

WHAT IS A SERVICES MESH?

*"The term **service mesh** is used to describe the **network** of **microservices** that make up such applications **and the interactions between them**."*

@MARCELV <https://istio.io/docs/concepts/what-is-istio/#what-is-a-service-mesh>

WHAT PROBLEM IS SOLVED?

MICRO SERVICES COMPLEXITY

INGRESS

VERSIONS

OBSERVABILITY

AUDIT TRAIL

SECURITY

RELEASE STRATEGIES

LETS TAKE A STEP BACK

MICRO SERVICES

AUTONOMOUS BUSINESS CAPABILITY

INDIVIDUALLY SCALABLE

AVOID SERVICE DEPENDENCIES!

FAULT TOLERANT

CONTAINERS & MICRO SERVICES

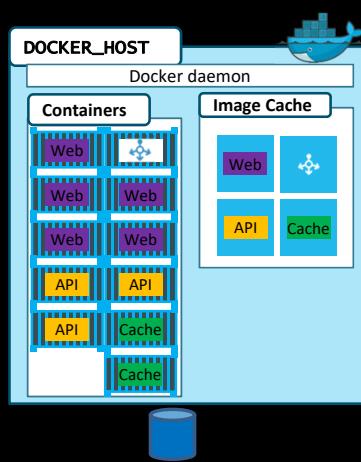
MARRIAGE MADE IN
HEAVEN

WHAT WAS SOLVED BY
KUBERNETES?

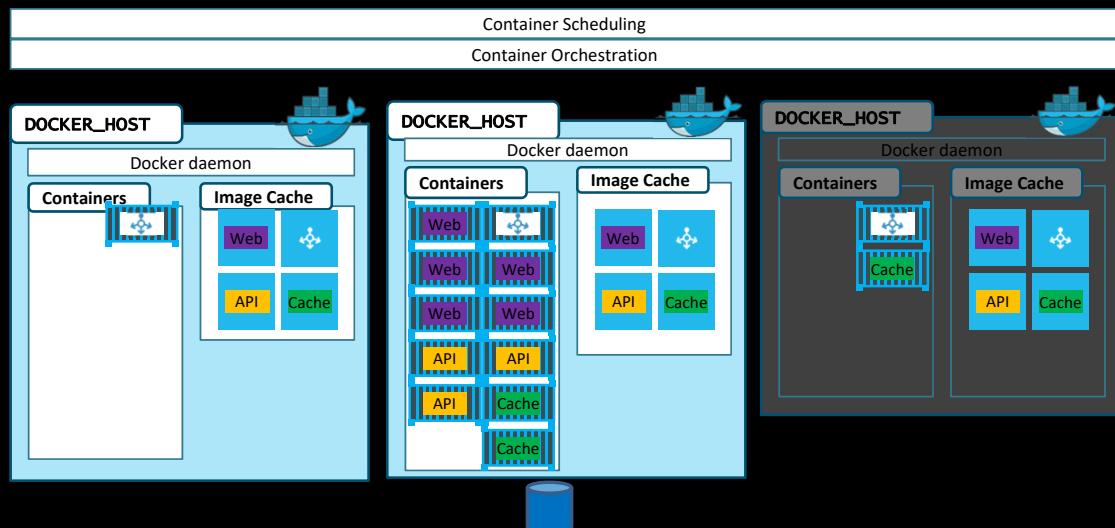


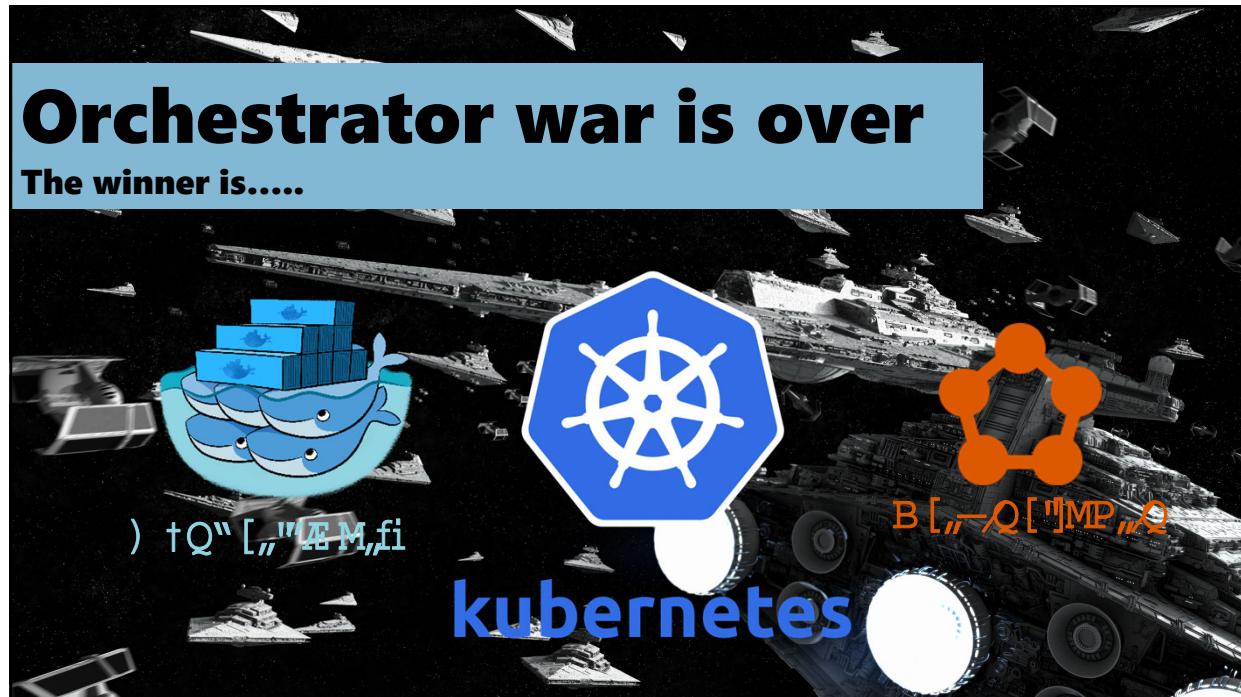
kubernetes

Container Orchestration & Scheduling



Container Orchestration & Scheduling





SOLVED INITIAL NEEDS

- INDIVIDUAL SCALING
- FAULT TOLERANCE
- 0-DOWNTIME DEPLOYMENTS
- INGRESS

NEEDS BEYOND K8S

TRAFIC POLICIES

TRAFIC TELEMETRY

TRAFIC MANAGEMENT

TRAFIC RESILIENCE

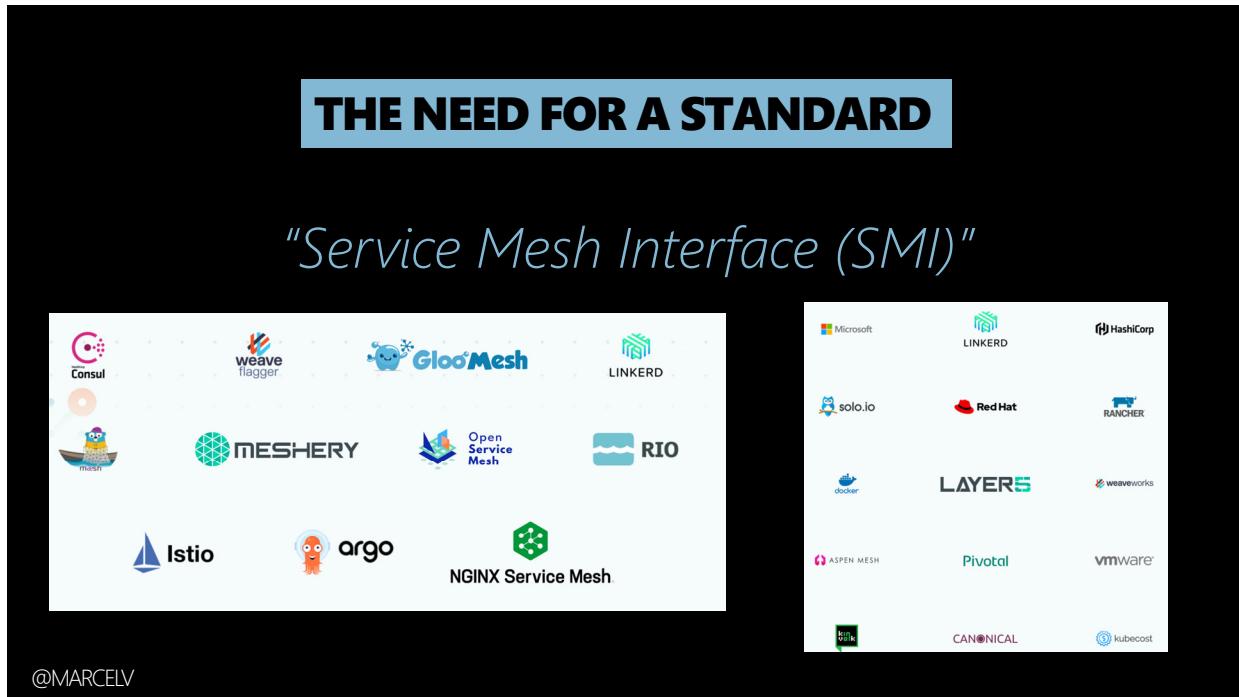
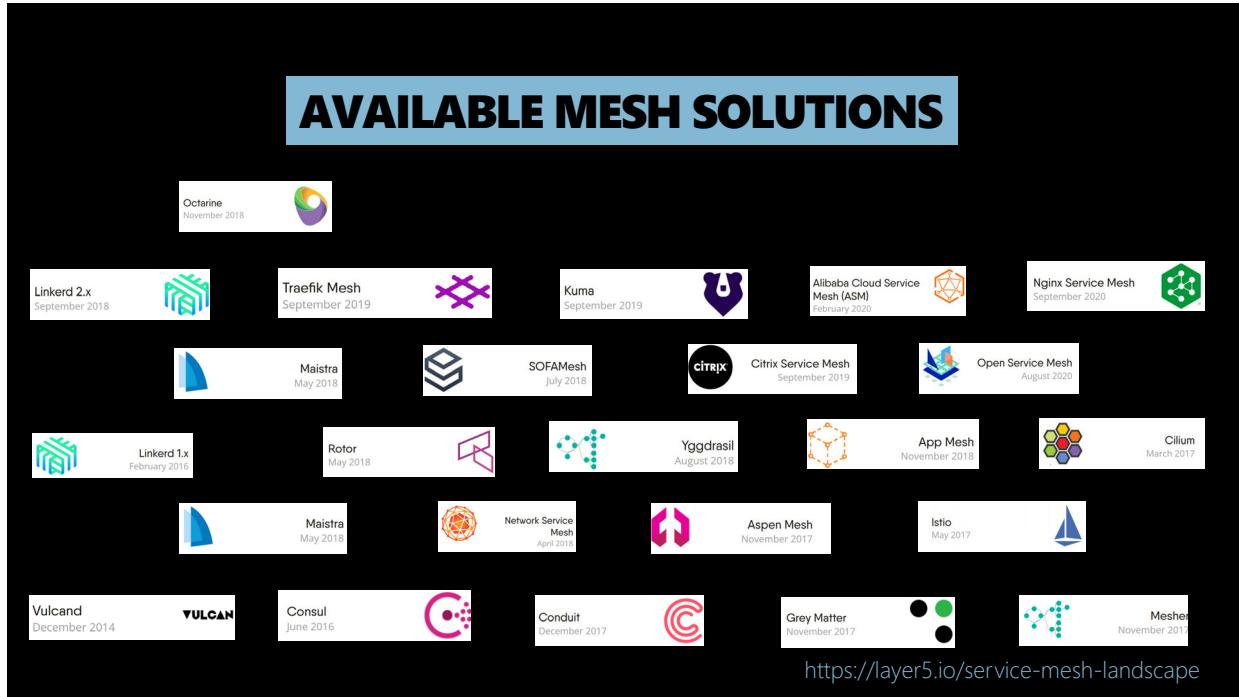
WHAT IS A SERVICES MESH?

*"The term *service mesh* is used to describe the **network** of **microservices** that make up such applications **and the interactions between them.**"*

@MARCELV

<https://istio.io/docs/concepts/what-is-istio/#what-is-a-service-mesh>

Visual Studio Live! Austin 2022



WHAT IS OSM ?

"Open Service Mesh (OSM) is a lightweight and extensible cloud native service mesh.

OSM takes a simple approach for users to uniformly manage, secure, and get out-of-the box observability features for highly dynamic microservice environments.

Using the CNCF Envoy project, OSM implements Service Mesh Interface (SMI) for securing and managing your microservice applications."

<https://openservicemesh.io/>

WHAT IS ISTIO ?

*"It is a completely open source **service mesh** that layers transparently onto existing distributed applications. It is also a **platform**, including APIs that let it integrate into any logging platform, or telemetry or policy system."*

@MARCELV

<https://istio.io/docs/concepts/what-is-istio/>

ISTIO, SERVICE MESH FOR K8S

The slide features four cards, each representing a pillar of Istio:

- Connect:** Shows a network of nodes with various connection paths. Description: "Intelligently control the flow of traffic and API calls between services, conduct a range of tests, and upgrade gradually with red/black deployments."
- Secure:** Shows a sailboat on waves. Description: "Automatically secure your services through managed authentication, authorization, and encryption of communication between services."
- Control:** Shows a central node with multiple arrows pointing outwards. Description: "Apply policies and ensure that they're enforced, and that resources are fairly distributed among consumers."
- Observe:** Shows a network of nodes with a vertical line and arrows indicating monitoring. Description: "See what's happening with rich automatic tracing, monitoring, and logging of all your services."

@MARCELV <https://istio.io/docs/concepts/what-is-istio/>

ISTIO, SERVICE MESH FOR K8S

The slide illustrates the Istio architecture with two main components:

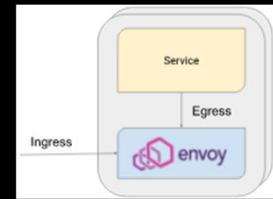
- Pilot:** Described as a "Central authority Service registry Manages all Sidecars". It contains an "Abstract model", "Rules API", "Network API", and the "Envoy API". It interacts with "Platform adapters" (Kubernetes, Mesos, CloudFoundry, etc.) and the "user".
- Sidecar:** Described as "Envoy proxy (in Pod) Manages all TCP networking of a Pod Created automatically, or explicitly Can restrict communication between Services". It shows four instances of the Envoy proxy running in pods.

@MARCELV

ENVOY PROXY

Sidecar

Envoy proxy injected in your Pod
Manages all TCP networking of a Pod
Created automatically, or explicitly
Can restrict communication between Services

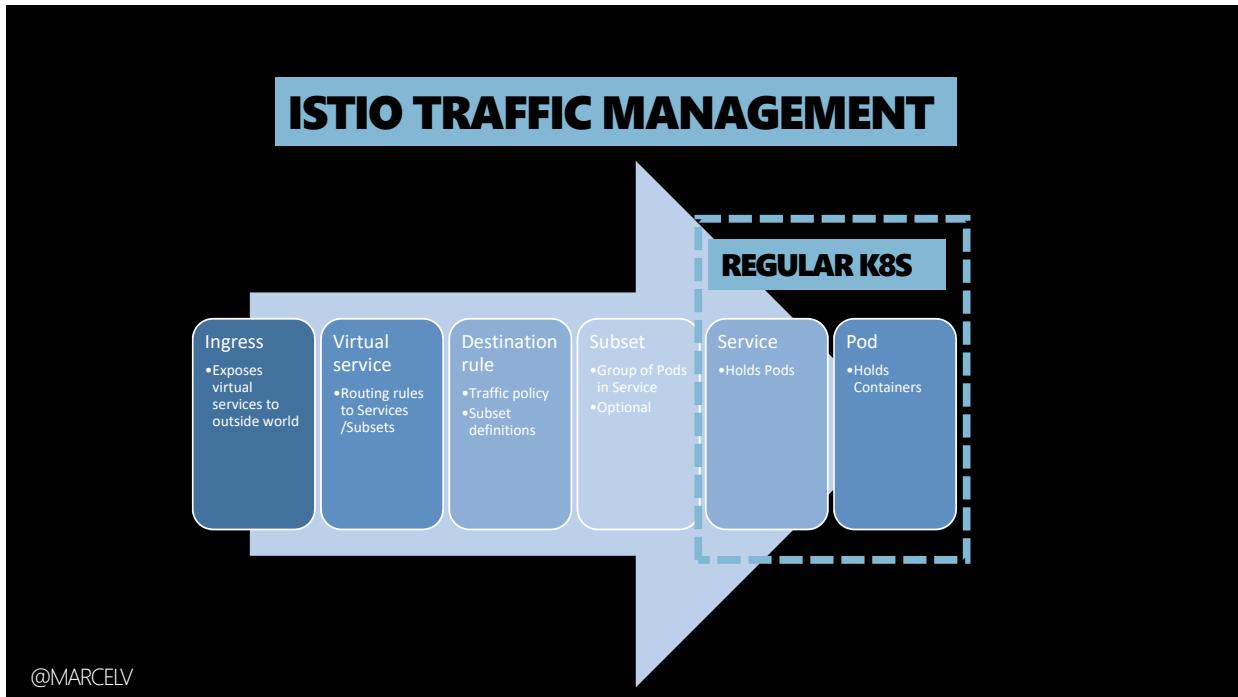


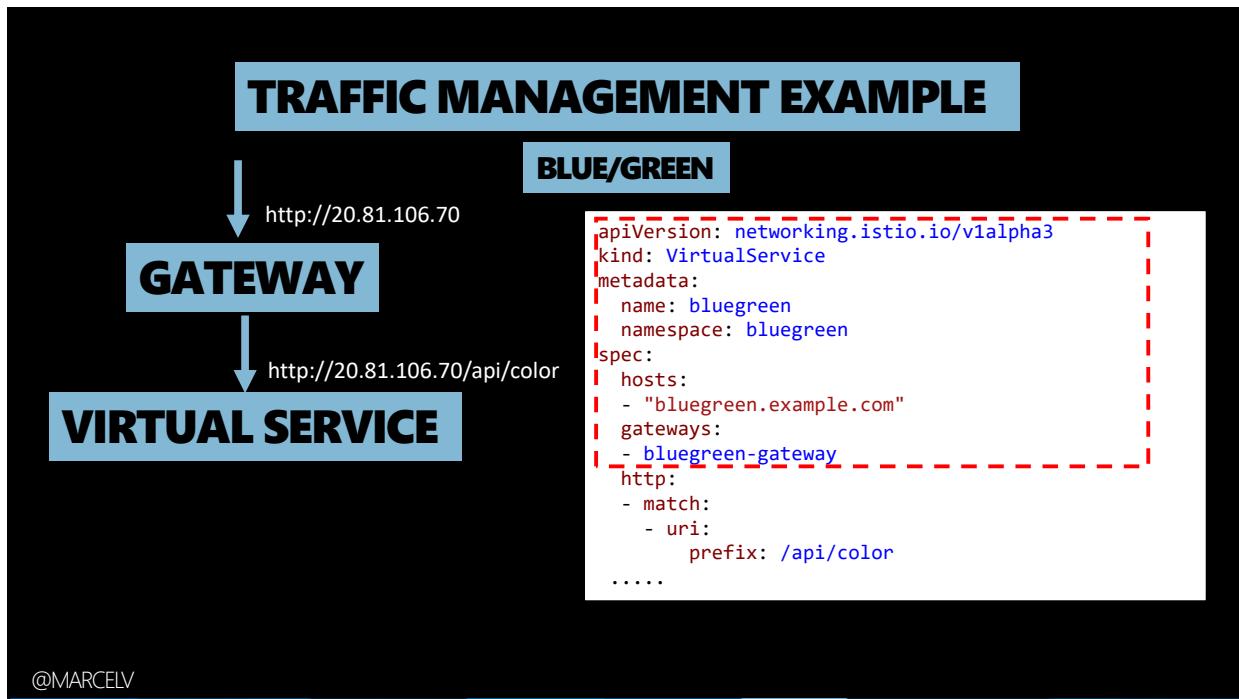
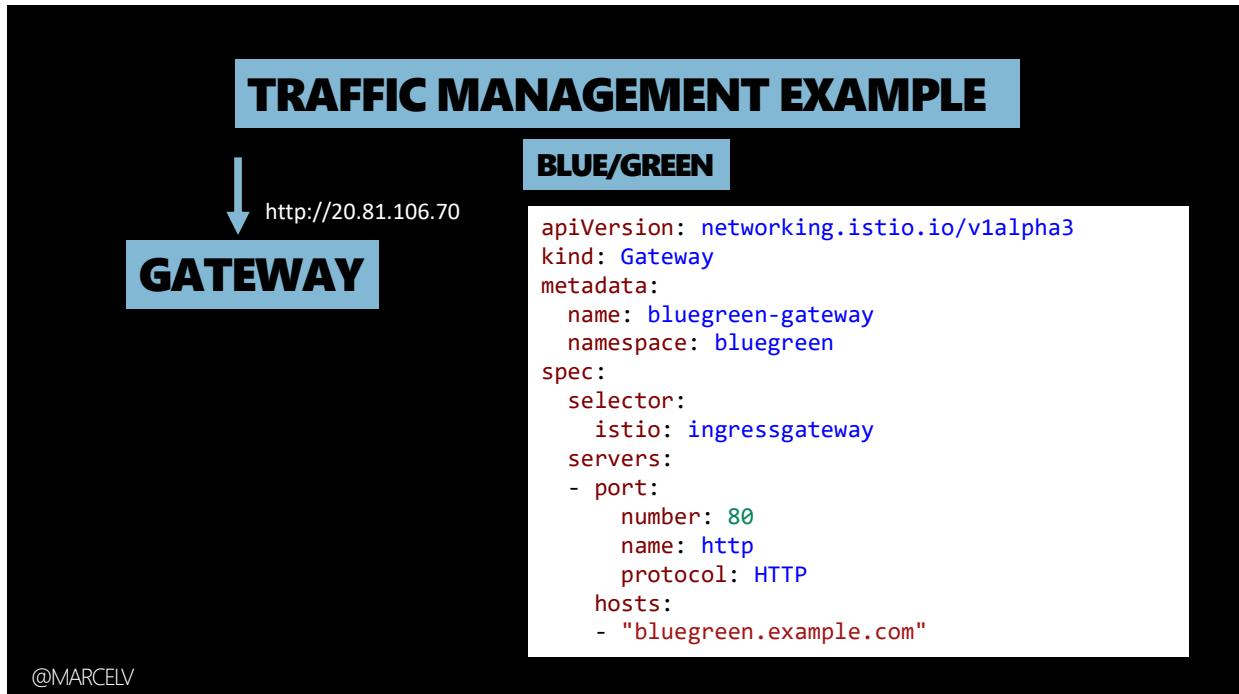
Envoy features

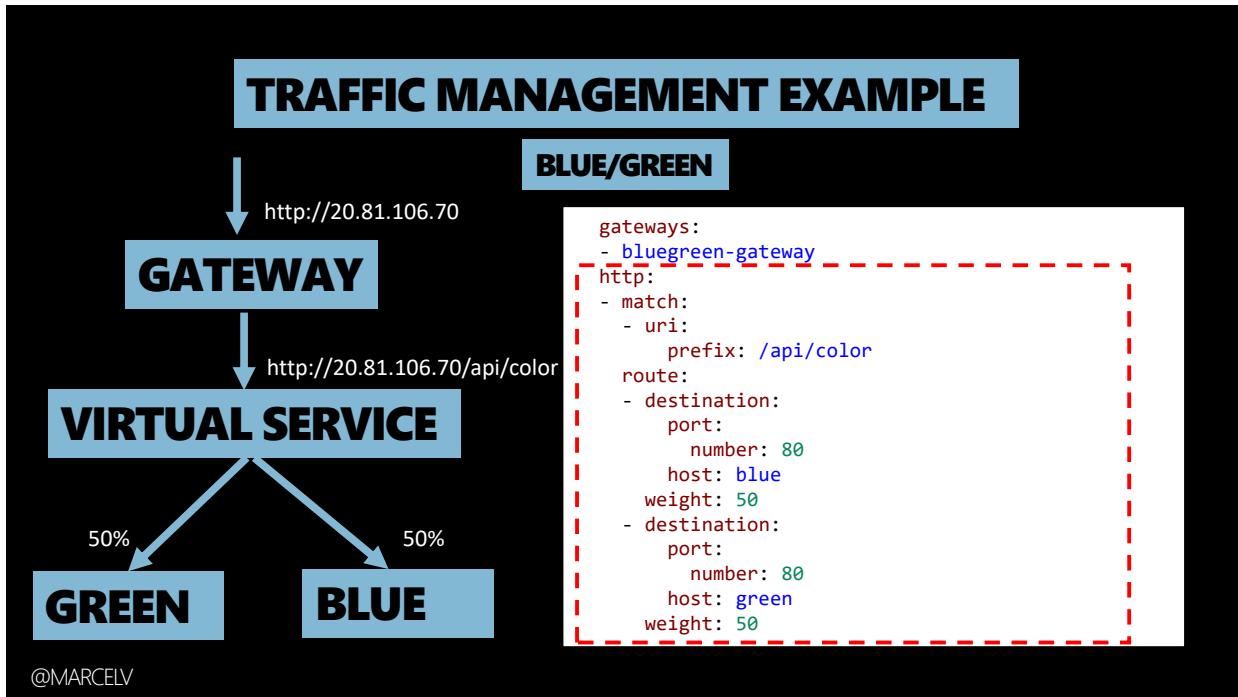
- **Statistics**
- **Throttling both rate and bandwidth**
- **Resilience**
- **Security**

@MARCELV

SIDECAR INJECTION







ISTIO, TRAFFIC MANAGEMENT

NETWORK RESILIENCE AND TESTING

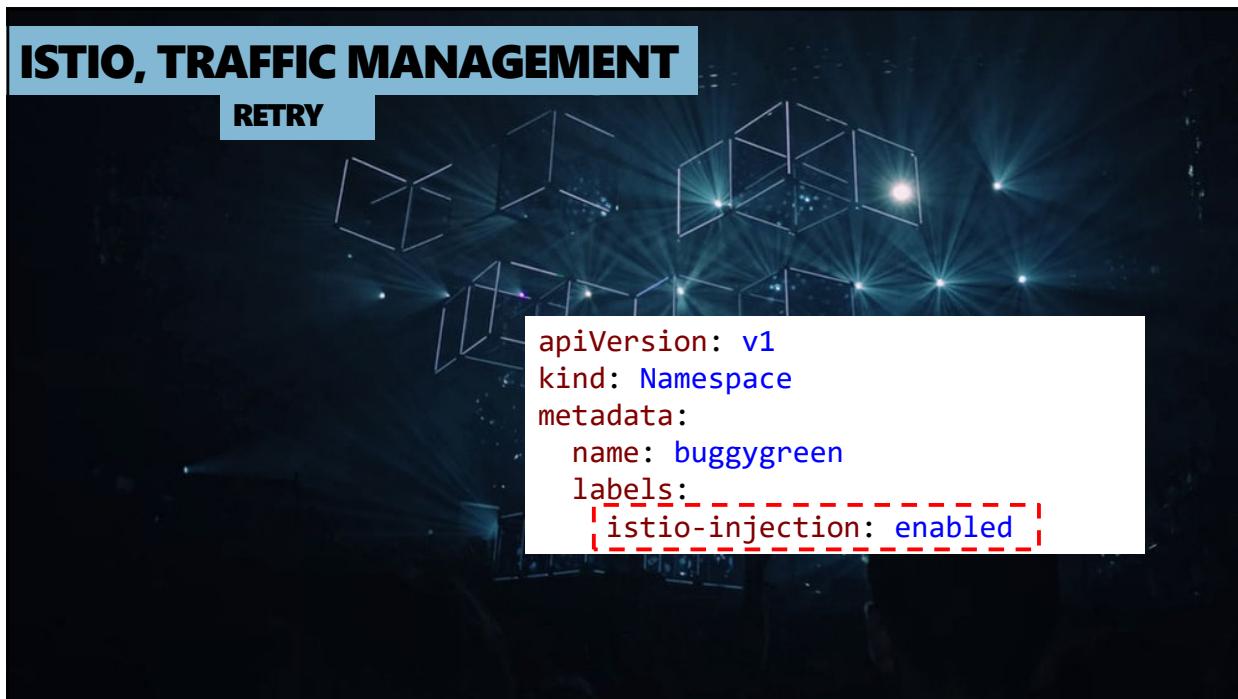
- TIMEOUTS
- RETRIES
- CIRCUIT BREAKERS
- FAULT INJECTION



ISTIO, TRAFFIC MANAGEMENT

RETRY

```
apiVersion: v1
kind: Namespace
metadata:
  name: buggygreen
  labels:
    istio-injection: enabled
```



ISTIO, TRAFFIC MANAGEMENT

RETRY

BUGGY V2

GREEN V1

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: buggy
  namespace: buggygreen
spec:
  replicas: 1
  selector:
    matchLabels:
      app: buggygreen-app
      version: v2
  template:
    metadata:
      labels:
        app: buggygreen-app
        version: v2
    spec:
      containers:
        - name: blue
          image: xpiritbv/bluegreen:buggy
          ports:
            - containerPort: 8080
          imagePullPolicy: Always
```

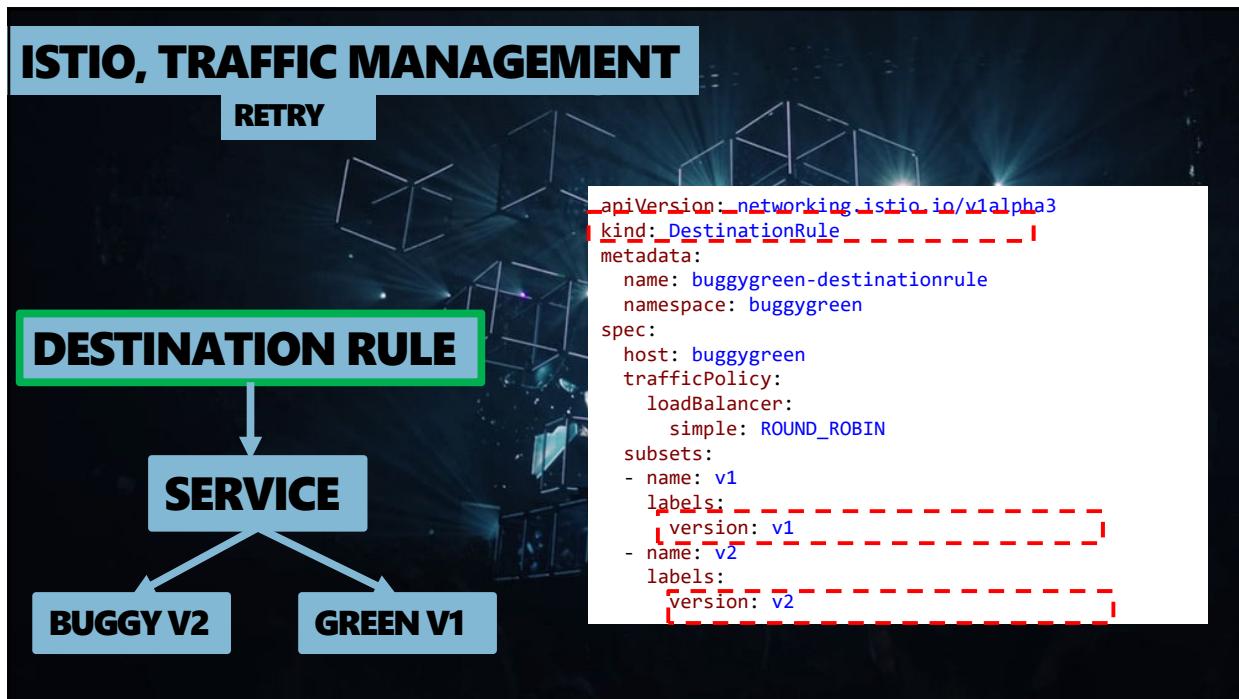
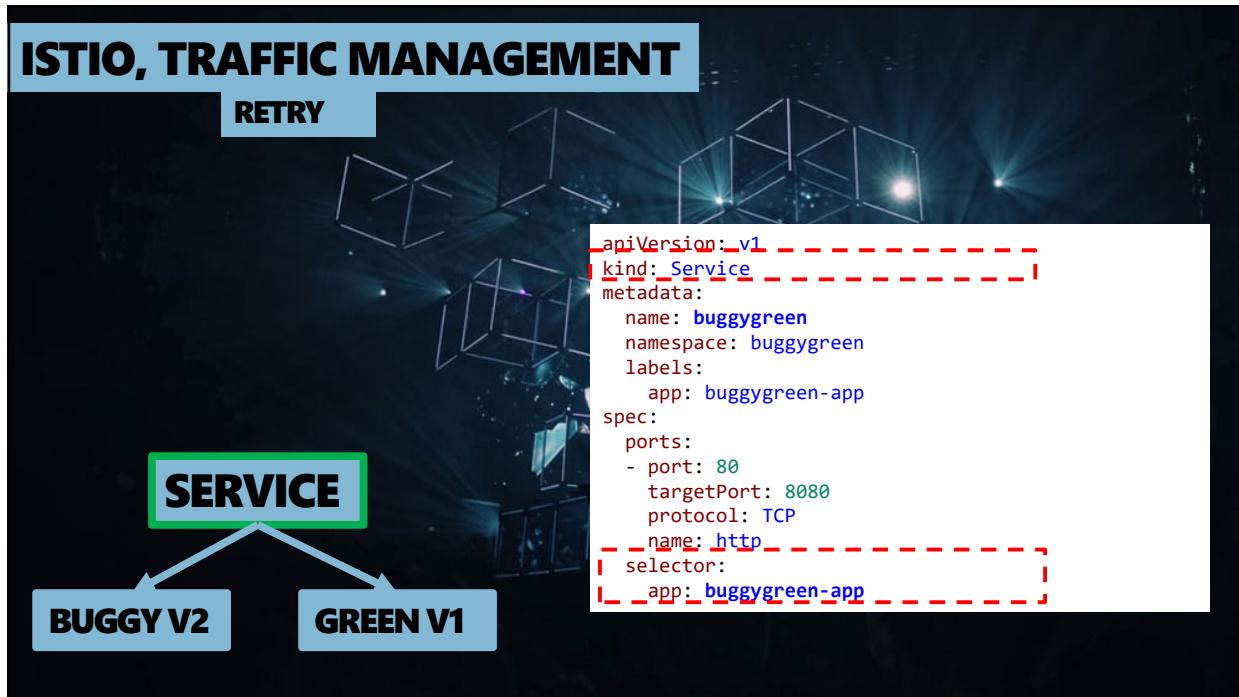
ISTIO, TRAFFIC MANAGEMENT

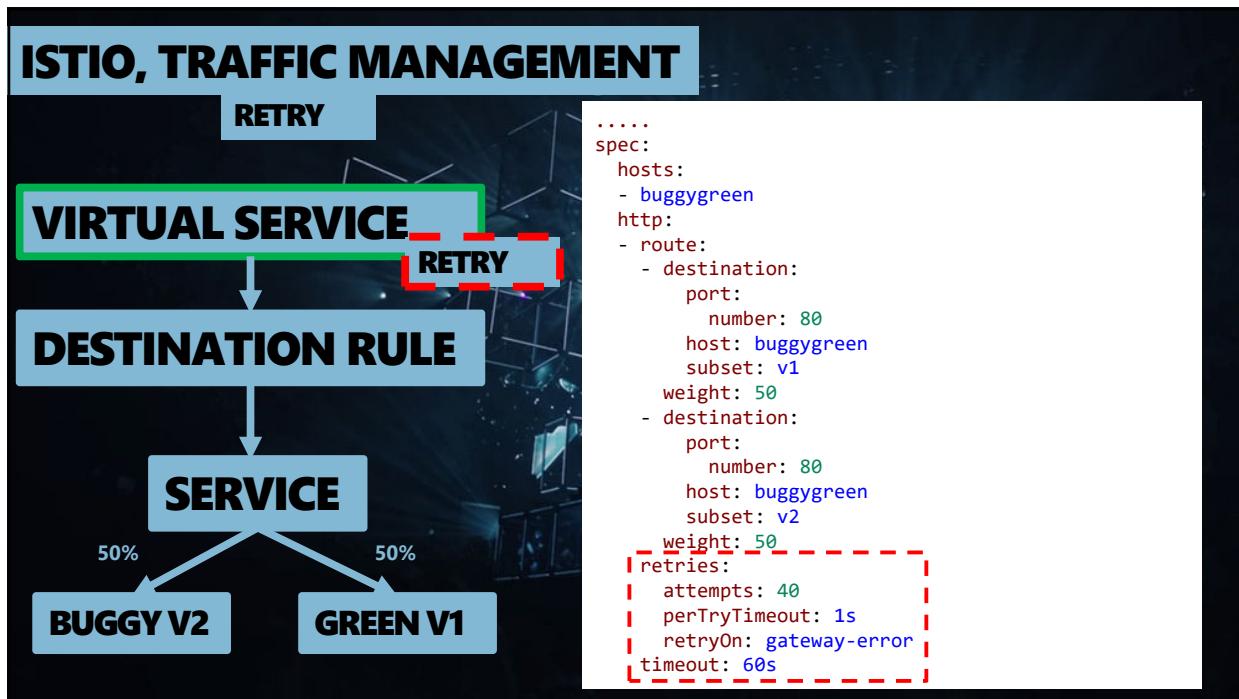
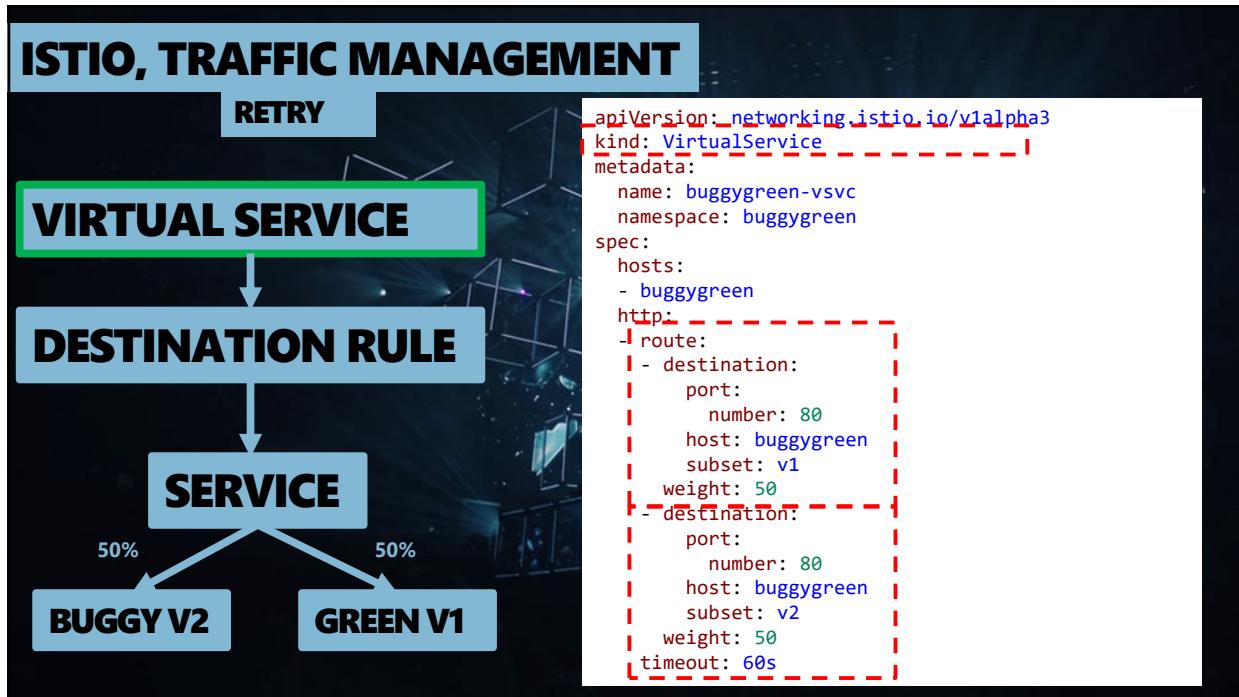
RETRY

BUGGY V2

GREEN V1

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: green
  namespace: buggygreen
spec:
  replicas: 1
  selector:
    matchLabels:
      app: buggygreen-app
      version: v1
  template:
    metadata:
      labels:
        app: buggygreen-app
        version: v1
    spec:
      containers:
        - name: green
          image: xpiritbv/bluegreen:green
          ports:
            - containerPort: 8080
```





RETRY IN ACTION

SECURITY

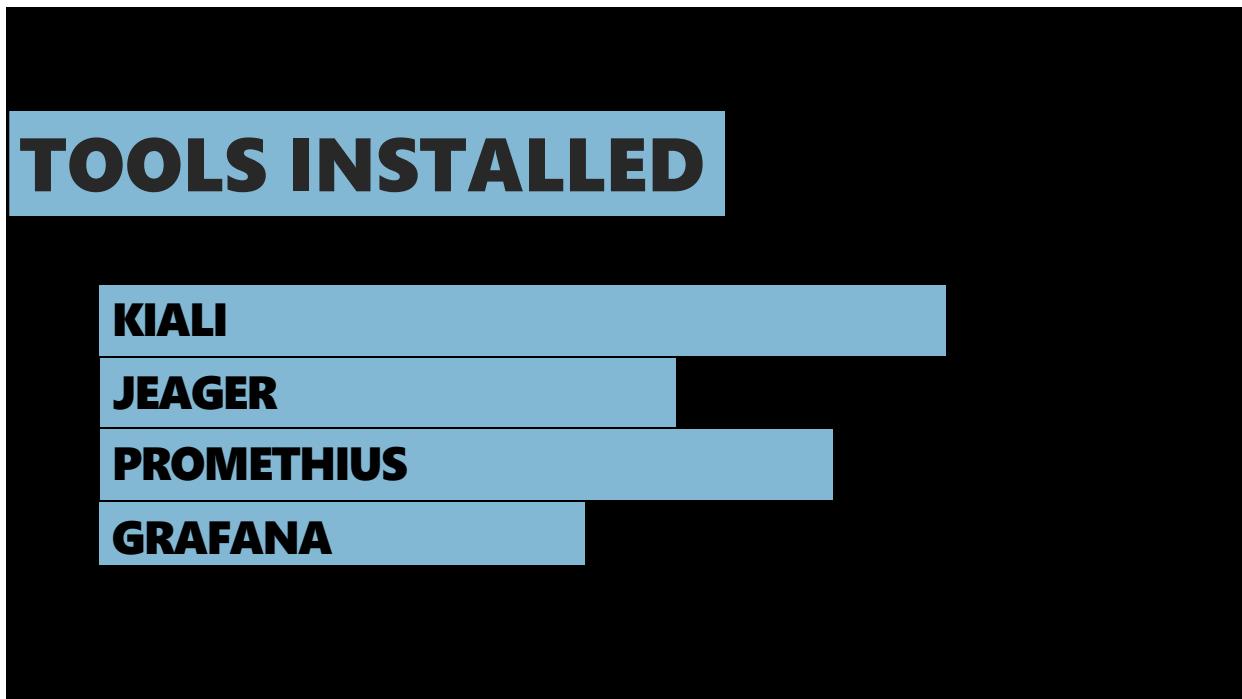
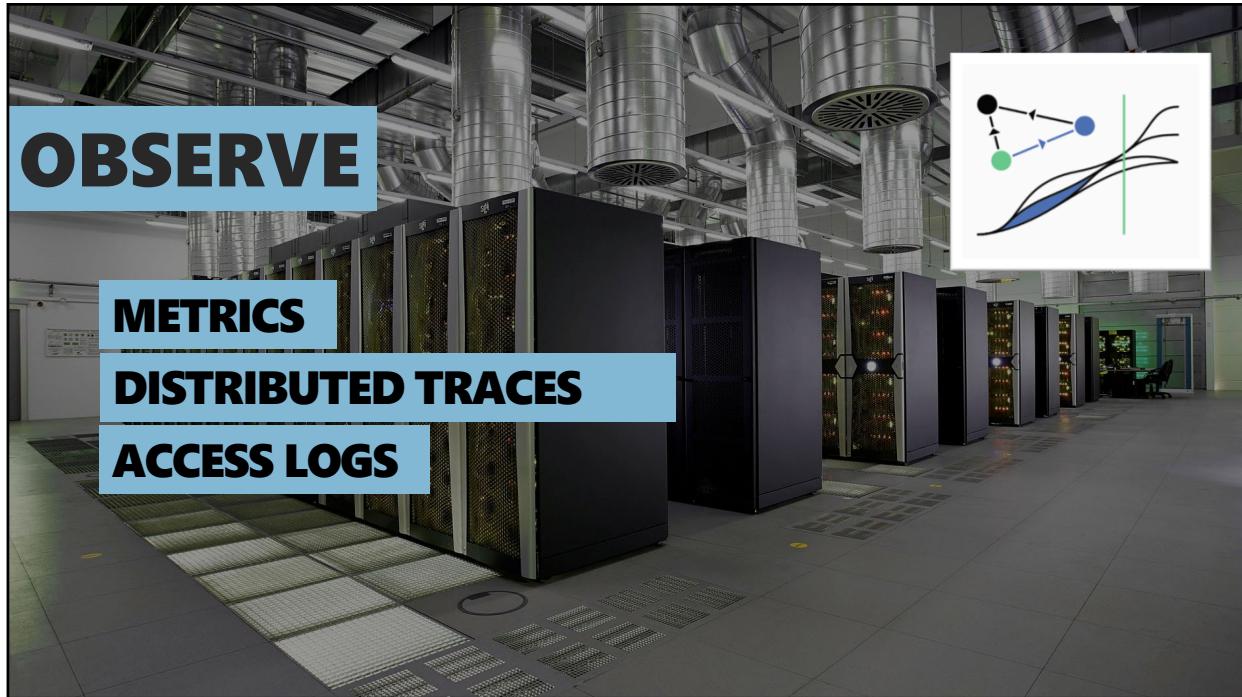


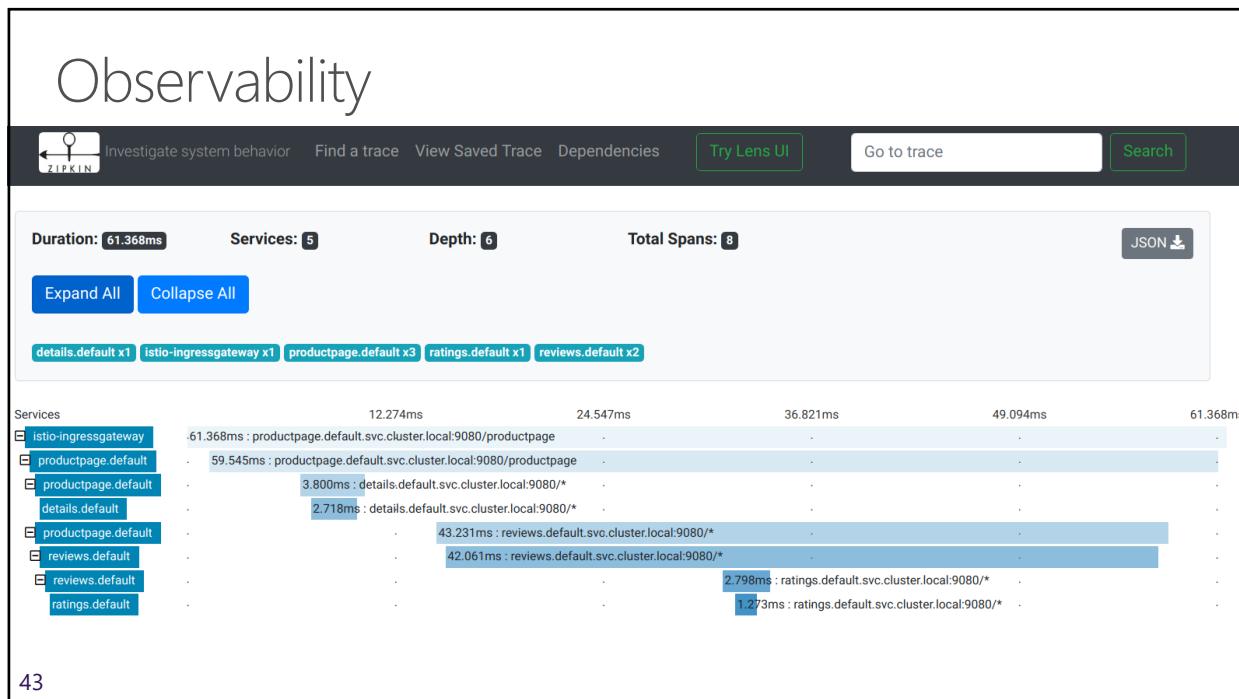
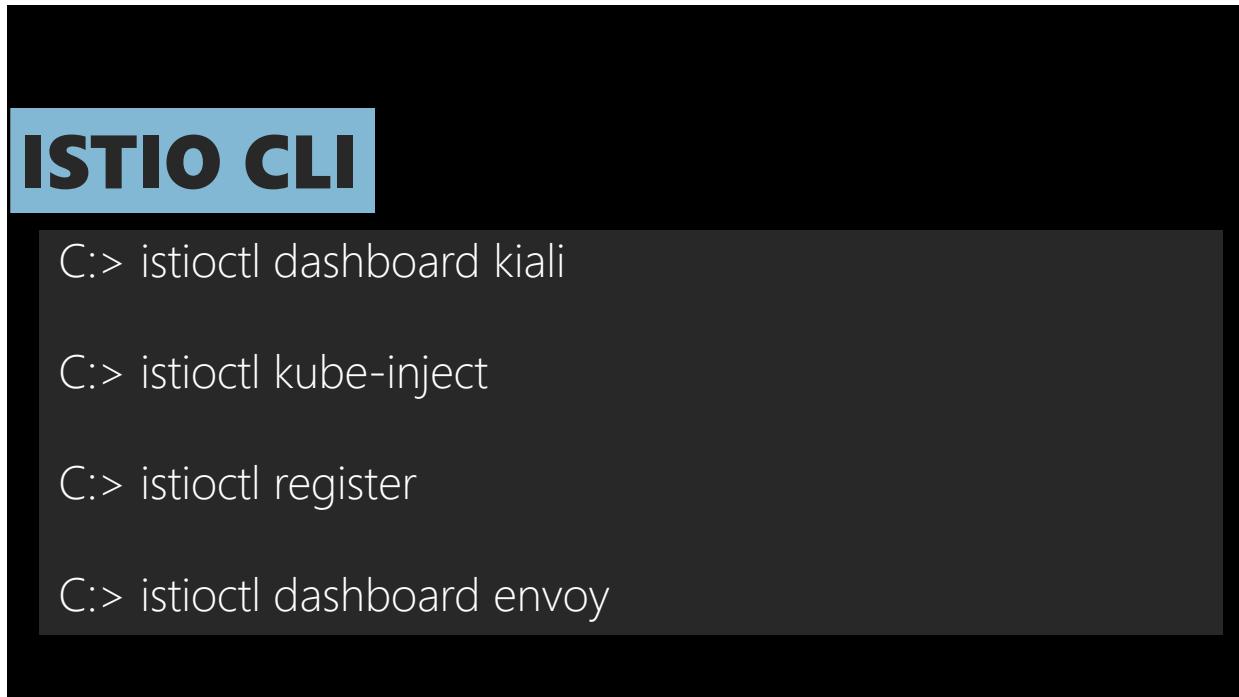
SECURE TRAFFIC BETWEEN SERVICES

MUTUAL TLS

AUTHENTICATION

AUDIT TRAIL





43

SHOW TRAFIC FLOW IN CLUSTER

WRAPPING UP

Service Mesh

Support Micro Service

Traffic Management

Observability

Security

SMI (Service Mesh Interface)

Istio

One of the many service meshes available

Adds about 10% overhead!

- Azure DevOps the big picture
- Getting Started with UI Testing with Appium
- Application Testing with Azure DevOps and Visual Studio 2019
- Continuous Delivery and DevOps with Azure DevOps, Managing Builds
- Continuous Delivery and DevOps with Azure DevOps, Managing releases
- Feature Toggles, Package Management and Versioning with Azure DevOps and Visual Studio 2019
- Continuous Delivery and DevOps with Azure DevOps: Source Control with Git
- Introduction to Docker on Windows with Visual Studio 2017
- Building a Continuous Delivery Pipeline with TFS and Visual Studio 2015

Marcel de Vries
CTO
Xpirit Netherlands
@marcelv <http://fluentbytes.com>

Attributions
Unsplash.com, pictures used for backgrounds
giphy.com for animated gifs



THANK
YOU!