керування контейнерними додатками в хмарі;
- ◆ Надає варіанти створення додатку, компоненту або сервера;
- ◆ Легке встановлення сервісної сітки з даними за замовчуванням;
- ◆ Вебконсоль з інформацією про використання потужностей серверу, процесору, оперативної пам'яті, кількість використаної пам'яті сервісом та багато іншого;
- ◆ Реалізація конвеєра CI/CD;
- ◆ Вбудовані можливості Red Hat Platform та підтримка решти продуктів Red Hat.

## Основні недоліки Red Hat OpenShift Service Mesh:

- ◆ Важкість та складність платформи, а інколи й недоречність її використання.
- ◆ Велике навантаження на CPU (88% +).
- ◆ Реліз платформи Red Hat OpenShift Service Mesh був у 2019 році, тому проект є доволі молодим і містить чимало дефектів при роботі.

---

**Дякую за увагу.**