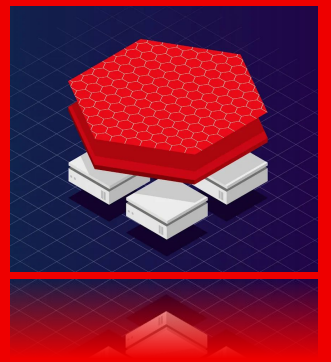


An Introduction to Service Mesh and Istio

(a platform for microservices)

Jason Dudash
Specialist Solutions Architect
Emerging Technology

github.com/dudash

Chris Kang
Specialist Solutions Architect
Emerging Technology

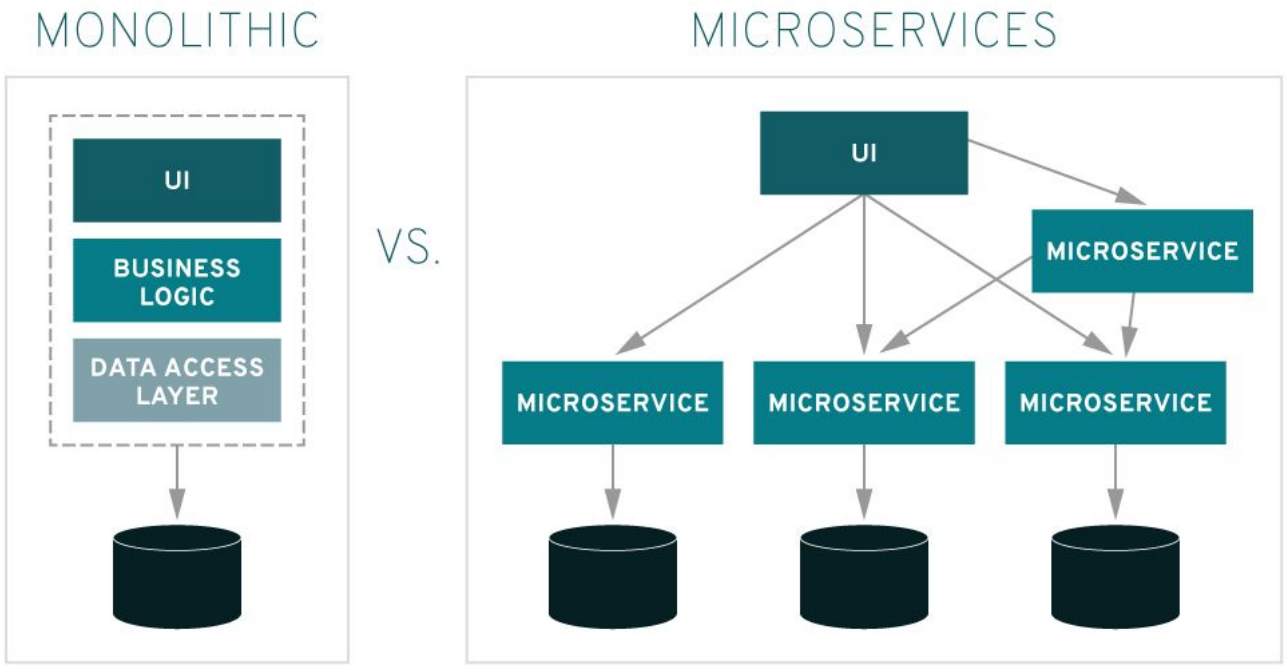
github.com/theckang

Red Hat

What are Microservices?

an architectural style that structures an application as a collection of services

- Single purpose
- Independently deployable
- Have their context bound to a biz domain
- Owned by a small team
- Often stateless



Red Hat

Benefits of Microservices

Agility
Deliver updates faster and react faster to new business demands

Highly scalable
Scale independently to meet temporary traffic increases, complete batch processing, or other business needs

Can be purpose-built
Use the languages and frameworks best suited for the service's domain

Resilience
Improved fault isolation restricts service issues, such as memory leaks or open database connections, to only affect that specific service

Many orgs have had success with Microservices - Netflix, Amazon, eBay, The Guardian

There is inherent complexity in adopting microservices

Some common areas where organizations stumble when adopting microservices

Tolerance to Faults

Cascading failure, partial outages, traffic spikes

DevOps and Deployments

More failure surface, version incompatibility, untracked svcs

Services Communication Needs

Latency, concurrence, distributed transactions

Inability to Monitor & Understand Performance

More to monitor & different types of monitoring required

Securing Services

Malicious requests, DoS, id & access control

Highly Distributed Logs

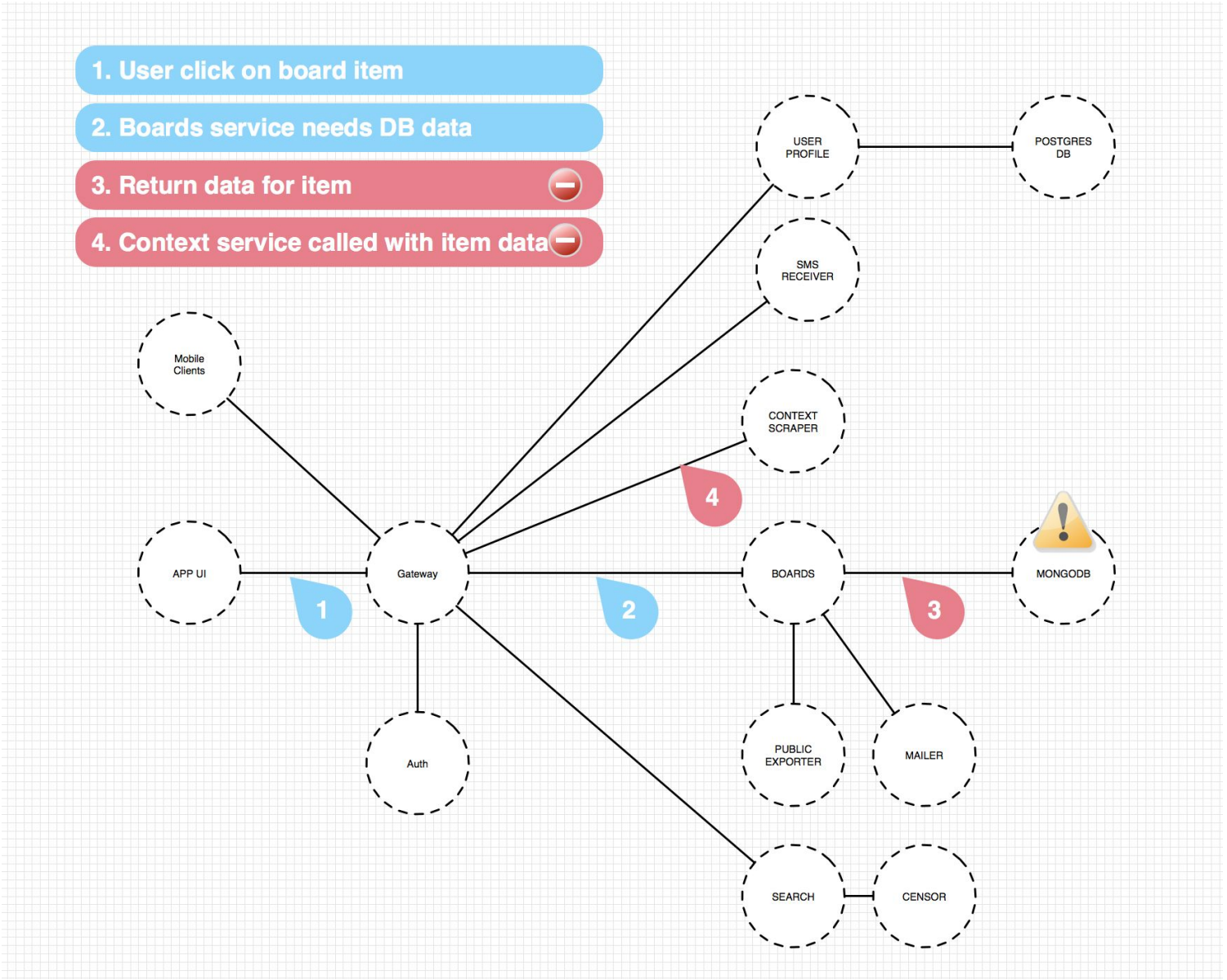
Scattered logs, lots more logs to manage, access control

Biggest challenge:
What used to be internal now needs to go across a network

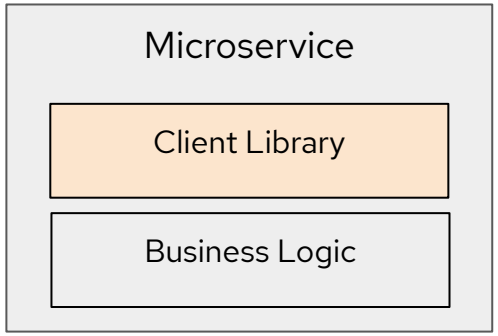
Partial Failure → Cascading Failures

- Route service's database goes offline
- Now no item data can be returned
- Default timeout for developer's HTTP framework is 2 minutes
  - This wait is happening repeatedly
- Item data is needed to pass to context scraper service - so it fails too
- User experience is poor

- Leads to unexpected case of users repeatedly mashing the refresh button
- Now the boards service begins to get more requests than than it can handle



Language Specific Libraries and Tools



Narrow Scope

Built to address a specific problem such as fault tolerance

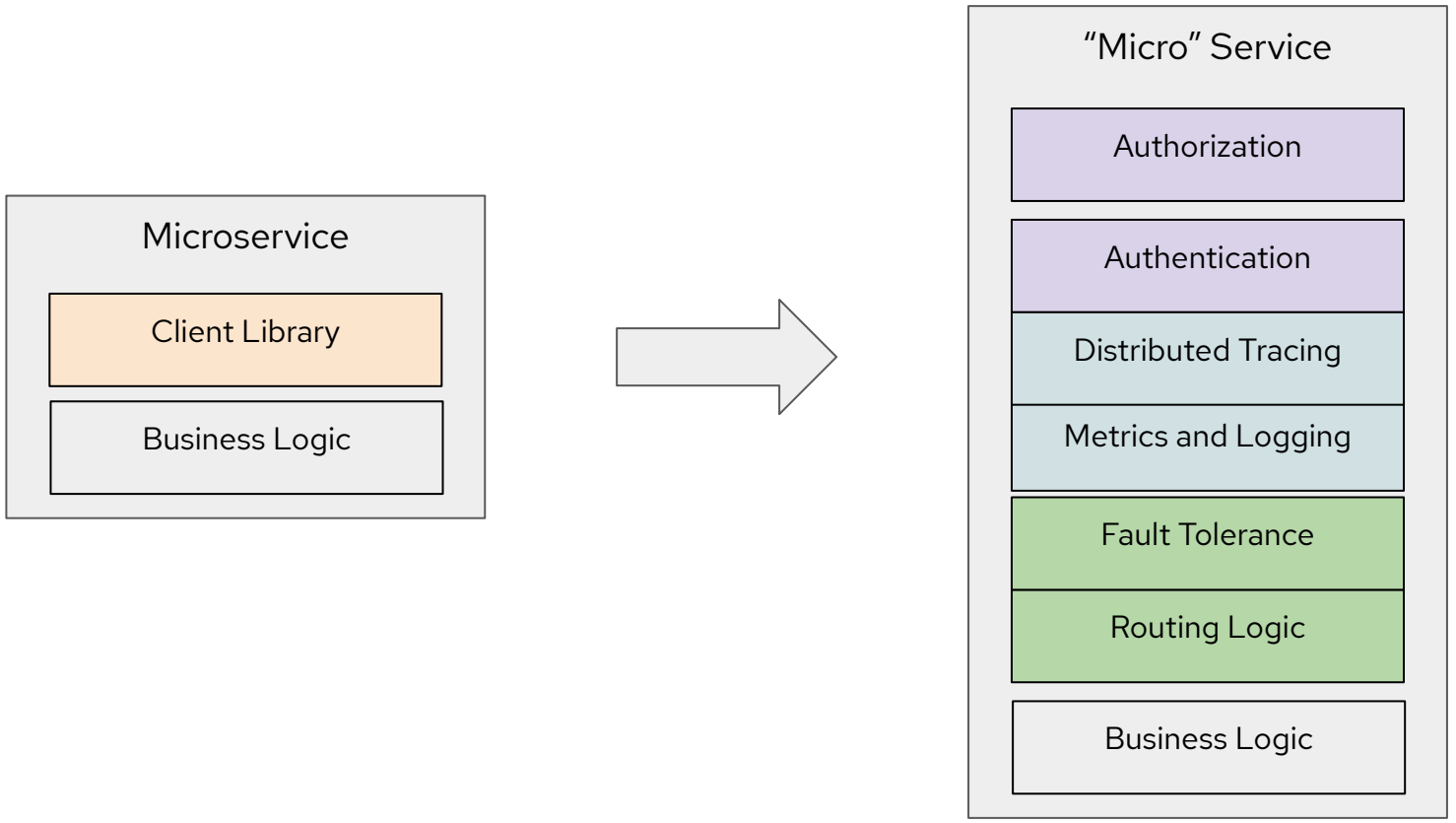
Language Specific

Client libraries are tied to programming language

Thick Clients

Solution results in clients with bulk of capabilities

Is This a “Micro” Service?



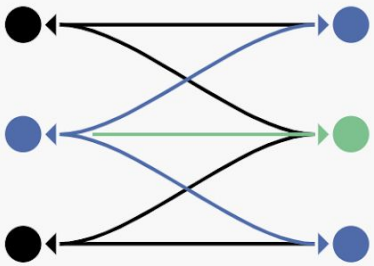
Does it *really* make sense to push operational challenges to developers to deal with?

Don’t force extra work on developers

There is a better way

Istio Service Mesh

A modern way to manage the complexity of microservice applications

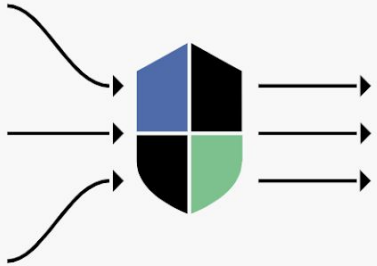


Connect

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

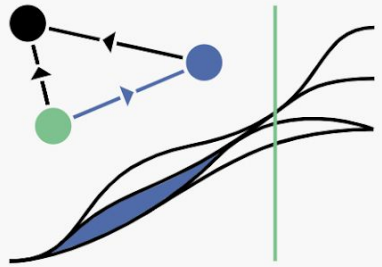
Secure

Automatically secure your services through managed authentication, authorization, and encryption of communication between services.



Control

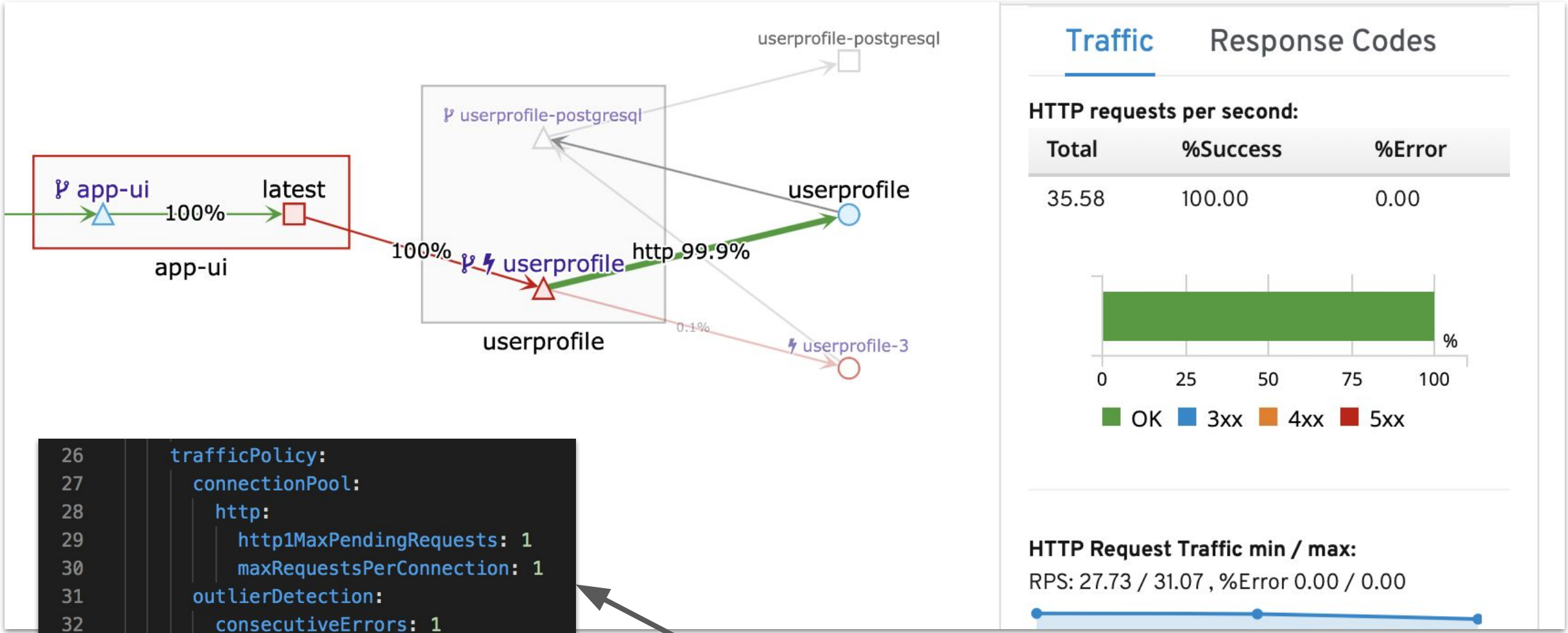
Apply policies and ensure that they're enforced, and that resources are fairly distributed among consumers.



Observe

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

Handling Partial Failures with the Service Mesh

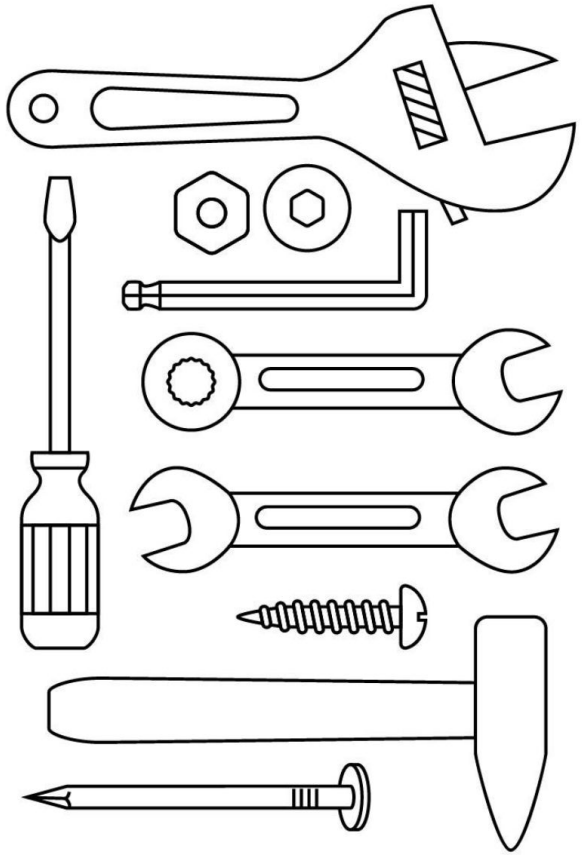



```
26      trafficPolicy:
27        connectionPool:
28          http:
29            http1MaxPendingRequests: 1
30            maxRequestsPerConnection: 1
31        outlierDetection:
32          consecutiveErrors: 1
33          interval: 1s
34          baseEjectionTime: 10m
35          maxEjectionPercent: 100
```

YAML configurable,
(Kubernetes native)

Red Hat

Features needed to meet the challenges of microservices



CIRCUIT BREAKING AND BULKHEADS

RATE LIMITING

MIRRORING / TRAFFIC SHIFTING

VERSION BASED ROUTING

AUTOSCALING

STAGED ROLLOUTS

CANARY DEPLOYMENTS

BLUE/GREEN DEPLOYMENTS

DISTRIBUTED TRACING

VISUAL SERVICE HEALTH

DISTRIBUTED LOGGING

COLLECTING AND VISUALIZING METRICS

CHAOS ENGINEERING

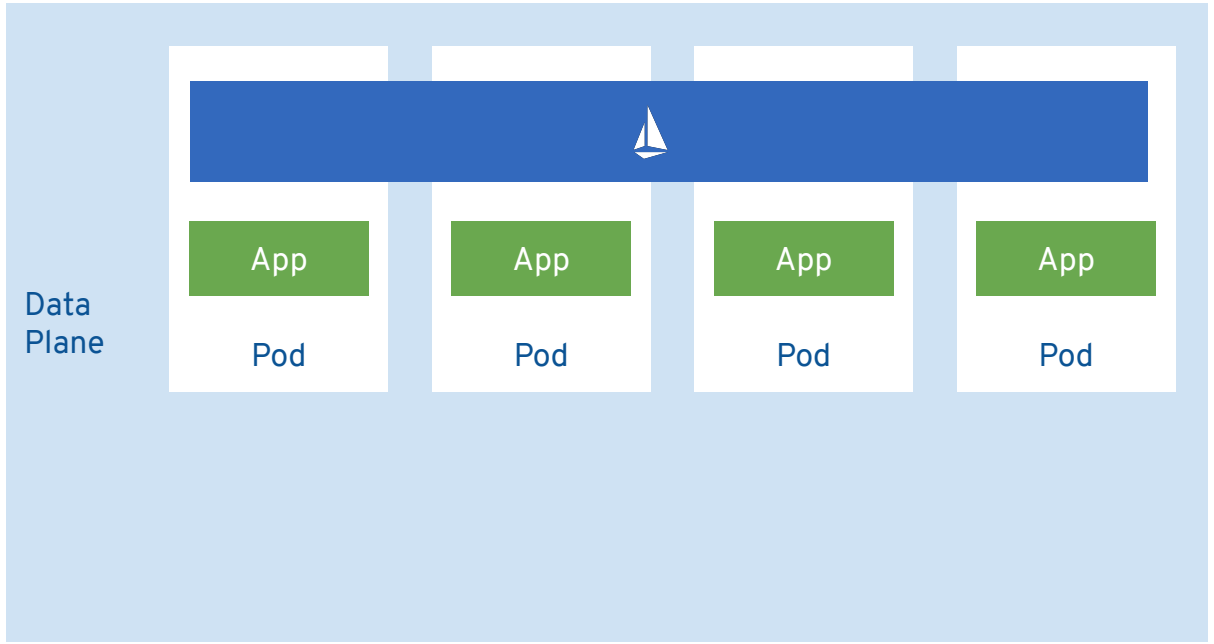
SERVICE SECURITY

CONTAINER BUILD AUTOMATION

Table Stakes for doing microservices

The service mesh is critical in addressing the inherent complexity of microservices

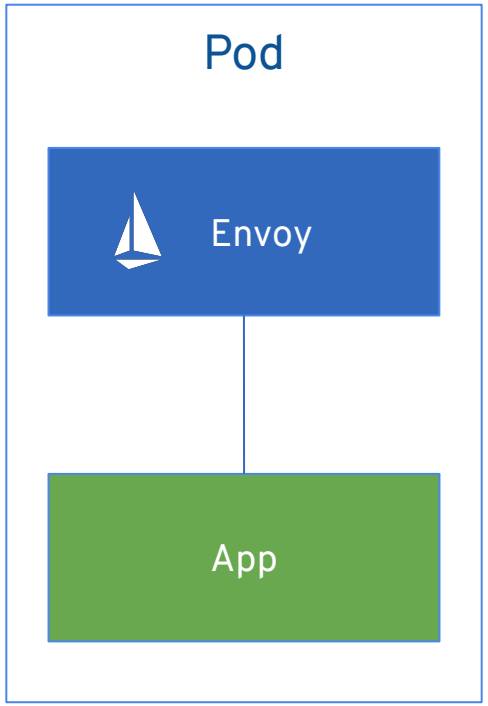
Your Services are in a “Data Plane”



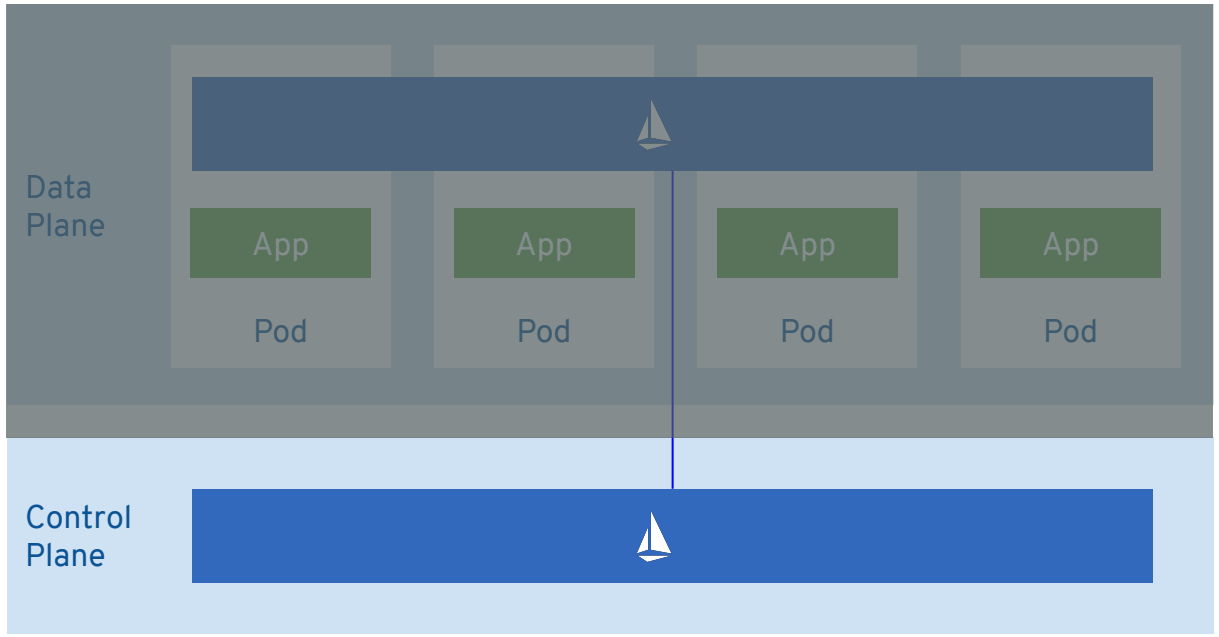
RED HAT® OPENSHIFT

Red Hat

What's a Sidecar have to do with containers?



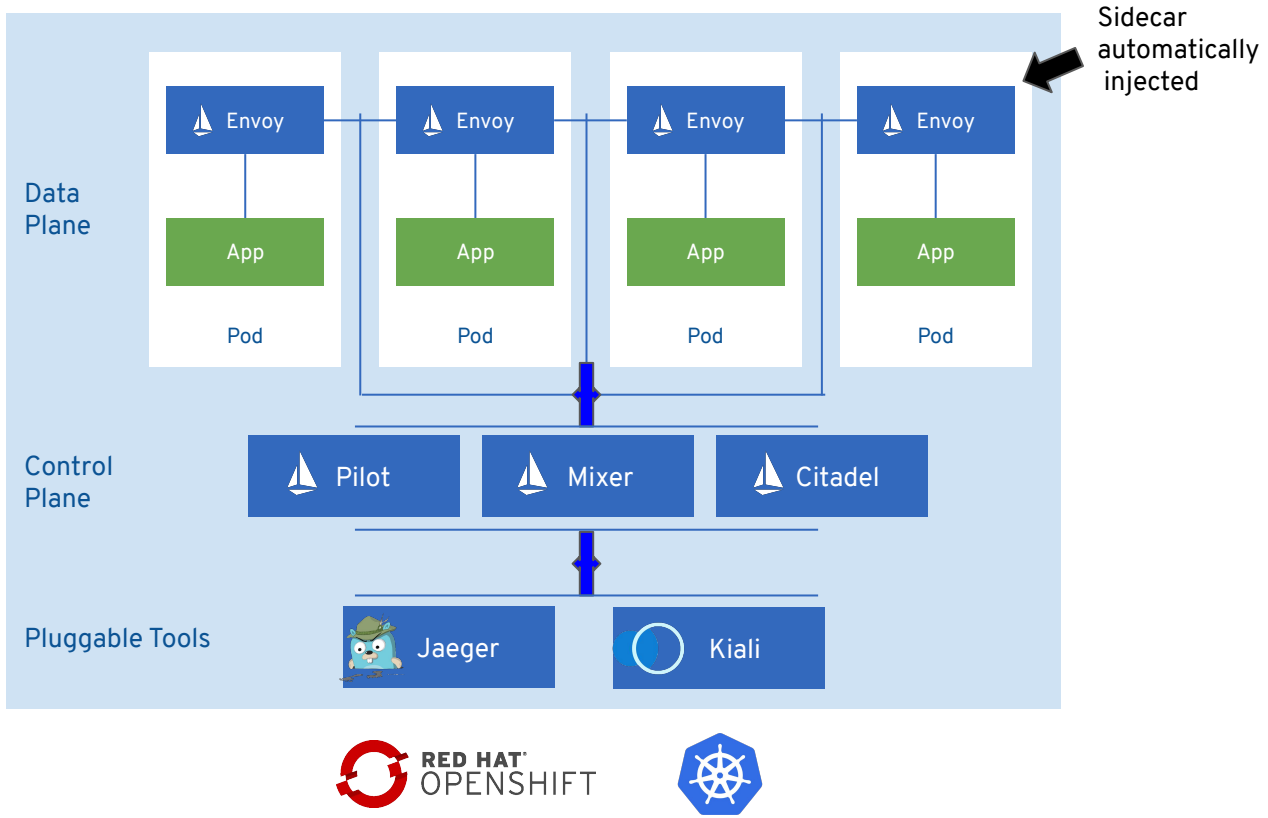
Your Policy Makes Up a “Control Plane”



RED HAT® OPENSHIFT

Red Hat

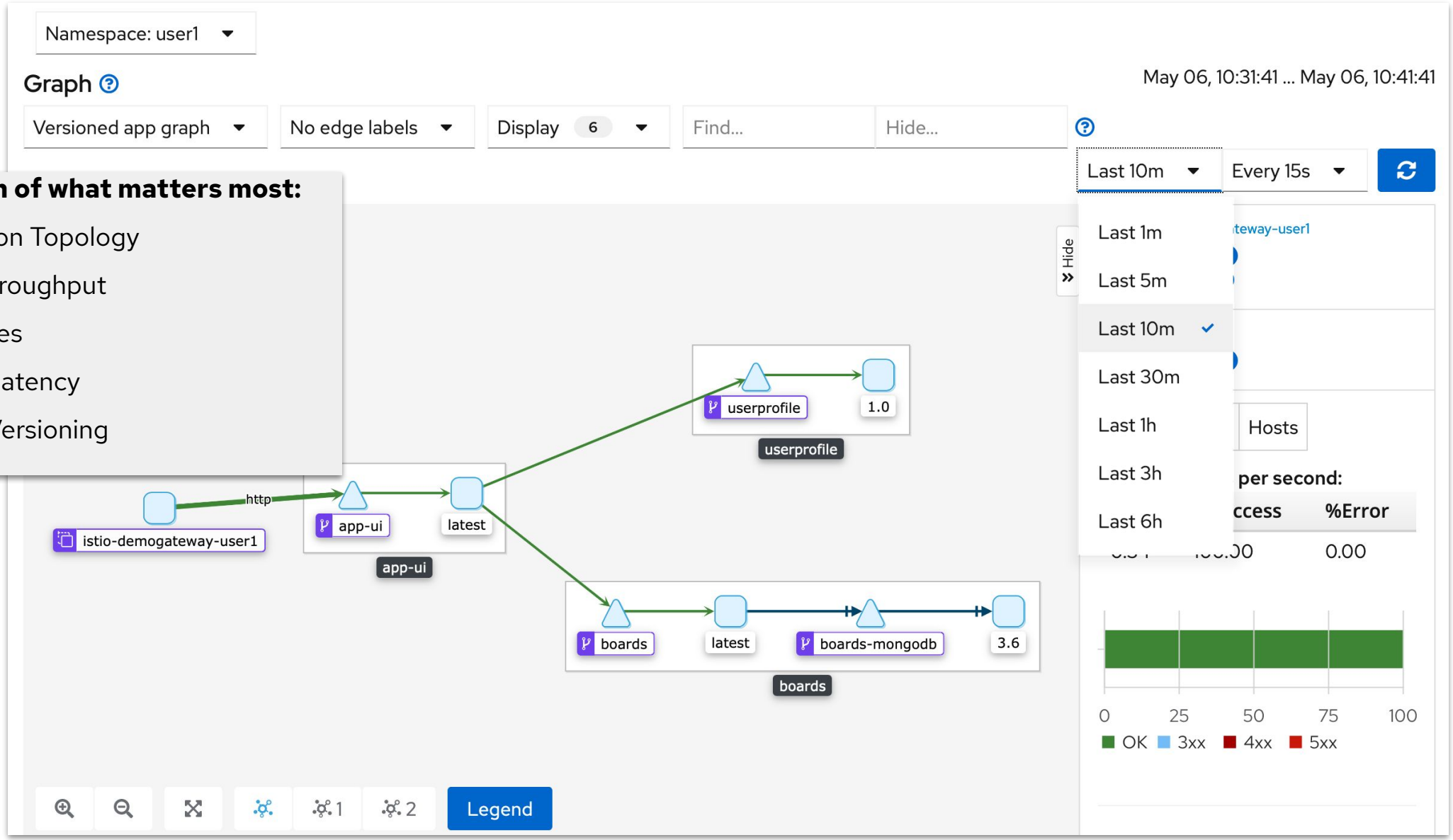
Let's Break it Down a Little Bit More



Red Hat

Visualization of what matters most:

- Application Topology
- Traffic throughput
- Error Rates
- Service Latency
- Service Versioning



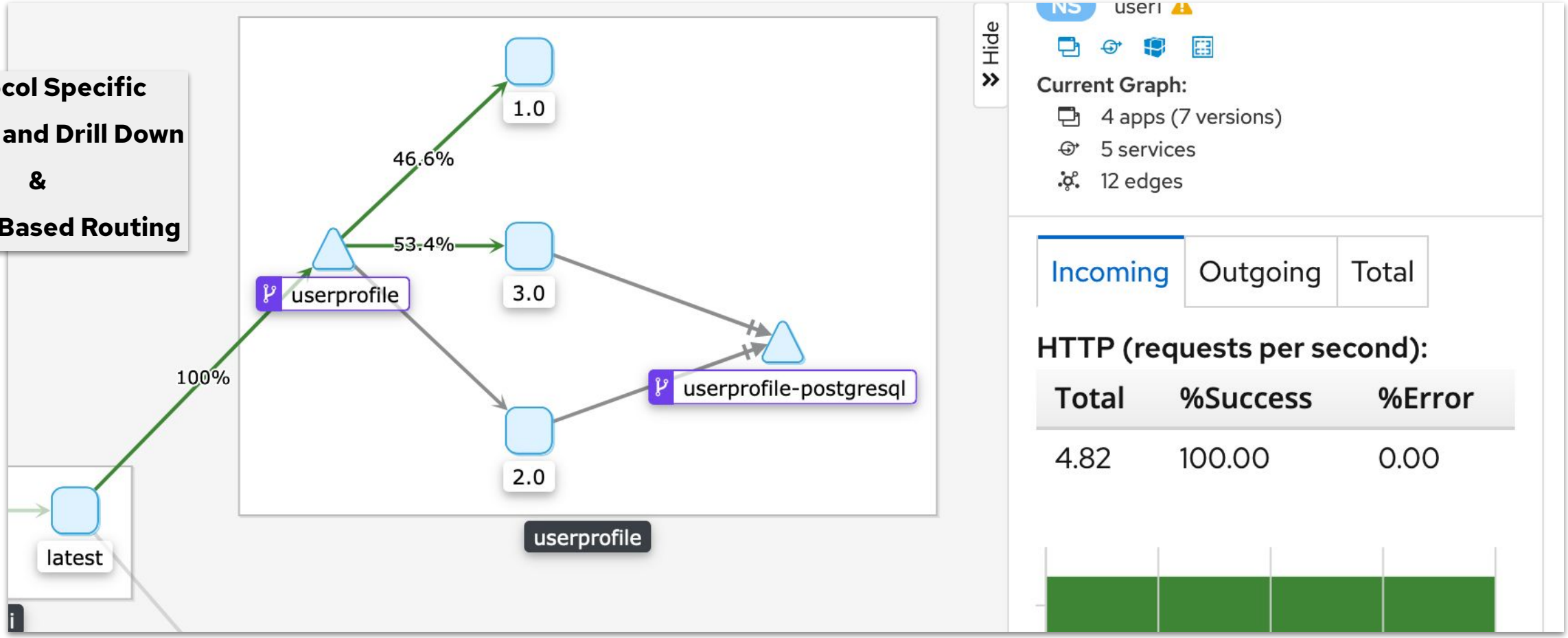
Red Hat

Protocol Specific

Analysis and Drill Down

&

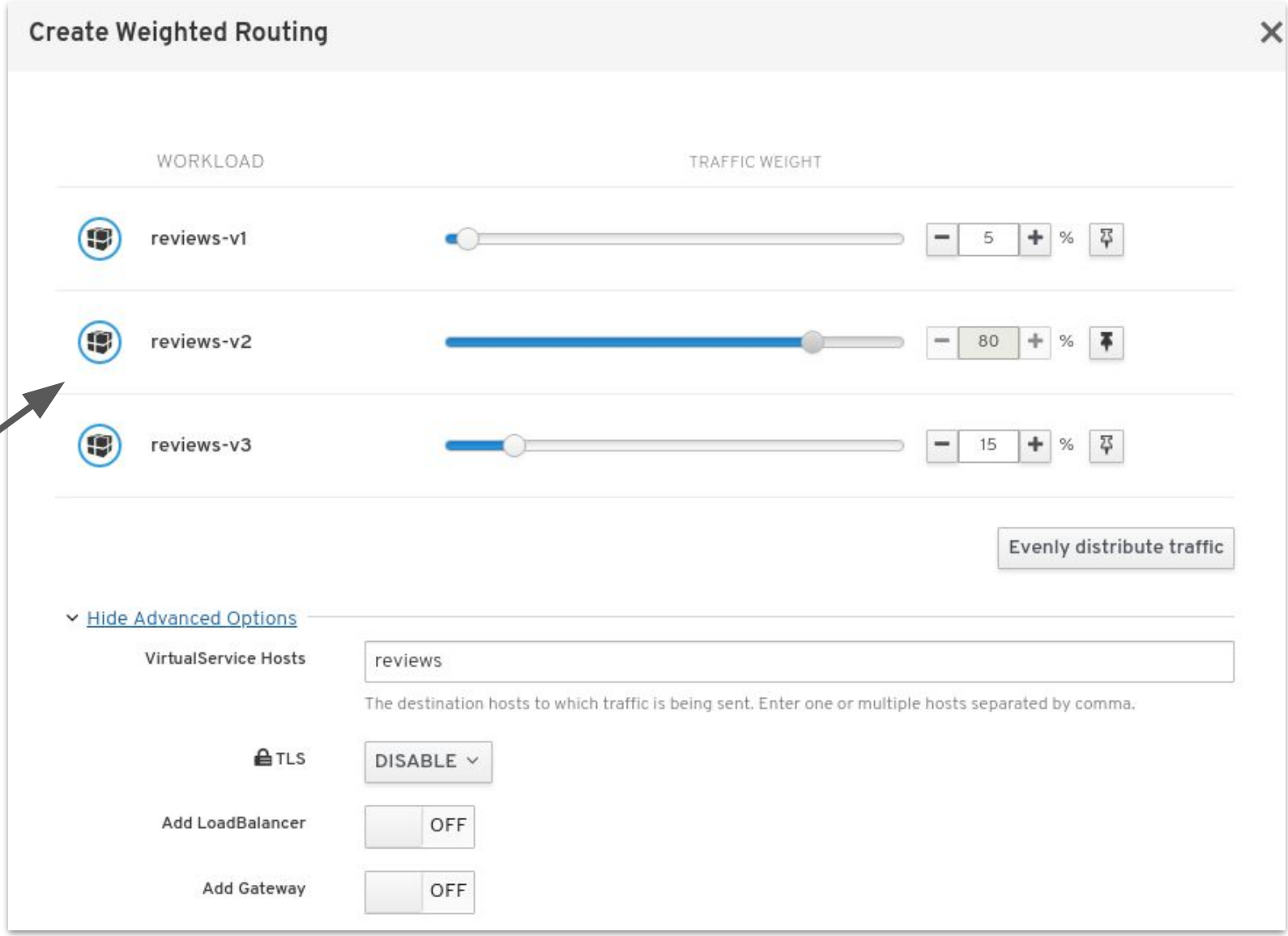
Version Based Routing



Red Hat

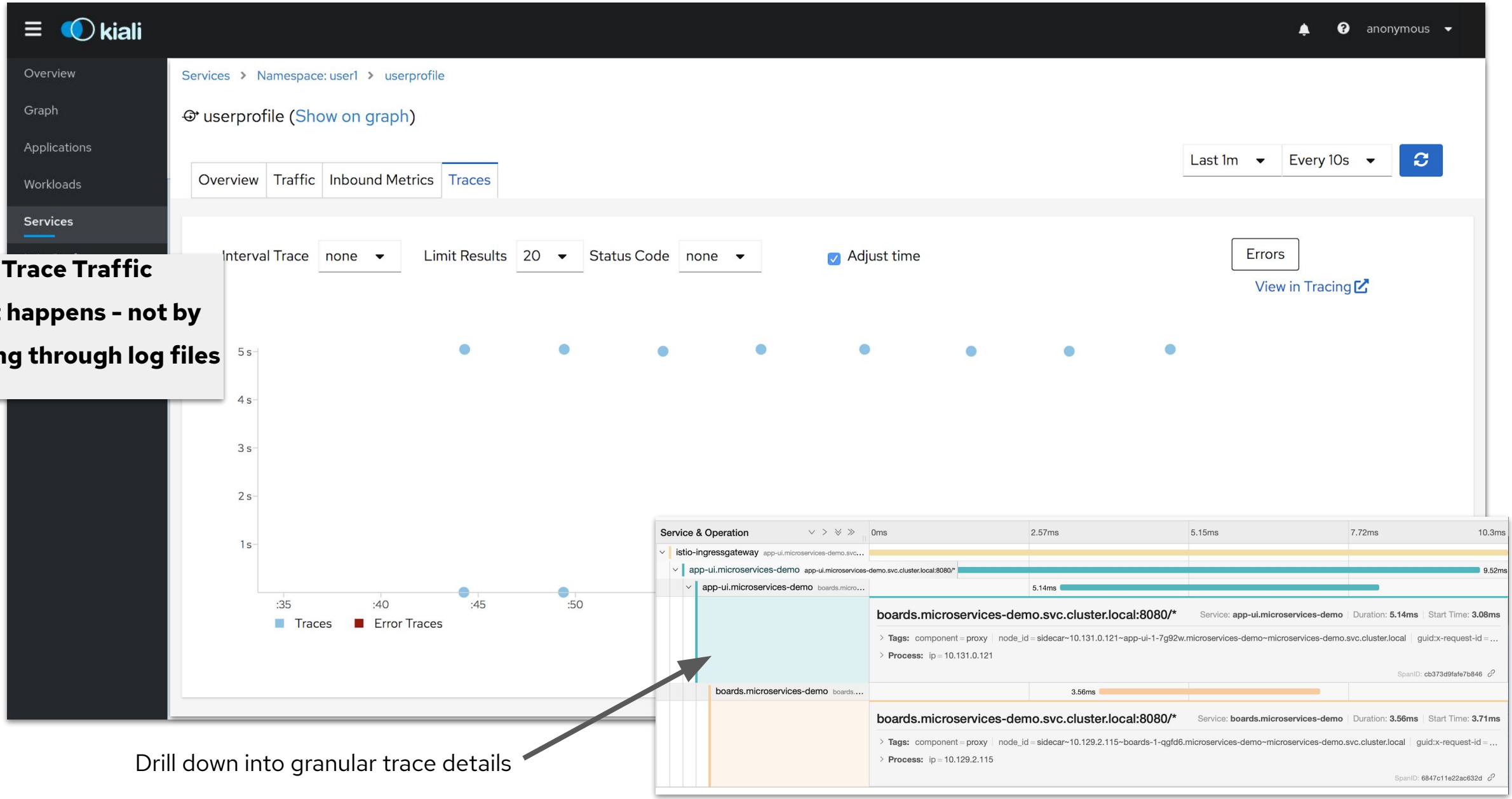
Guided Config of Traffic Policies

That's real-time version based routing!



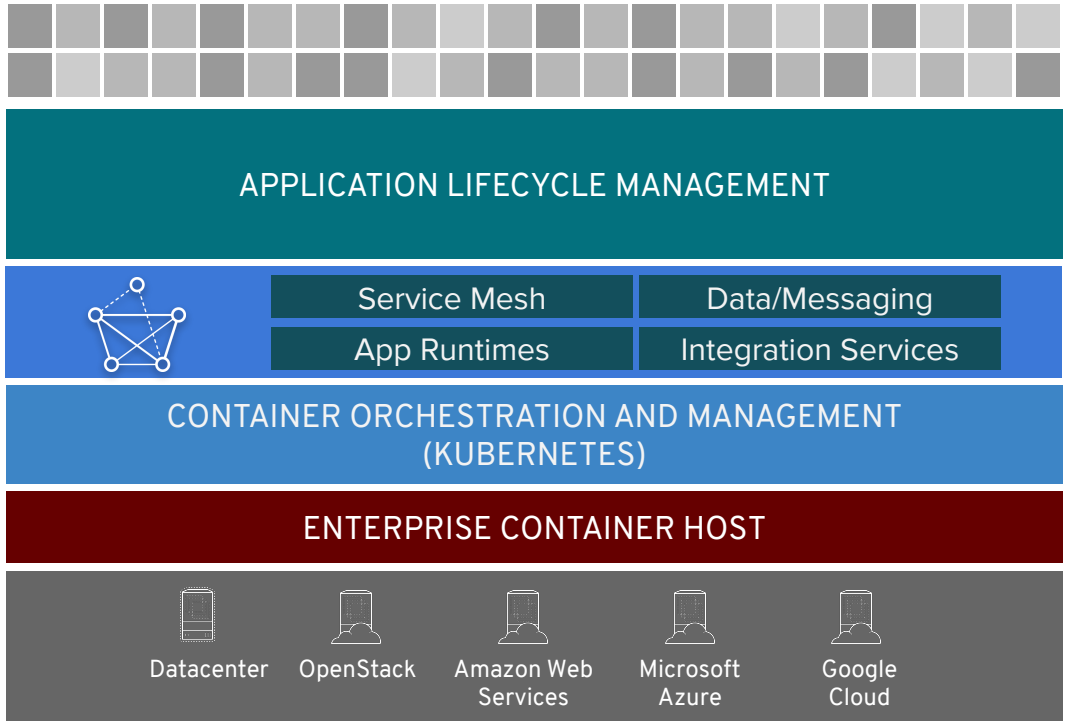
Red Hat

Trace Traffic as it happens - not by digging through log files



Drill down into granular trace details

Bundled Platform for Microservices



ANY CONTAINER

RED HAT® OPENSHIFT

ANY INFRASTRUCTURE

Red Hat

The future is Istio

Summary and Resources

- Microservices are great but they come with challenges
- Expecting developers to deal with all the challenges of microservices is unrealistic
- Service Mesh can reduce the development and operational complexity of microservices based applications
- Service Mesh 1.1 out now! (included with OpenShift)
- Self paced hands-on: https://learn.openshift.com/servicemesh/
- Guided technical workshop: Reach out to your Red Hat account rep.

A Service Mesh is critical to run microservices and requires little to no changes to your code

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.


linkedin.com/company/red-hat


youtube.com/user/RedHatVideos


facebook.com/redhatinc


twitter.com/RedHat

Red Hat