Intro to Istio

Marek Sadowski
Developer Advocates | IBM San Francisco

@blumareks

IBM **Code**

ibm.biz/201910-istio-webinar

Java – Swift – Mobile – Containers – Serverless – AI (Watson) with Visual Recognition, Voice UI, NLU, ...

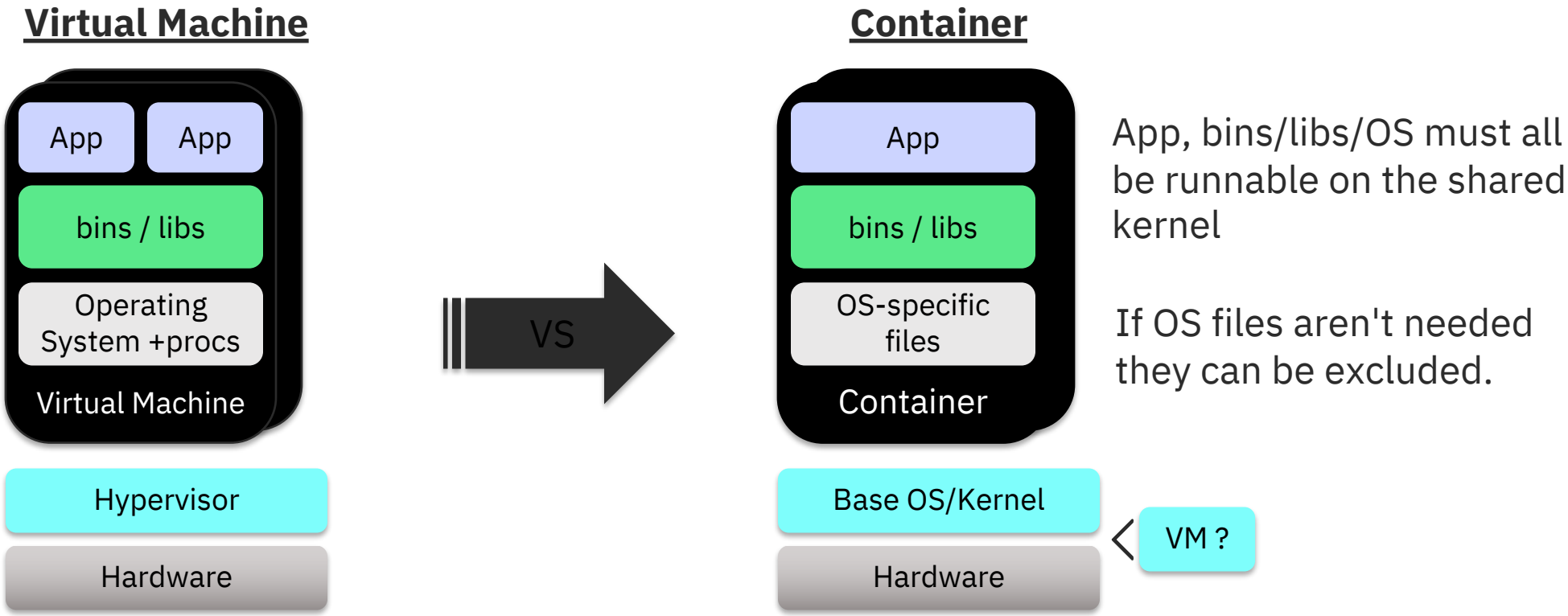
Rescue Scuba Diver – Snowboarder – Robotics

Agenda

- Containers
  - Docker
  - Kubernetes
- Istio
- Workshop
- Wrap-up
- Q&A

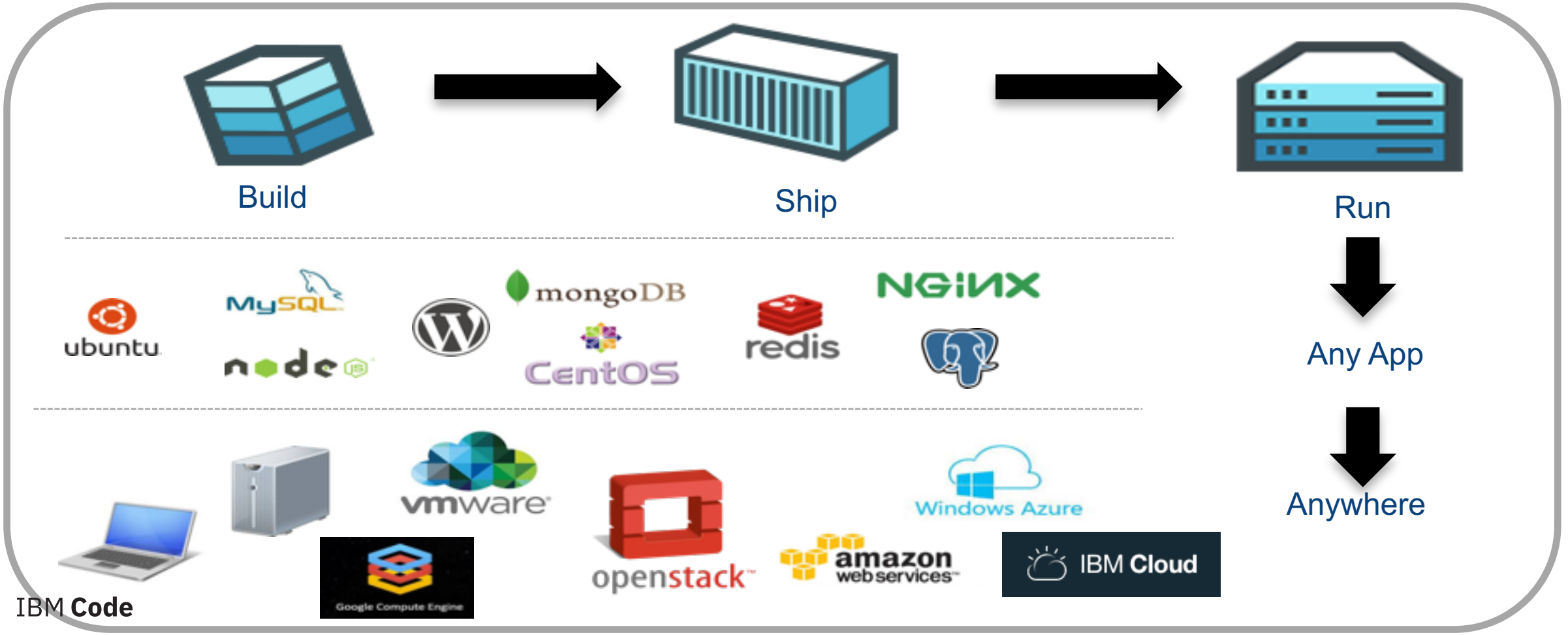
ibm.biz/201910-istio-webinar
github.com/blumareks/istio-2019

VM vs Container



Docker Mission

Docker is an **open platform** for building distributed applications for **developers** and **system administrators**



Dev vs. Ops

Dev	Ops

- Code
- Libraries
- Configuration
- Server runtime
- OS

- Logging
- Remote access
- Network configuration
- Monitoring

Separation of concerns

A container separates and bridges the Dev and Ops in DevOps

- Dev focuses on the application environment
- Ops focuses on the deployment environment

Docker Component Overview

Docker Engine

- Manages containers on a host
- Accepts requests from clients
  - REST API
- Maps container ports to host ports
  - E.g. 80 → 3582

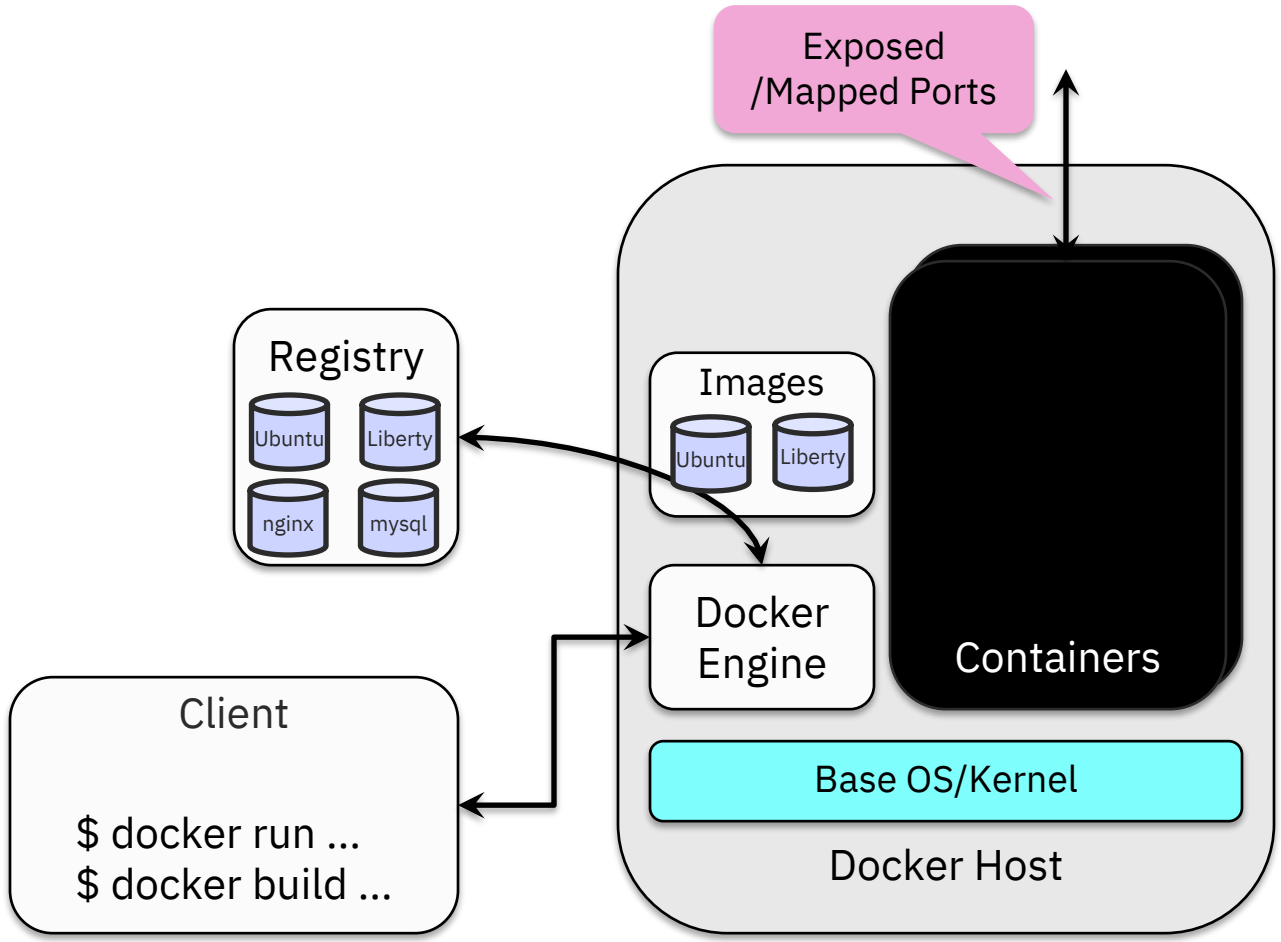
Images

Docker Client

- Drives engine
- Drives "builder" of Images

Docker Registry

- Image DB



Containers

Everyone’s container journey starts with one container....

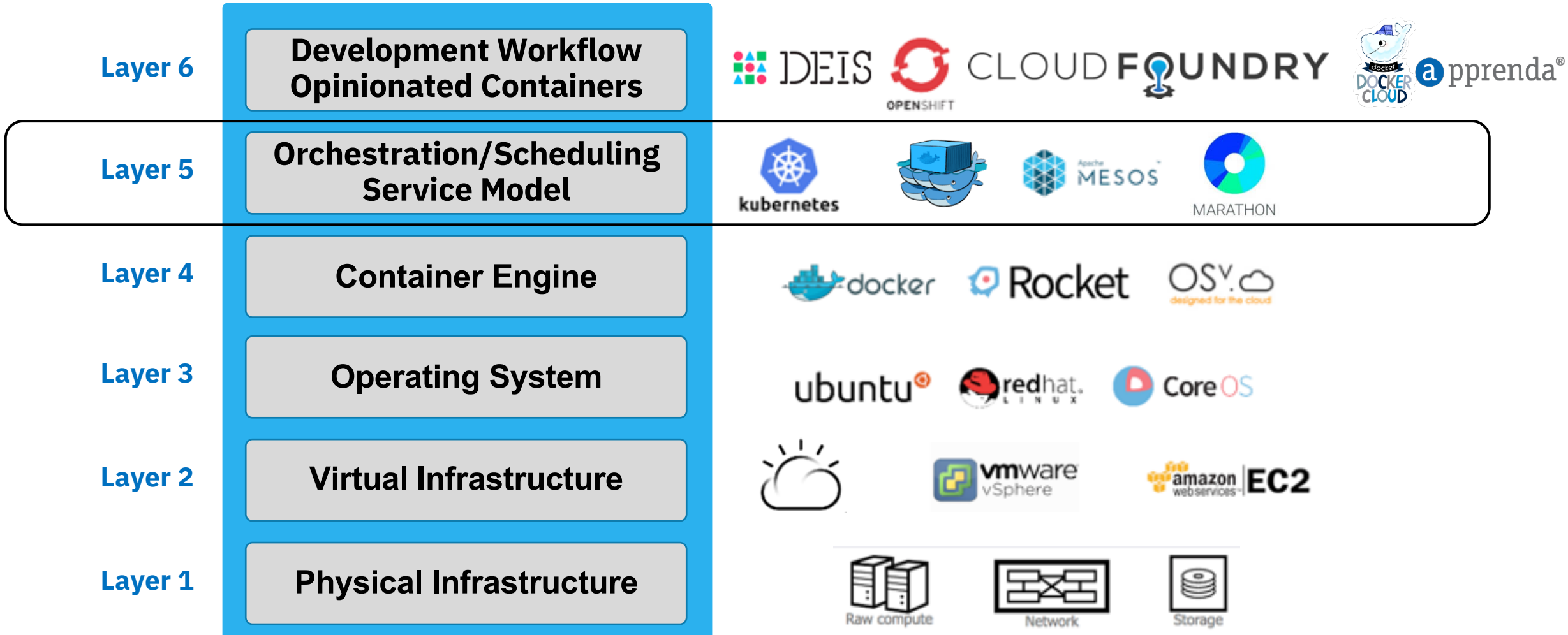
Containers

At first the growth is easy to handle....

Containers

But soon it is overwhelming... chaos reigns

Container Ecosystem Layers

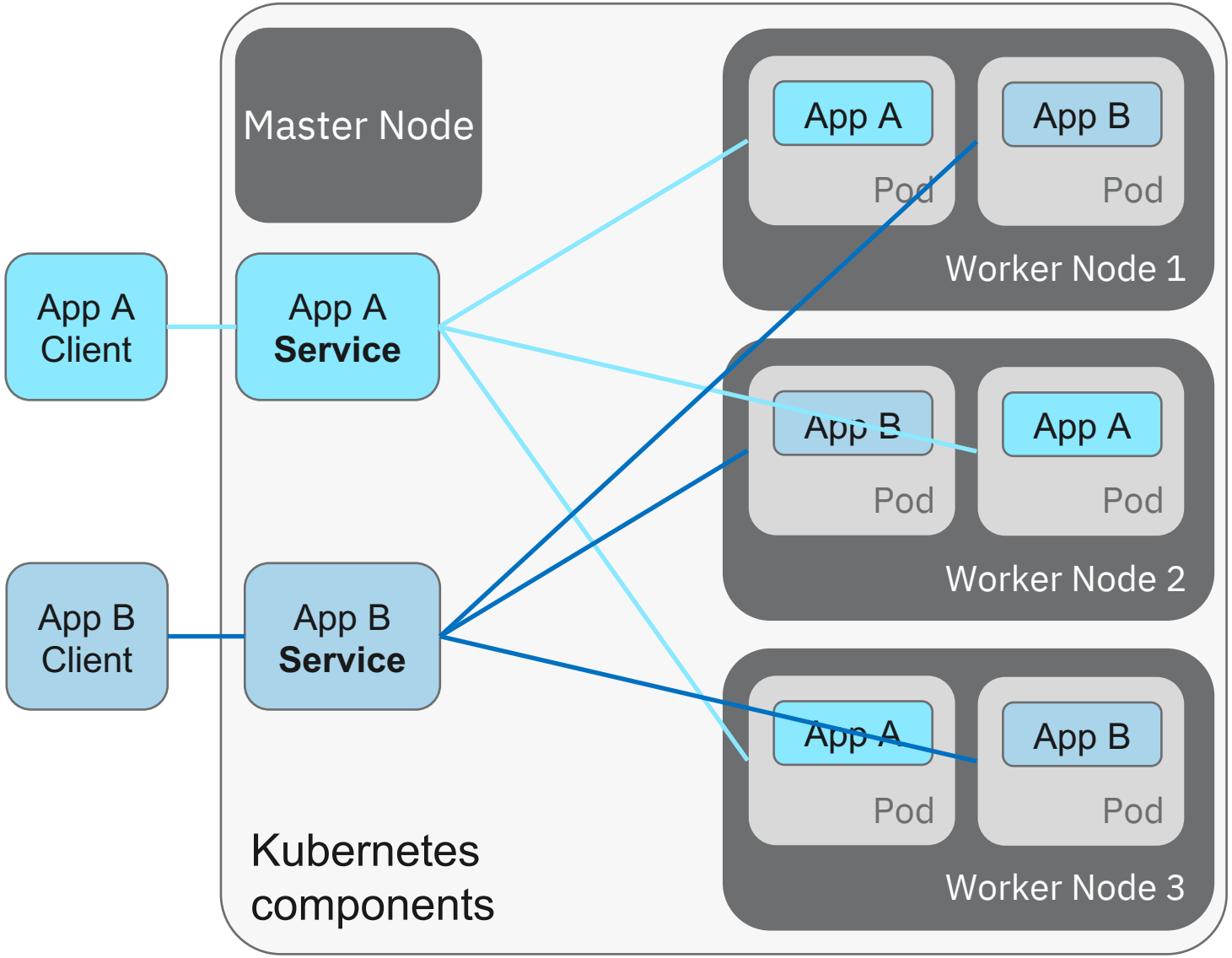


What is Kubernetes?

- Container orchestrator
- Manage applications, **not machines**
- Designed for extensibility
- Open source project managed by the Linux Foundation

Kubernetes Architecture: Workloads

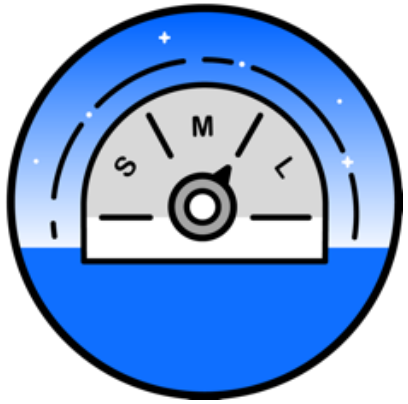
- Container
  - Packaging of an app
- Pod
  - Unit of deployment
- Service
  - Fixed endpoint for 1+ pods



Kubernetes

Intelligent Scheduling

Self-healing



Horizontal scaling

Service discovery & load balancing

Automated rollouts and rollbacks

Secret and configuration management

Helm & Helm Chart

- Helm
  - Package manager for Kubernetes
  - Used to manage Kubernetes applications
- Helm Chart
  - Used to define, install, and upgrade complex Kubernetes applications
  - Easy to create, version, share and publish
  - Expressed in “Yet Another Markup Language” (YAML) files

Challenges with Microservices

Security

Canary deployments

A/B testing

Retries and Circuit breaking

Rate limiting

Fault injection

Policy management

Telemetry

Istio

An open platform to connect, manage, and secure microservices

IBM Code

http://istio.io

ibm.biz/20190823-html5

Istio community partners

- RedHat
- Pivotal
- WeaveWorks
- Tigera
- Datawire
- Scytale (SPIFFE)
- Microsoft
- Uber (Jaeger)

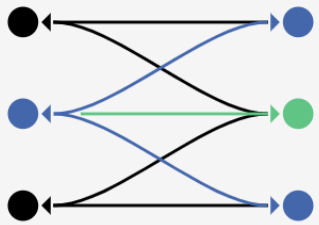
What is a ‘Service Mesh’ ?

A network for services, not bytes

- Observability
- Resiliency
- Traffic Control
- Security
- Policy Enforcement

Istio

Connect, secure, control, and observe services.

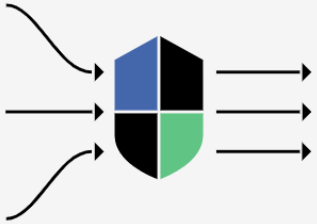


Connect

Intelligently control the flow of traffic and API calls between services, conduct a range of tests, and upgrade gradually with red/black deployments.

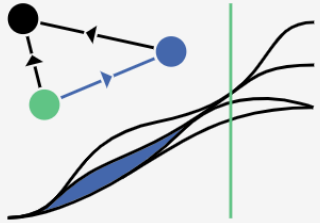
Secure

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.



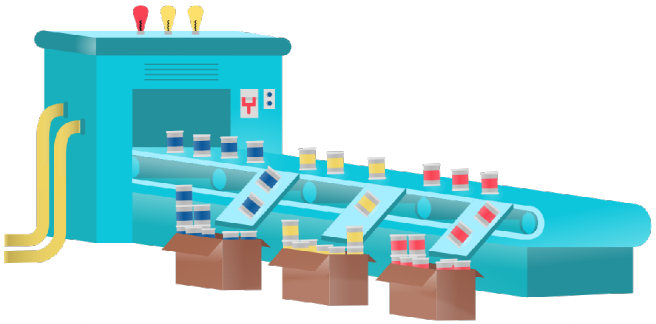
Observe

See what's happening with rich automatic tracing, monitoring, and logging of all your services.

Istio

WHERE ISTIO FINDS ITS PLACE

Intelligent Routing and Load Balancing



Fleet Wide Policy Enforcement

Resiliency across Languages and Platforms

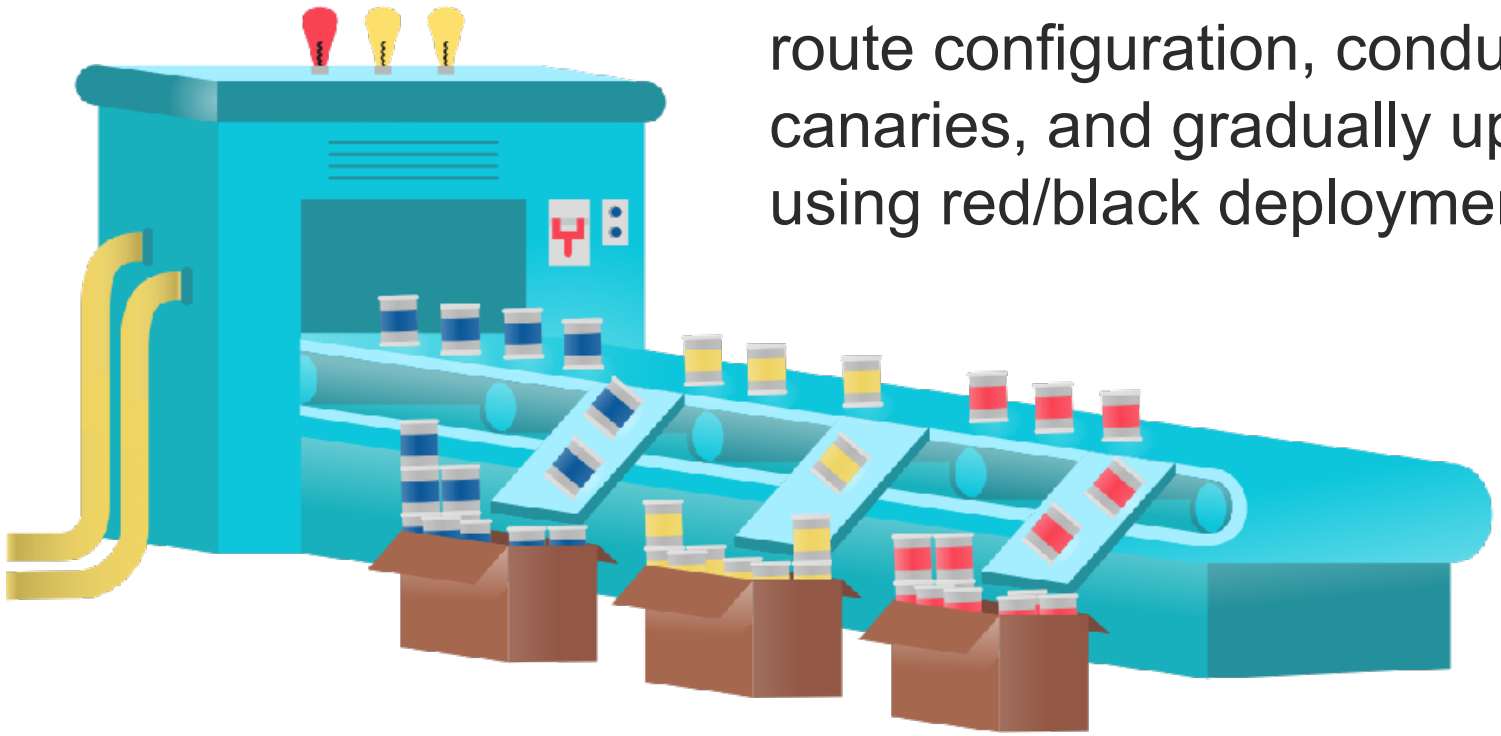
In-Depth Telemetry and Reporting

IBM **Code**

Istio

WHERE ISTIO FINDS ITS PLACE

Intelligent Routing and Load Balancing

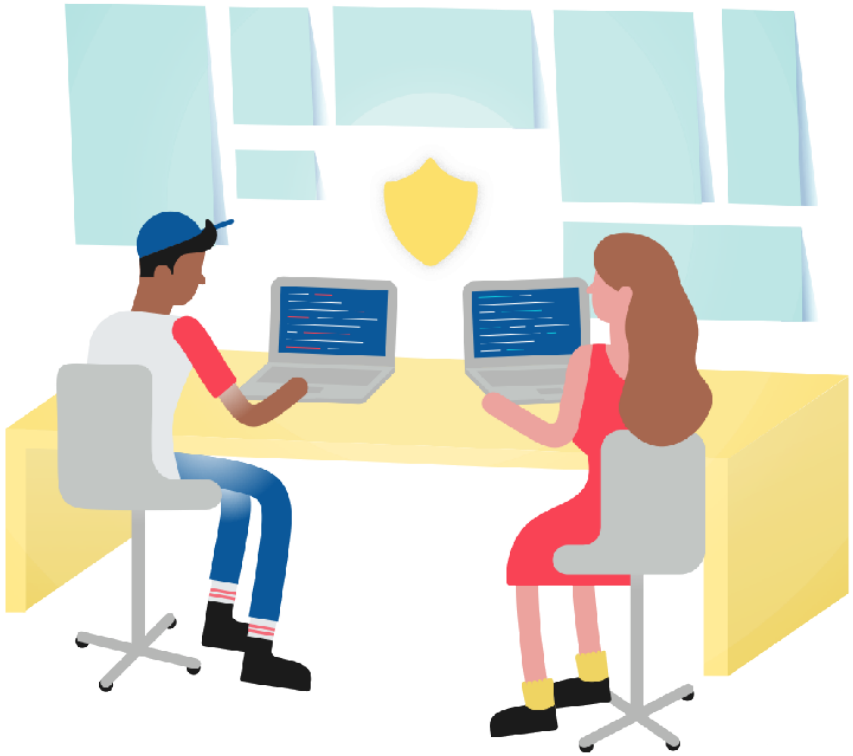


Control traffic between services with dynamic route configuration, conduct A/B tests, release canaries, and gradually upgrade versions using red/black deployments.

Istio

WHERE ISTIO FINDS ITS PLACE

Resiliency across Languages and Platforms



Increase reliability by shielding applications from flaky networks and cascading failures in adverse conditions.

Istio

WHERE ISTIO FINDS ITS PLACE

Fleet Wide Policy Enforcement

Apply organizational policy to the interaction between services, ensure access policies are enforced and resources are fairly distributed among consumers.

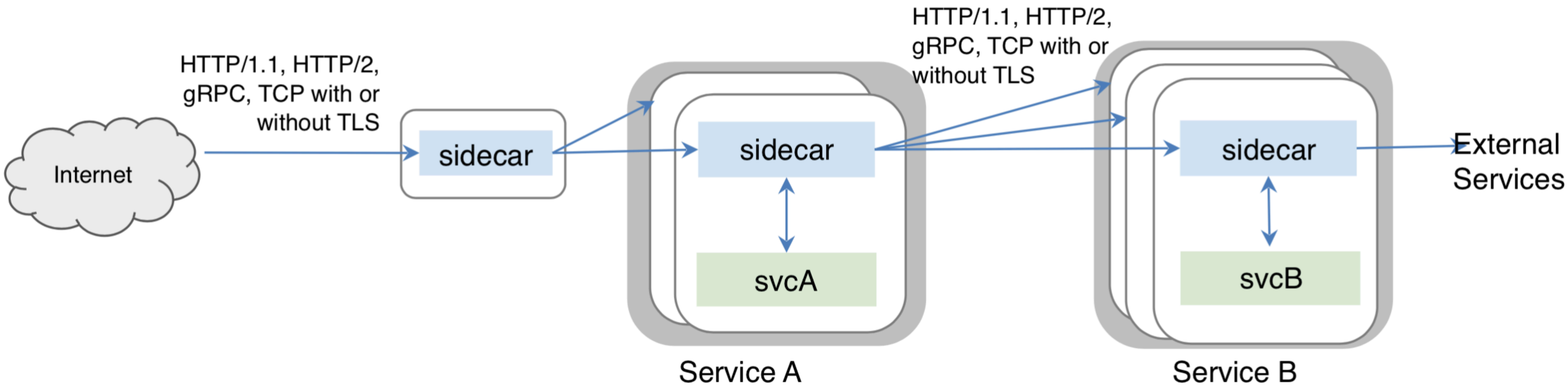
Istio

WHERE ISTIO FINDS ITS PLACE

In-Depth Telemetry and Reporting

Understand the dependencies between services, the nature and flow of traffic between them and quickly identify issues with distributed tracing.

Istio – original concept - Sidecar



Outbound features:

- Service authentication
- Load balancing
- Retry and circuit breaker
- Fine-grained routing
- Telemetry
- Request Tracing
- Fault Injection

Inbound features:

- Service authentication
- Authorization
- Rate limits
- Load shedding
- Telemetry
- Request Tracing
- Fault Injection

IBM **Code**

Envoy by Lyft – sidecar of choice

- A C++ based L4/L7 proxy
- Low memory footprint
- Battle-tested @ Lyft
  - 100+ services
  - 10,000+ VMs
  - 2M req/s

Goodies:

- HTTP/2 & gRPC
- Zone-aware load balancing w/ failover
- Health checks, circuit breakers, timeouts, retry
- budgets
- No hot reloads - API driven config updates

Istio's contributions:

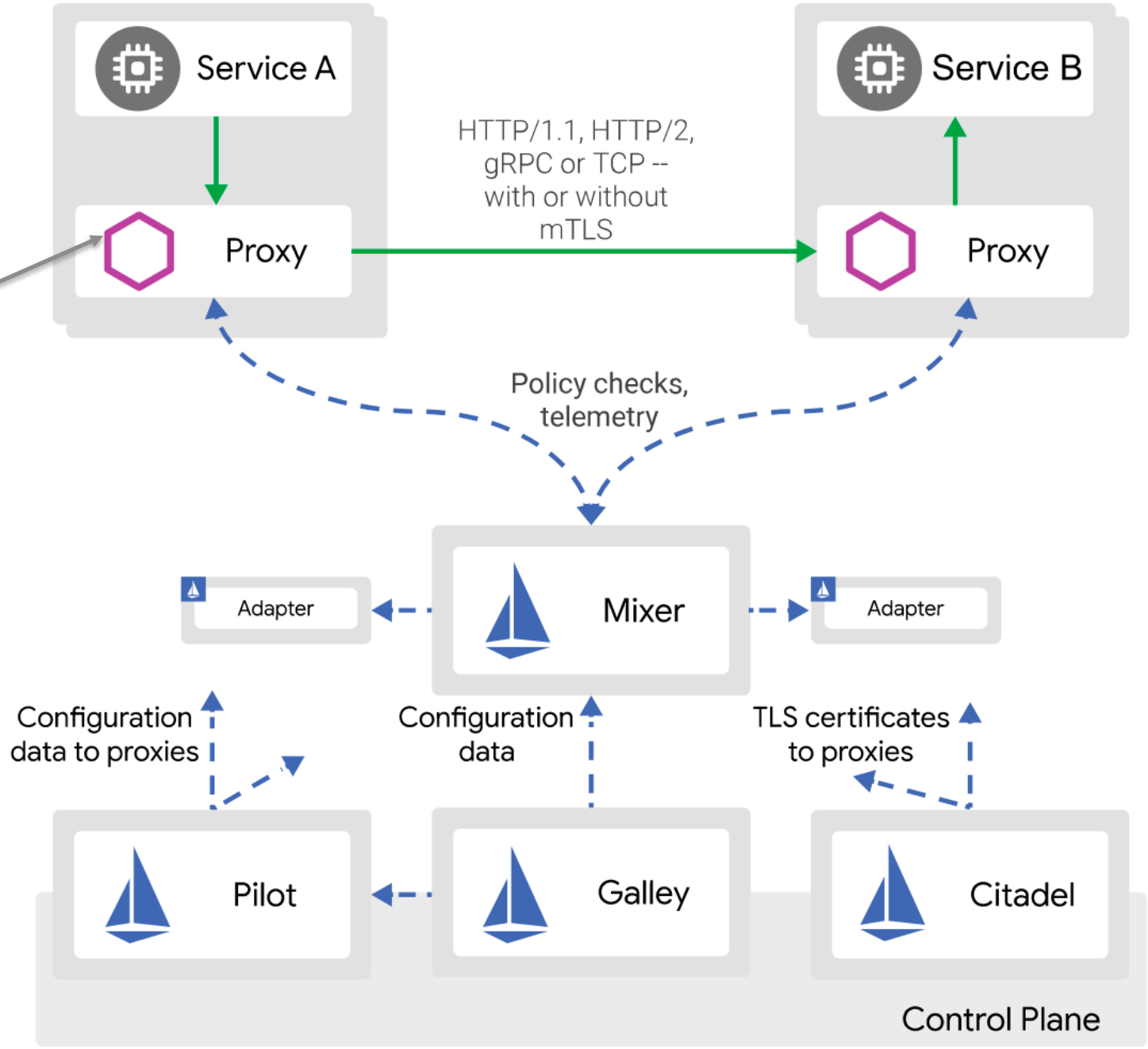
- Transparent proxying w/ SO_ORIGINAL_DST
- Traffic routing and splitting
- Request tracing using Zipkin
- Fault injection

Lyft's Envoy: From monolith to service mesh
Matt Klein, Software Engineer @Lyft

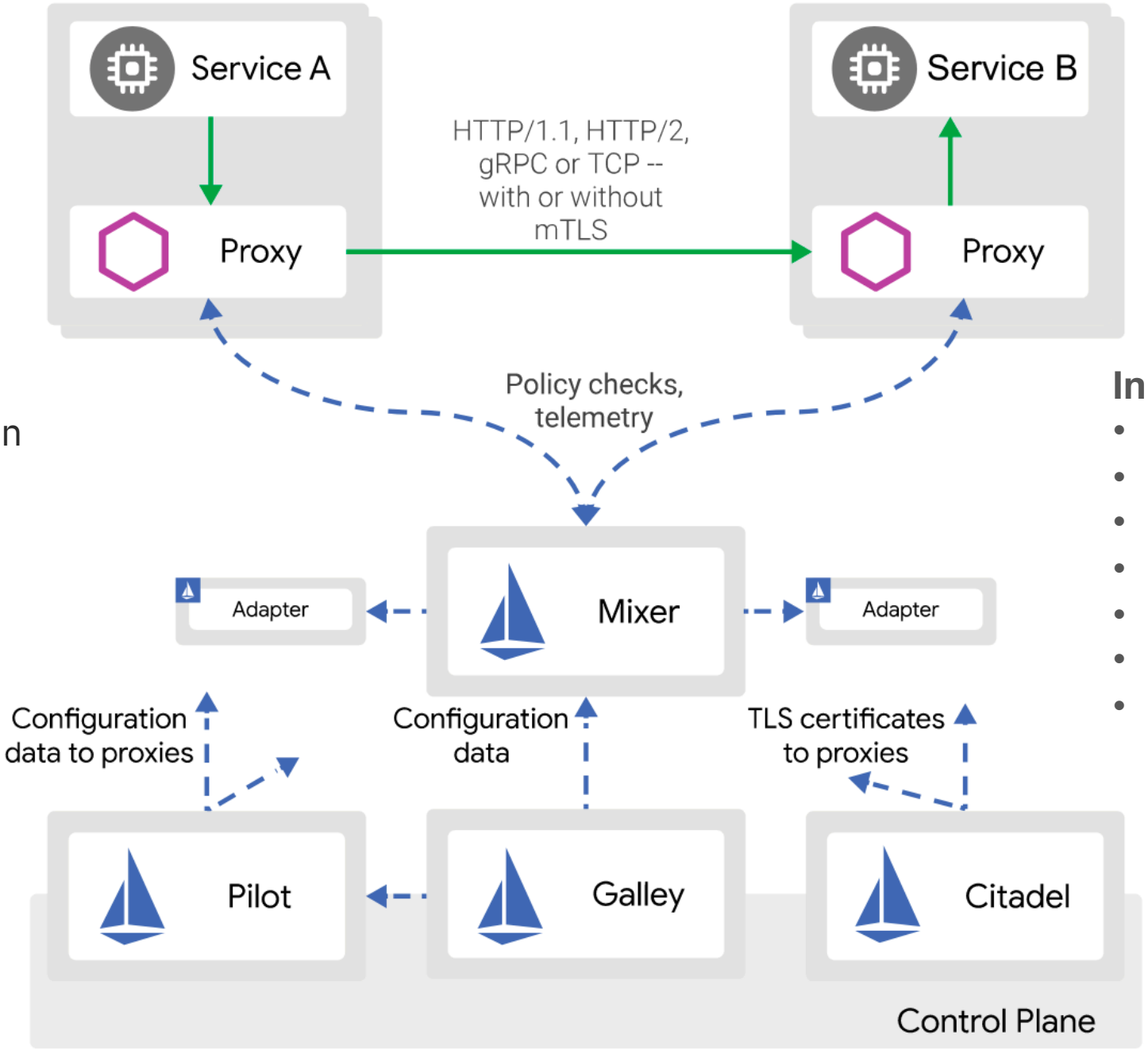
https://www.youtube.com/watch?v=RVZX4CwKhGE

Istio Architecture

Traffic is transparently intercepted and proxied. An App is unaware of Envoy's presence



Istio Architecture



Outbound features:

- Service authentication
- Load balancing
- Retry and circuit breaker
- Fine-grained routing
- Telemetry
- Request Tracing
- Fault Injection

Inbound features:

- Service authentication
- Authorization
- Rate limits
- Load shedding
- Telemetry
- Request Tracing
- Fault Injection

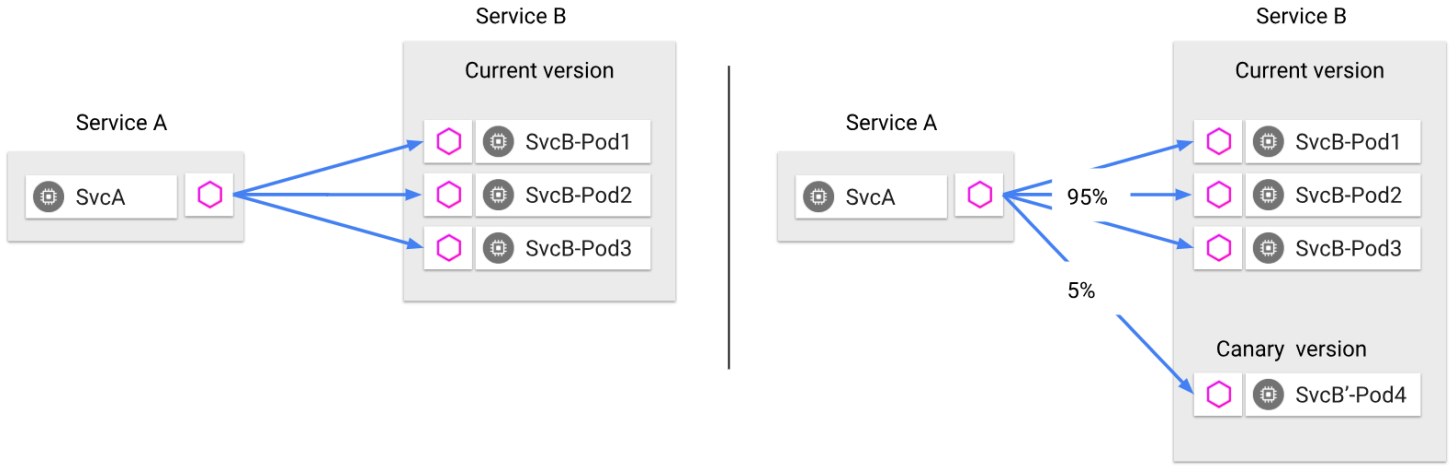
Istio Traffic Management Overview

The traffic management model decouples traffic flow and infrastructure scaling giving you the option of specifying via rules and Pilot how traffic should flow

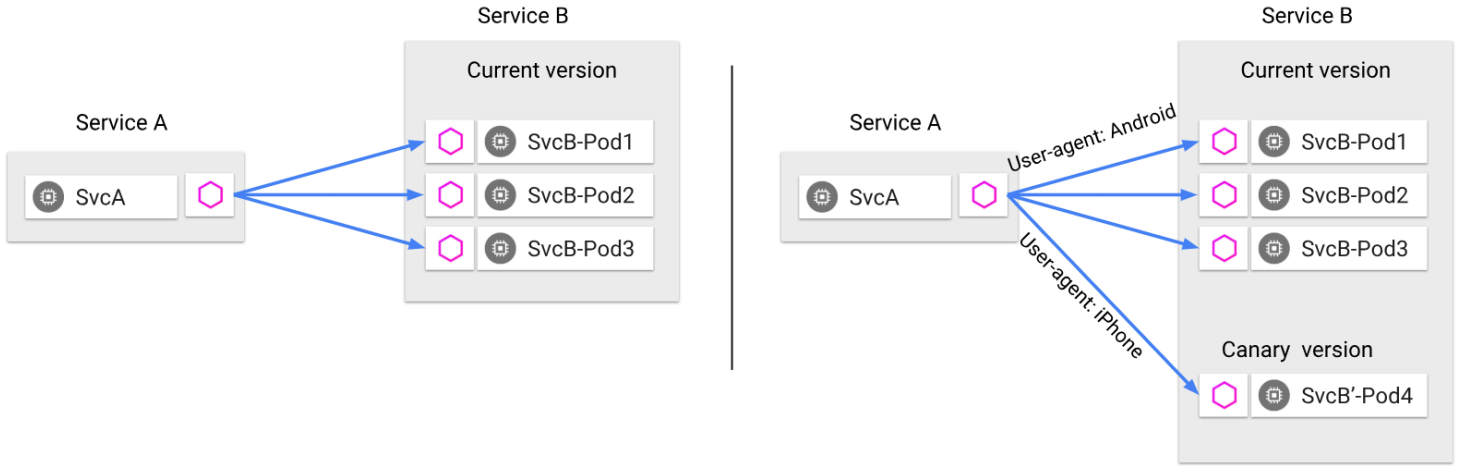
For example, you can direct a percentage of traffic for a particular service to a canary service or only direct to the canary based upon the content of the request

Decoupling traffic flow from scaling of infrastructure allows for traffic management features outside of the application code including failure recovery via timeouts, retries, circuit breakers and fault injection to test failure recovery procedures

Traffic splitting decoupled from infrastructure scaling



Content based traffic steering



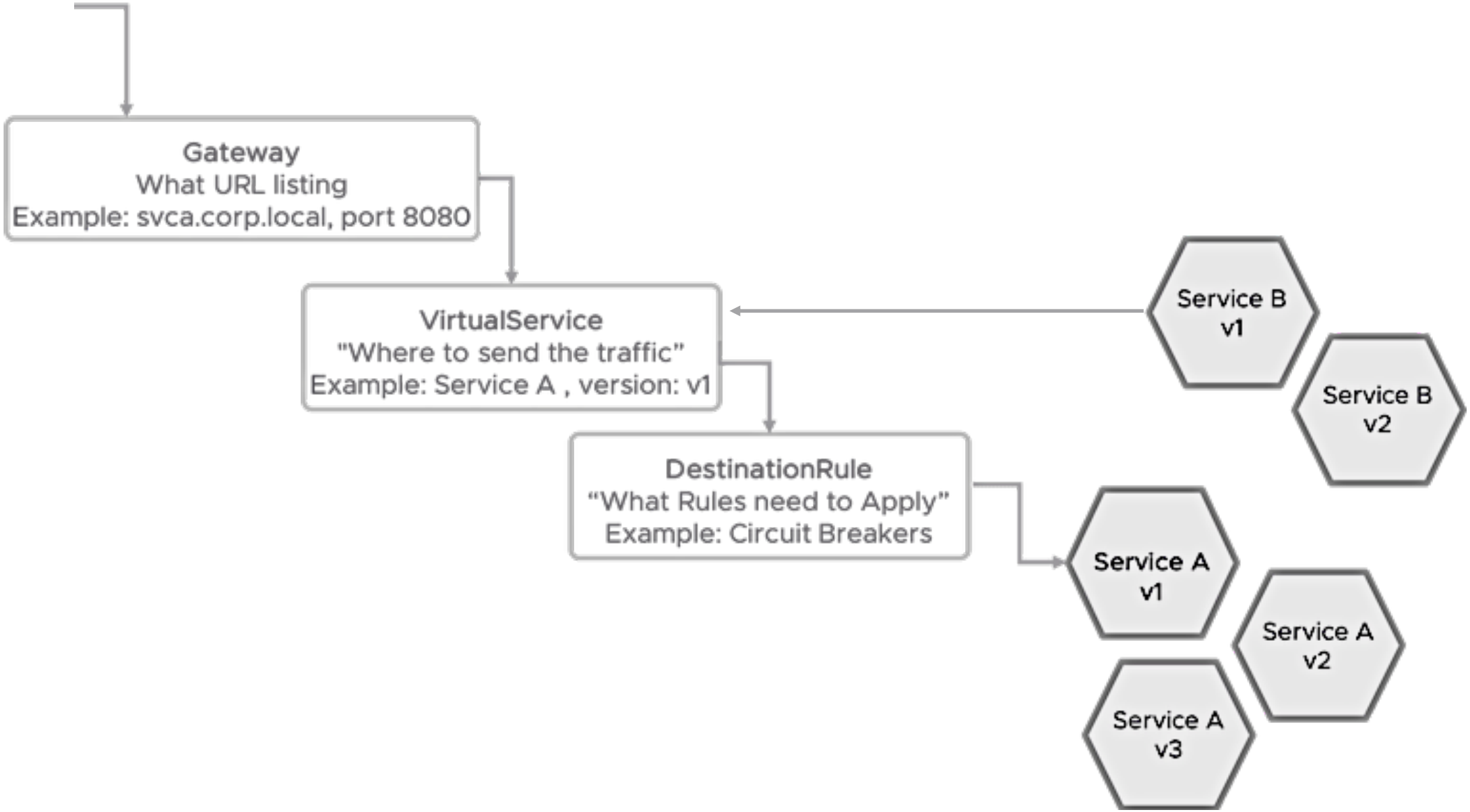
Rule Configuration

Istio provides a simple configuration model to control how API calls and layer-4 traffic flow across various services in an application deployment

The configuration model allows you to configure service-level properties such as circuit breakers, timeouts, and retries, as well as set up common continuous deployment tasks such as canary rollouts, A/B testing, staged rollouts with %-based traffic splits, etc.

There are three traffic management configuration resources in Istio
Gateway , VirtualService, and **DestinationRule** :

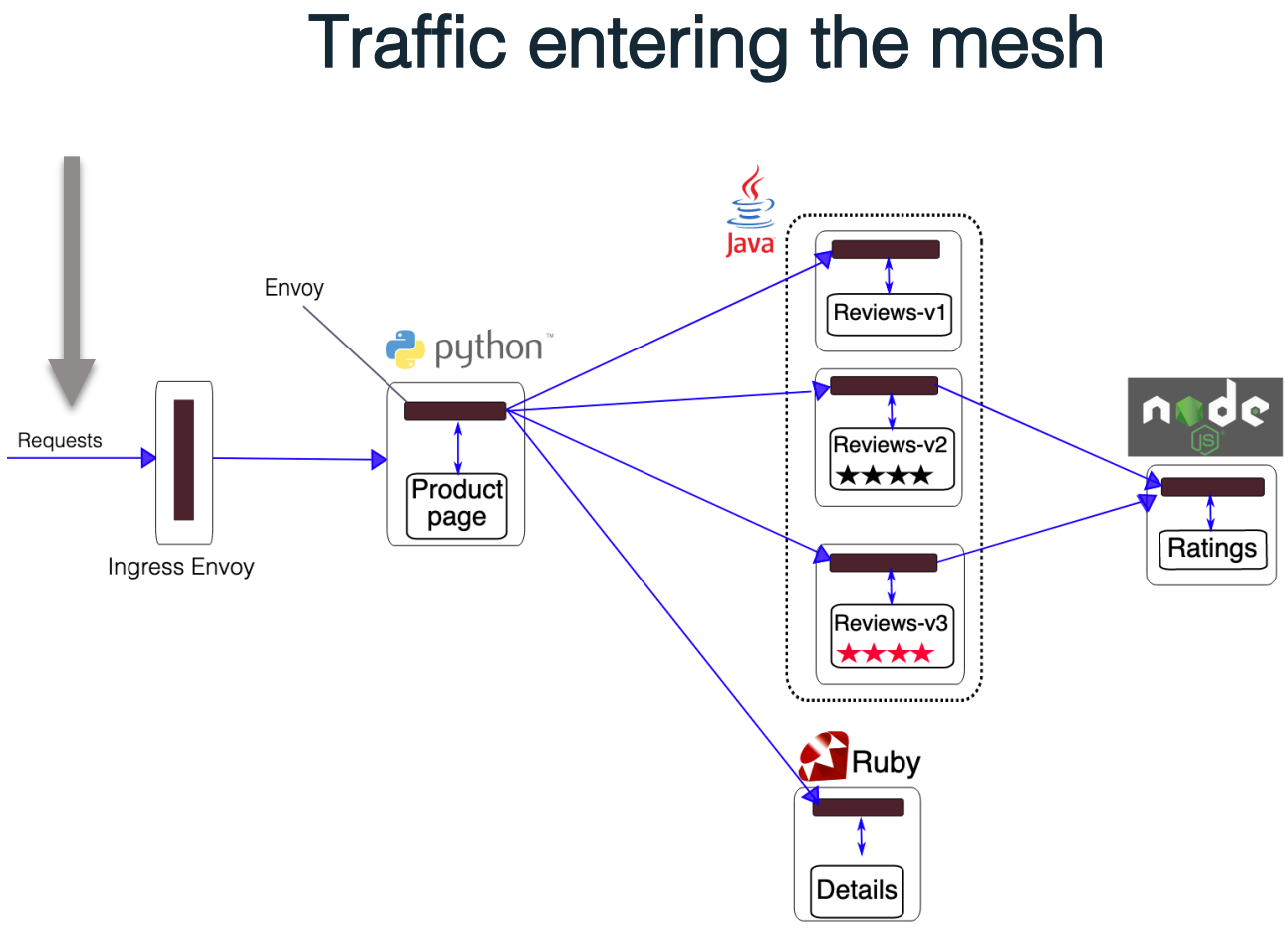
- A Gateway configures a load balancer for HTTP/TCP traffic, most commonly operating at the edge of the mesh to enable ingress traffic for an application
- A VirtualService defines the rules that control how requests for a service are routed within an Istio service mesh
- A DestinationRule configures the set of policies to be applied to a request after VirtualService routing has occurred



http://www.routetocloud.com/2019/04/introduction-to-istio-service-mesh/

Gateways (Ingress)

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

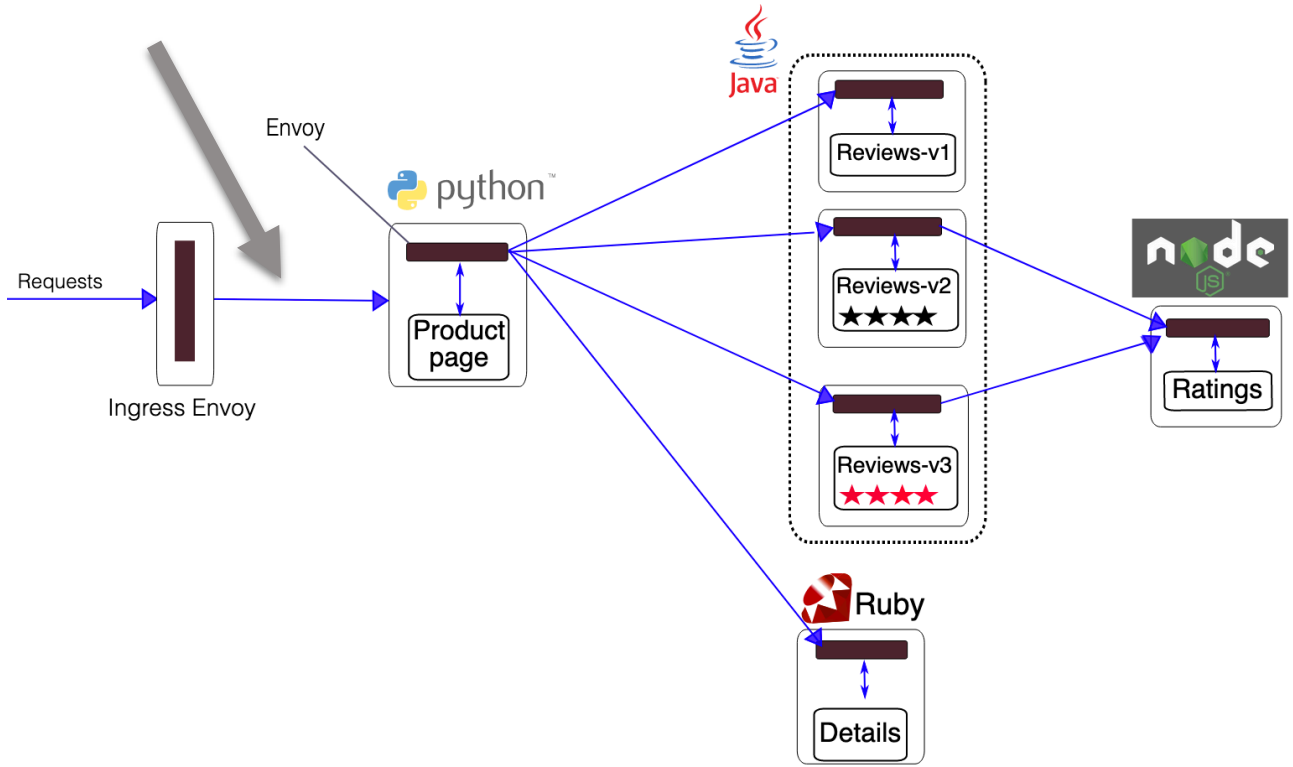


VirtualService – match conditions from Gateway

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: bookinfo
spec:
  hosts:
  - "*"
  gateways:
  - bookinfo-gateway
  http:
  - match:
    - uri:
        exact: /productpage
    - uri:
        prefix: /static
    - uri:
        exact: /login
    - uri:
        exact: /logout
    - uri:
        prefix: /api/v1/products
    route:
    - destination:
        host: productpage
        port:
          number: 9080
```

Bind to a gateway

Apply routing rules



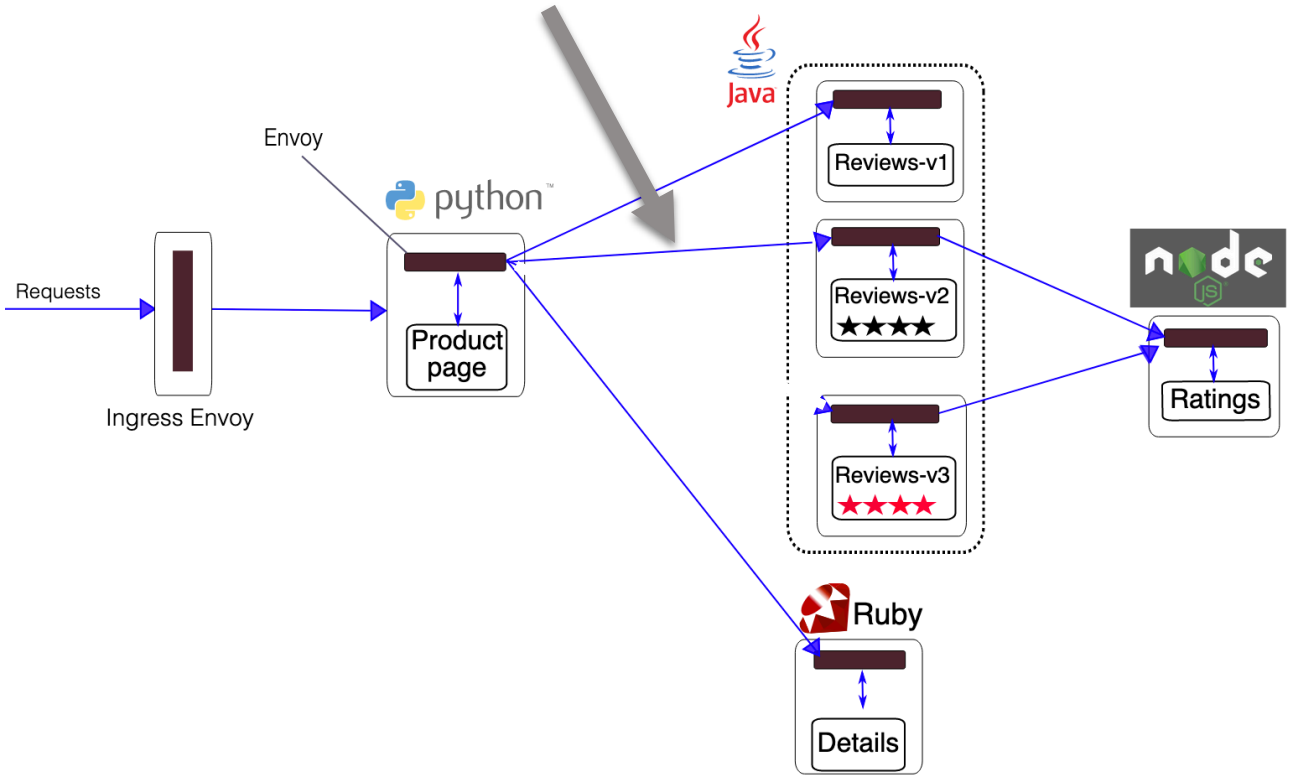
Virtual Service - Request Routing

```yaml
apiVersion: networking.istio.io/v1alpha
kind: VirtualService
metadata:
  name: reviews
spec:
  hosts:
    - reviews
  http:
  - match:
    - headers:
        end-user:
          exact: jason
    route:
    - destination:
        host: reviews
        subset: v2
  - route:
    - destination:
        host: reviews
        subset: v1
```

Canary Testing:

Route user:jason to reviews:v2

Others still get reviews:v1

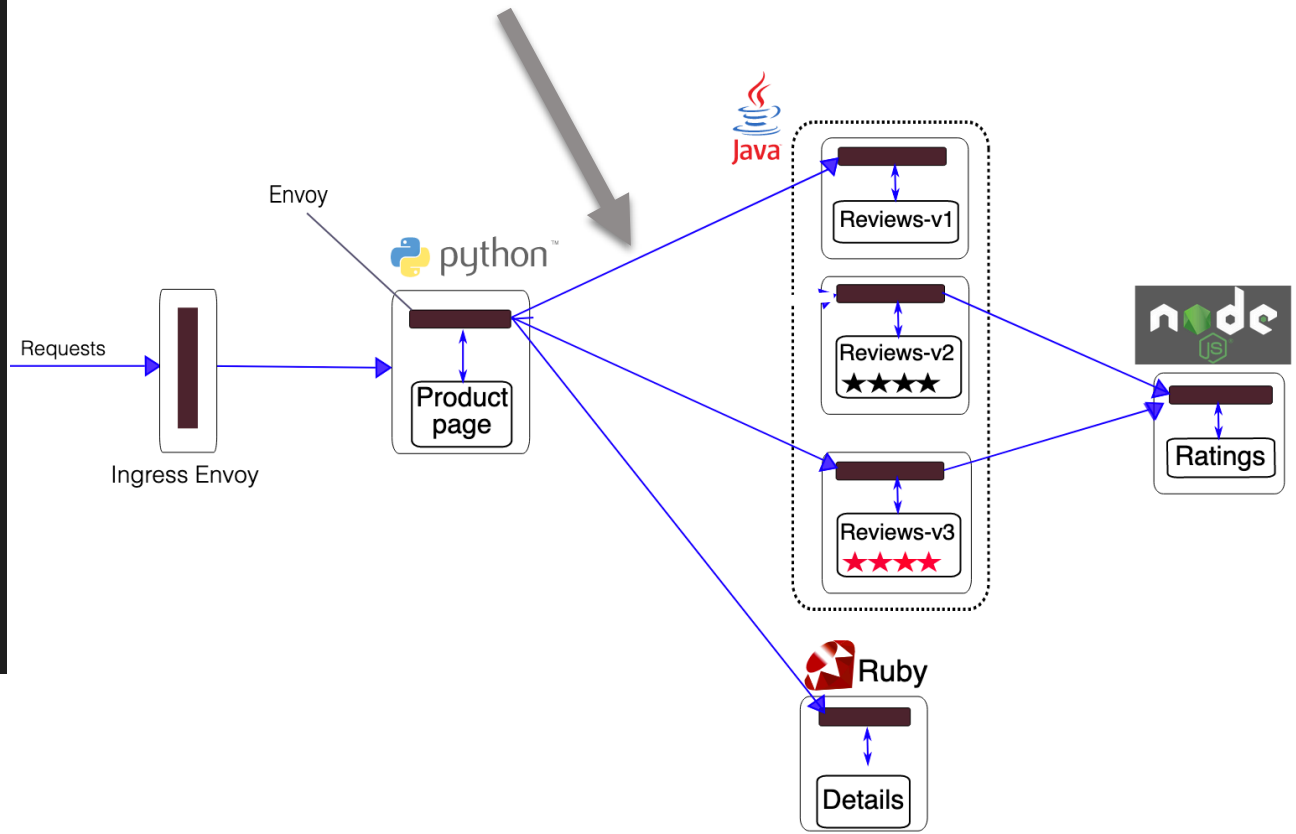


Virtual Service - Traffic Shifting

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: reviews
spec:
  hosts:
    - reviews
  http:
  - route:
    - destination:
        host: reviews
        subset: v1
      weight: 95
    - destination:
        host: reviews
        subset: v3
      weight: 5
```

95% -> v1

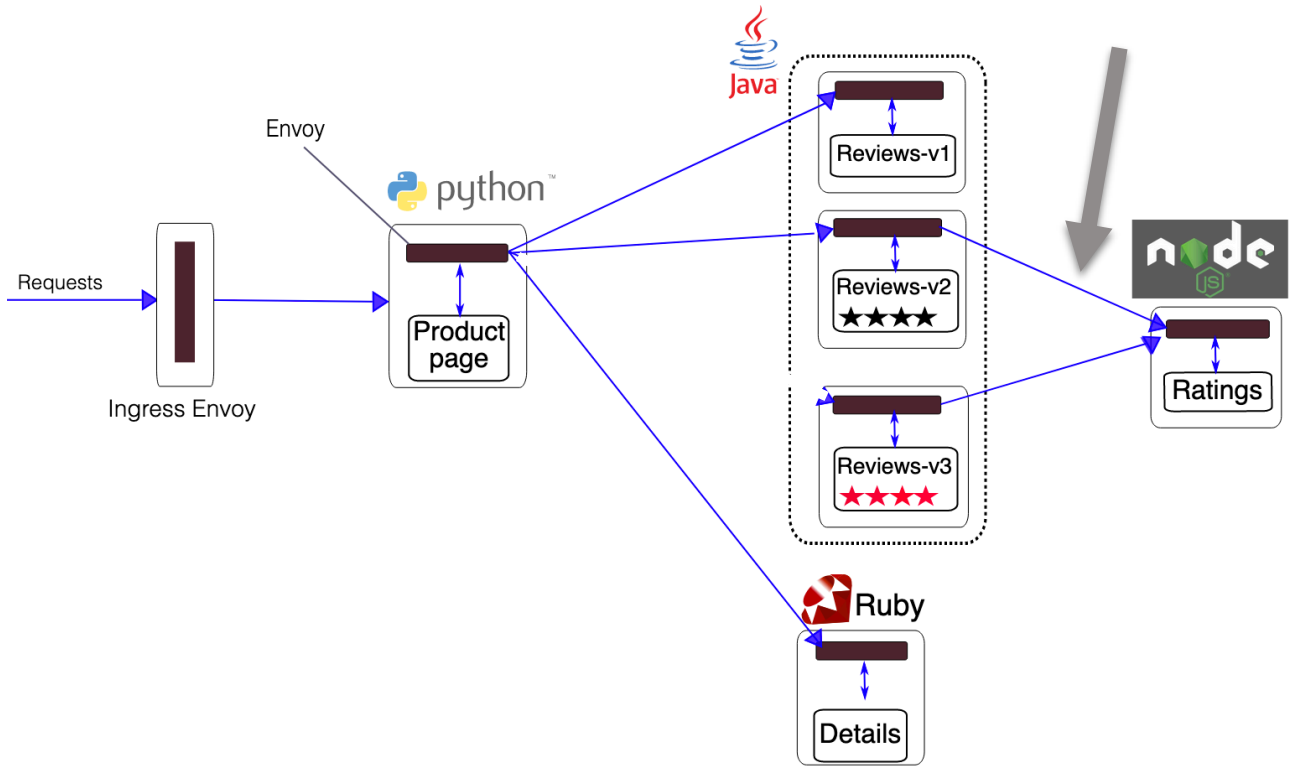
5% -> v3



Virtual Service - Delay & Fault Injection

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings
spec:
  hosts:
  - ratings
  http:
  - match:
    - headers:
        end-user:
          exact: jason
    fault:
      delay:
        percent: 100
        fixedDelay: 7s
    route:
    - destination:
        host: ratings
        subset: v1
  - route:
    - destination:
        host: ratings
        subset: v1
```

Inject 7 second delay

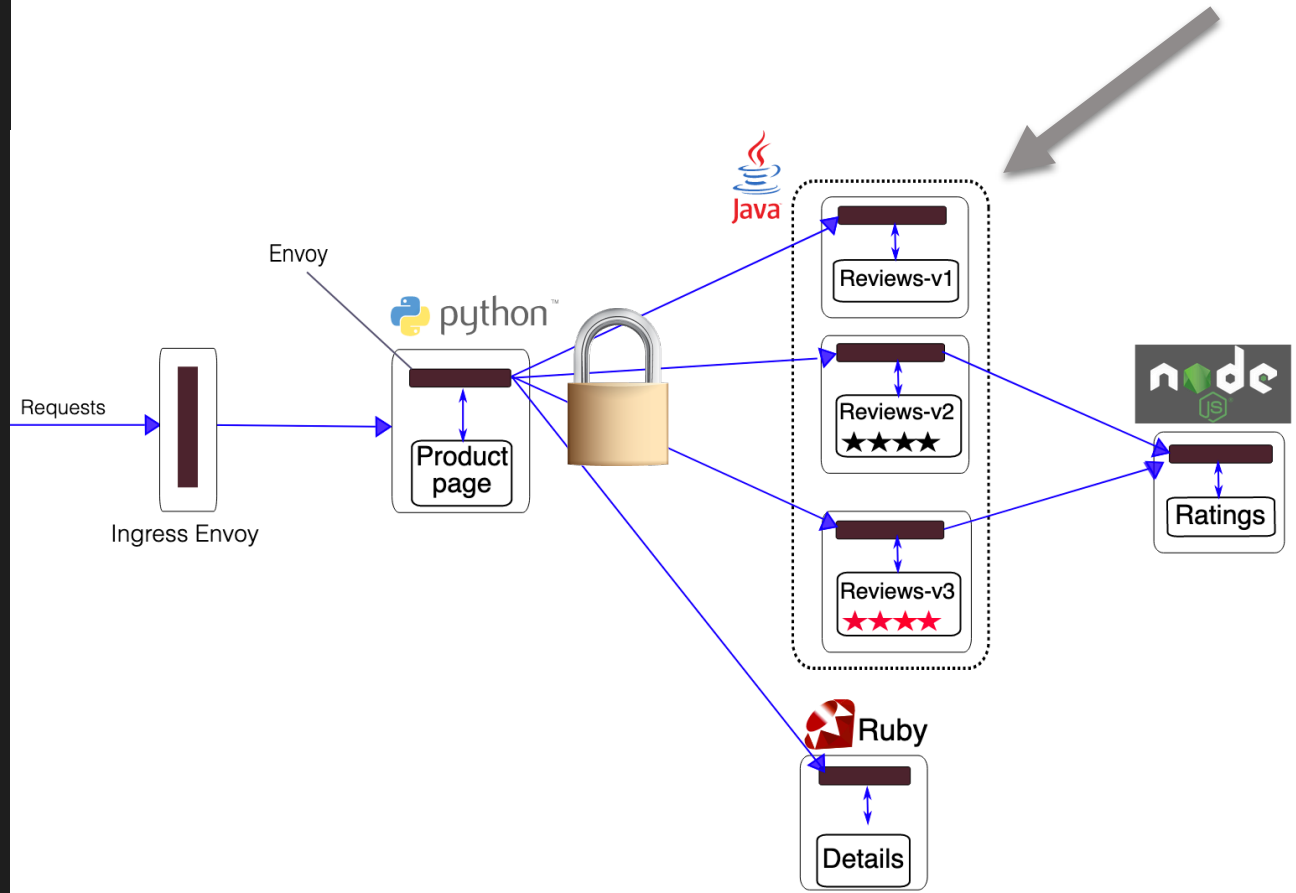


Destination Rules – subsets and tls

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: reviews
spec:
  host: reviews
  trafficPolicy:
    tls:
      mode: ISTIO_MUTUAL
  subsets:
  - name: v1
    labels:
      version: v1
  - name: v2
    labels:
      version: v2
  - name: v3
    labels:
      version: v3
```

Define subsets (versions)

Tell clients to talk TLS with reviews



Rate Limits

```yaml
apiVersion: "config.istio.io/v1alpha2"
kind: memquota
metadata:
  name: handler
  namespace: istio-system
spec:
  quotas:
  - name: requestcount.quota.istio-system
    maxAmount: 5000
    validDuration: 1s
    # The first matching override is applied.
    # A requestcount instance is checked against override dimensions.
    overrides:
    # The following override applies to 'ratings' when
    # the source is 'reviews'.
    - dimensions:
        destination: ratings
        source: reviews
      maxAmount: 1
      validDuration: 1s
    # The following override applies to 'ratings' regardless
    # of the source.
    - dimensions:
        destination: ratings
      maxAmount: 100
```

Cluster wide limits

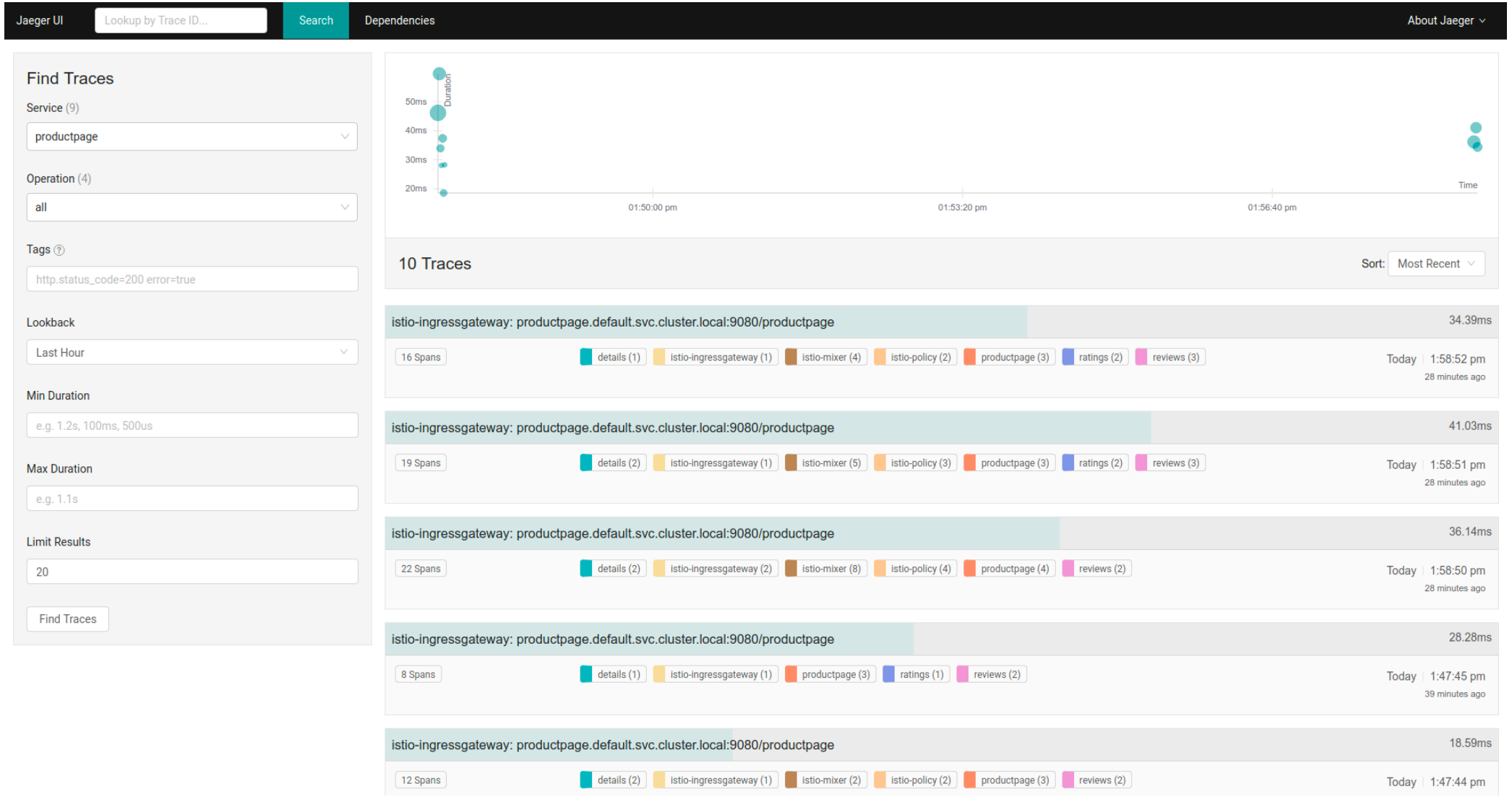
Overrides for services

Control Access

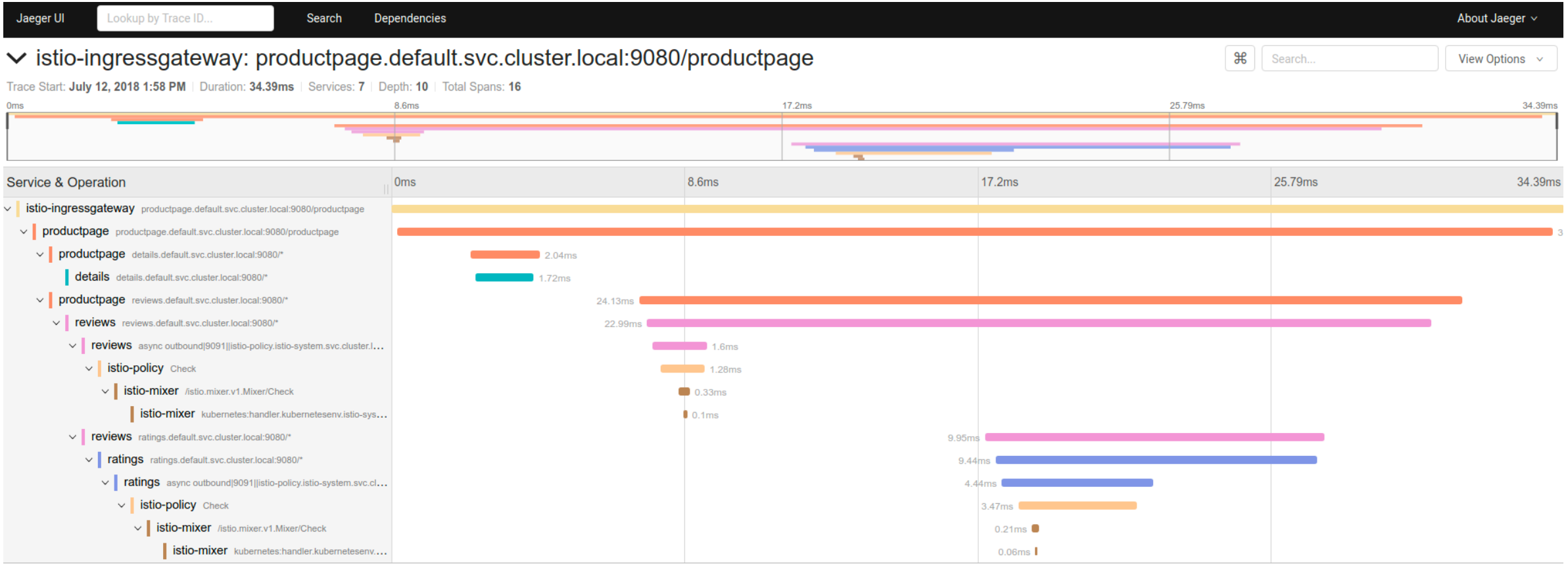
```yaml
apiVersion: "rbac.istio.io/v1alpha1"
kind: ServiceRole
metadata:
  name: analyzer-viewer
  namespace: default
spec:
  rules:
  - services: ["analyzer.default.svc.cluster.local"]
    methods: ["POST"]
---
apiVersion: "rbac.istio.io/v1alpha1"
kind: ServiceRoleBinding
metadata:
  name: bind-analyzer
  namespace: default
spec:
  subjects:
  - user: "cluster.local/ns/default/sa/guestbook"
  roleRef:
    kind: ServiceRole
    name: "analyzer-viewer"
```

namespace-level, service-level, or method-level access control

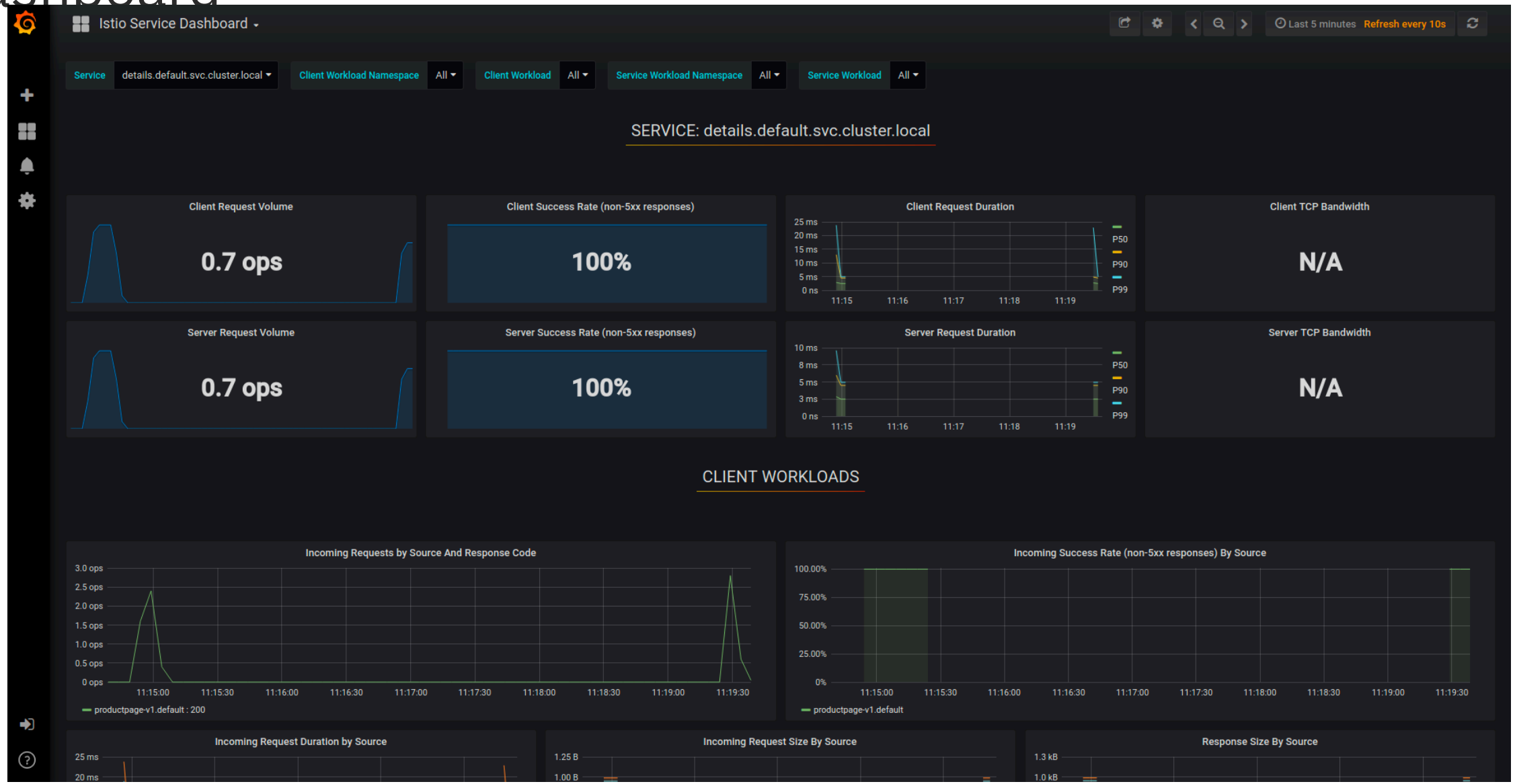
Telemetry: Tracing Dashboard



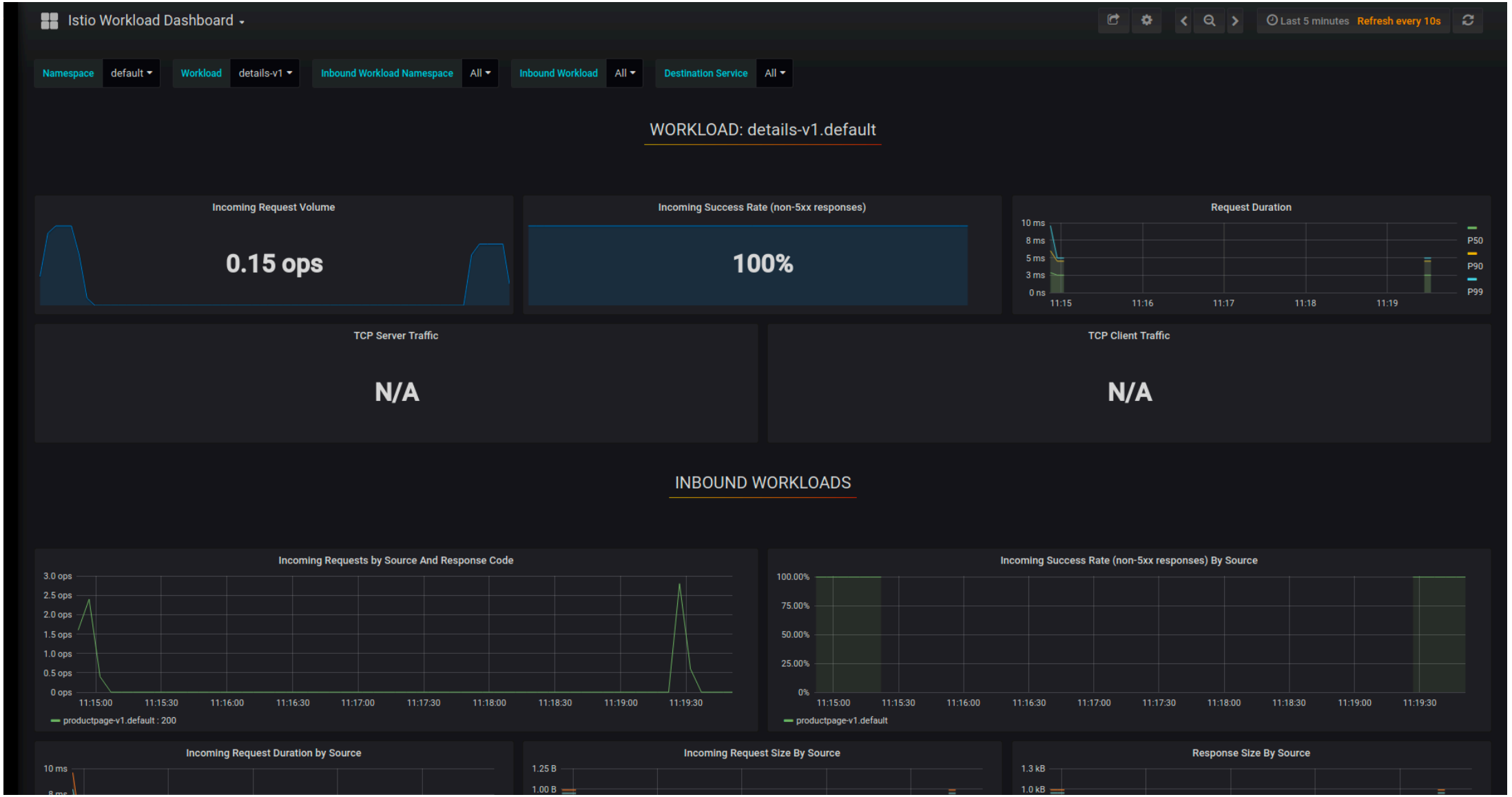
Telemetry: Tracing Detail



Telemetry: Visualize Service Dashboard



Telemetry: Tracing Detail



Example: Injecting an HTTP Delay Fault

To test the an application's microservices for resiliency you could inject a delay between services for a specific user

In the case shown here, traffic initiated by Jason to the ratings version 1 service will receive a 7 second delay

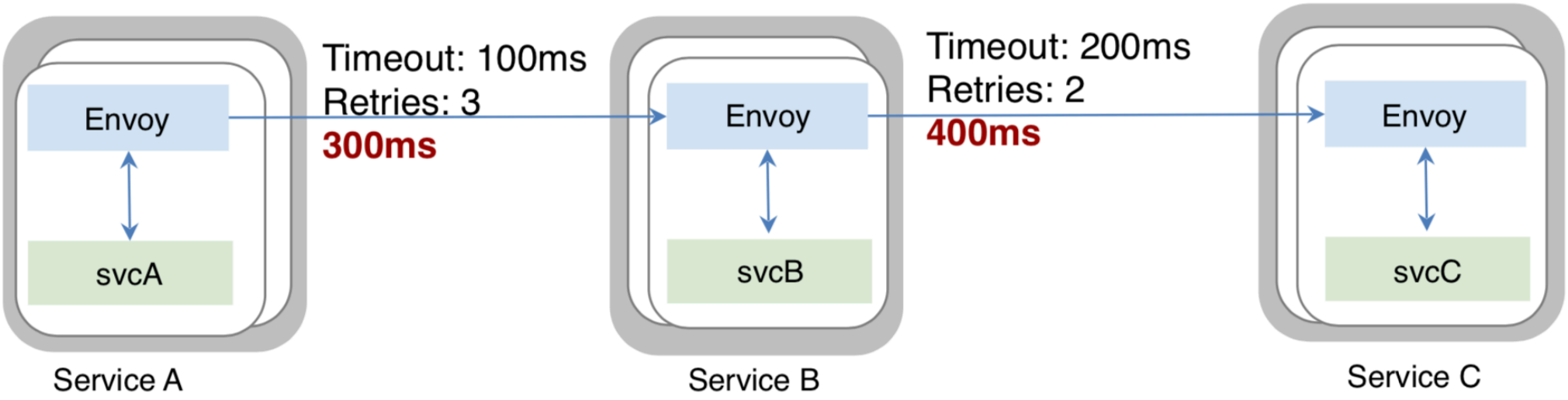
```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings
spec:
  hosts:
  - ratings
  http:
  - match:
    - headers:
        end-user:
          exact: jason
    fault:
      delay:
        percent: 100
        fixedDelay: 7s
    route:
    - destination:
        host: ratings
        subset: v1
  - route:
    - destination:
        host: ratings
        subset: v1
```

Resiliency testing

Systematic fault injection to identify weaknesses in failure recovery policies:

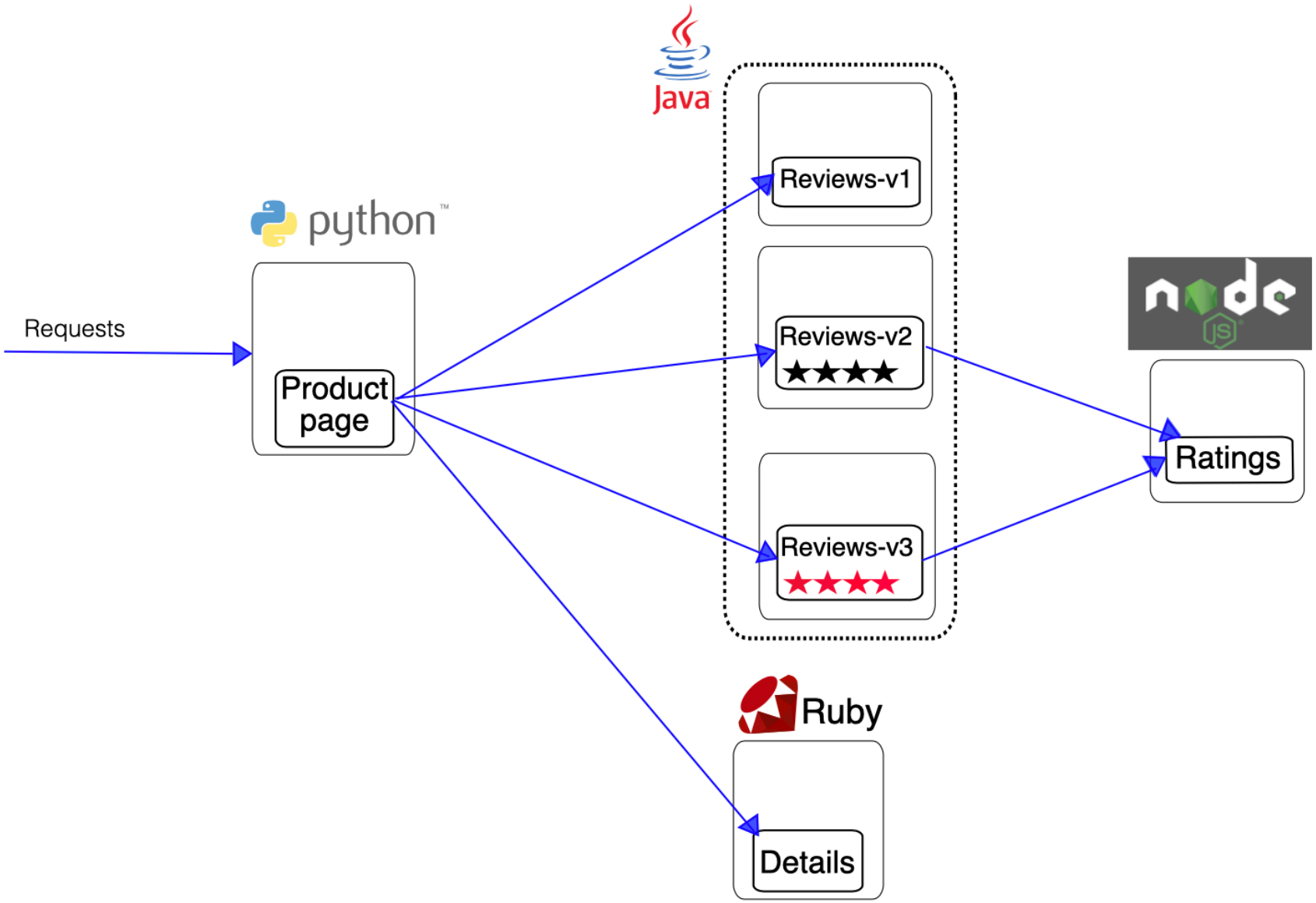
HTTP/gRPC error codes

Delay injection



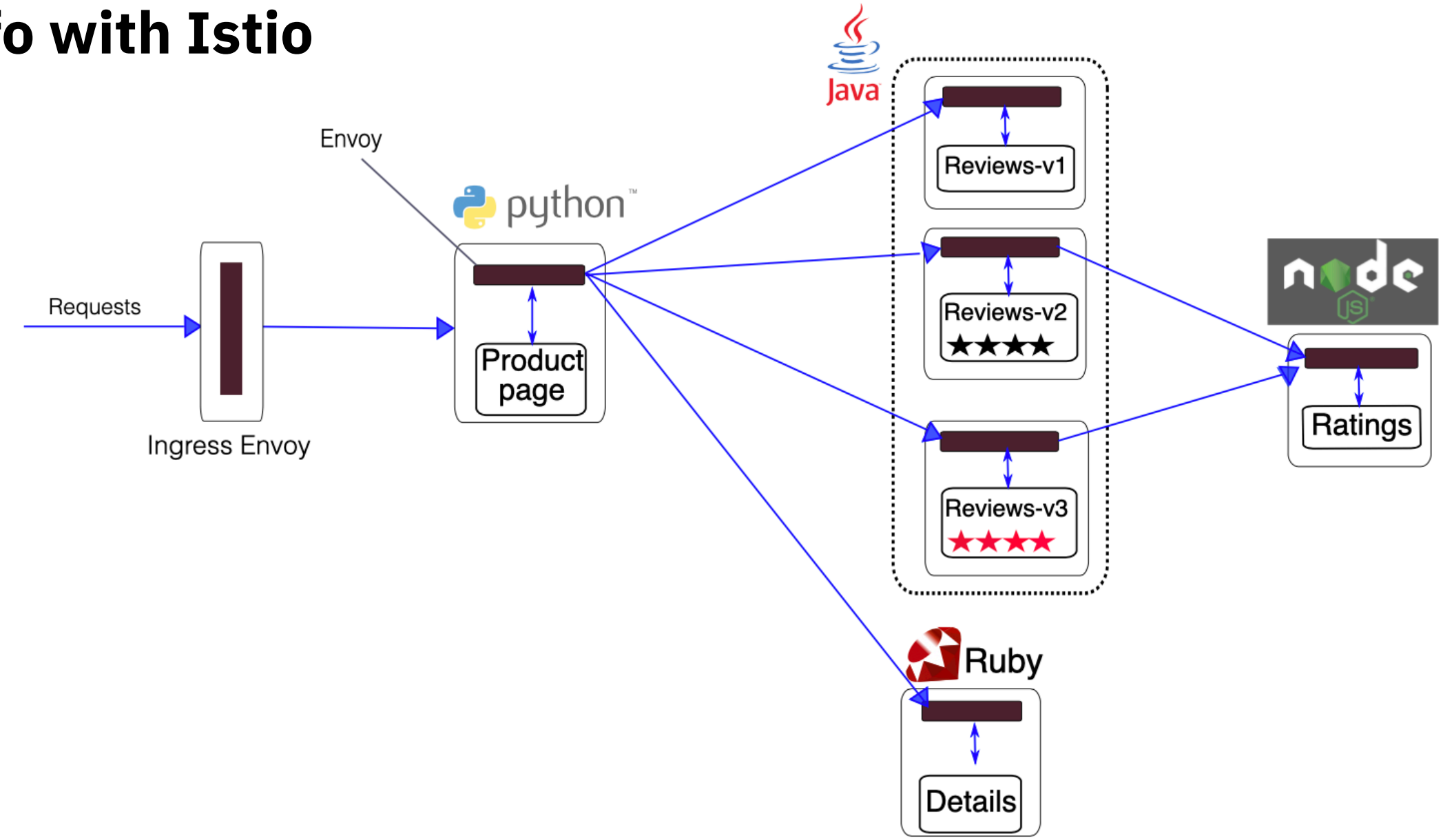
IBM **Code**

Bookinfo



Bookinfo Application without Istio

Bookinfo with Istio



Bookinfo Application

Example

ibm.biz/201910-istio-webinar

Conclusion

- Docker containers
- Container orchestration
- Istio – beyond smart routing and resiliency

- Get started with Istio and IBM Cloud Kubernetes Service

github.com/blumareks/istio-2019

meetup.com/**IBM-Developer-SF-Bay-Area-Meetup/**

Start a new group | Log in | Sign up

Build Smart

Part of **IBM Developer – 40 groups**

IBM Developer SF Bay Area

San Francisco, CA

8,072 members · Public group

Organized by **Angie K** and **6 others**

IBM **Developer**

IBM

Share:

WED, OCT 23, 6:00 PM

Meetup: OpenShift on IBM Cloud Workshop

Hacker Dojo

Red Hat OpenShift on IBM Cloud is an extension of the IBM Cloud Kubernetes Service, where IBM manages OpenShift Container Platform for you. With Red Hat OpenShift on IBM Cloud developers have a fast and secure way to containerize and deploy enterprise workloads in Kubernetes...

32 attendees

Attend

IBM Partners

Enabling Independent Software Vendors (ISVs) and tech companies for growth

Target audience

- ISVs and tech companies building and selling cloud solutions
- New to IBM Cloud
- Startups who aspire to build and sell their own solutions

Offers to help you get started

Build with up to $12,000 of free IBM Cloud™ credits ($1,000 per month for 12 months)

Integrate your solutions with leading-edge IBM Cloud technologies to deliver more innovation and value to your clients. Access more than **130 unparalleled services** including Watson™, Analytics and Security.

Build with 10TB of IBM Cloud Object Storage at no charge

Build data capability into your offering. IBM Cloud Object Storage is designed for high durability, resiliency and security.

Build with IBM Watson Assistant with a 1-year free trial

Receive access to 100K API calls per month plus 10 workspaces. Build and deploy chatbots quickly and efficiently with IBM Watson Assistant's advanced capabilities and seamless interface.

Build with IBM Cloud Kubernetes Service with a 1-year free trial

Containerize your solution with 1TB of block storage. Ship all your applications in one agile, well-defined structure with IBM Cloud Kubernetes Service.

Build with IBM Blockchain with a 6-month free trial

Build a network with up to 3 organizations to prototype. Build a secure business transaction network for your clients using blockchain and smart contracts.

Finished building and testing? Go-to-market with IBM

Access Provider Workbench, attend an orientation session and join the premier network of over 400 partners who are already listing their solutions on the IBM Marketplace.

Is your business a Startup? Build with up to $120,000 in IBM Cloud credits

If your business revenue in the last 12 months is less than $1M and you've been in business for fewer than five years, then you may qualify for Startup with IBM.

Get started

Experience IBM's countless partner benefits. Start building and selling with IBM today.

Learn more and access offers at ibm.com/partners/start

Thank you

twitter.com/blumareks

github.com/blumareks/istio-2019

developer.ibm.com/code