Microservices Observability

sensedia

Cláudio E. de Oliveira

claudio.oliveira@sensedia.com
+55 19 3705-5775

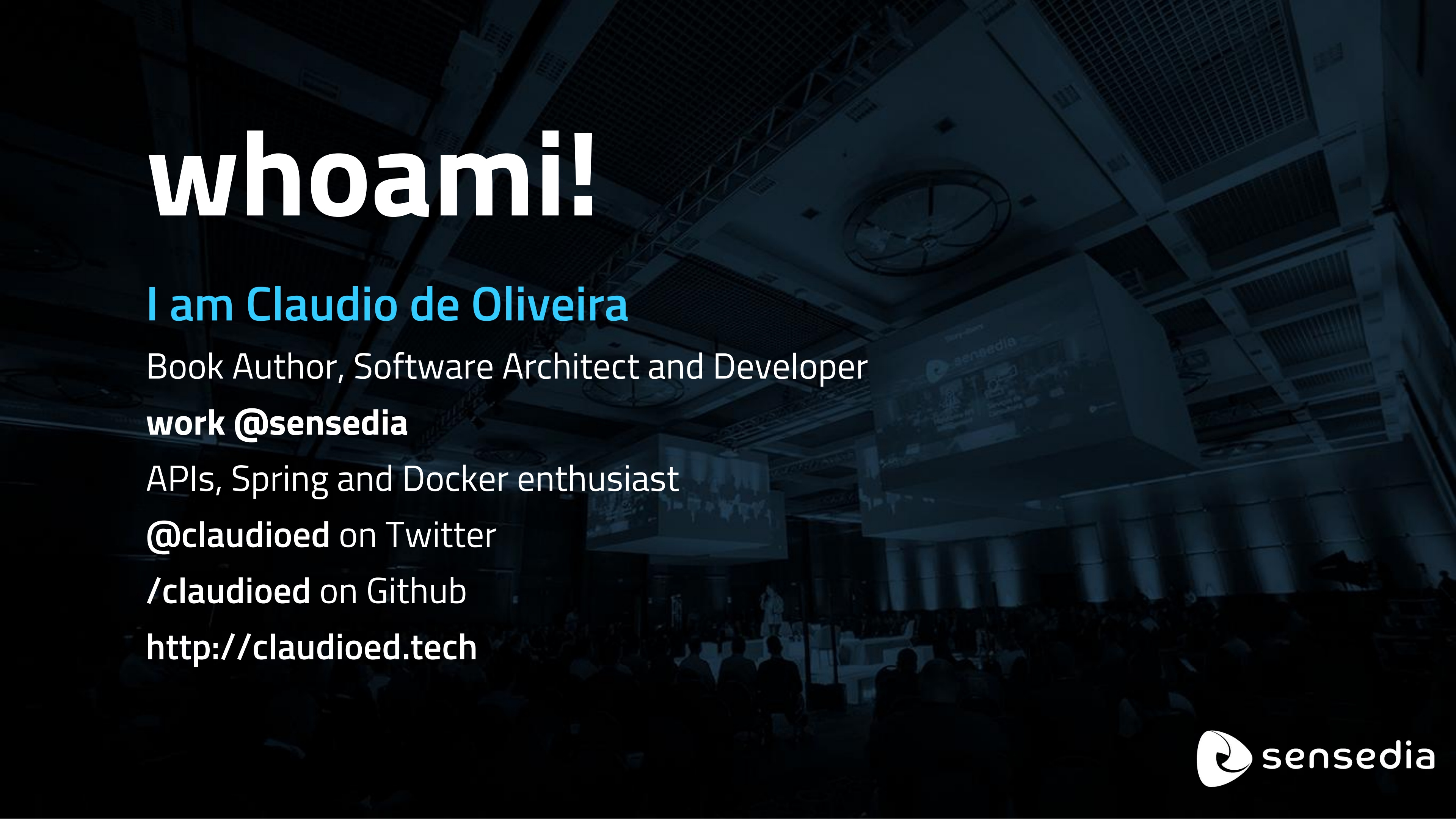
AGENDA

1. Microservices Drawbacks
2. Kubernetes
3. Open Tracing
4. Spring Cloud Sleuth
5. Service Mesh
6. ISTIO

APIX
API EXPERIENCE

Agenda

- Microservices Drawbacks
- Kubernetes
- Open Tracing
- Spring Cloud Sleuth
- Service Mesh
- ISTIO



whoami!

I am Claudio de Oliveira

Book Author, Software Architect and Developer

work @sensedia

APIs, Spring and Docker enthusiast

@claudioed on Twitter

/claudioed on Github

http://claudioed.tech

sensedia

Claudio Eduardo de Oliveira

Spring 5.0 By Example

Grasp the fundamentals of Spring 5.0 to build modern, robust, and scalable Java applications

Packt>

sensedia

Glossary

- **Telemetry**

  How to collect data that will provide observability *(sensors)*

- **Observability**

  Monitoring, Alerting and Visualizations, Distributed Tracing and Log Aggregation.

MICROSERVICES DRAWBACKS

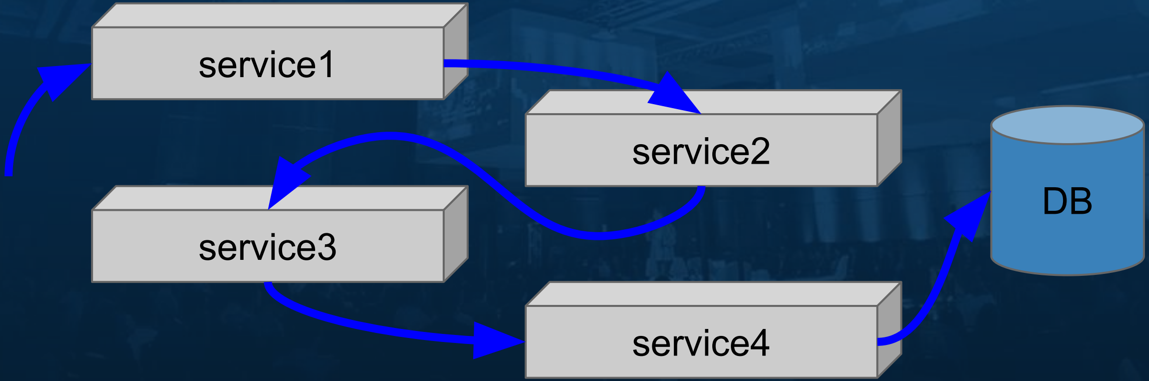
Fallacies of distributed computing

- the network is reliable
- latency is zero
- bandwidth is infinite

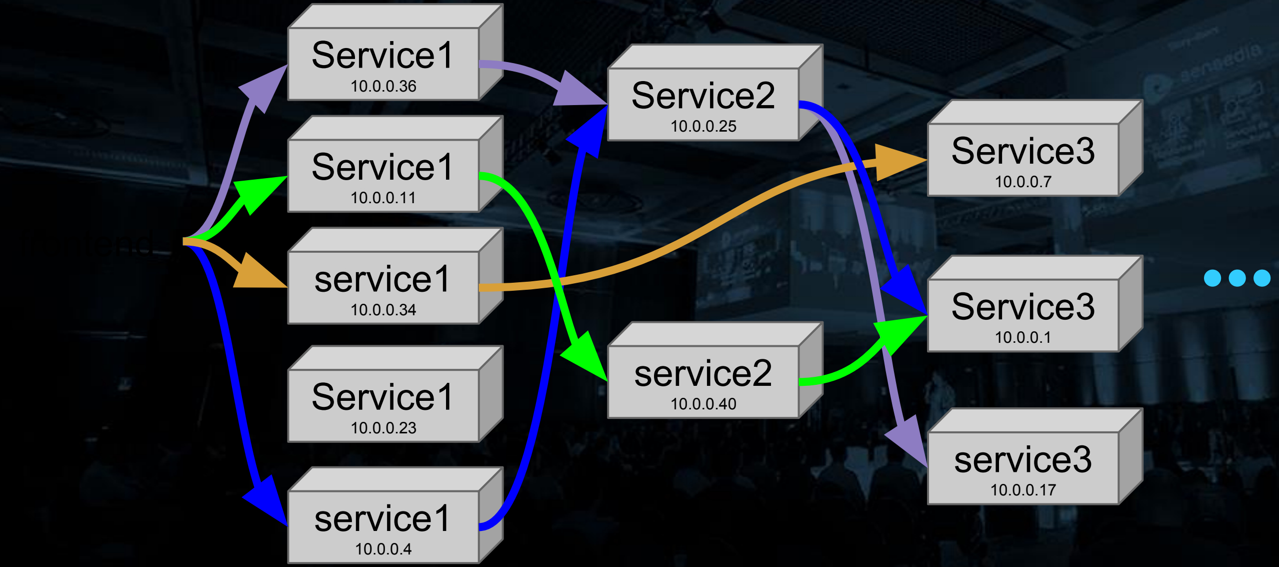
Microservices Drawbacks

- How microservices connect each other
- Network Latencies can be a bottleneck
- Network can be unreliable
- Control the UP and Running instances
- It increase the non-functional requirements

your system doesn't look like this:



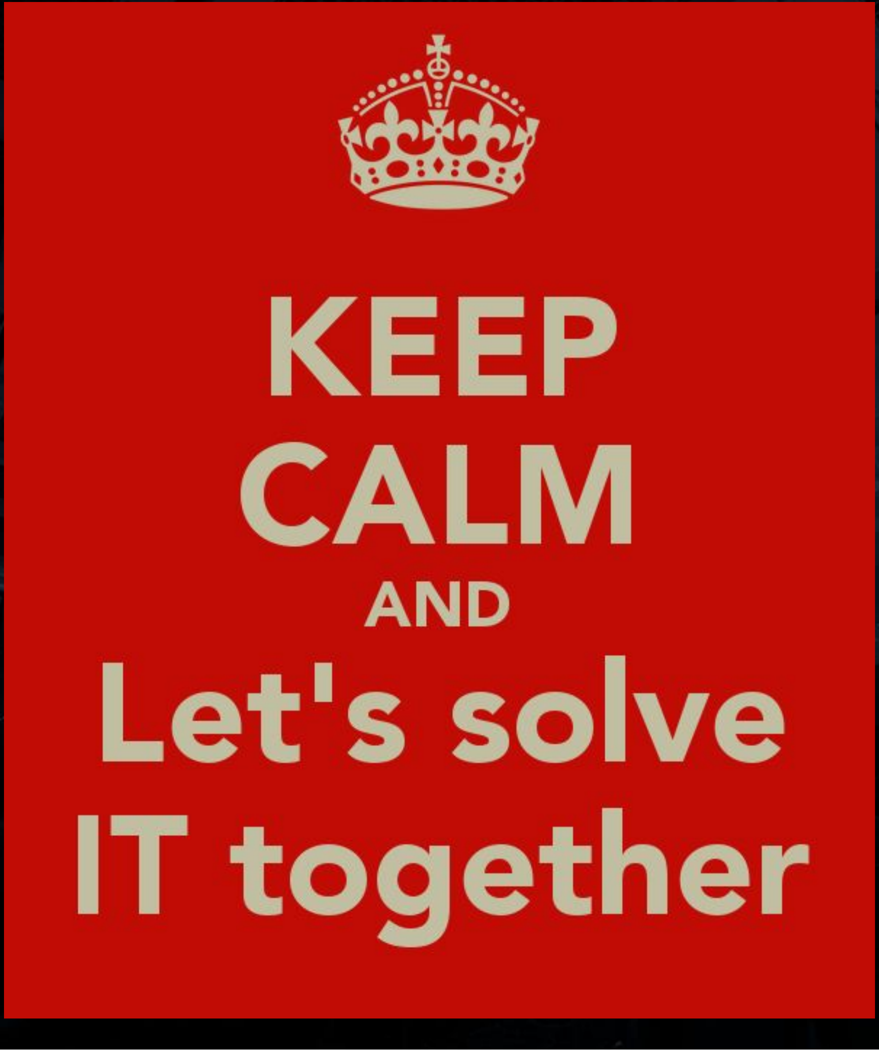
...it looks more like this:



sensedia

Network Latency Problems

- Which microservice was responsible?
- How to find out where the defect is?
- How know what is generally happening during a request?

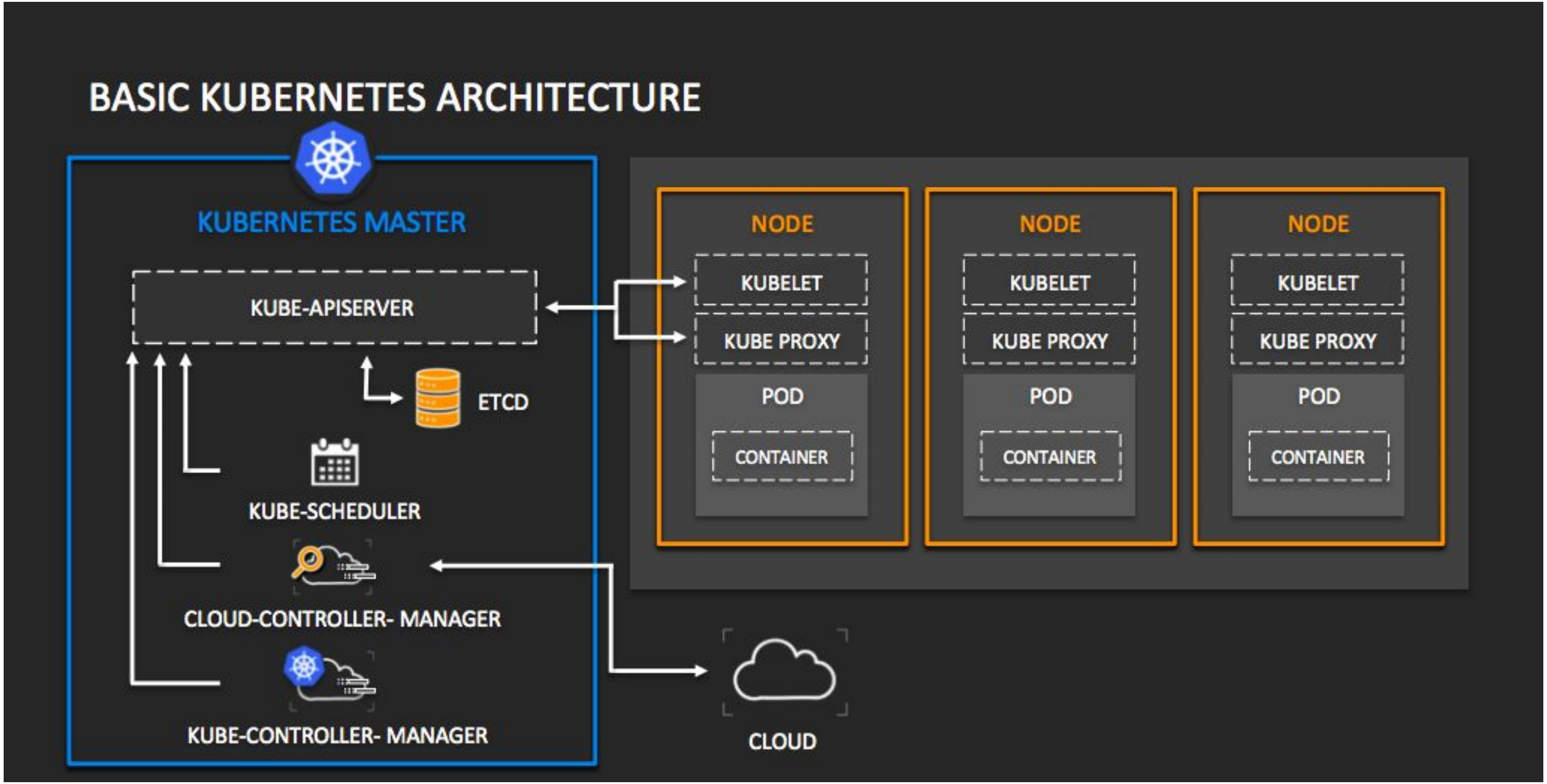


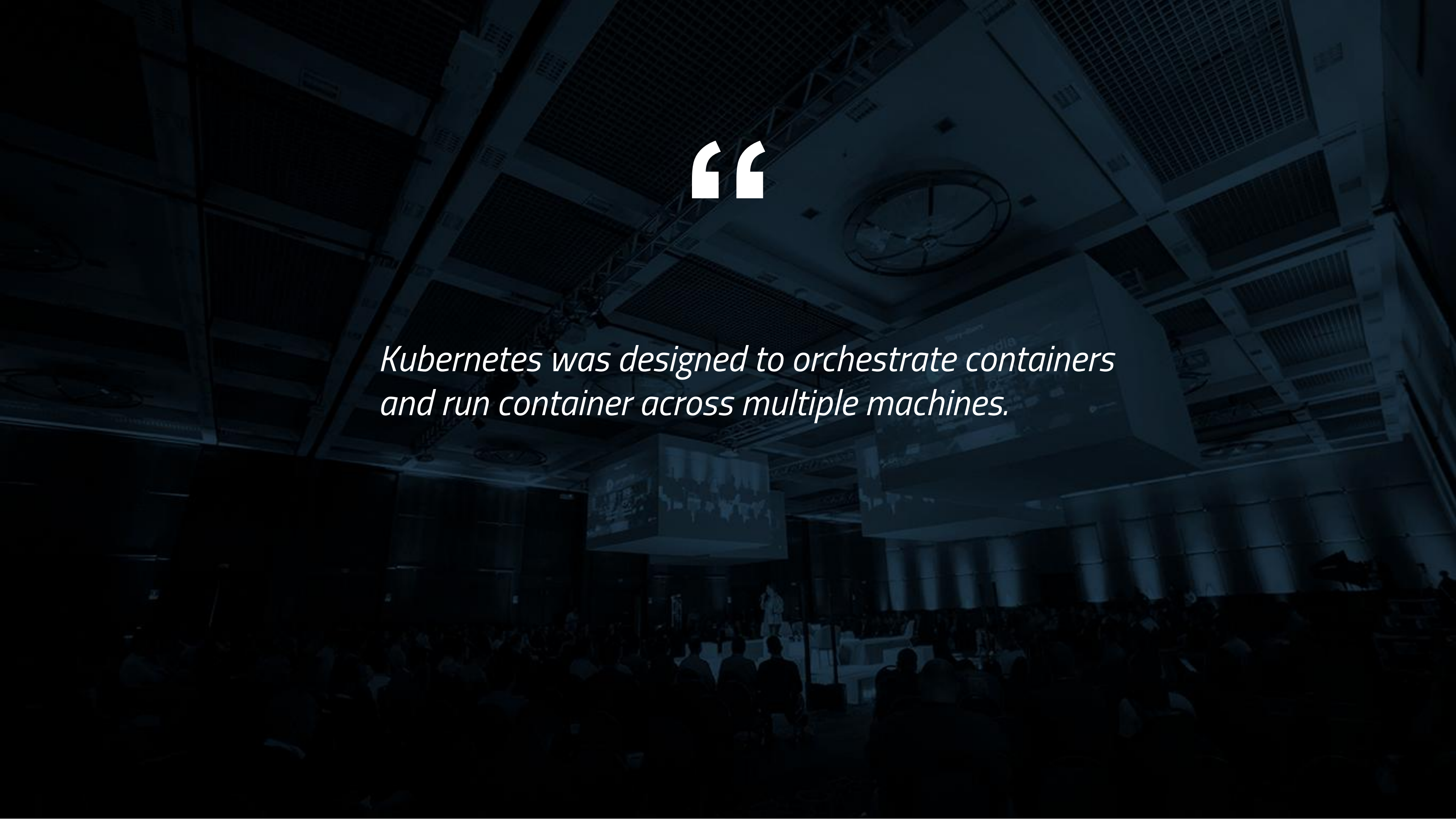
KEEP
CALM
AND
Let's solve
IT together

Introducing Kubernetes

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.

Kubernetes Architecture Diagram



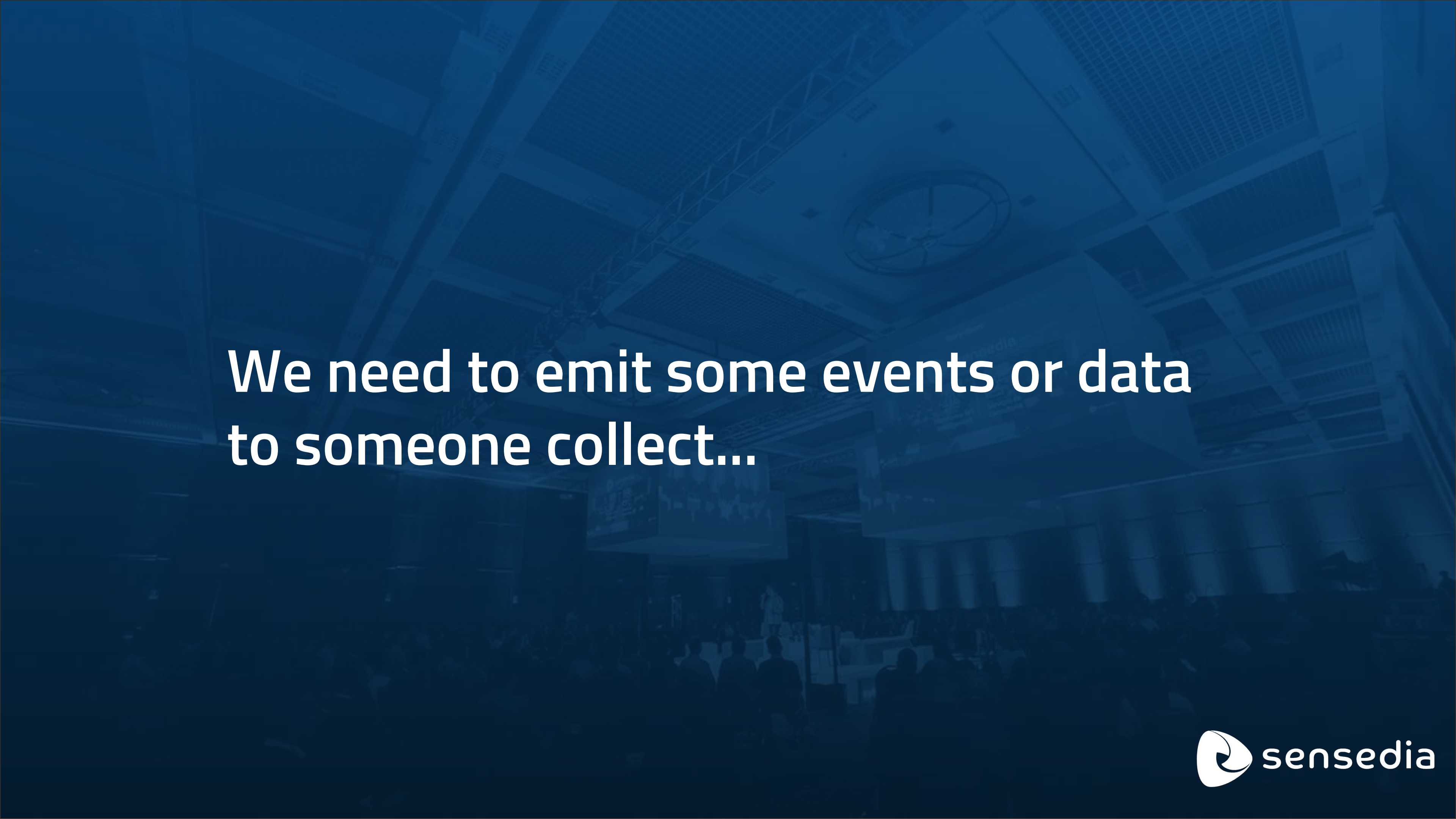


Kubernetes was designed to orchestrate containers and run container across multiple machines.

Now we are ready to talk about **Observability...**

The OpenTracing standard

- Cloud Native Computing Foundation (CNCF)
- it standardizes the instrumentation of applications for distributed tracing
- its says a trace tells the story of a transaction
- while a span represents a single call
- distributed tracing systems collect and organize all this information in a nice interface and Zipkin is one of them



We need to emit some events or data to someone collect...

sensedia

SPRING CLOUD SLEUTH

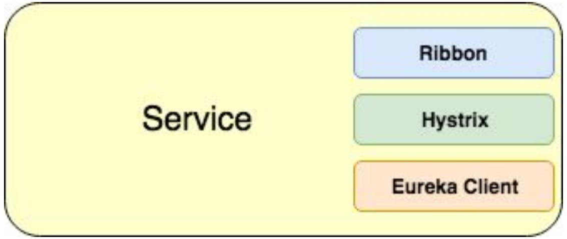
Services interactions
instrumented automatically

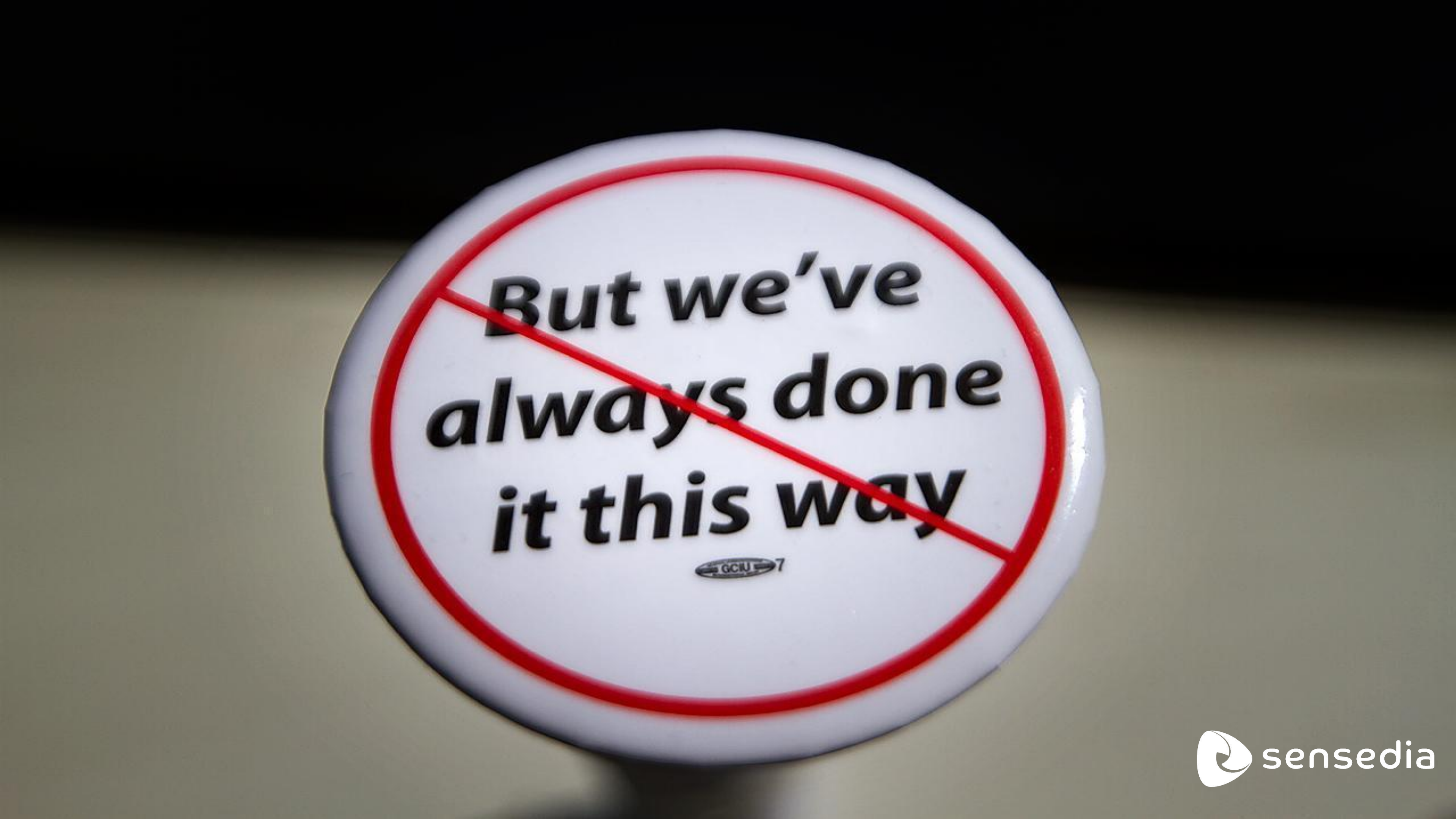
Features

- Adds trace and span ids to the SLF4J Mapped Diagnostic Context (MDC)
- Automatically instruments common ingress and egress points from Spring applications (servlet filter, rest template, feign client)
- just add **spring-cloud-sleuth-zipkin** dependency and the service will generate and send traces to Zipkin via HTTP

Ways to emit data

Popularized by Netflix OSS and Spring Cloud
Spring Framework enable it in Spring Boot
All these features works only in JVM languages





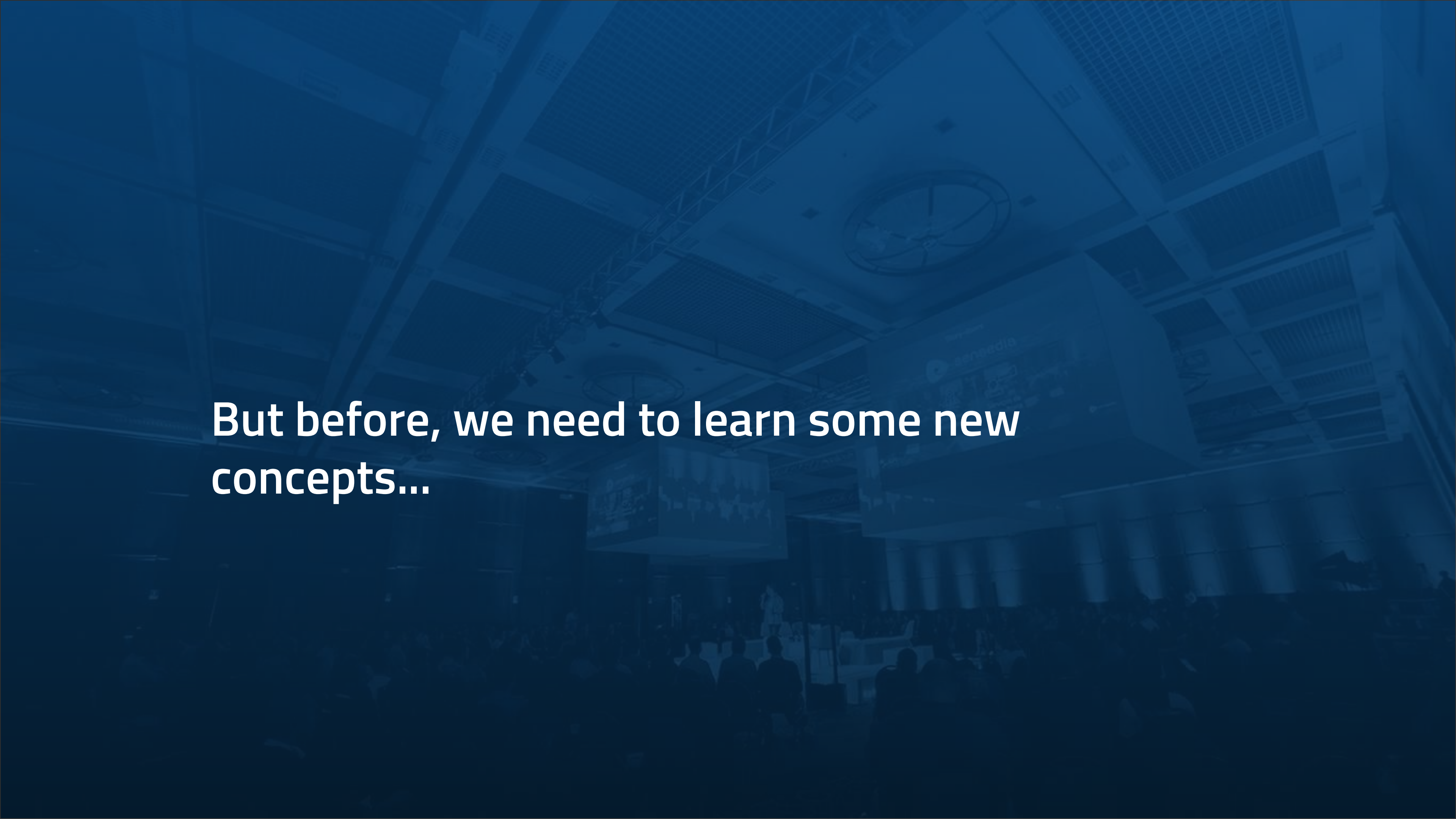
sensedia

Let’s think a little bit differently…

What about the platform infrastructure for service communications ???

sensedia

ISTIO



But before, we need to learn some new concepts...

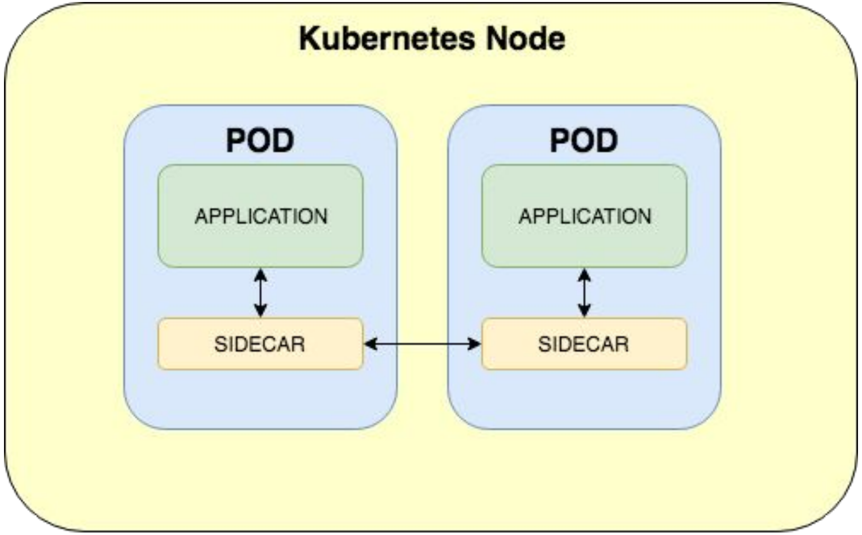
Sidecar Pattern

Sidecar Pattern

A service which can enrich the main service with utility tasks. Normally these tasks are cross-cutting and infrastructure related.

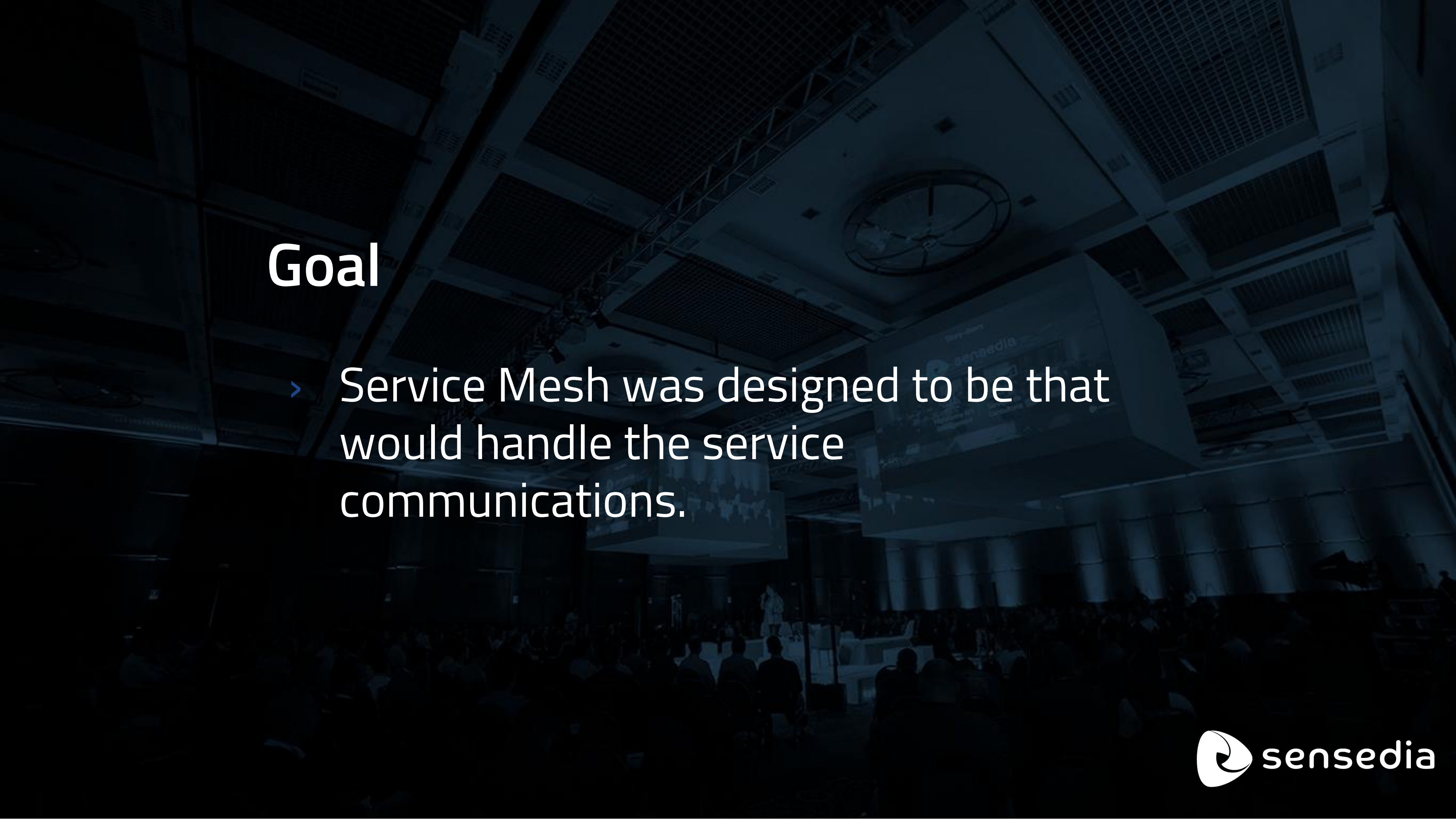
sensedia

Sidecar Pattern



Service Mesh

A service mesh is a configurable infrastructure layer for a microservices application.



Goal

- Service Mesh was designed to be that would handle the service communications.

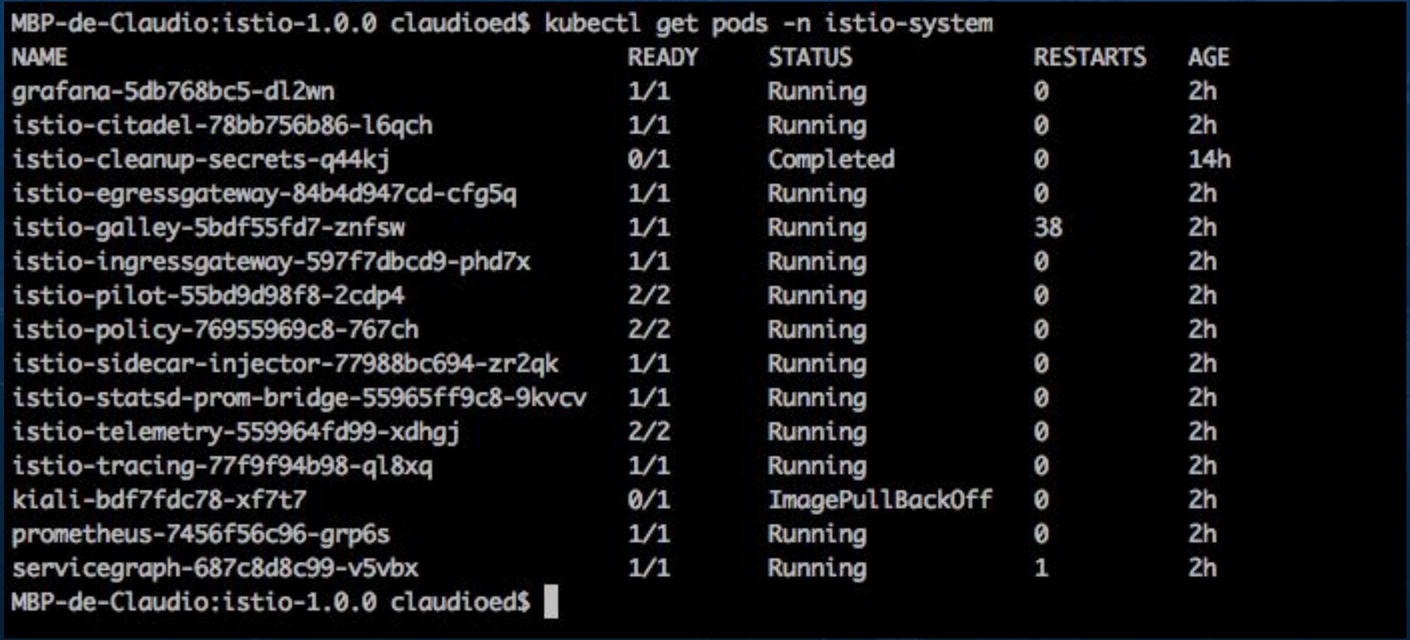
This approach can give us some advantages

- We can take off some responsibilities from our microservices
- Ribbon, Hystrix, Metrics and Tracer (OSS)
- The platform can provide it for us.
- **Polyglot**

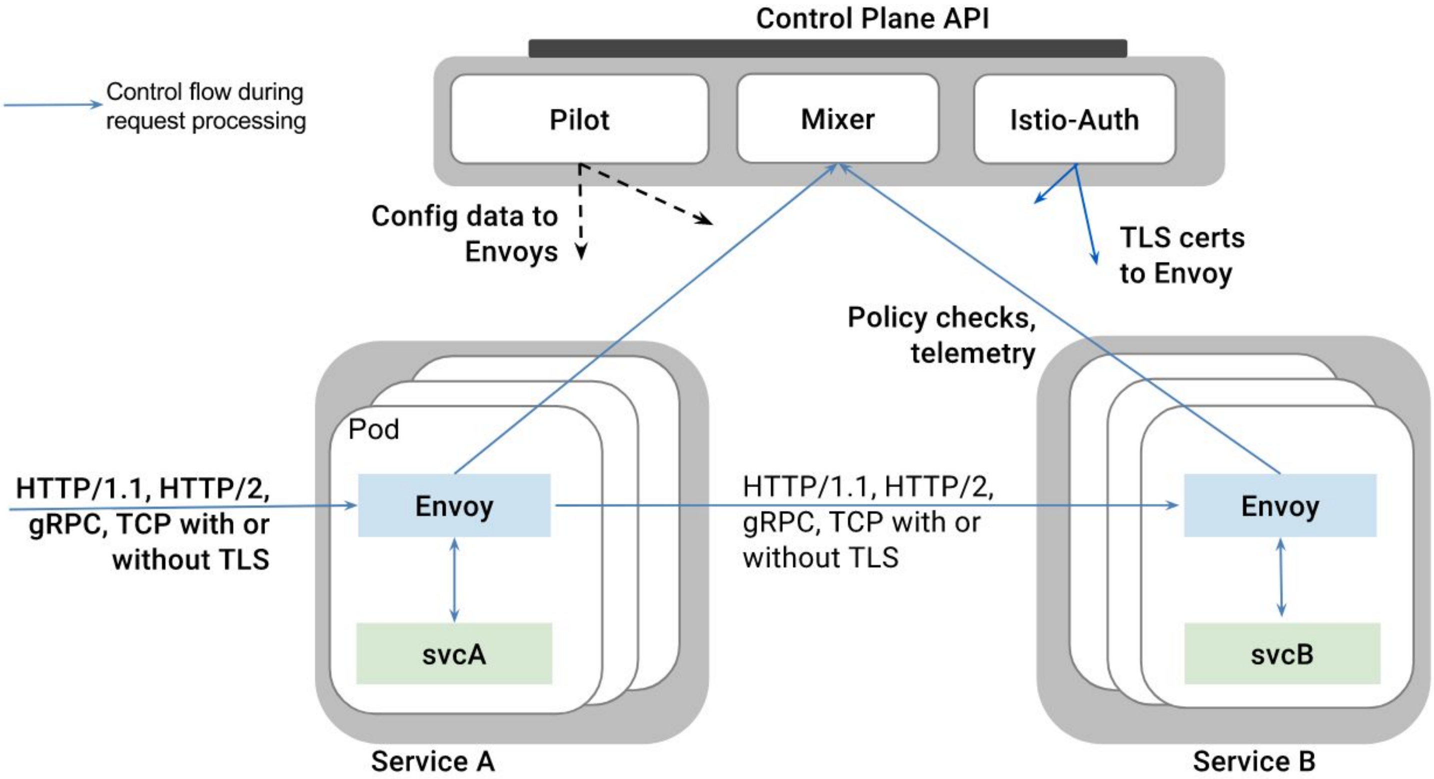
Service Mesh Features

- Sidecar Proxy
- Service Discovery & Load Balancing
- Circuit Breaking
- Fault Injection
- Can be used in legacy systems

Istio Deployment



```
MBP-de-Claudio:istio-1.0.0 claudioed$ kubectl get pods -n istio-system
NAME                                        READY   STATUS             RESTARTS   AGE
grafana-5db768bc5-dl2wn                     1/1     Running            0          2h
istio-citadel-78bb756b86-l6qch              1/1     Running            0          2h
istio-cleanup-secrets-q44kj                 0/1     Completed          0          14h
istio-egressgateway-84b4d947cd-cfg5q        1/1     Running            0          2h
istio-galley-5bdf55fd7-znfsw                1/1     Running            38         2h
istio-ingressgateway-597f7dbcd9-phd7x       1/1     Running            0          2h
istio-pilot-55bd9d98f8-2cdp4                2/2     Running            0          2h
istio-policy-76955969c8-767ch               2/2     Running            0          2h
istio-sidecar-injector-77988bc694-zr2qk     1/1     Running            0          2h
istio-statsd-prom-bridge-55965ff9c8-9kvcv   1/1     Running            0          2h
istio-telemetry-559964fd99-xdhgj            2/2     Running            0          2h
istio-tracing-77f9f94b98-ql8xq              1/1     Running            0          2h
kiali-bdf7fdc78-xf7t7                       0/1     ImagePullBackOff   0          2h
prometheus-7456f56c96-grp6s                 1/1     Running            0          2h
servicegraph-687c8d8c99-v5vbx               1/1     Running            1          2h
MBP-de-Claudio:istio-1.0.0 claudioed$
```



Istio Architecture

PILOT

Component that programs the Envoy proxies, responsible for service discovery, load balancing, and routing.

MIXER

Component responsible for enforcing access control and usage policies across the service mesh and collecting telemetry data from Envoy and other services

ISTIO ADD-ONs

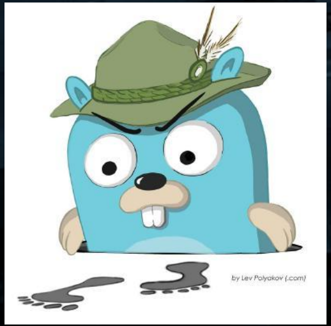
Open-Source components which will work together to provide some additional functionalities

Open Source

Prometheus

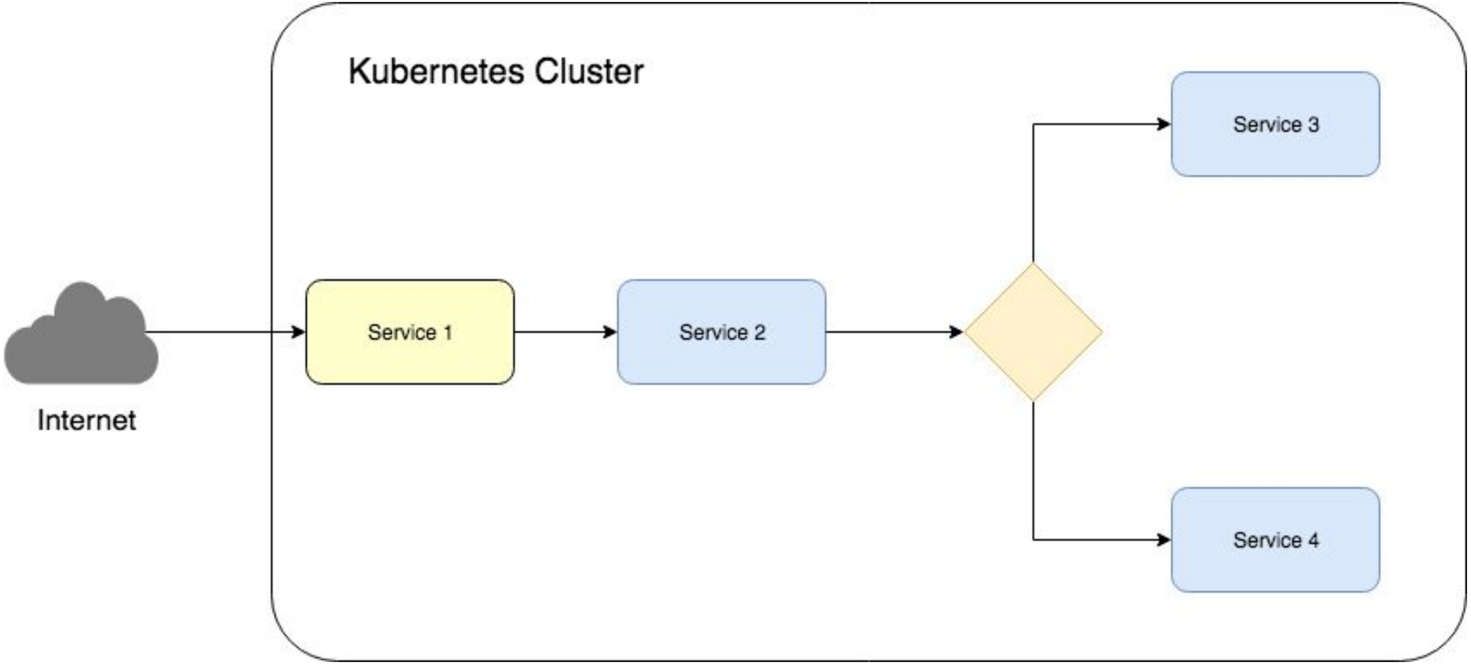
Grafana

Jaeger



Demo Time

Use Case - Architecture Diagram



Obrigado!

Cláudio E. de Oliveira

claudio.oliveira@sensedia.com
+55 19 3705-5775