



Construir microservicios en Python. v3.0

Expectativas

VS

REALIDAD



Microservicios en el mundo real

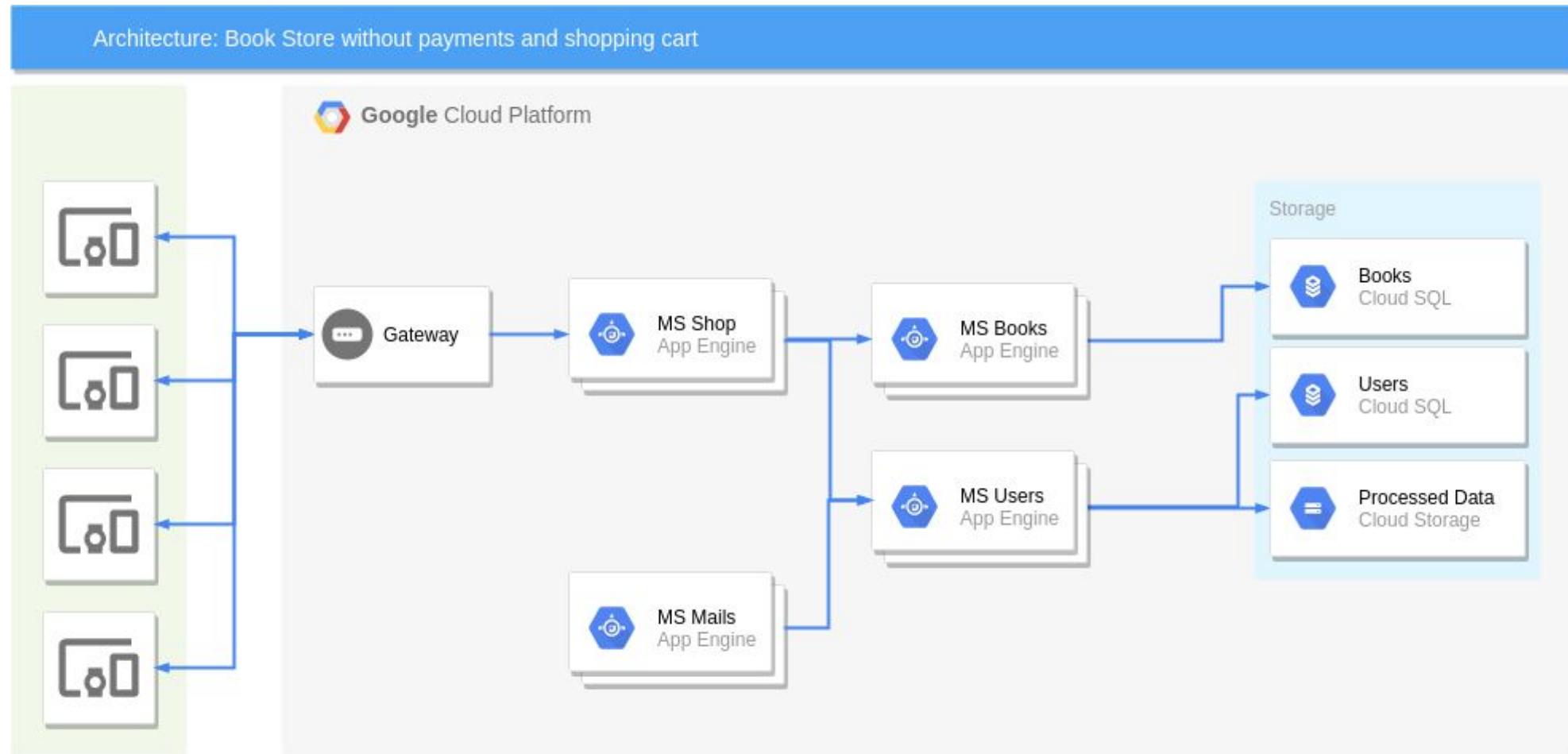
## Blogs y tutoriales de internet al buscar “Build microservice in XXX”

- Añade una ruta GET: /users/
- Añade otra ruta PUT: /users/
- devuelve un JSON:

```
[{  
    "id": 1,  
    "name": "ImTheBoss"  
}]
```

- Actualiza LinkedIn con “Microservice Architect”
- Wait...

## El cliente que te pide hacer esto:

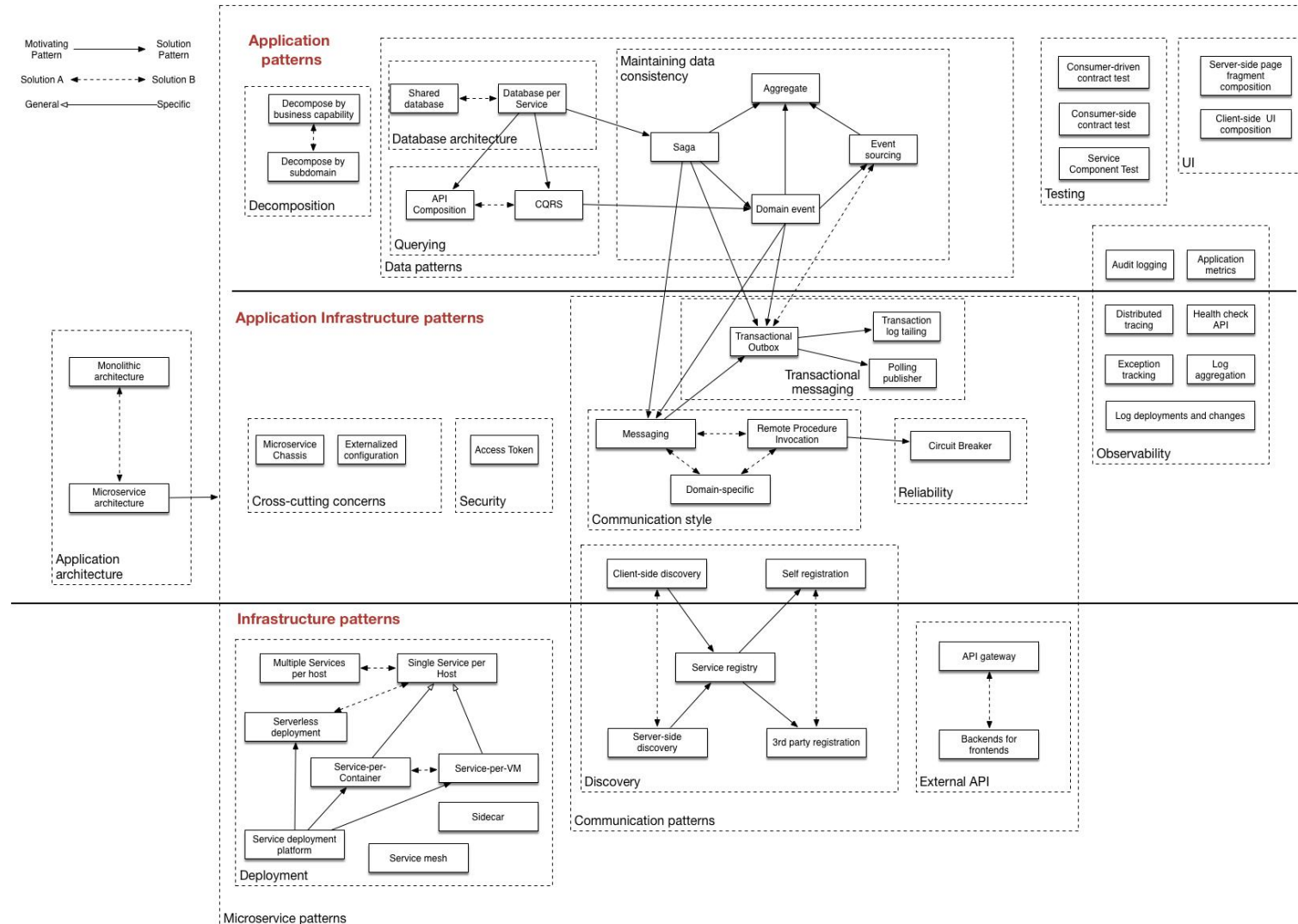


## El cliente que te pide hacer esto:





## Patrones de microservicios





## Patrón: Microservice chassis

- Documentación
- Configuración externalizada
- Trazabilidad de peticiones
- logging
- Health checks
- Metrics



## github.com/python-microservices

PyMS, la librería construida y basada en Flask que unifica todas las librerías necesarias para construir un microservicio. Entre ellas:

- Flask (obvio).
- Connexion y Swagger.
- Prometheus.
- Opentracing y Jaegger (WIP: Opentelemetry)
- Anyconfig.
- Consul.
- Cryptography.



Más literatura de cómo llegué a esto (DRY):

[paradigmadigital.com/dev/como-construir-microservicios-en-python-1-2/](http://paradigmadigital.com/dev/como-construir-microservicios-en-python-1-2/)

[paradigmadigital.com/techbiz/microservice-chassis-pattern-python-2-2/](http://paradigmadigital.com/techbiz/microservice-chassis-pattern-python-2-2/)



## Librería y arquetipos: cómo contribuir

Librería Patrón Chasis para Microservicios:

[github.com/python-microservices/pyms](https://github.com/python-microservices/pyms)

Arquetipo en el que nos hemos basado:

[github.com/python-microservices/microservices-scaffold](https://github.com/python-microservices/microservices-scaffold)

Template con Cookiecutter:

[github.com/python-microservices/cookiecutter-pyms](https://github.com/python-microservices/cookiecutter-pyms)



## Ejemplos

Ejemplos sencillos:

[github.com/python-microservices/pyms/tree/master/examples](https://github.com/python-microservices/pyms/tree/master/examples)

Ejemplo con Kubernetes:

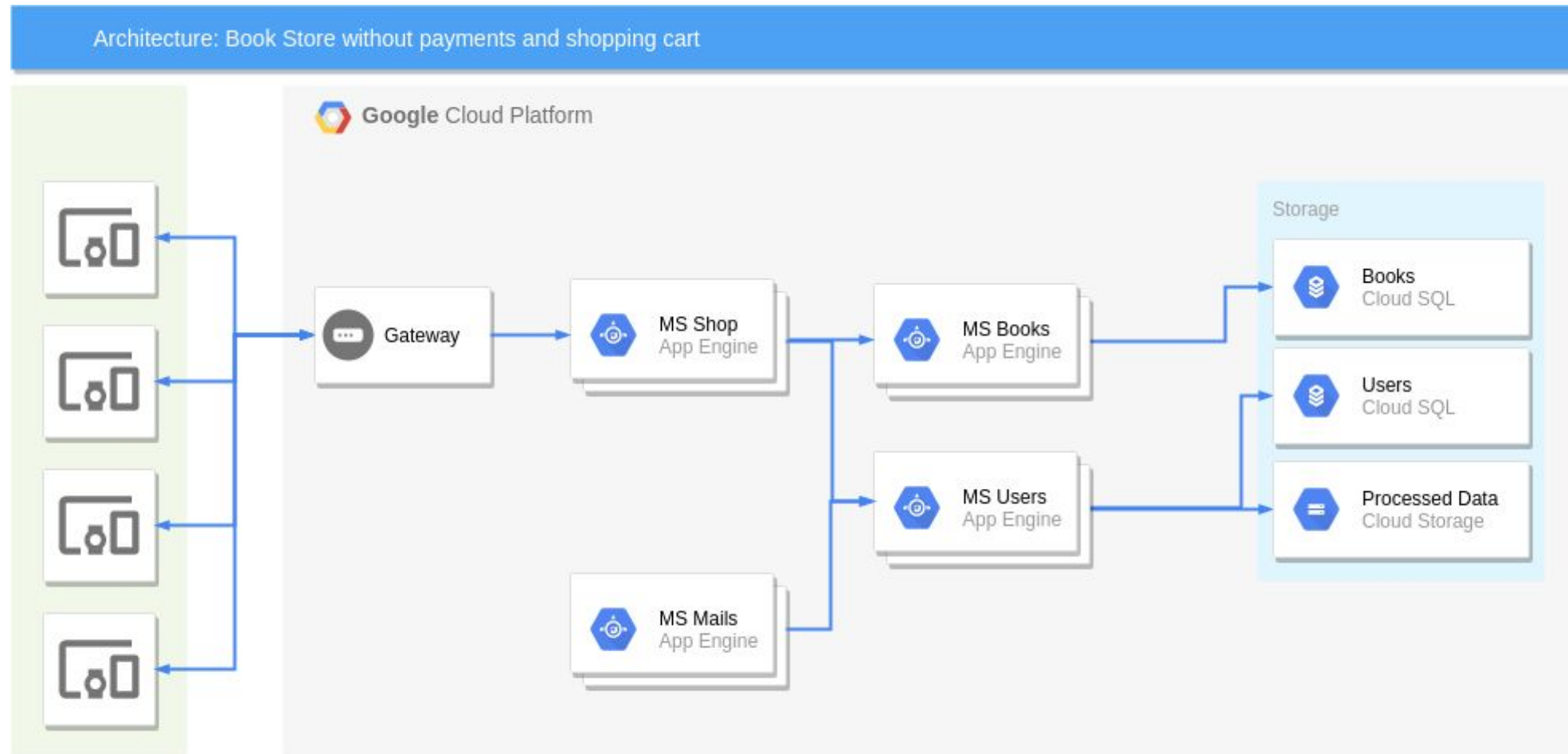
[github.com/python-microservices/microservices-chat](https://github.com/python-microservices/microservices-chat)

Ejemplo con Docker Compose:

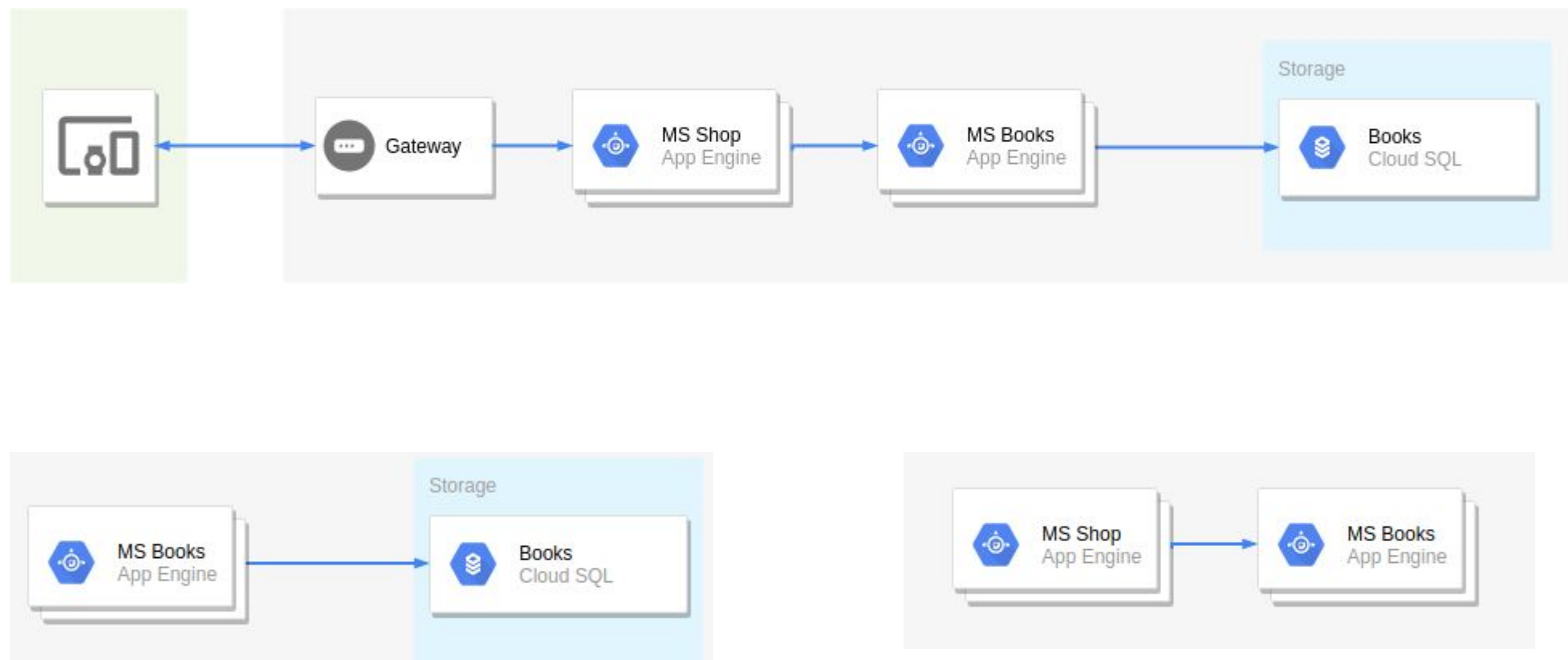
[github.com/avara1986/pivoandcode-2019-11-15](https://github.com/avara1986/pivoandcode-2019-11-15)



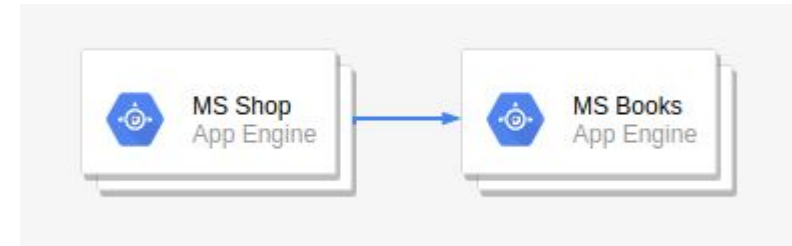
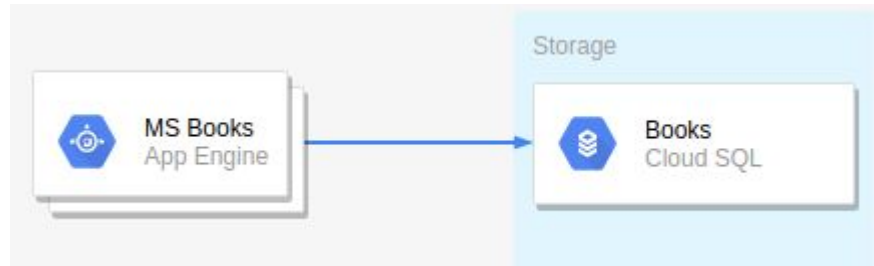
## El cliente que te pide hacer esto:



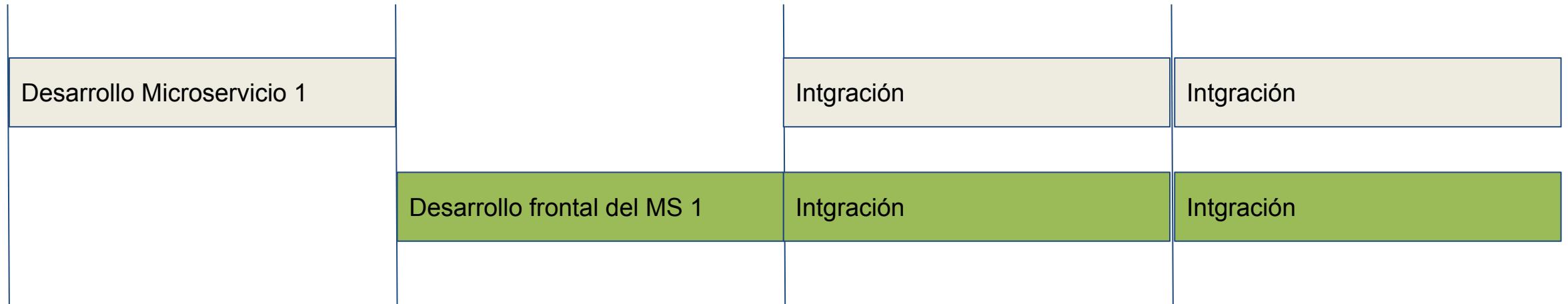
## Problema



## Problema: Documentación



## Problema: Documentación





## Problema: Documentación

### OpenAPI

Estándar para la descripción de APIs

### Swagger

Tecnología más popular para las documentaciones de API

### ~~API First~~ Spec First

Documentar primero antes de trabajar

## Problema: Documentación

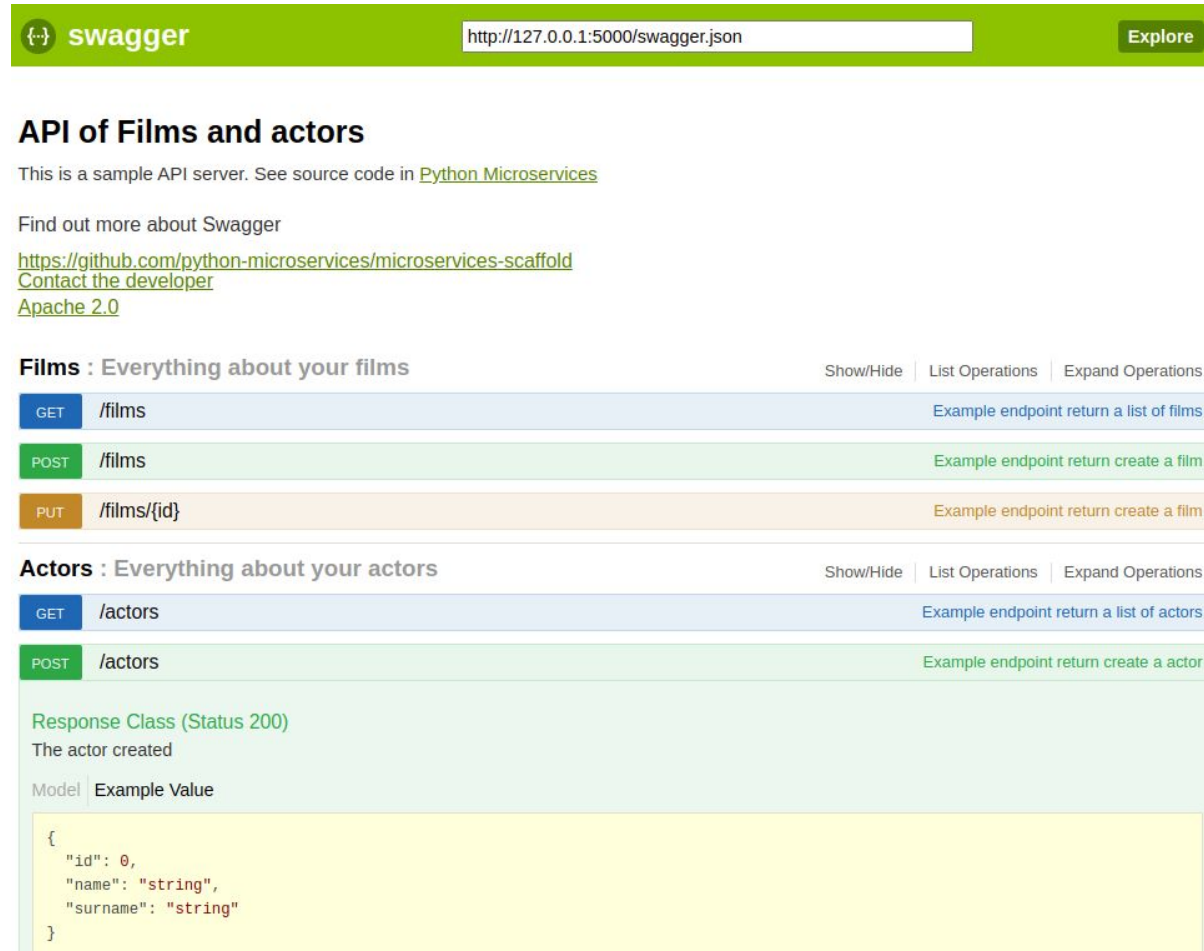
### OpenAPI + Swagger

```
get:
  tags:
  - "Actors"
  summary: "Example endpoint return a list of actors"
  description: ""
  consumes:
  - "application/json"
  produces:
  - "application/json"
  responses:
    "200":
      description: "A list of actors"
      schema:
        $ref: '#/definitions/Actors'
    "400":
      description: "Invalid ID supplied"
    "404":
      description: "Actor not found"
    "405":
      description: "Validation exception"
```

```
post:
  tags:
  - "messages"
  summary: "Example endpoint return create a messages"
  description: ""
  operationId: "create_view"
  consumes:
  - "application/json"
  produces:
  - "application/json"
  parameters:
    - name: user_id
      in: formData
      type: string
      required: true
      description: "Id of the user"
    - name: username
      in: formData
      type: string
      required: true
```

# Problema: Documentación

## OpenAPI + Swagger



The image shows the Swagger UI interface for an API. At the top, there's a green bar with the Swagger logo, a text input field containing 'http://127.0.0.1:5000/swagger.json', and an 'Explore' button. Below this, the title 'API of Films and actors' is displayed, followed by a description: 'This is a sample API server. See source code in [Python Microservices](#)'. There are also links to 'Find out more about Swagger', 'https://github.com/python-microservices/microservices-scaffold', 'Contact the developer', and 'Apache 2.0'.

The main content is divided into two sections: 'Films : Everything about your films' and 'Actors : Everything about your actors'. Each section has a 'Show/Hide', 'List Operations', and 'Expand Operations' link. Under 'Films', there are three operations: GET /films (Example endpoint return a list of films), POST /films (Example endpoint return create a film), and PUT /films/{id} (Example endpoint return create a film). Under 'Actors', there are two operations: GET /actors (Example endpoint return a list of actors) and POST /actors (Example endpoint return create a actor). The POST /actors operation is expanded, showing a 'Response Class (Status 200)' with the message 'The actor created'. Below this, there's a 'Model' section with an 'Example Value' showing a JSON object: { "id": 0, "name": "string", "surname": "string" }.



# Problema: Documentación


Documentar	Desarrollo Microservicio 1	Desarrollo Microservicio 1	Integración		
Documentar	Desarrollo frontal del MS	Desarrollo frontal del MS	Integración		





# Problema: Documentación

Documentar	Desarrollo Microservicio 1	Desarrollo Microservicio 1	Integración	
Documentar	Desarrollo frontal del MS	Desarrollo frontal del MS	Integración	



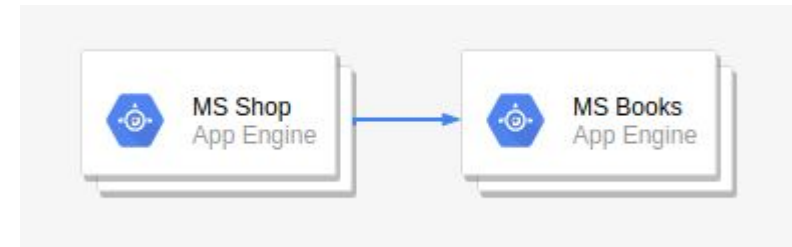
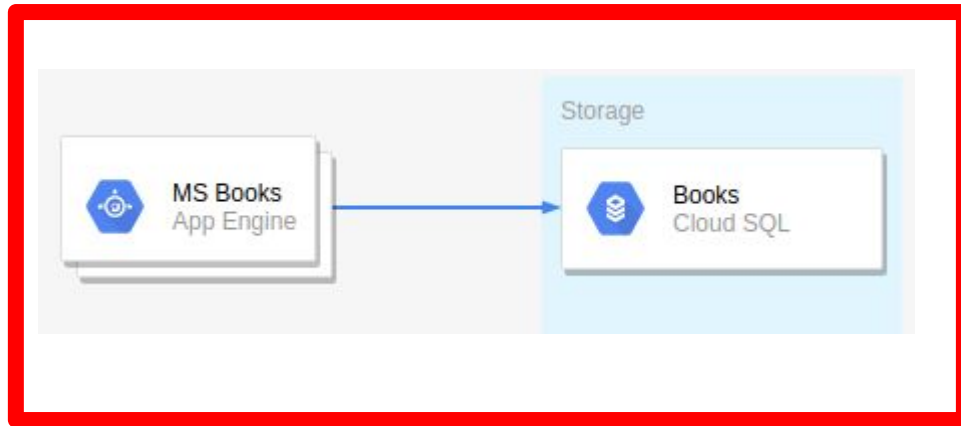
## Problema: Documentación

### Proyectos y librerías OpenAPI + Swagger

- Connexion [connexion.readthedocs.io](https://connexion.readthedocs.io)
- django-rest-framework [www.django-rest-framework.org](https://www.django-rest-framework.org)
- flask-restplus [flask-restplus.readthedocs.io](https://flask-restplus.readthedocs.io)
- flask-restful [flask-restful.readthedocs.io](https://flask-restful.readthedocs.io)
- FastAPI [fastapi.tiangolo.com](https://fastapi.tiangolo.com)



## Problema: Configuración y desarrollo en local



## Problema: Configuración y desarrollo en local



## Problema: Configuración y desarrollo en local

```
class Config:
    DEBUG = False
    TESTING = False
    APP_NAME = "Template"
    SQLALCHEMY_DATABASE_URI = "sqlite:///..."

class DevConfig(Config):

class TestConfig(Config):

class PreConfig(Config):

class ProConfig(Config):
```

## Problema: Configuración y desarrollo en local

### Uso de variables de entorno. config.py

```
class Config:
    DEBUG = False
    TESTING = False
    APP_NAME = "Template"
    SQLALCHEMY_DATABASE_URI = os.getenv("DB")
```

### Configuración externalizada. config.yaml

```
ms:

    DEBUG: false

    TESTING: false

    APP_NAME: Template

    SQLALCHEMY_DATABASE_URI : sqlite:/...
```

# Problema: Configuración y desarrollo en local

## Docker compose

```
services:

  postgresql:

    image: postgres

  books:

    image:

    depends_on:

      - mysql

  shop:

    image:

    depends_on:

      - books
```

## Kubernetes (Minikube) y Helm

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ include "chat_db.fullname" . }}
  labels:
spec:
  replicas: {{ .Values.replicaCount }}
  template:
    metadata:
      labels:
        app.kubernetes.io/name: {{ include "chat_db.name" . }}
    spec:
      containers:
        - name: {{ .Chart.Name }}
          image: "{{ .Values.image.repository }}:{{ .Values.image.tag }}"
          ports:
            - name: http
              containerPort: 8080
              protocol: TCP
```



## Problema: Configuración y desarrollo en local

### Proyectos y librerías

- PyYAML <https://pyyaml.org/>
- anyconfig [github.com/ssato/python-anyconfig](https://github.com/ssato/python-anyconfig)
- Vault [github.com/hvac/hvac](https://github.com/hvac/hvac)
- Kubernetes Secrets [kubernetes.io/es/docs/concepts/configuration/secret/](https://kubernetes.io/es/docs/concepts/configuration/secret/)





## Ejemplos

Ejemplo con Kubernetes:

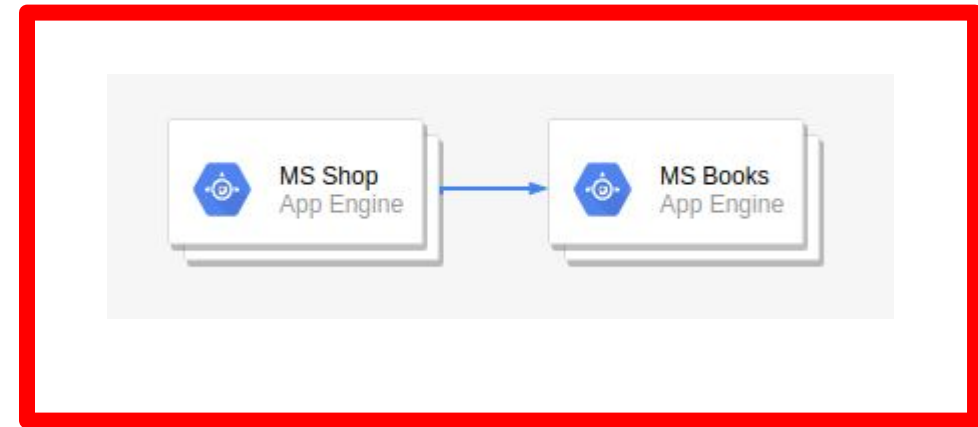
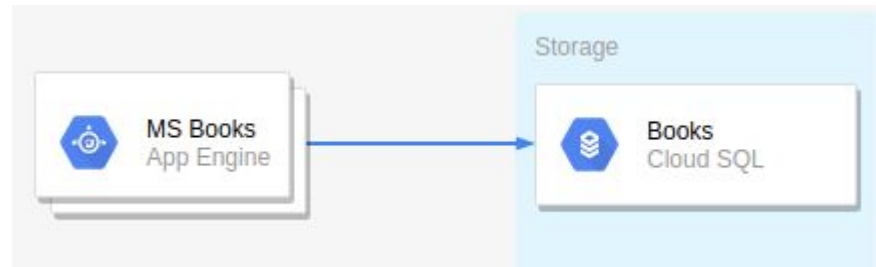
[github.com/python-microservices/microservices-chat](https://github.com/python-microservices/microservices-chat)

Ejemplo con Docker Compose:

[github.com/avara1986/pivoandcode-2019-11-15](https://github.com/avara1986/pivoandcode-2019-11-15)



## Problema: Trazabilidad y Logging



## Trazabilidad Distribuida

En una frase: poder hacer seguimiento de una request a través de los diferentes microservicios por donde pasa

Pero la trazabilidad engloba muchísimo más

### Más Info:

Trazabilidad distribuida [paradigmadigital.com/dev/trazabilidad-distribuida-con-opentracing-y-jaeger/](https://paradigmadigital.com/dev/trazabilidad-distribuida-con-opentracing-y-jaeger/)

Patrón de Trazabilidad distribuida [microservices.io/patterns/observability/distributed-tracing](https://microservices.io/patterns/observability/distributed-tracing)

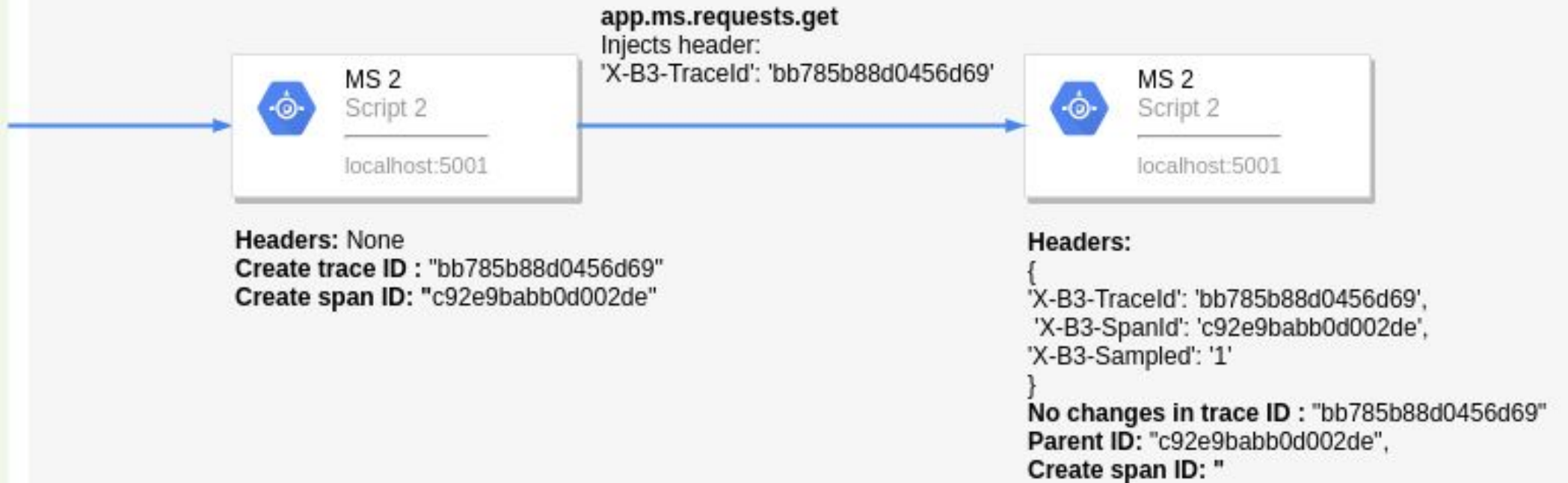
Ejemplo de trazabilidad: [python-microservices.github.io/tutorials/tutorial\\_propagate\\_traces/](https://python-microservices.github.io/tutorials/tutorial_propagate_traces/)

## Trazabilidad Distribuida



PyMS: Distributed tracing, <https://py-ms.readthedocs.io>

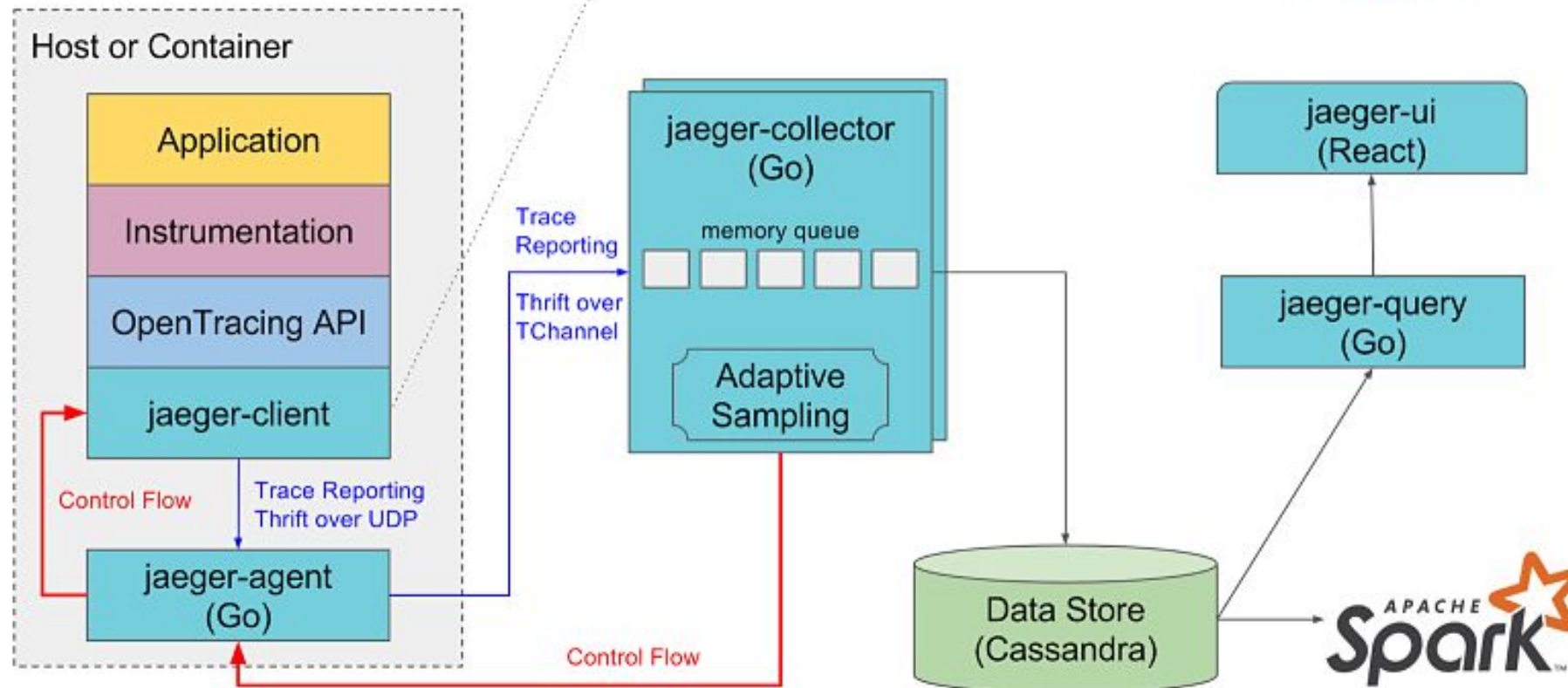
CURL request



# Jaeger

OPENTRACING

## JAEGER



## Problema: Trazabilidad

### Clientes

- jaeger [jaegertracing.io](https://jaegertracing.io)
- LightStep [lightstep.com](https://lightstep.com)
- Datadog [datadoghq.com](https://datadoghq.com)
- ElasticAPM [elastic.co](https://elastic.co)
- Opentracing [opentracing.io](https://opentracing.io)
- OpenTelemetry [opentelemetry.io](https://opentelemetry.io)



# Healthcheck y Metrics

## Metrics

<http://localhost:8000/metrics>

```
# HELP go_gc_duration_seconds A summary of the GC invocation durations.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 3.291e-05
go_gc_duration_seconds{quantile="0.25"} 4.3849e-05
go_gc_duration_seconds{quantile="0.5"} 6.2452e-05
go_gc_duration_seconds{quantile="0.75"} 9.8154e-05
go_gc_duration_seconds{quantile="1"} 0.011689149
go_gc_duration_seconds_sum 3.451780079
go_gc_duration_seconds_count 13118
```

## Healthcheck

<http://localhost:8000/healthcheck>

```
curl -IX GET "http://localhost:8000/healthcheck"
HTTP/1.1 200
Content-Type: text/html; charset=utf-8
Date: Sun, 15 Sep 2019 12:32:22 GMT
```



# Healthcheck y Metrics

## Proyectos y librerías

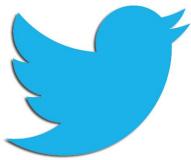
- Prometheus [github.com/prometheus/client\\_python](https://github.com/prometheus/client_python)
- Graphana [grafana.com](https://grafana.com)

¡GRACIAS POR VUESTRO TIEMPO!

[github.com/avara1986/pycones-2021](https://github.com/avara1986/pycones-2021)



[github.com/avara1986](https://github.com/avara1986)



[twitter.com/avara1986](https://twitter.com/avara1986)



Hdiv

¡Hdiv estamos contratando!  
[careers@hdivsecurity.com](mailto:careers@hdivsecurity.com)

LinkedIn

[linkedin.com/in/albertovara/](https://linkedin.com/in/albertovara/)



[a.vara.1986@gmail.com](mailto:a.vara.1986@gmail.com)