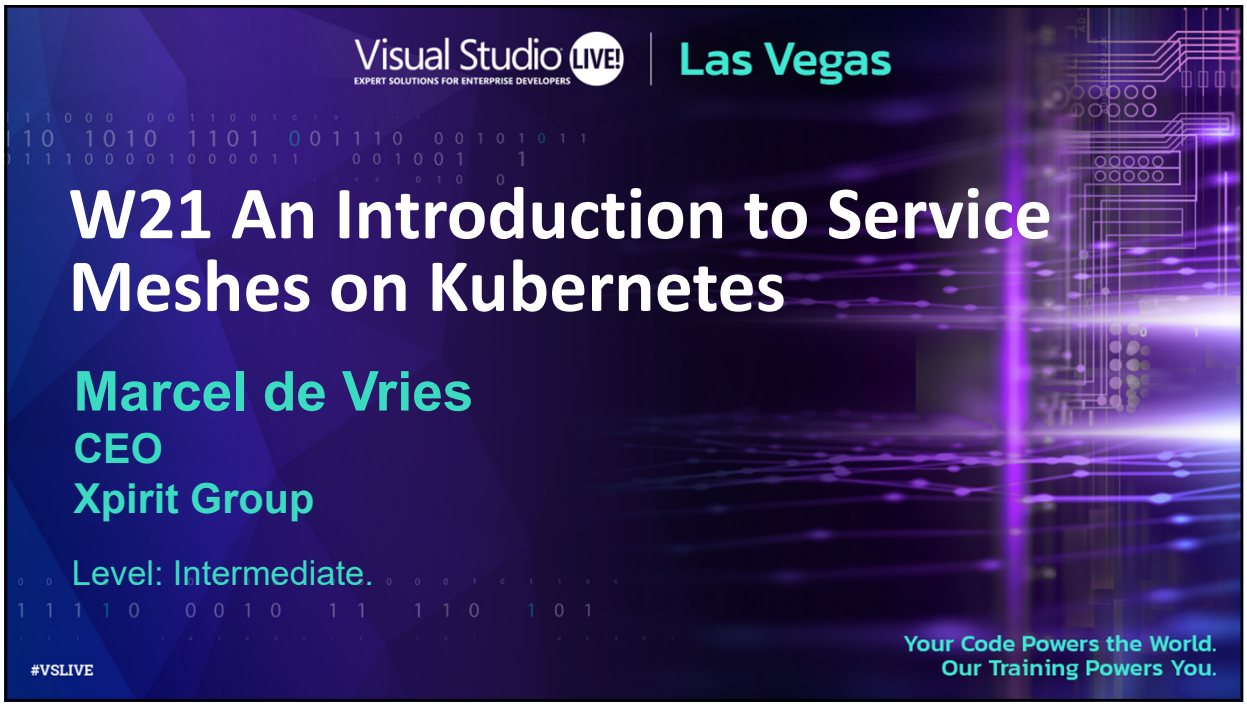
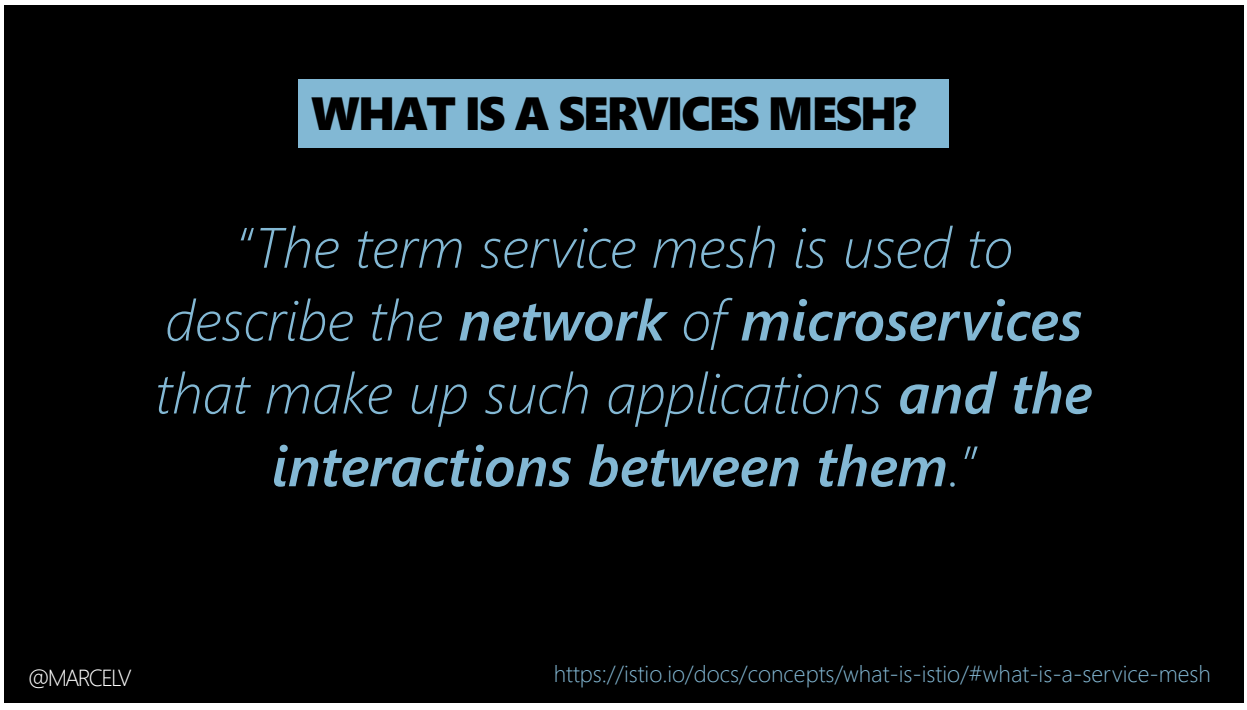
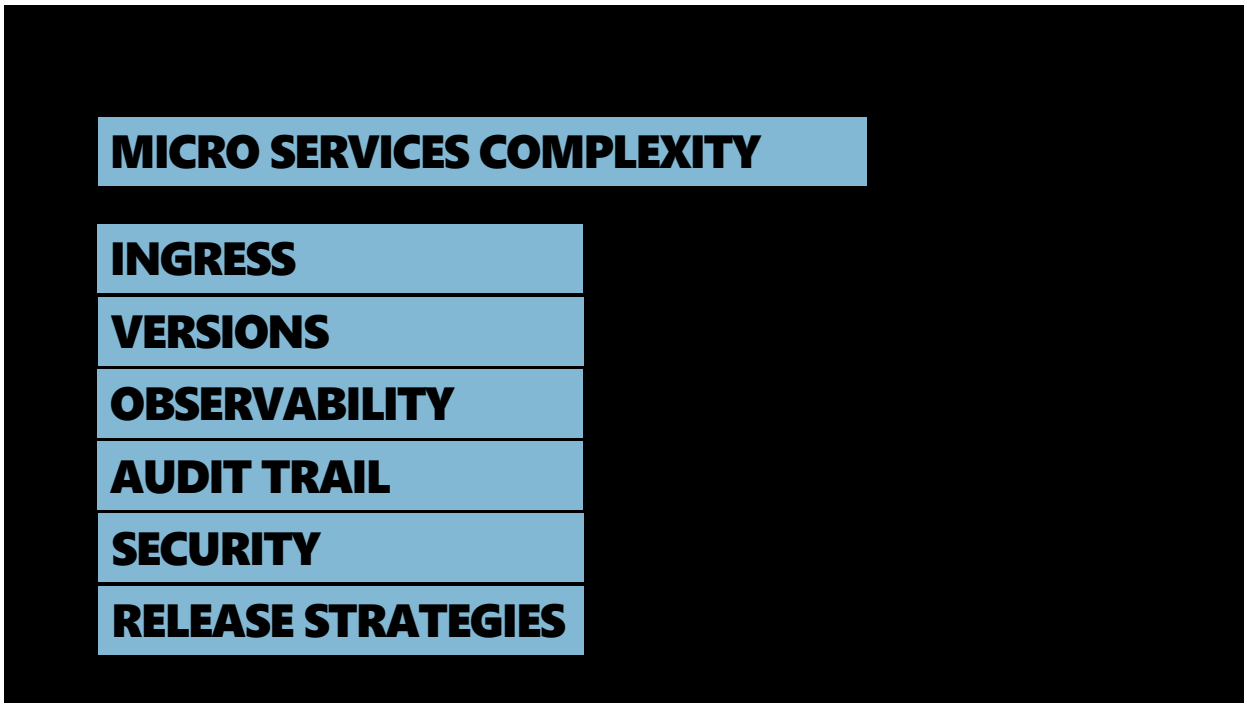

Visual Studio LIVE!
EXPERT SOLUTIONS FOR ENTERPRISE DEVELOPERS
Las Vegas
W21 An Introduction to Service Meshes on Kubernetes
Marcel de Vries
CEO
Xpirit Group
Level: Intermediate.
#VSLIVE
Your Code Powers the World.
Our Training Powers You.


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

LET'S TAKE A STEP BACK

MICRO SERVICES

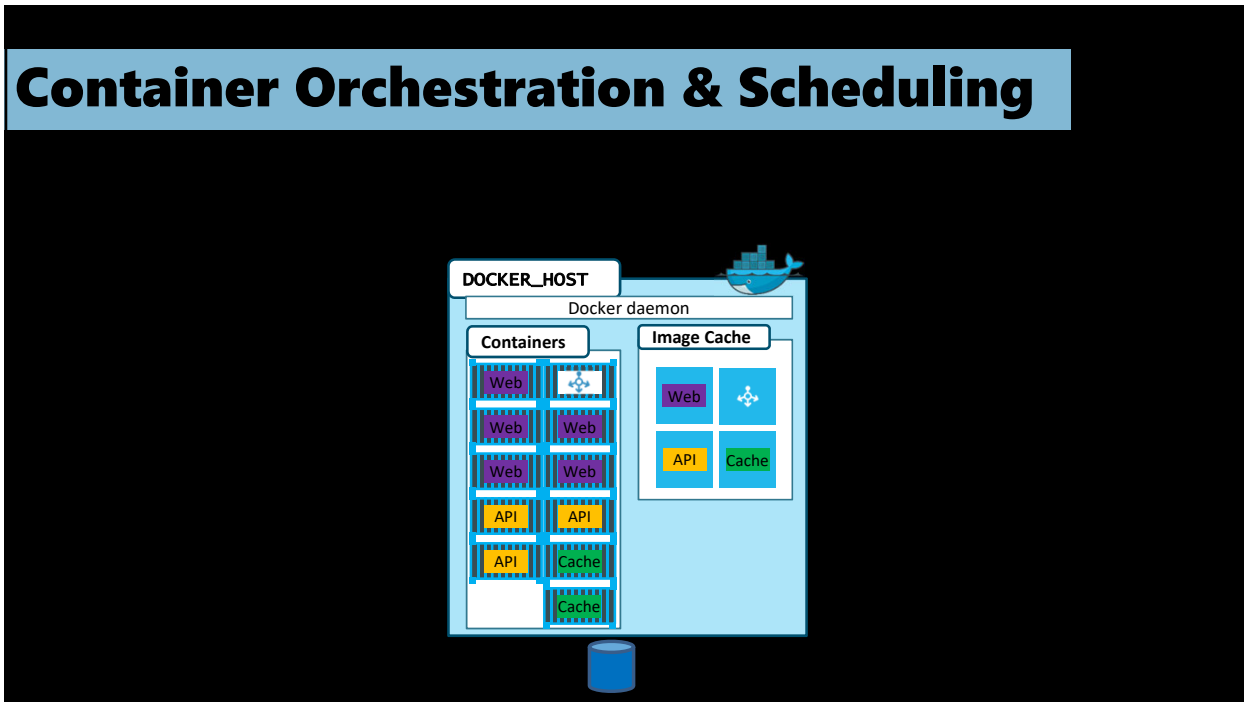
- AUTONOMOUS BUSINESS CAPABILITY
- INDIVIDUALY SCALABLE
- AVOID SERVICE DEPENDENCIES!
- FAULT TOLERANT

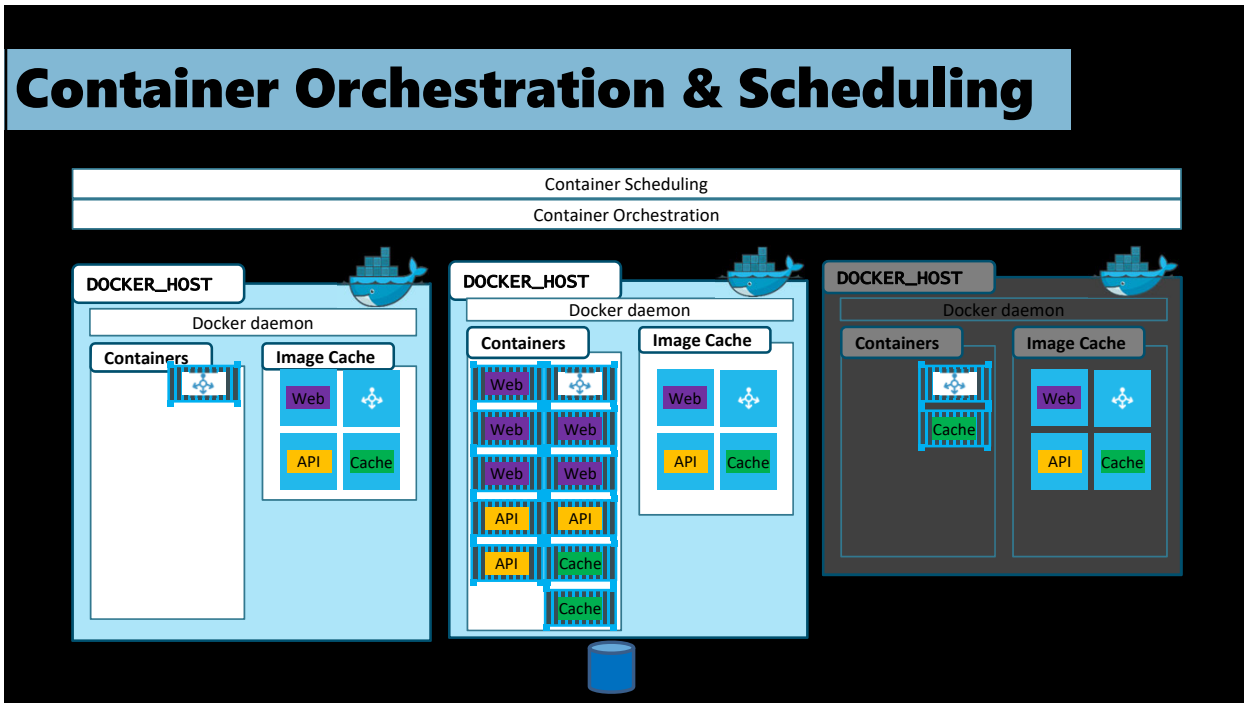
CONTAINERS & MICRO SERVICES

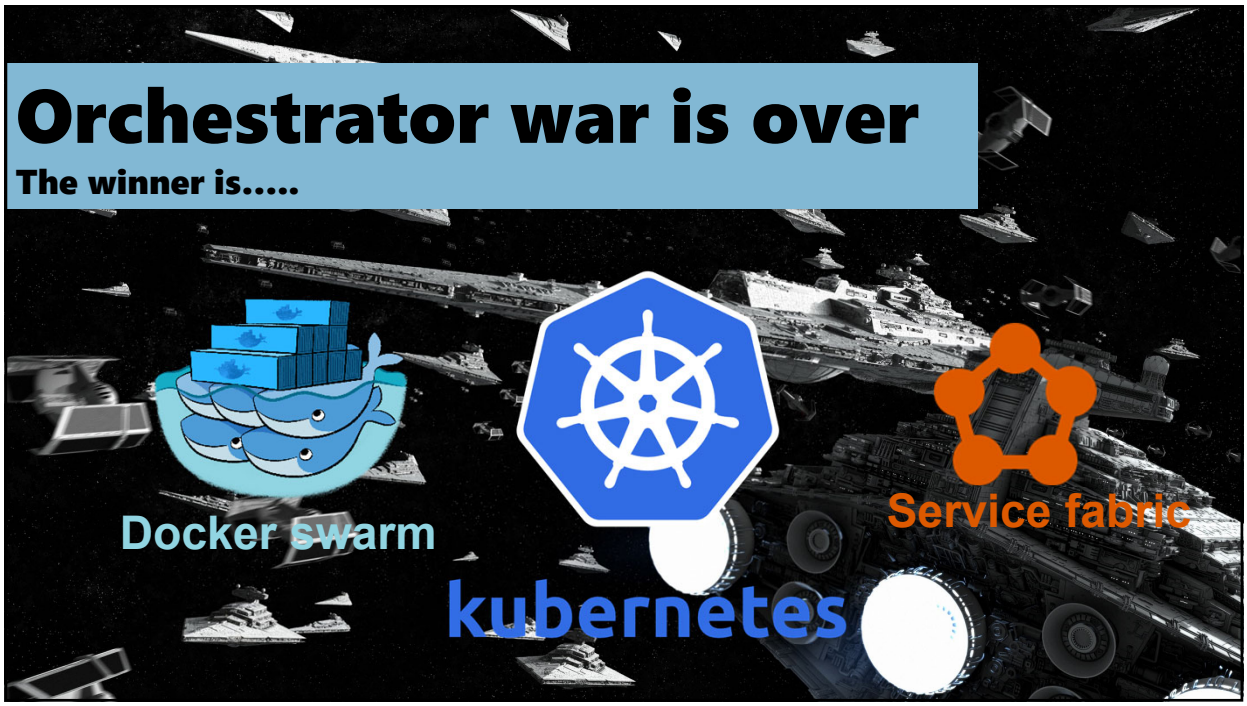
MARRIAGE MADE IN HEAVEN

WHAT WAS SOLVED BY KUBERNETES?

kubernetes


Container Orchestration & Scheduling
DOCKER_HOST
Docker daemon
Containers
Image Cache
Web
API
Cache


Container Orchestration & Scheduling
Container Scheduling
Container Orchestration
DOCKER_HOST
Docker daemon
Containers
Image Cache
Web
API
Cache



SOLVED INITIAL NEEDS

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

NEEDS BEYOND K8S

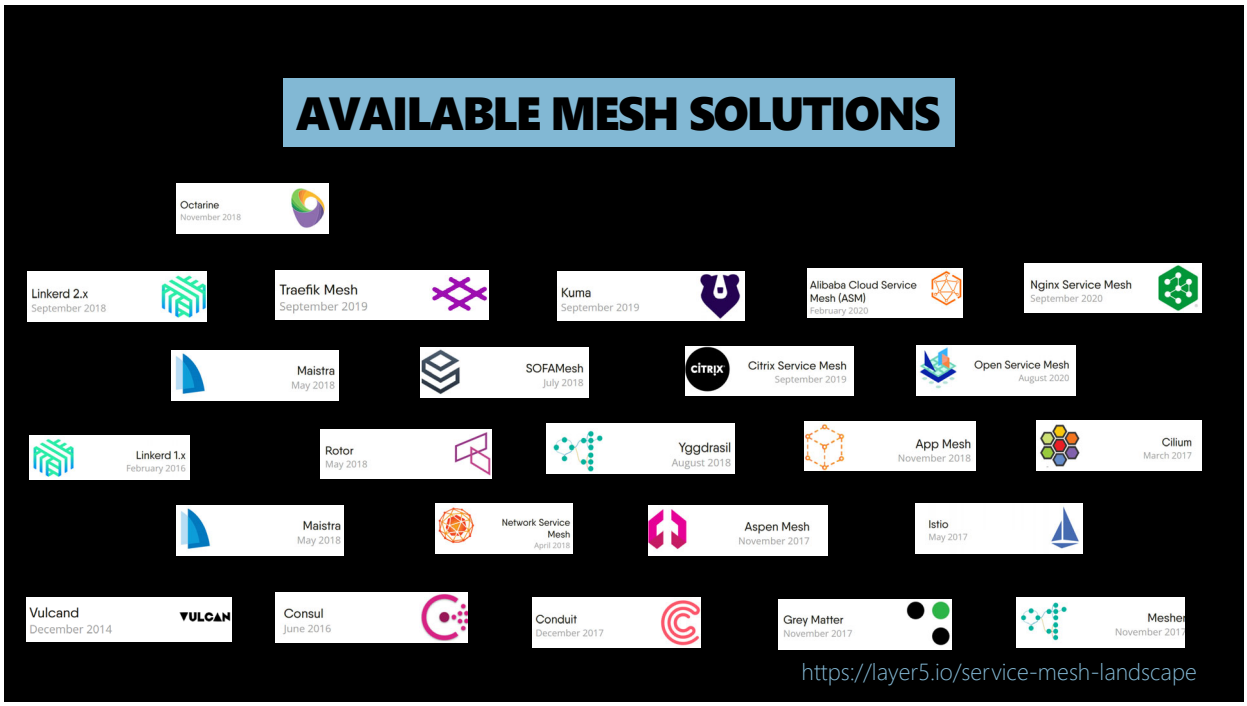
- TRAFIC POLICIES
- TRAFIC TELEMETRY
- TRAFIC MANAGEMENT
- TRAFIC RESILIENCE

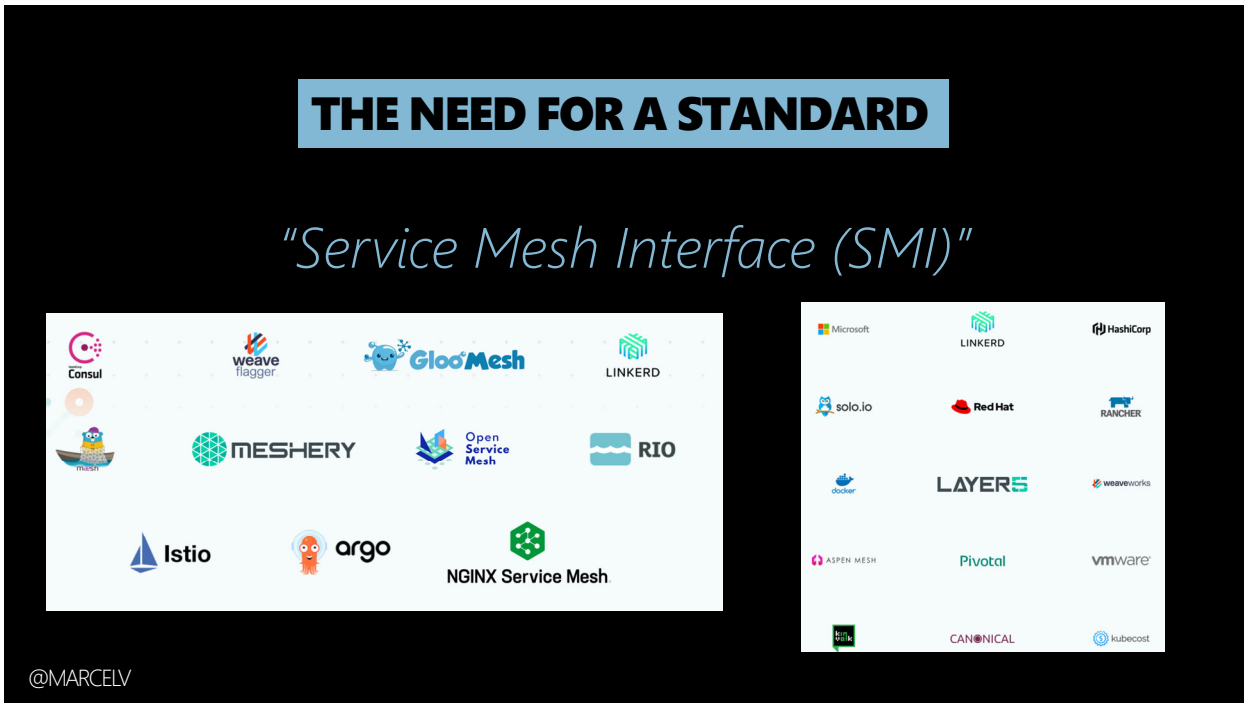
WHAT IS A SERVICES MESH?

*"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


AVAILABLE MESH SOLUTIONS
Octarine
November 2018
Linkerd 2.x
September 2018
Traefik Mesh
September 2019
Kuma
September 2019
Alibaba Cloud Service Mesh (ASM)
February 2020
Nginx Service Mesh
September 2020
Maistra
May 2018
SOFAMesh
July 2018
Citrix Service Mesh
September 2019
Open Service Mesh
August 2020
Linkerd 1.x
February 2016
Rotor
May 2018
Yggdrasil
August 2018
App Mesh
November 2018
Cilium
March 2017
Maistra
May 2018
Network Service Mesh
April 2018
Aspen Mesh
November 2017
Istio
May 2017
Vulcand
December 2014
Consul
June 2016
Conduit
December 2017
Grey Matter
November 2017
Mesher
November 2017
https://layer5.io/service-mesh-landscape


THE NEED FOR A STANDARD
"Service Mesh Interface (SMI)"
Consul
weave flagger
GlooMesh
LINKERD
MESHERY
Open Service Mesh
RIO
Istio
argo
NGINX Service Mesh
Microsoft
LINKERD
HashiCorp
solo.io
Red Hat
RANCHER
docker
LAYER5
weaveworks
ASPEN MESH
Pivotal
vmware
CANONICAL
kubecost
@MARCELV

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."

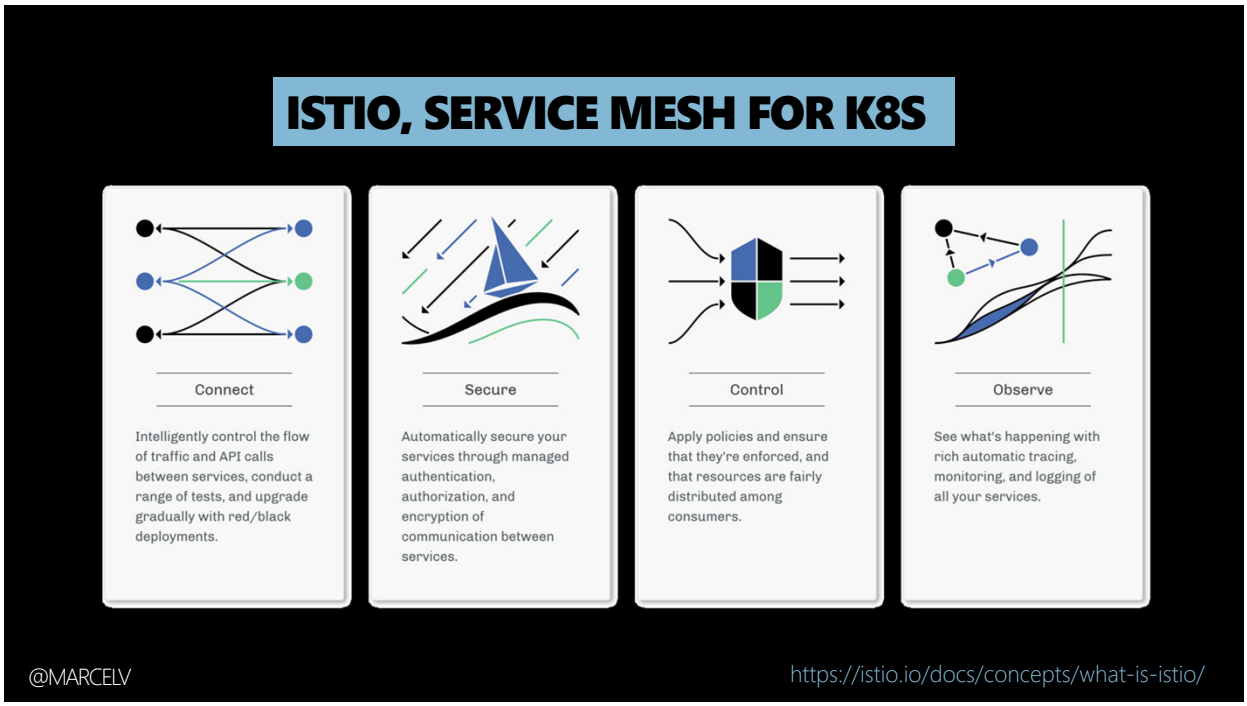
https://openservicemesh.io/

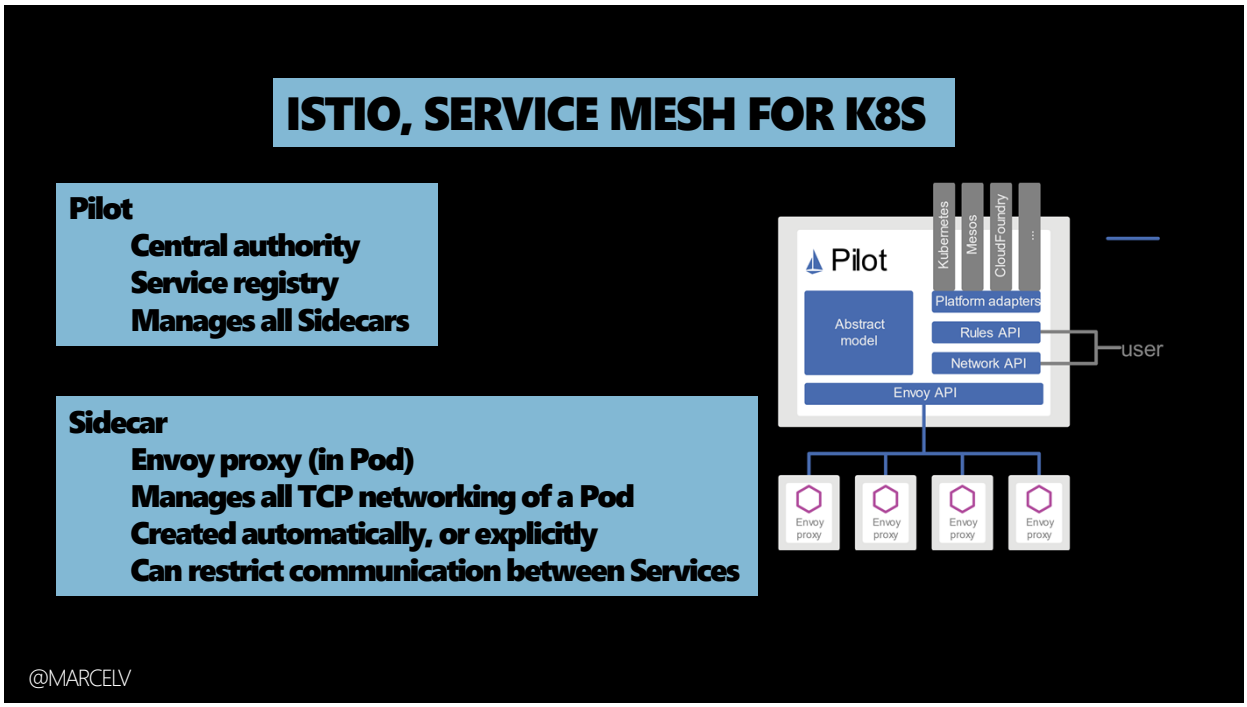
WHAT IS ISTIO ?

*"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/

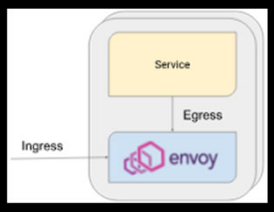

ISTIO, SERVICE MESH FOR K8S
Connect
Intelligently control the flow of traffic and API calls between services, conduct a range of tests, and upgrade gradually with red/black deployments.
Secure
Automatically secure your services through managed authentication, authorization, and encryption of communication between services.
Control
Apply policies and ensure that they're enforced, and that resources are fairly distributed among consumers.
Observe
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
Pilot
Central authority
Service registry
Manages all Sidecars
Sidecar
Envoy proxy (in Pod)
Manages all TCP networking of a Pod
Created automatically, or explicitly
Can restrict communication between Services
Pilot
Kubernetes
Mesos
CloudFoundry
...
Platform adapters
Abstract model
Rules API
Network API
Envoy API
user
Envoy proxy
Envoy proxy
Envoy proxy
Envoy proxy
@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

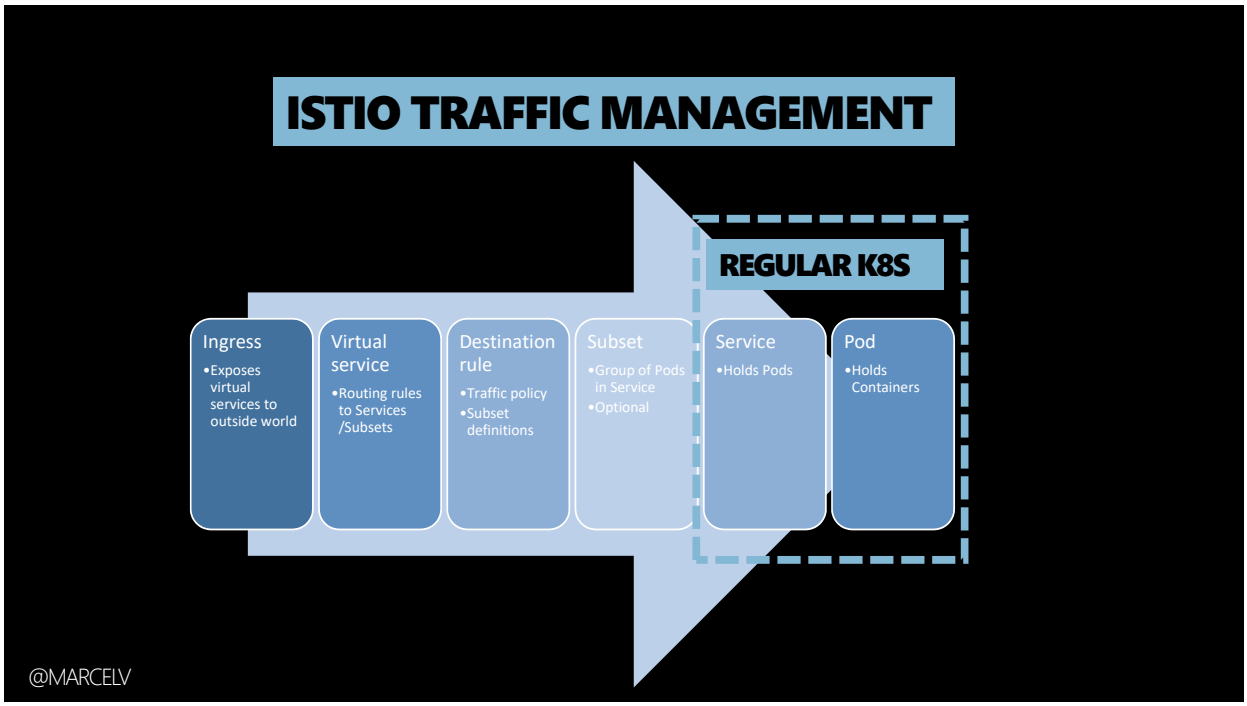


Envoy features

- Statistics
- Throttling both rate and bandwith
- Resilience
- Security

@MARCELV

SIDECAR INJECTION


ISTIO TRAFFIC MANAGEMENT
REGULAR K8S
Ingress
•Exposes virtual services to outside world
Virtual service
•Routing rules to Services /Subsets
Destination rule
•Traffic policy
•Subset definitions
Subset
•Group of Pods in Service
•Optional
Service
•Holds Pods
Pod
•Holds Containers
@MARCELV


ISTIO, TRAFFIC MANAGEMENT
ENABLES EASY TRAFIC SHIFTING
E.G. BLUE/GREEN
CANARY RELEASE
FOCUSED CANARY
@MARCELV

TRAFFIC MANAGEMENT EXAMPLE

http://20.81.106.70

GATEWAY

You use a gateway to manage inbound and outbound traffic for your mesh, letting you specify which traffic you want to enter or leave the mesh.

BLUE/GREEN

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: Gateway
metadata:
  name: bluegreen-gateway
  namespace: bluegreen
spec:
  selector:
    istio: ingressgateway
  servers:
  - port:
      number: 80
      name: http
      protocol: HTTP
    hosts:
    - "bluegreen.example.com"
```

@MARCELV

TRAFFIC MANAGEMENT EXAMPLE

BLUE/GREEN

http://20.81.106.70

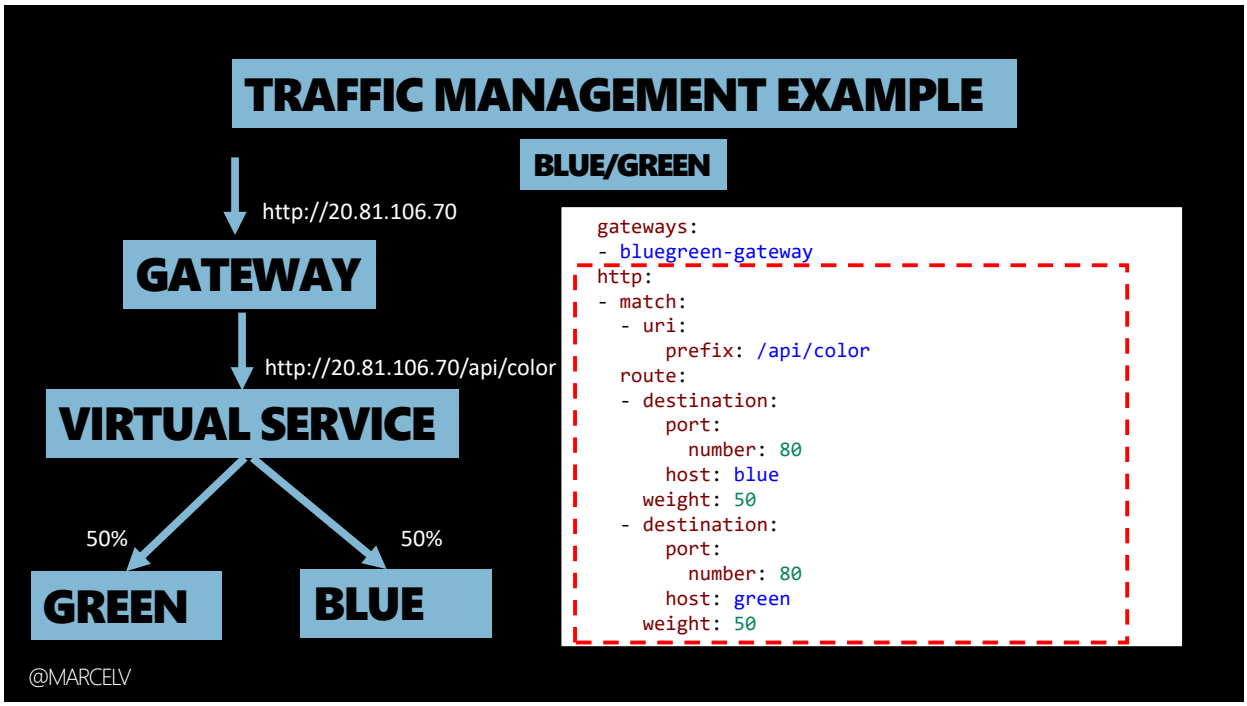
GATEWAY

http://20.81.106.70/api/color

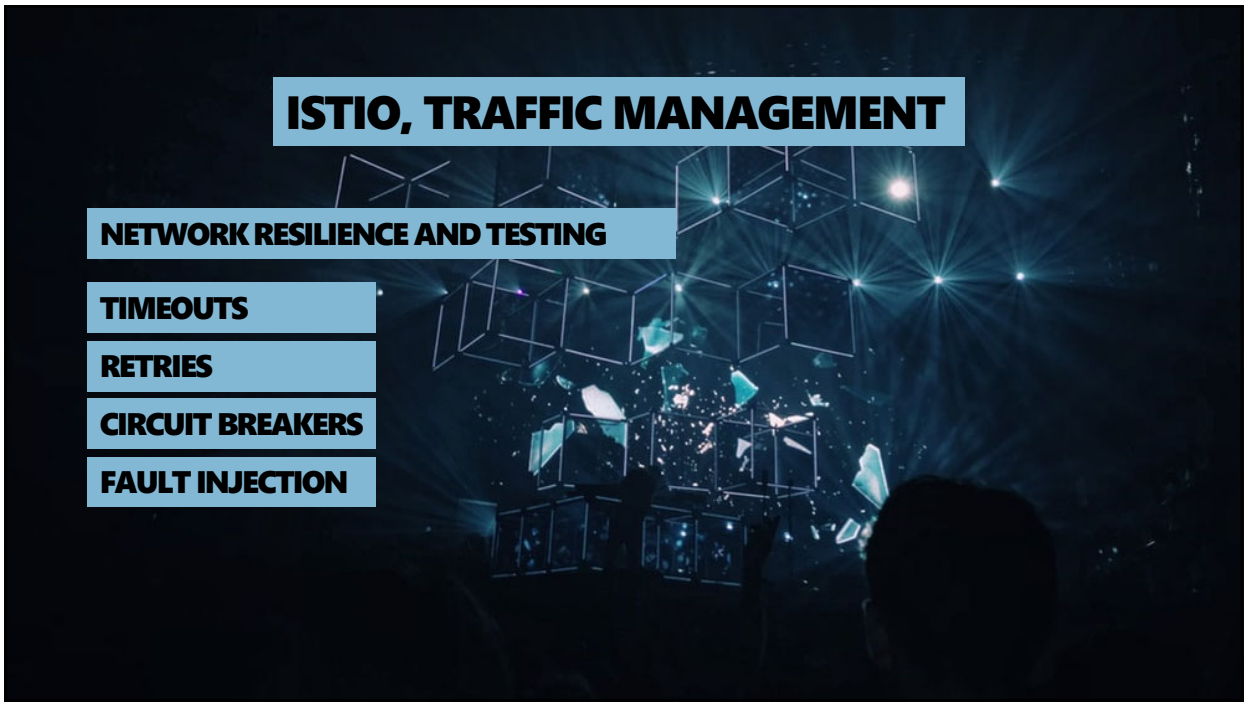
VIRTUAL SERVICE

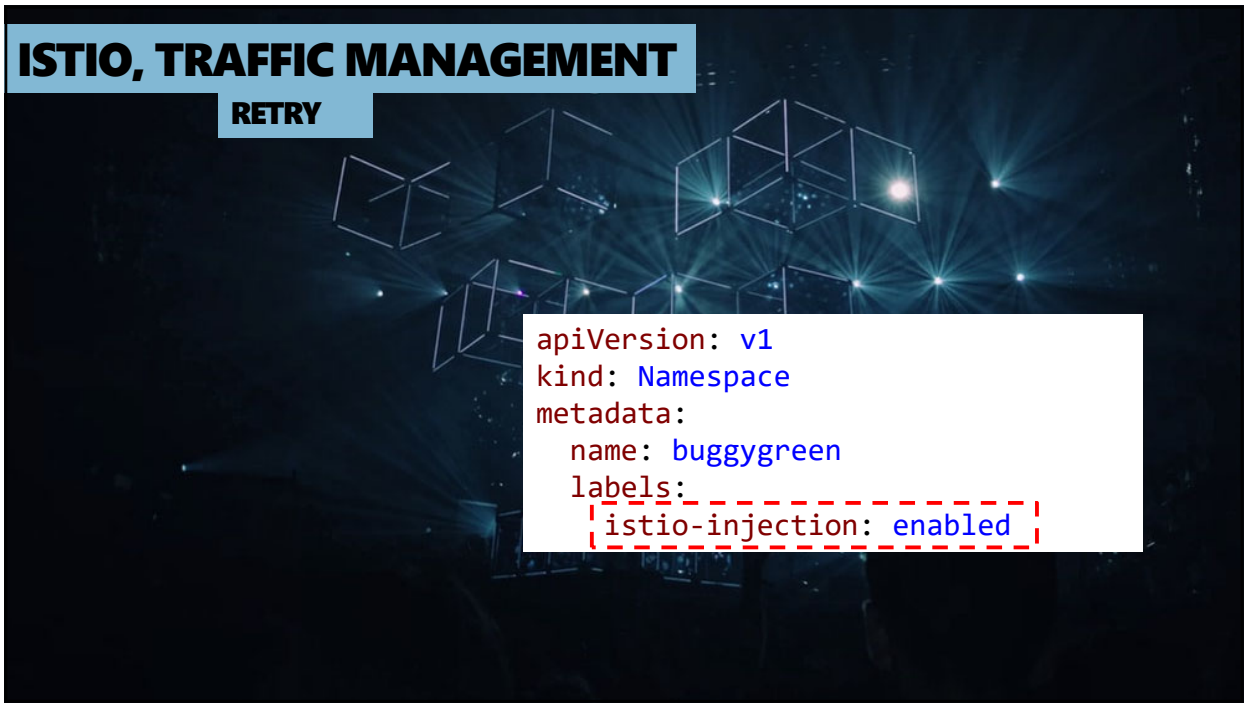
```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: bluegreen
  namespace: bluegreen
spec:
  hosts:
  - "bluegreen.example.com"
  gateways:
  - bluegreen-gateway
  http:
  - match:
    - uri:
        prefix: /api/color
 .....
```

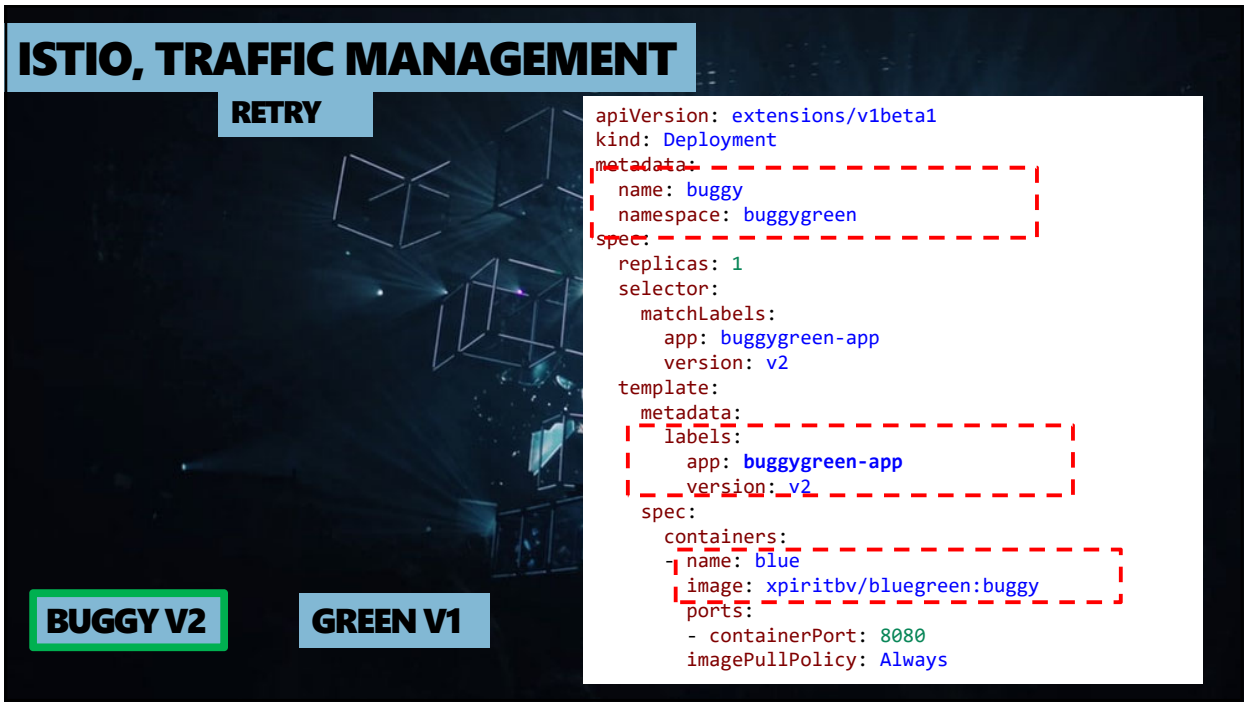
@MARCELV

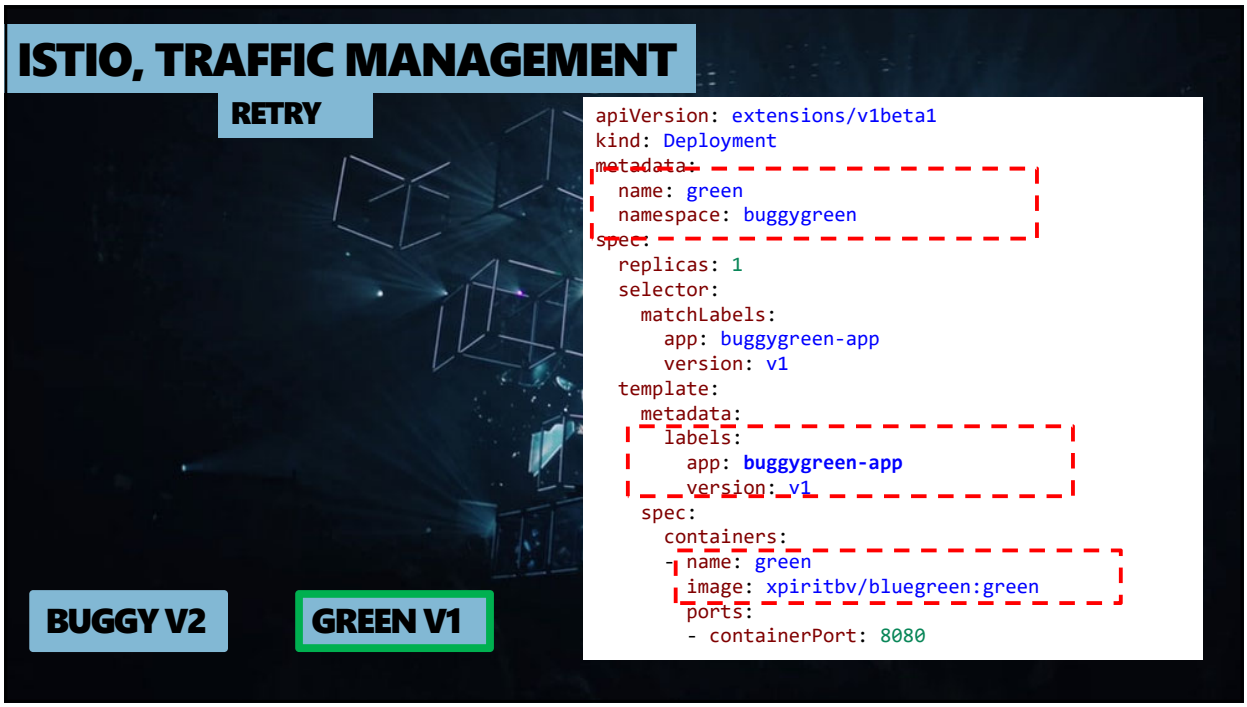

TRAFFIC MANAGEMENT EXAMPLE
BLUE/GREEN
http://20.81.106.70
GATEWAY
http://20.81.106.70/api/color
VIRTUAL SERVICE
50%
50%
GREEN
BLUE
@MARCELV
gateways:
- bluegreen-gateway
http:
- match:
- uri:
prefix: /api/color
route:
- destination:
port:
number: 80
host: blue
weight: 50
- destination:
port:
number: 80
host: green
weight: 50

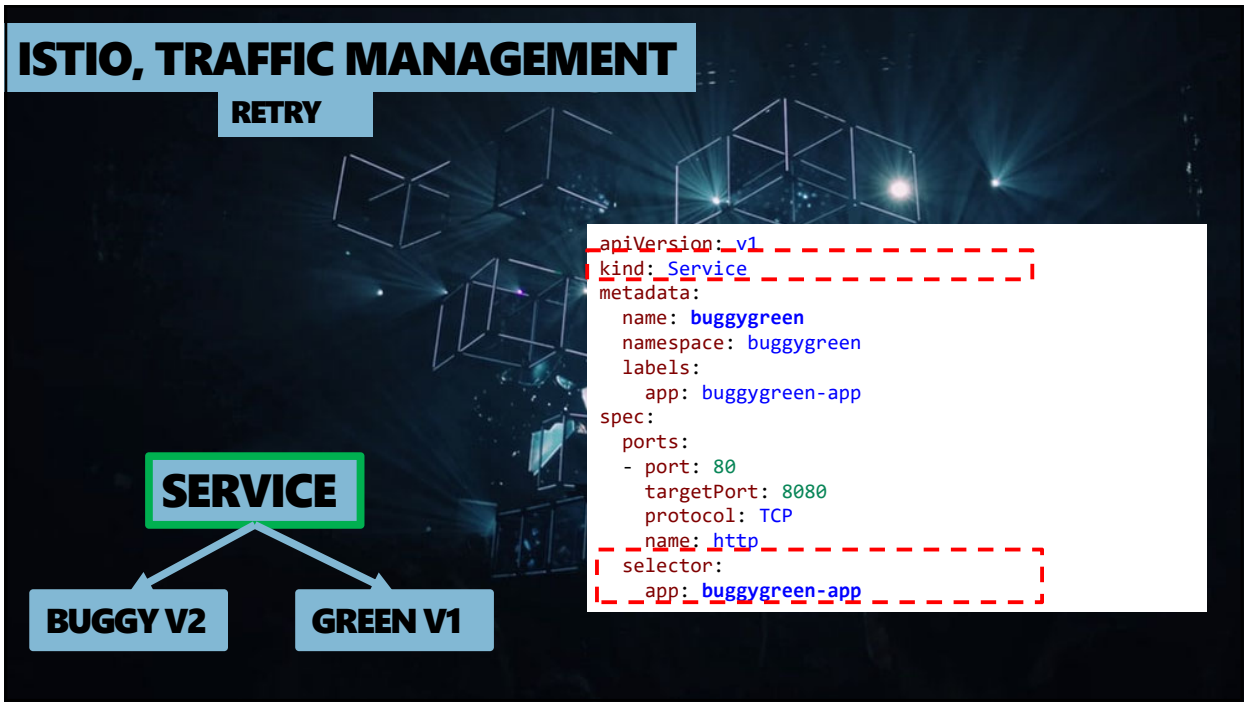

TRAFFIC SHIFT & CANARY

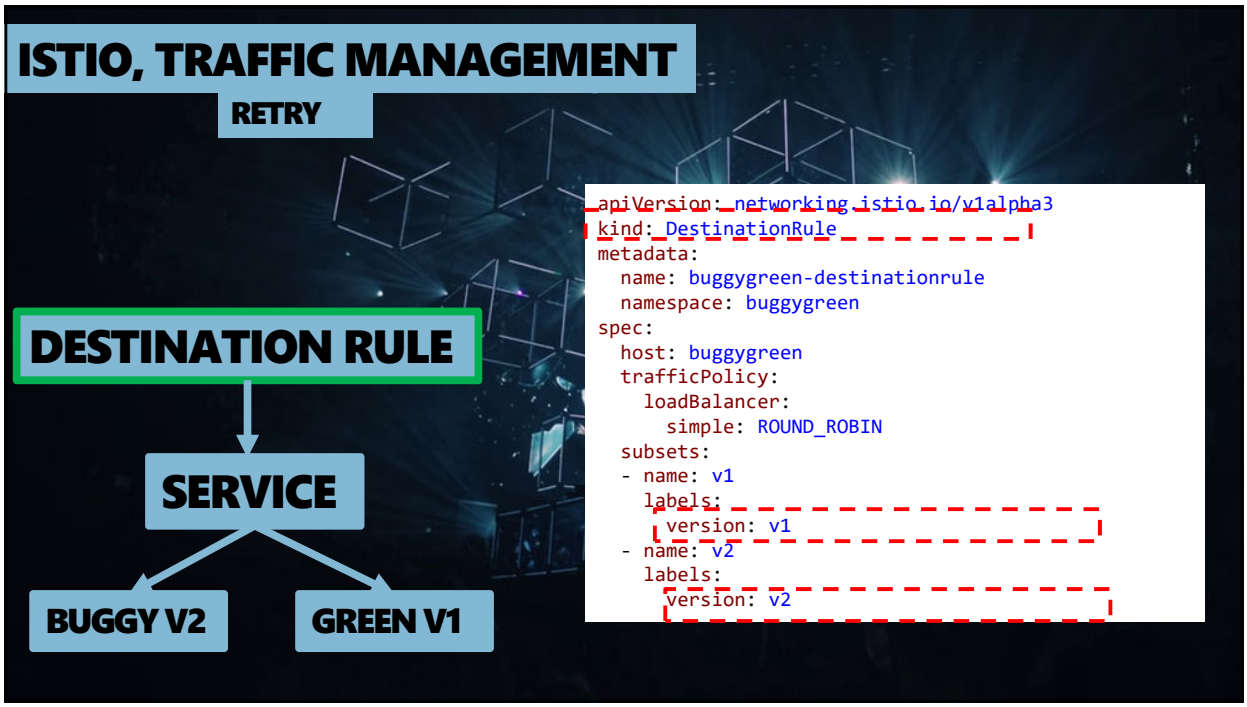

ISTIO, TRAFFIC MANAGEMENT
NETWORK RESILIENCE AND TESTING
TIMEOUTS
RETRIES
CIRCUIT BREAKERS
FAULT INJECTION

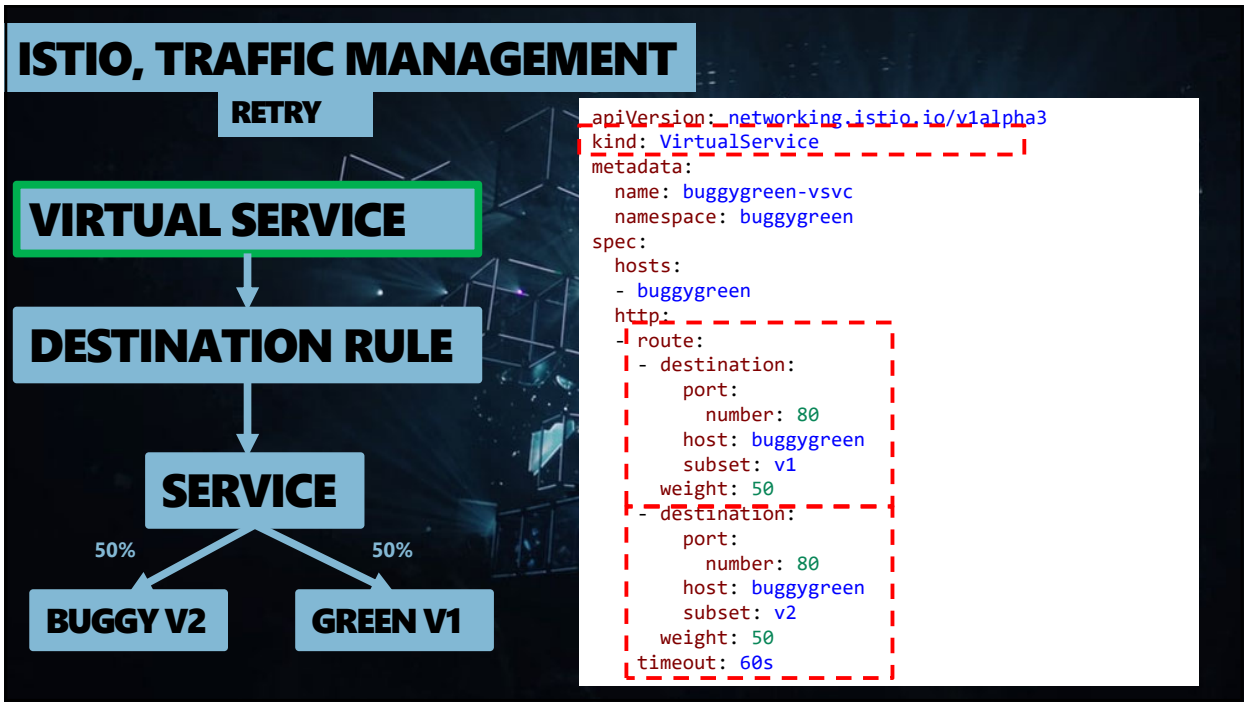

ISTIO, TRAFFIC MANAGEMENT
RETRY
apiVersion: v1
kind: Namespace
metadata:
name: buggygreen
labels:
istio-injection: enabled

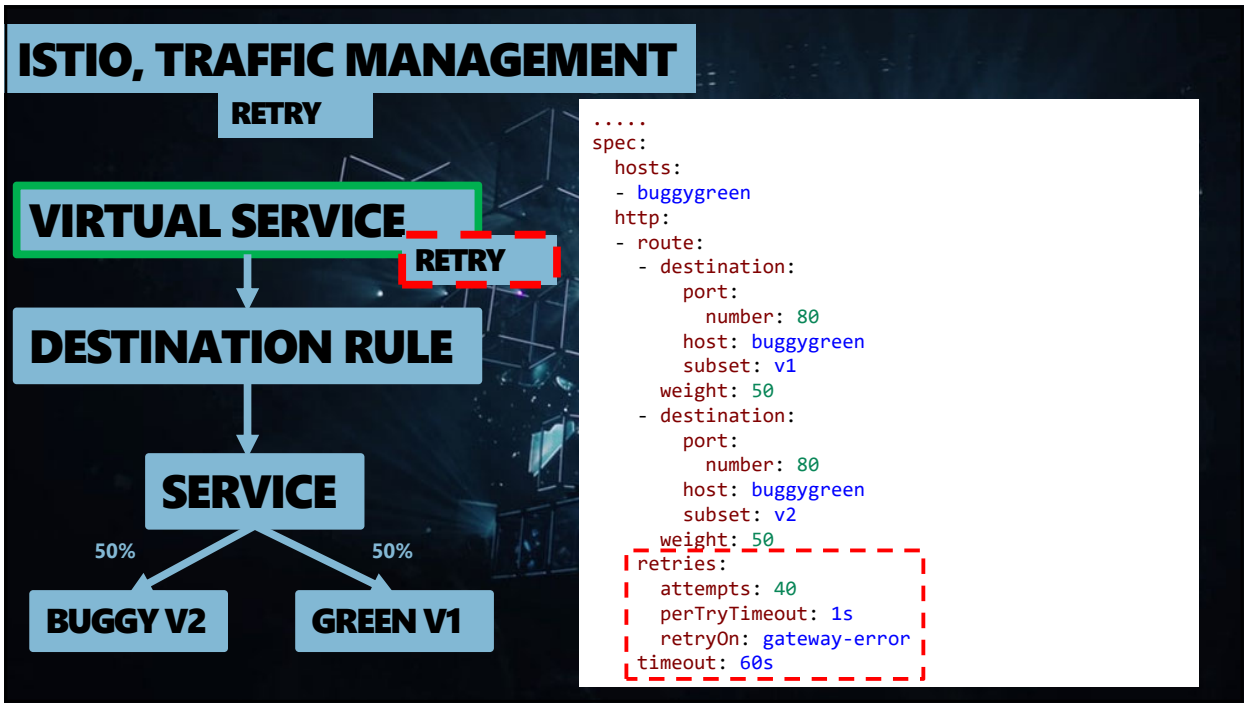

ISTIO, TRAFFIC MANAGEMENT
RETRY
BUGGY V2
GREEN V1
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
name: buggy
namespace: buggygreen
spec:
replicas: 1
selector:
matchLabels:
app: buggygreen-app
version: v2
template:
metadata:
labels:
app: buggygreen-app
version: v2
spec:
containers:
- name: 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


ISTIO, TRAFFIC MANAGEMENT
RETRY
SERVICE
BUGGY V2
GREEN V1
apiVersion: v1
kind: Service
metadata:
name: buggygreen
namespace: buggygreen
labels:
app: buggygreen-app
spec:
ports:
- port: 80
targetPort: 8080
protocol: TCP
name: http
selector:
app: buggygreen-app


ISTIO, TRAFFIC MANAGEMENT
RETRY
DESTINATION RULE
SERVICE
BUGGY V2
GREEN V1
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
name: buggygreen-destinationrule
namespace: buggygreen
spec:
host: buggygreen
trafficPolicy:
loadBalancer:
simple: ROUND_ROBIN
subsets:
- name: v1
labels:
version: v1
- name: v2
labels:
version: v2


ISTIO, TRAFFIC MANAGEMENT
RETRY
VIRTUAL SERVICE
DESTINATION RULE
SERVICE
50%
50%
BUGGY V2
GREEN V1
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
name: buggygreen-vsvc
namespace: buggygreen
spec:
hosts:
- buggygreen
http:
- route:
- destination:
port:
number: 80
host: buggygreen
subset: v1
weight: 50
- destination:
port:
number: 80
host: buggygreen
subset: v2
weight: 50
timeout: 60s


ISTIO, TRAFFIC MANAGEMENT
RETRY
VIRTUAL SERVICE
RETRY
DESTINATION RULE
SERVICE
50%
50%
BUGGY V2
GREEN V1
.....
spec:
hosts:
- buggygreen
http:
- route:
- destination:
port:
number: 80
host: buggygreen
subset: v1
weight: 50
- destination:
port:
number: 80
host: buggygreen
subset: v2
weight: 50
retries:
attempts: 40
perTryTimeout: 1s
retryOn: gateway-error
timeout: 60s

RETRY IN ACTION

SECURITY

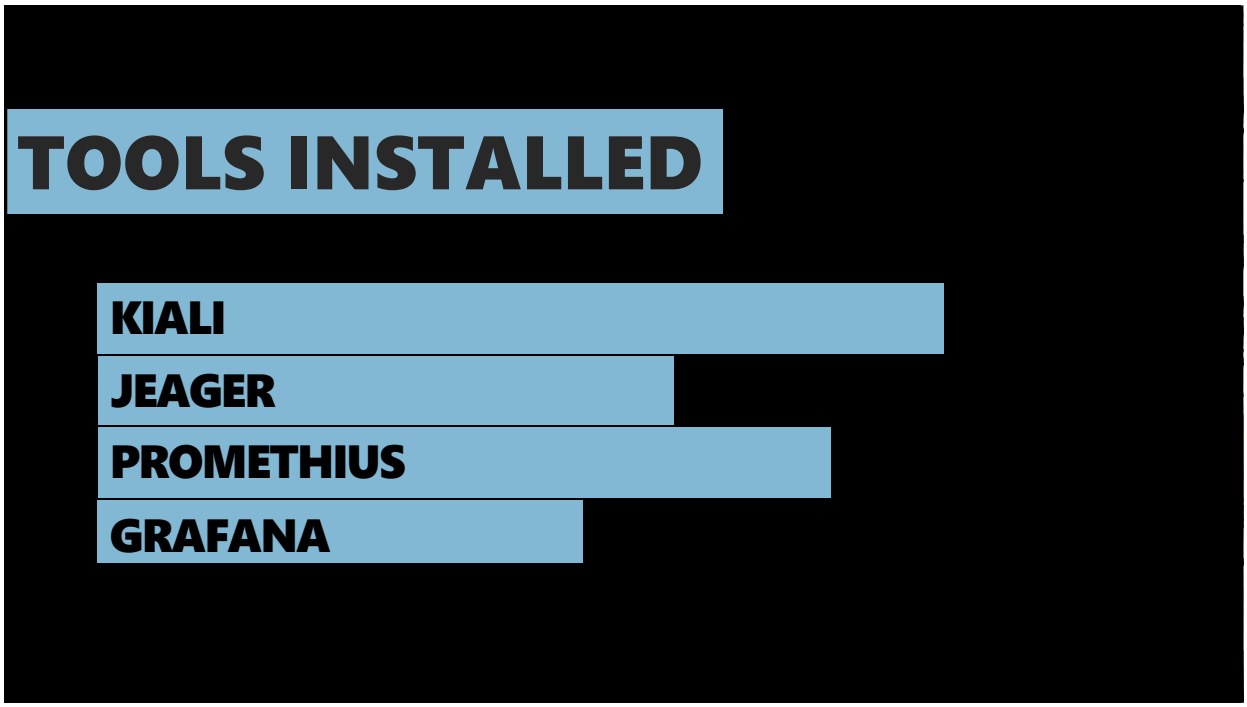
SECURE TRAFFIC BETWEEN SERVICES

MUTUAL TLS

AUTHENTICATION

AUDIT TRAIL


OBSERVE
METRICS
DISTRIBUTED TRACES
ACCESS LOGS


TOOLS INSTALLED
KIALI
JEAGER
PROMETHIUS
GRAFANA

ISTIO CLI

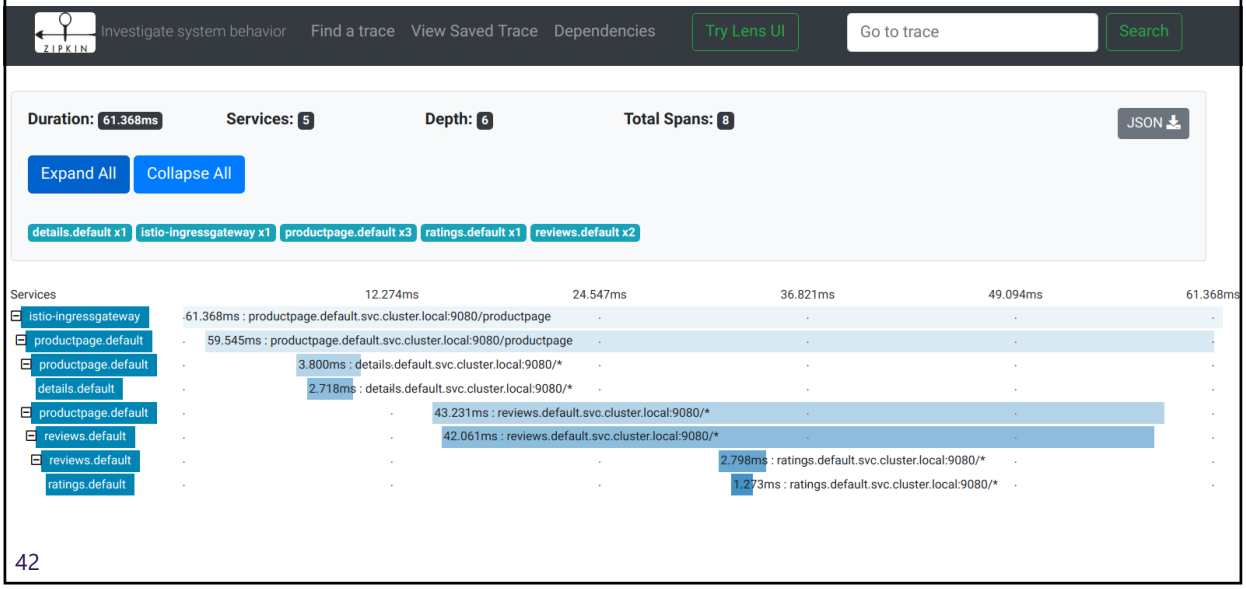
C:> istioctl dashboard kiali

C:> istioctl kube-inject

C:> istioctl register

C:> istioctl dashboard envoy

Observability



42

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 and GitHub the Big Picture
- Dapr 1 the Big Picture
- Azure DevOps the Big Picture
- Continuous Delivery and DevOps with Azure DevOps, Managing Builds
- Continuous Delivery and DevOps with Azure DevOps, Managing releases
- 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 2019
- And more …

THANK YOU!

Marcel de Vries
CEO Xpirit Group
@marcelv http://fluentbytes.com

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

Xpirit
Microsoft Regional Director
MVP Microsoft Most Valuable Professional
do epic shit