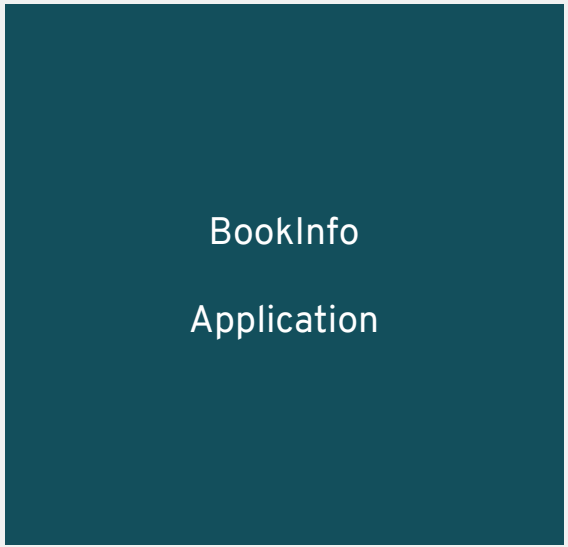
ISTIO SERVICE MESH

Joaquim Moreno
Senior Software Engineer
Red Hat

INTRODUCING OUR MONOLITHIC APPLICATION



BookInfo Application.

A simple Application that displays information about books from a catalog.

This information includes:

- Description
- ISBN
- Number of pages
- Reviews

BookInfo Sample

Sign in

The Comedy of Errors

Summary: Wikipedia Summary: The Comedy of Errors is one of **William Shakespeare's** early plays. It is his shortest and one of his most farcical comedies, with a major part of the humour coming from slapstick and mistaken identity, in addition to puns and word play.

Book Details

Type:
paperback
Pages:
200
Publisher:
PublisherA
Language:
English
ISBN-10:
1234567890
ISBN-13:
123-1234567890

Book Reviews

An extremely entertaining play by Shakespeare. The slapstick humour is refreshing!

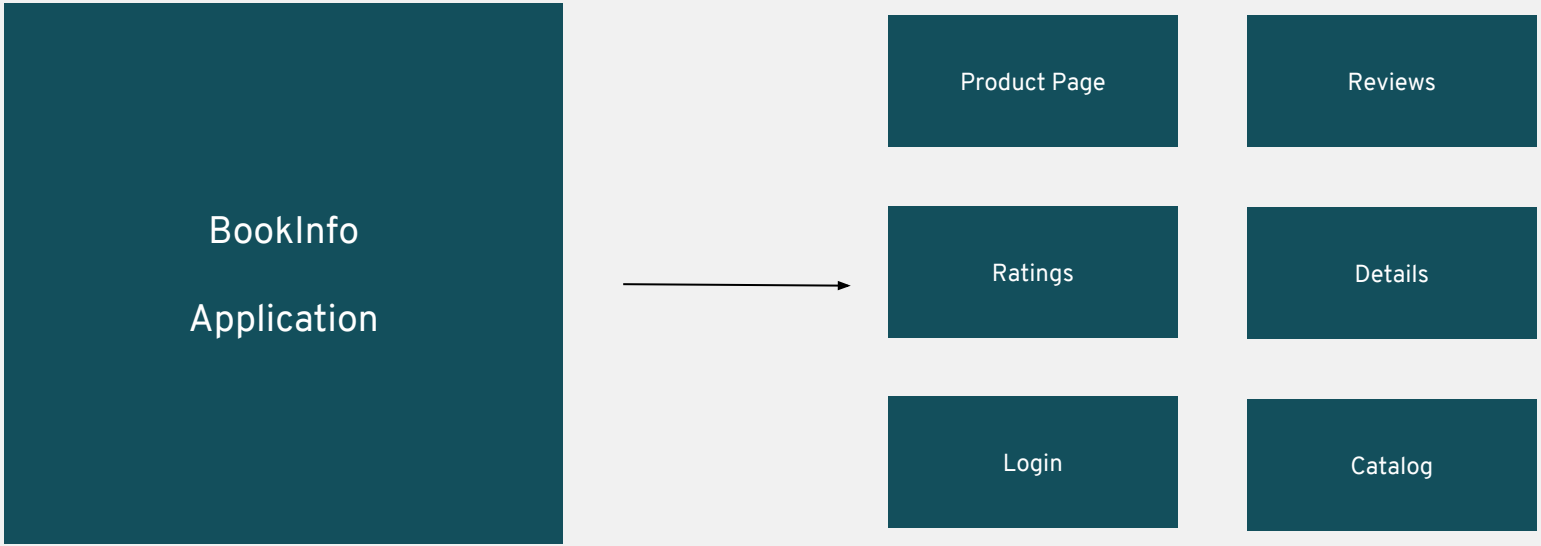
— Reviewer1

Absolutely fun and entertaining. The play lacks thematic depth when compared to other plays by Shakespeare.

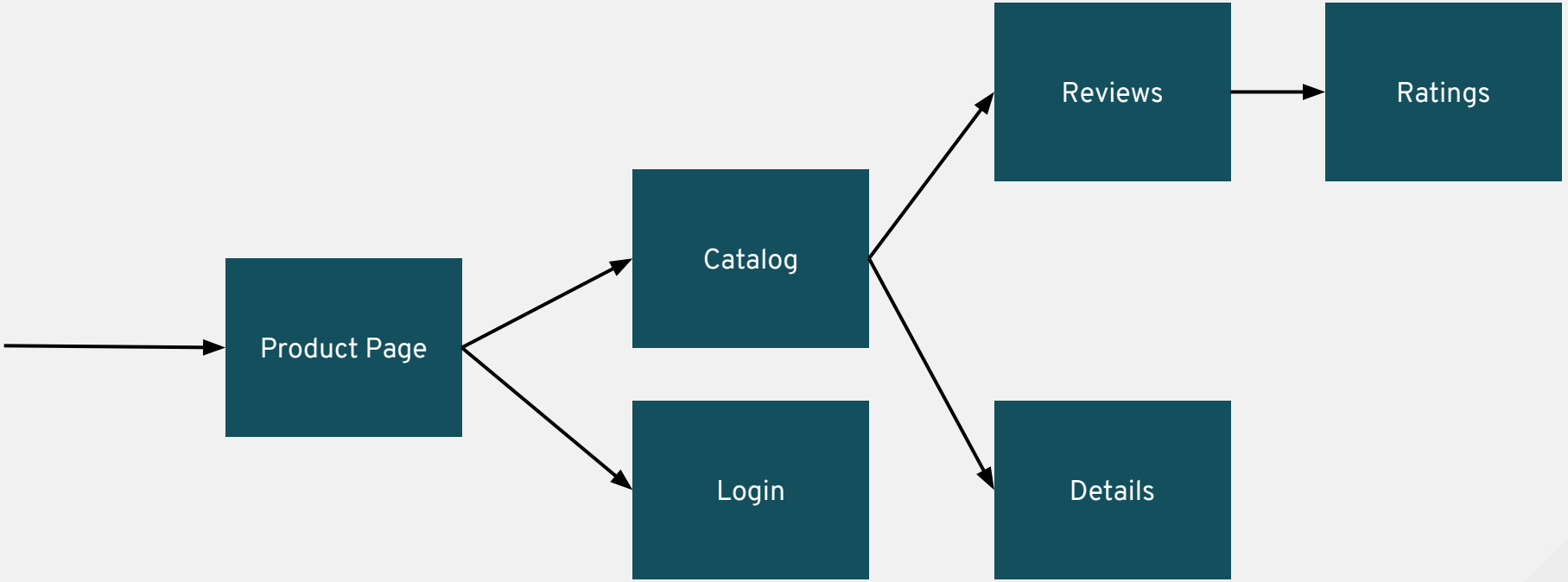
— Reviewer2

BUT COMPANIES EVOLVE AND REQUIREMENTS CHANGE...

BREAKING OUR MONOLITH INTO SMALLER SERVICES



OUR NEW MICROSERVICES APPLICATION



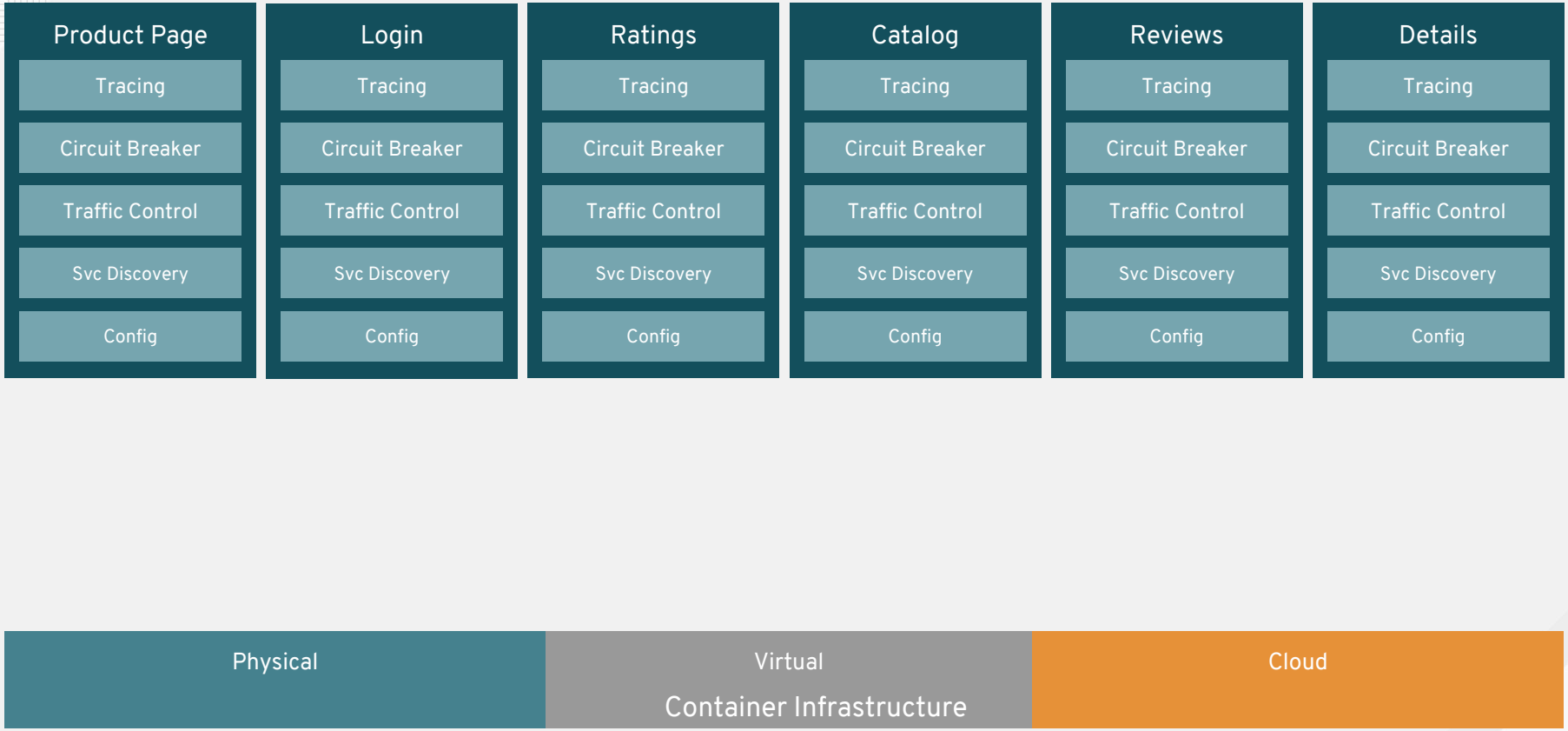
WITH MICROSERVICES NEW CHALLENGES APPEAR:

- How can my services know if another **service is up or down**?
- How can my services **find each other**?
- How can we **handle failures in a deployment**?
- How can we **auto scale our application**?
- What happens if any of those **internal requests between services fail?**

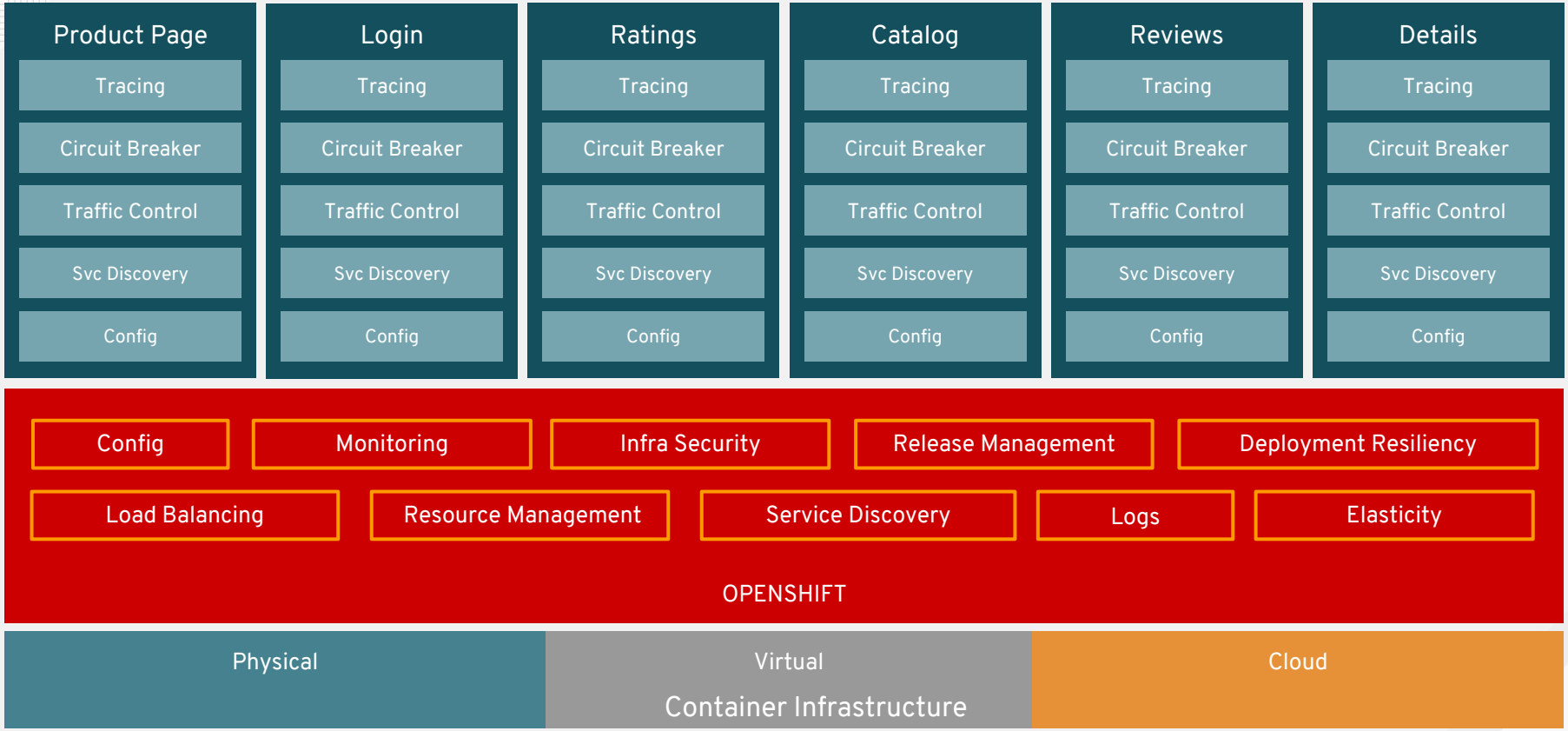
WITH MICROSERVICES NEW CHALLENGES APPEAR:

- Where are my **requests failing**?
- Why this **API endpoint is so slow**? **Which service is at fault?**
- That service is really prone to errors… Can we **retry those calls** if they fail?
- Someone is hammering this service every day at the same time, we should add some **rate limit** to avoid that!
- **Which services can talk with each other?**

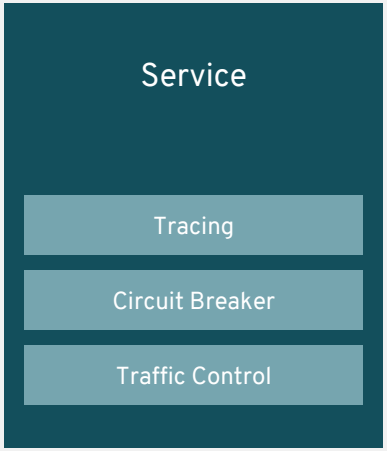
OUR NEW MICROSERVICES APPLICATION



OUR NEW MICROSERVICES APPLICATION



We are still missing some pieces...

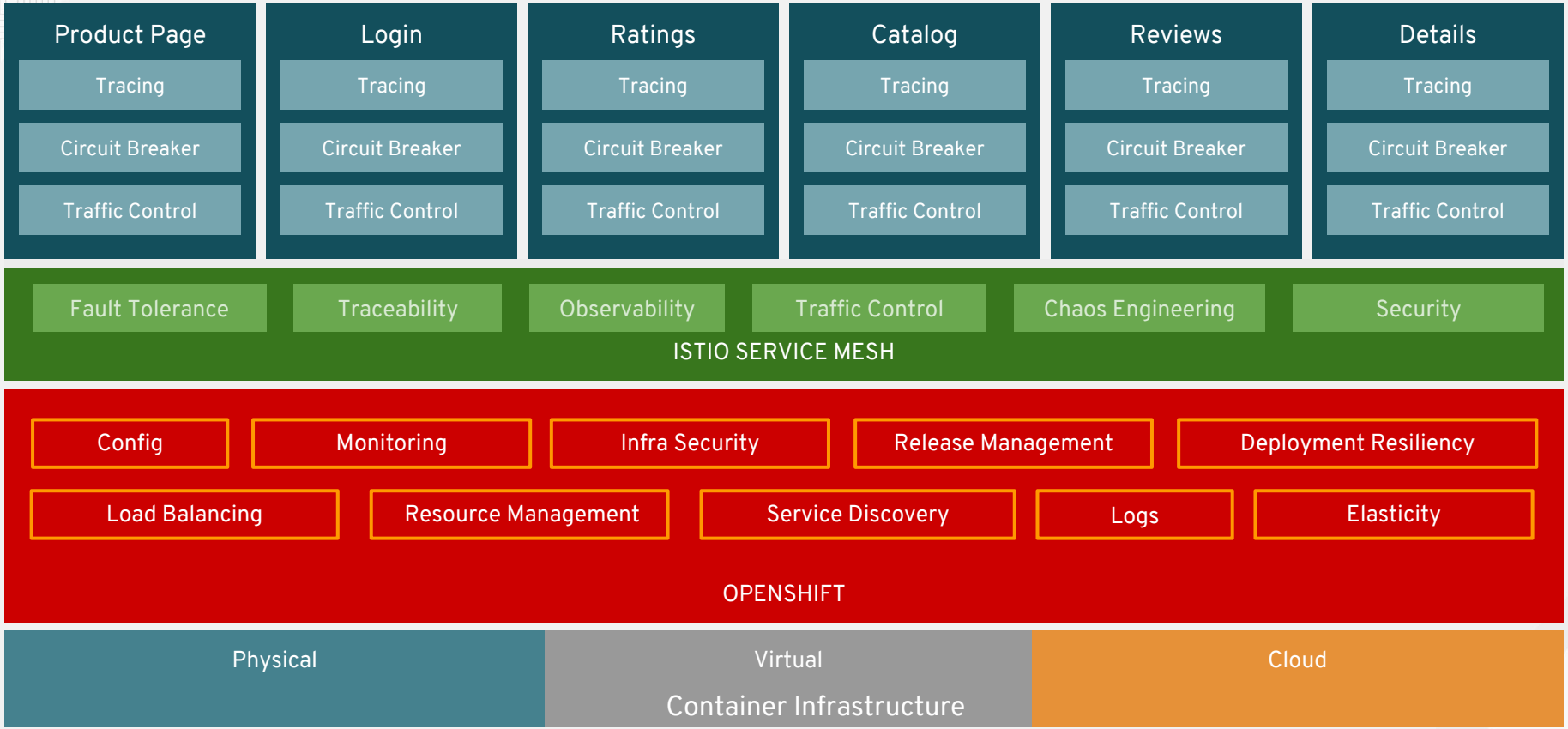


Those features will need to be implemented as part of the application

- ✖ Incompatible across languages and frameworks
- ✖ Existing apps require refactoring
- ✖ Upgrades needs tight coordinations libraries

THERE SHOULD BE A BETTER WAY

OUR NEW MICROSERVICES APPLICATION



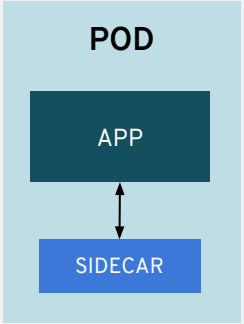
HOW ISTIO WORKS

WHAT DOES ISTIO GIVE ME?

- Fault Tolerance
- Traceability
- Observability
- Traffic control
- Security
- Chaos Engineering

SIDECAR PATTERN

- A utility container in the same pod to enhance the main container's functionality
- Share the same network and lifecycle
- Istio uses an Istio Proxy (L7 Proxy) sidecar to proxy all network traffic between apps

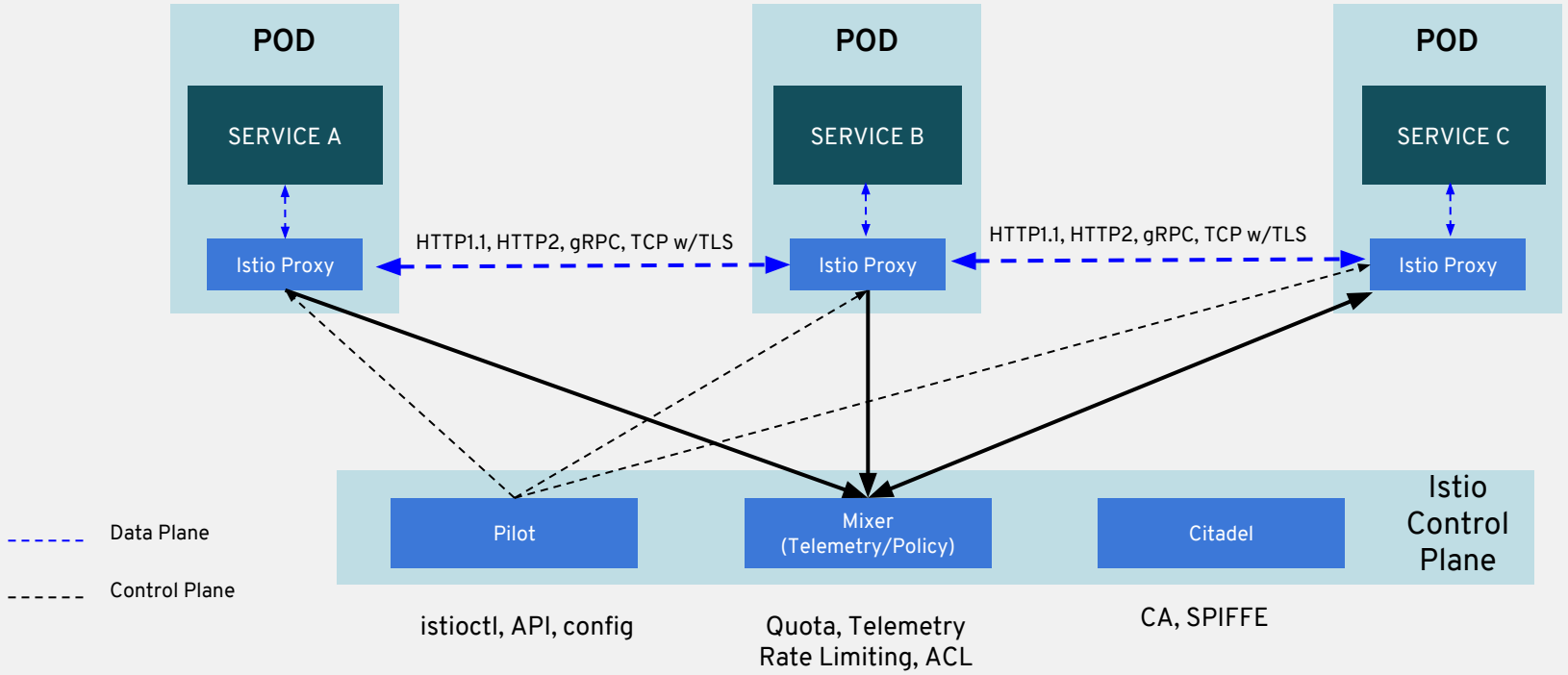


Source: http://blog.kubernetes.io/2015/06/the-distributed-system-toolkit-patterns.html

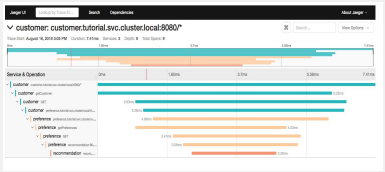
USE A SERVICE PROXY

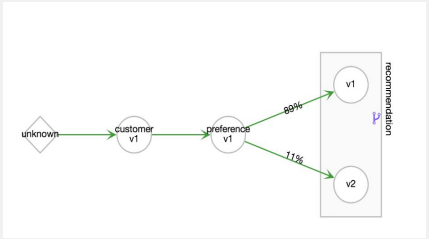
- Envoy Proxy (used in Istio as Istio proxy)
- L3/4 network filter, out of the box L7 filters (HTTP, HTTP2, gRPC)
- Service discovery, load balancing, circuit breaking, metrics collection, timeouts, retries, rate limiting, distributed tracking, et al
- Written in C++
- Dynamic configuration

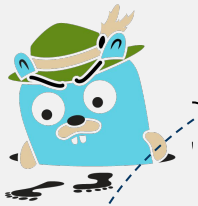
Istio Architecture



SERVICE MESH ECOSYSTEM







Jaeger

Secure

Observe

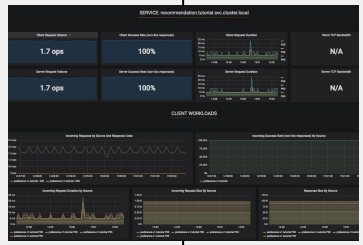
Connect

Control

Kiali

Prometheus

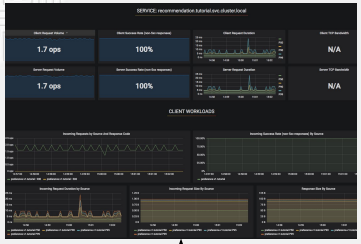
Observe



Grafana

OBSERVABILITY

ISTIO MONITORING



GRAFANA

PROMETHEUS

Prometheus Adapter

POD

SERVICE A

Istio Proxy

POD

SERVICE A

Istio Proxy

POD

SERVICE A

Istio Proxy

Check & Report

Logging | Telemetry | Auth | Quota

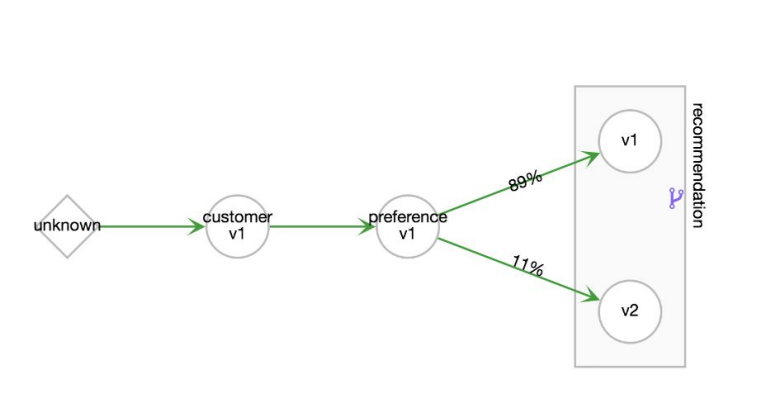
Adapter API

Mixer

Pilot

Citadel

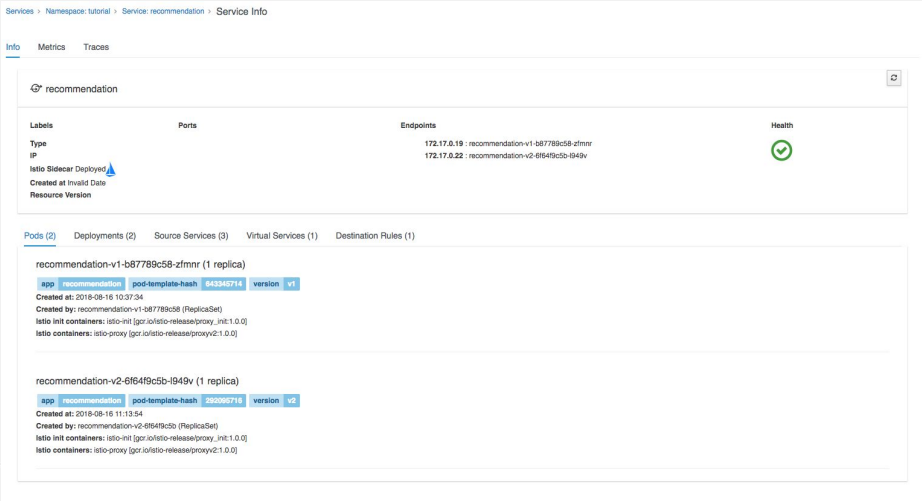
Control Plane

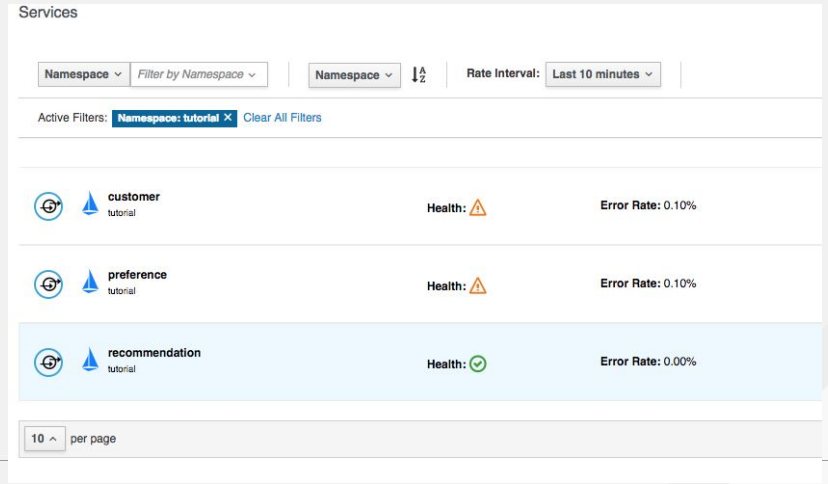


KIALI - SERVICE MESH OBSERVABILITY

FEATURES:

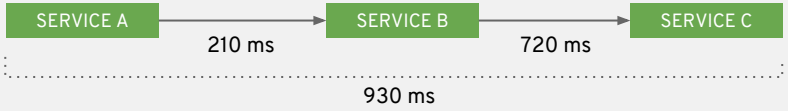
- Service graph representation
- Distributed tracing (via Jaeger)
- Metrics collection and graphs (from Prometheus)
- Configuration validation
- Health computation/display
- Service discovery

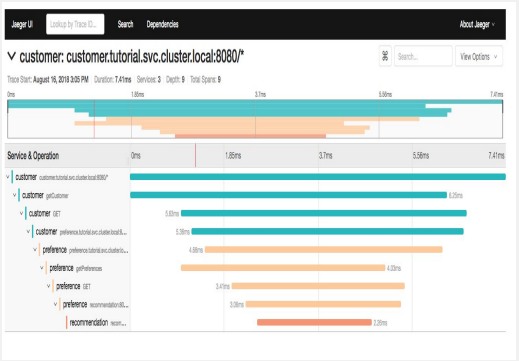


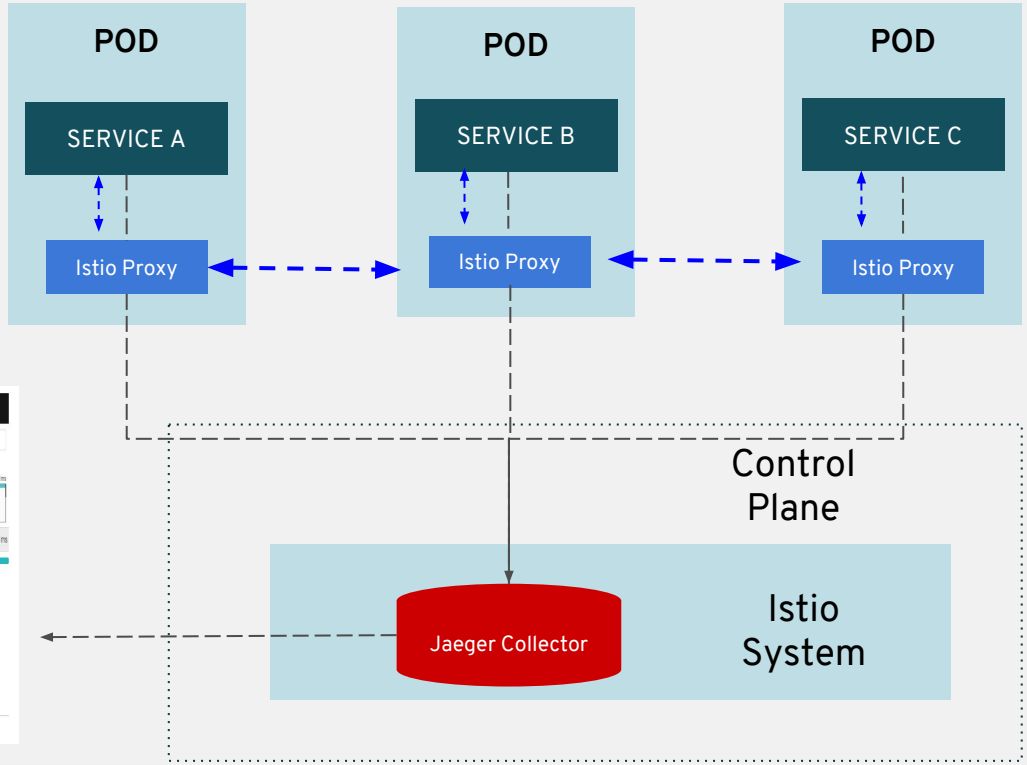


TRACING

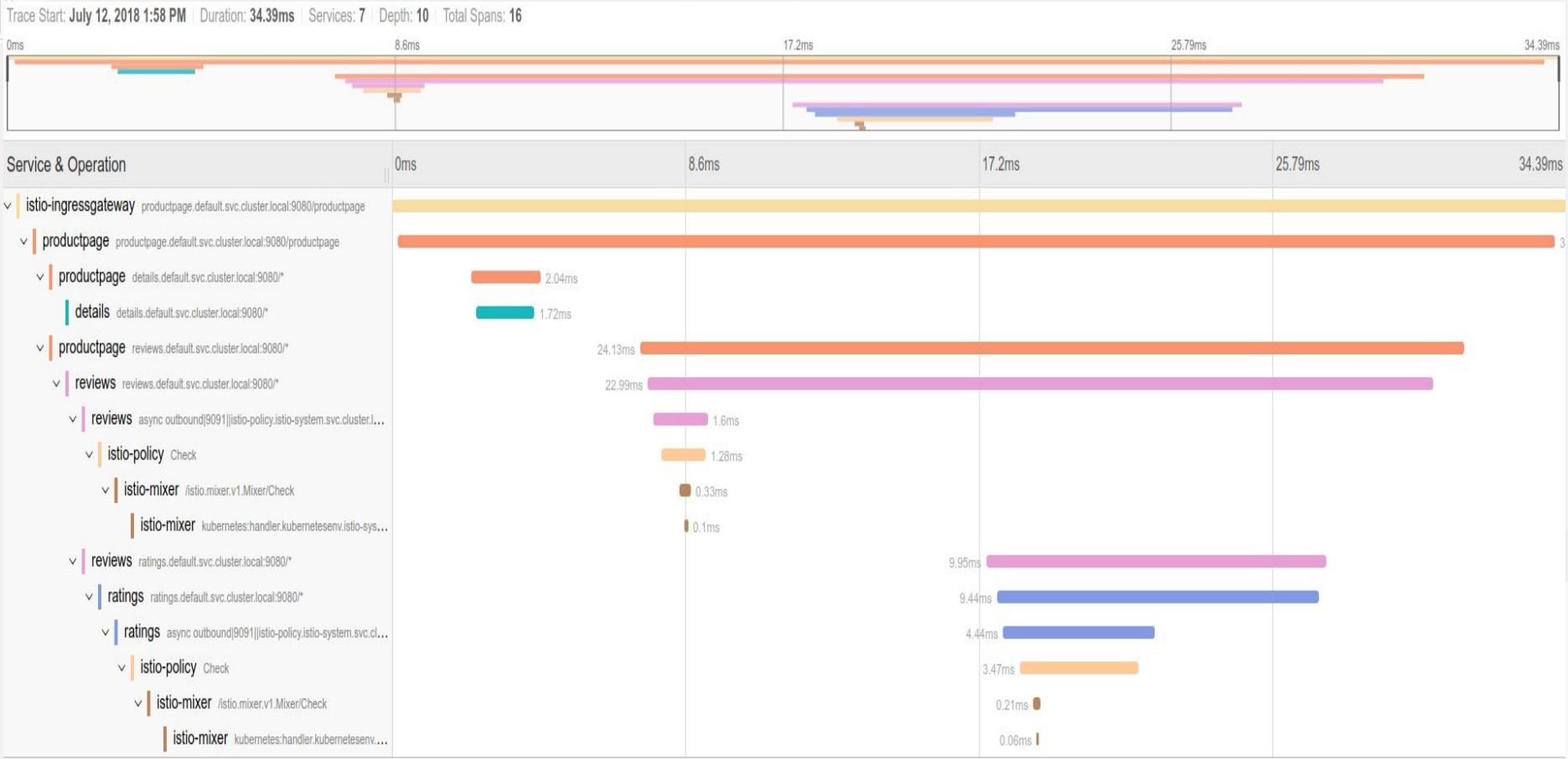
ISTIO TRACING



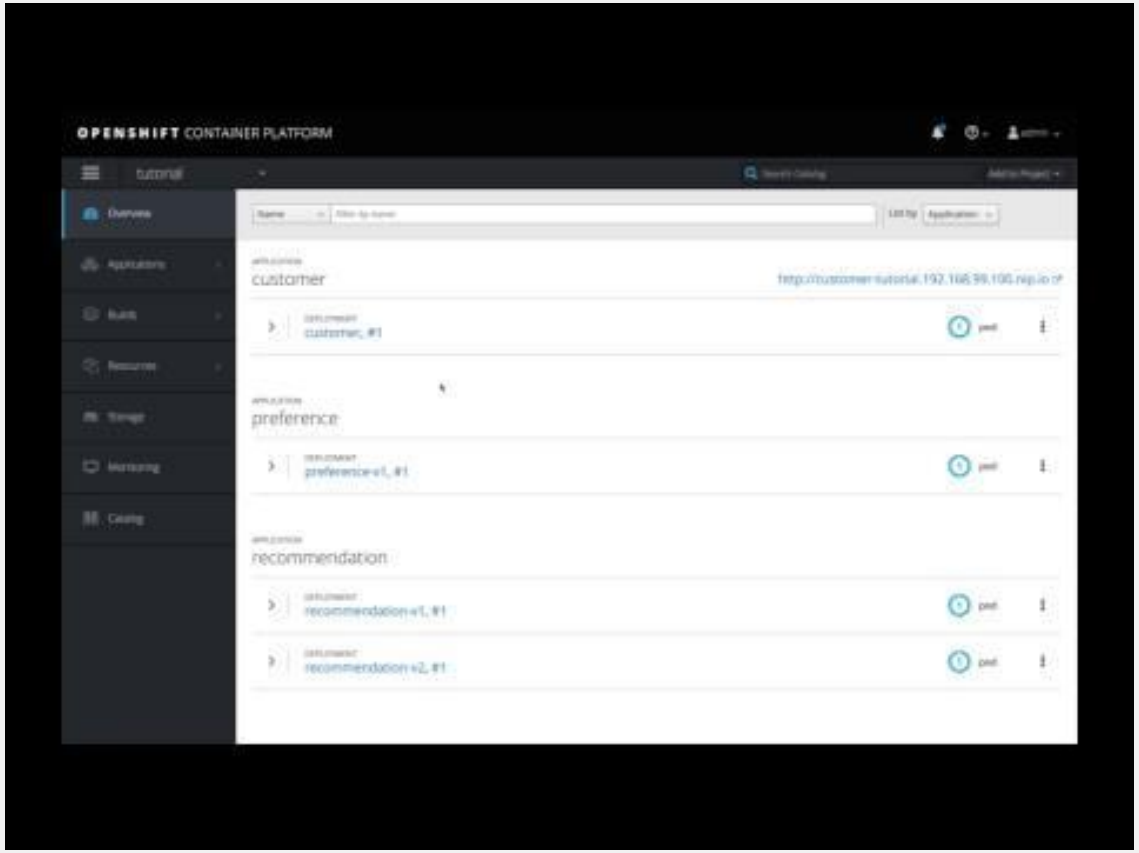




TRACING WITH ISTIO & JAEGER

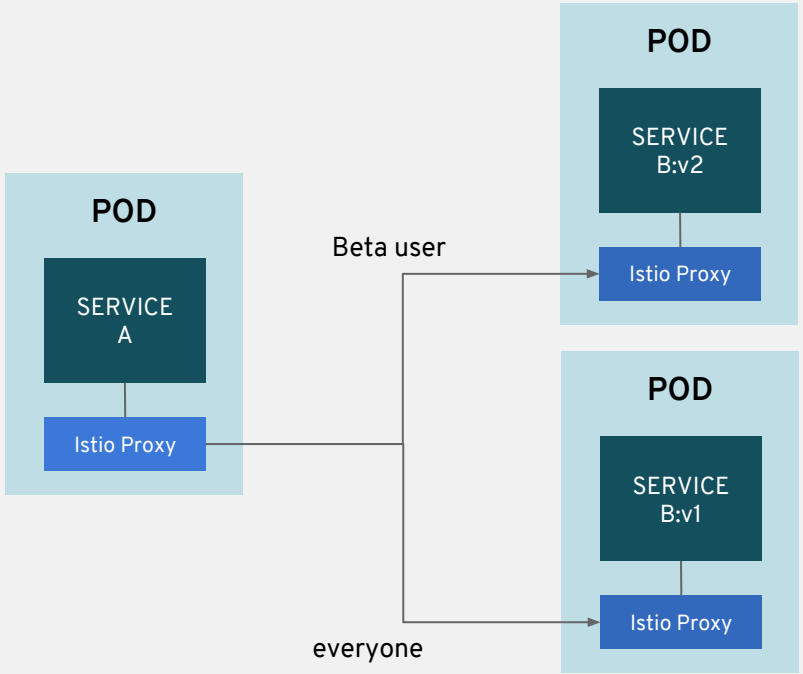


DEMO

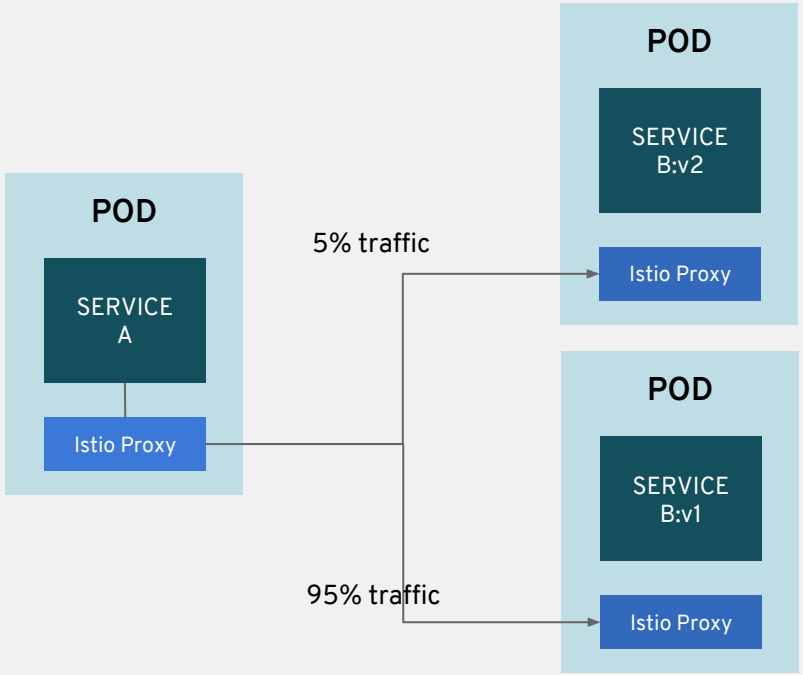

OPENSHIFT CONTAINER PLATFORM
tutorial
customer
preference
recommendation

TRAFFIC CONTROL

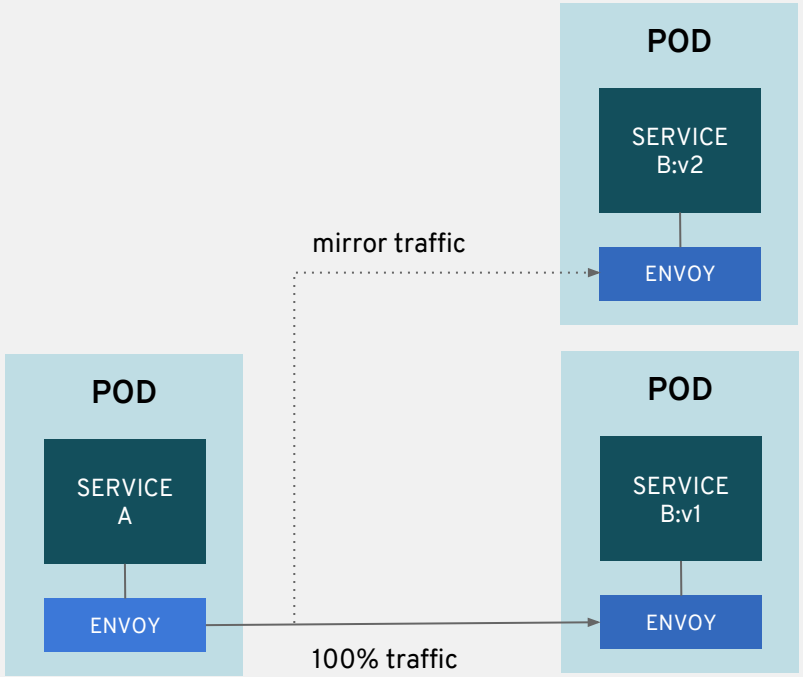
CANARY DEPLOYMENT WITH ISTIO



WEIGHTED ROUTING WITH ISTIO

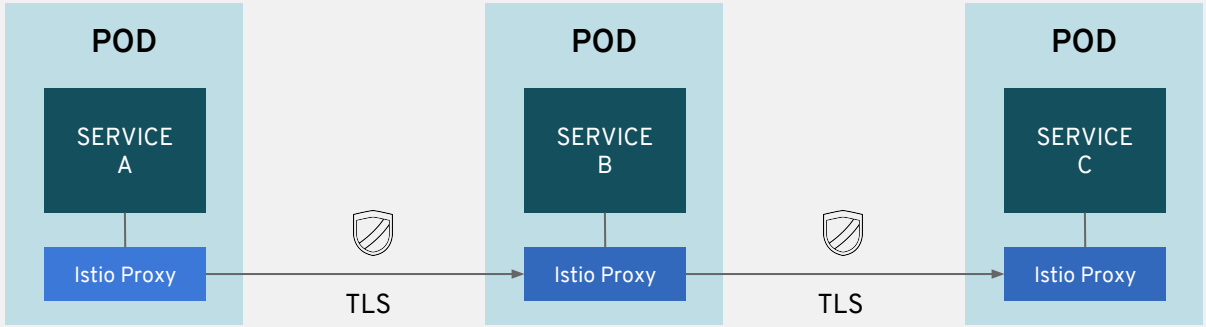


DARK LAUNCHES WITH ISTIO



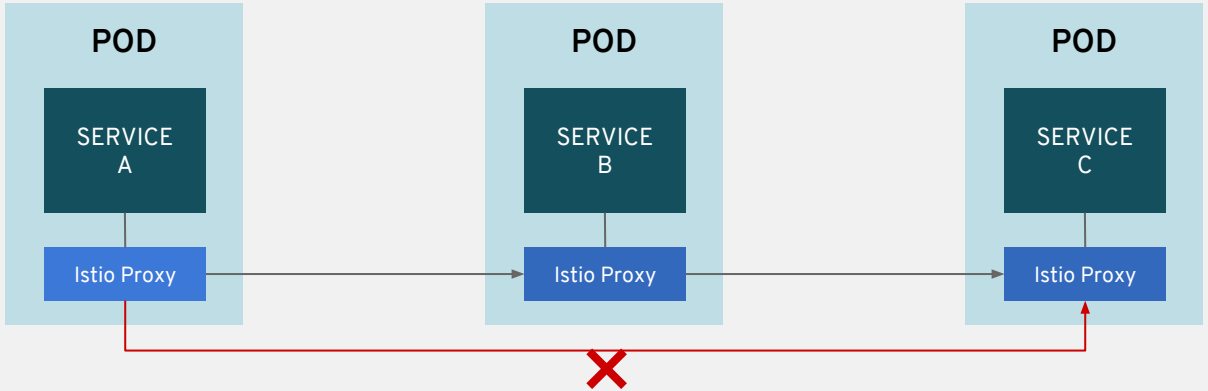
SERVICE SECURITY

SECURE COMMUNICATION WITH ISTIO



Automatic mutual TLS authentication, transparent to the services

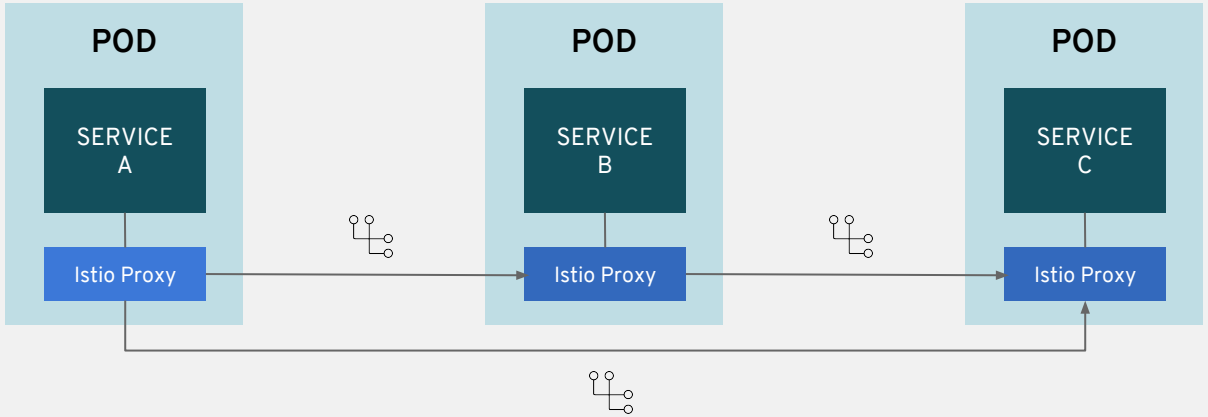
CONTROL SERVICE ACCESS WITH ISTIO



control the service access flow, transparent to the services

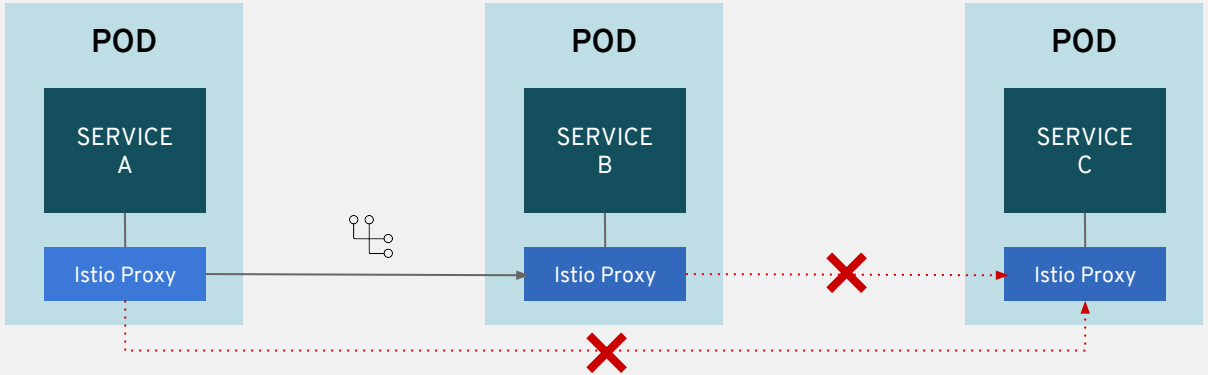
FAULT TOLERANCE

CIRCUIT BREAKERS WITH ISTIO



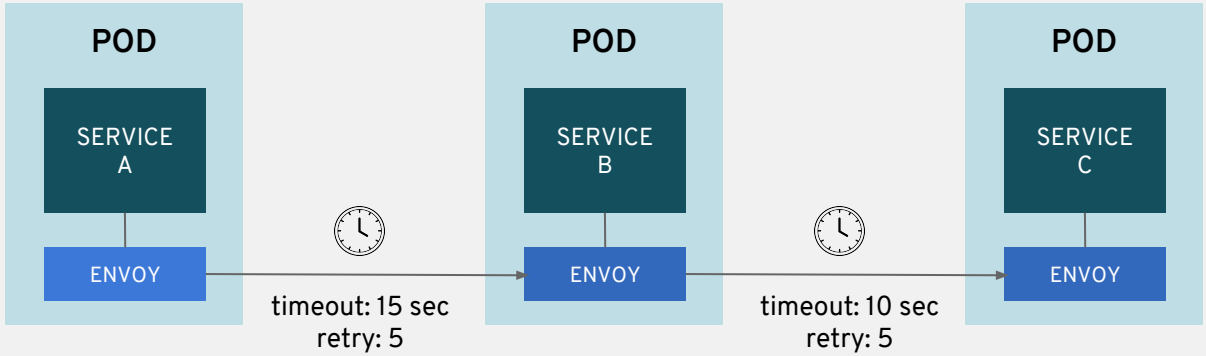
transparent to the services

CIRCUIT BREAKERS WITH ISTIO



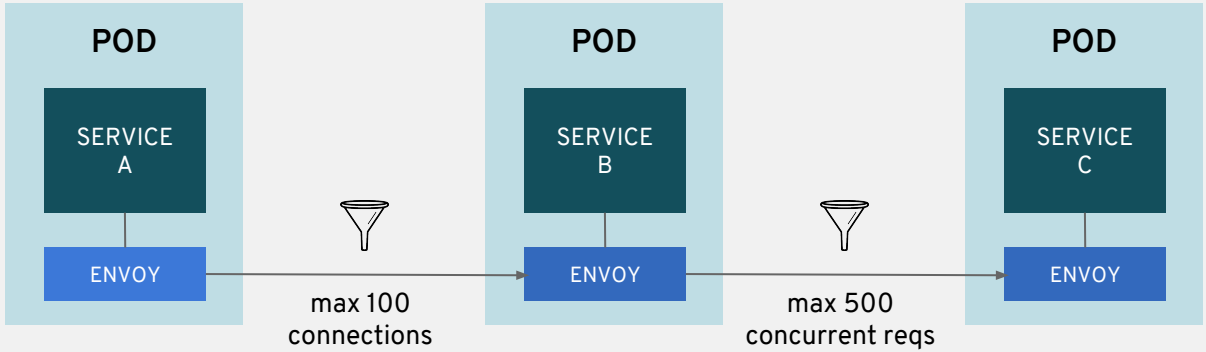
transparent to the services

TIMEOUTS AND RETRIES WITH ISTIO



configure timeouts and retries, transparent to the services

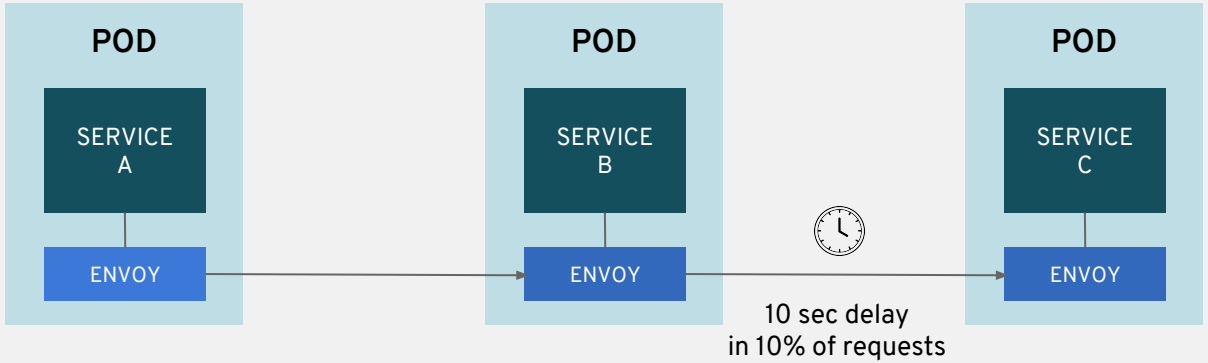
RATE LIMITING WITH ISTIO



limit invocation rates, transparent to the services

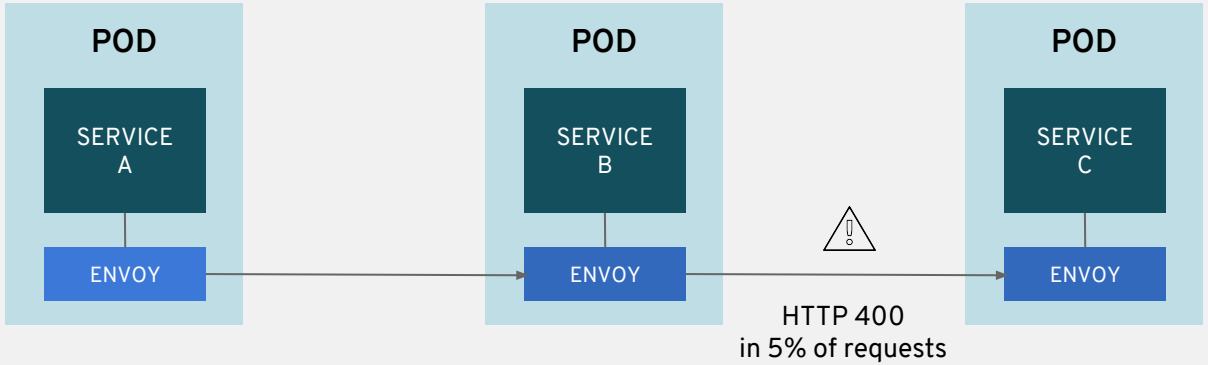
CHAOS ENGINEERING

CHAOS ENGINEERING WITH ISTIO



inject delays, transparent to the services

CHAOS ENGINEERING WITH ISTIO



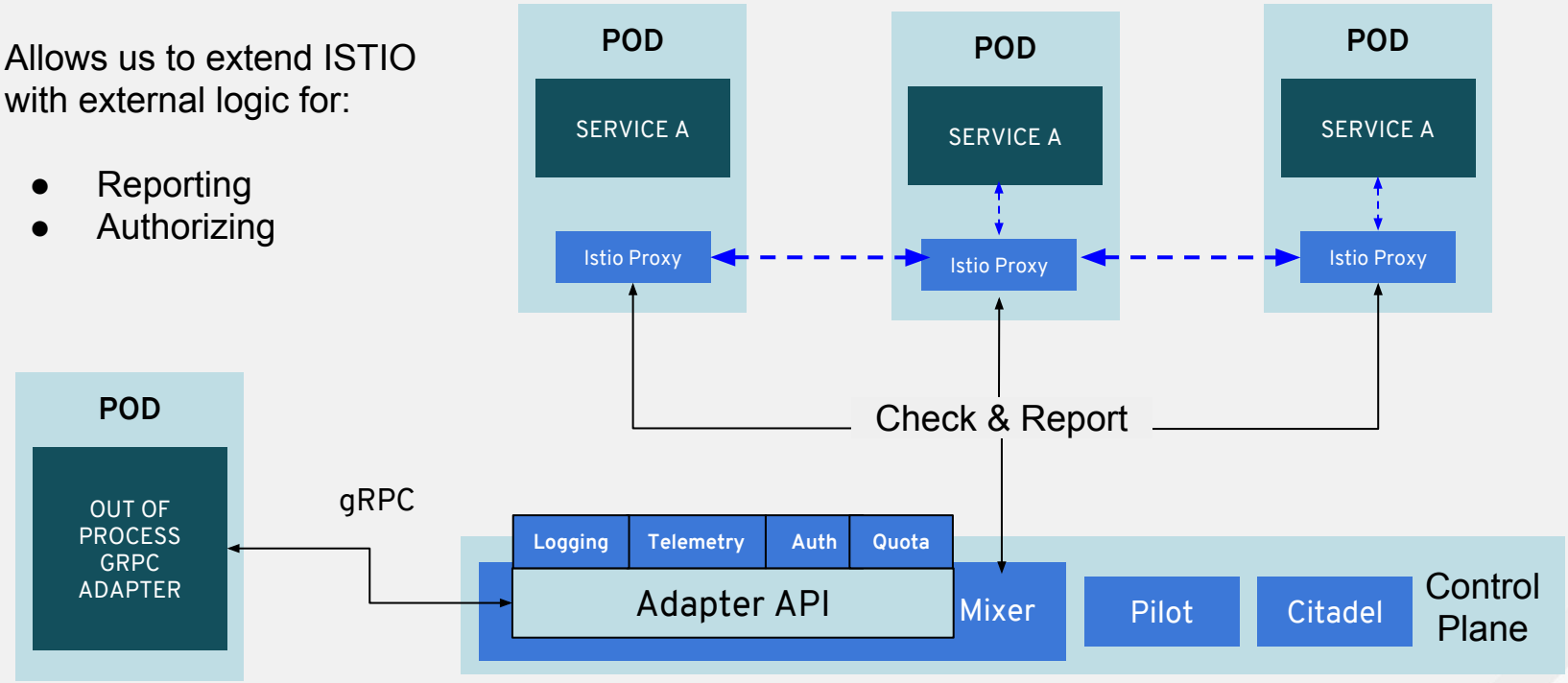
inject protocol-specific errors, transparent to the services

ADAPTERS

ADAPTER API

Allows us to extend ISTIO with external logic for:

- Reporting
- Authorizing



Q&A

Resources

- **Maistra, the Istio distribution of Red Hat:** https://maistra.io
- **Maistra Getting Started:** https://maistra.io/docs/
- Openshift + Service Mesh installation: https://docs.openshift.com/container-platform/3.11/servicemesh-install/servicemesh-install.html
- Red Hat Istio tutorial: https://github.com/redhat-developer-demos/istio-tutorial
- The Istio Project site: https://istio.io


THANKS
redhat.