

A vertical decorative graphic on the left side of the slide, rendered in various shades of red. It features a collage of icons representing cloud computing, networking, and data storage. Elements include a cloud with a keyhole, a database cylinder, a server rack, a computer monitor, and various arrows and geometric shapes like 'X' and 'O' marks, suggesting a complex system architecture.

Istio / Service Mesh

Understanding your microservice applications

January 22nd, 2020

Maliat Manzur mmanzur@redhat.com

Steve Tran stran@redhat.com

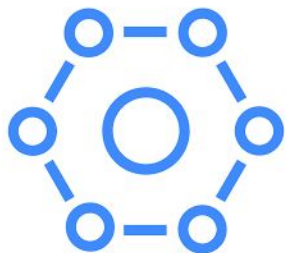


Who do we have in the room today

tinyurl.com/msaroles

Agenda

- How Istio Service Mesh fits into the context of microservices
- Which metrics provide insight into a microservices application
- How to use Istio to expose platform and application level metrics



What is Istio / Service Mesh

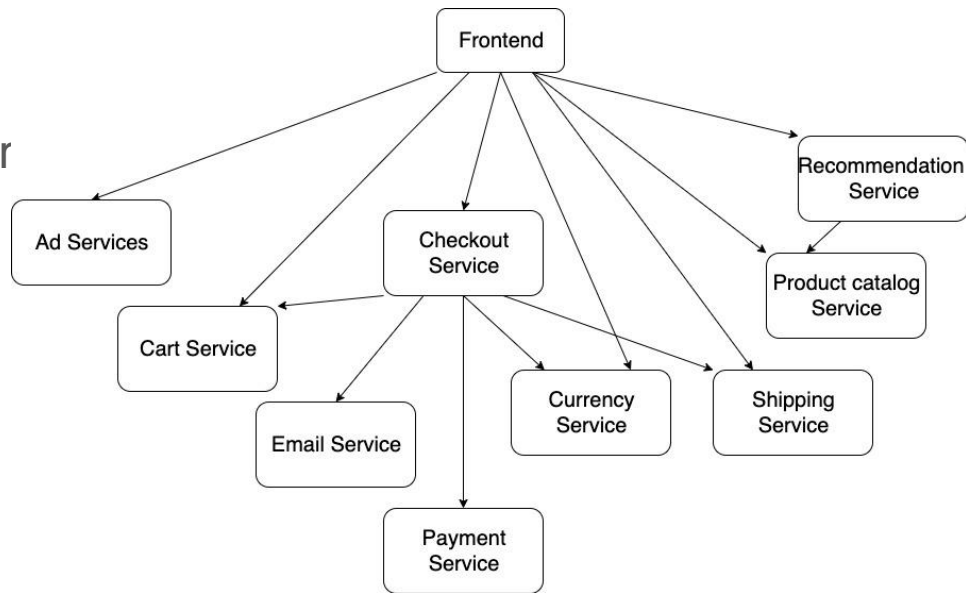
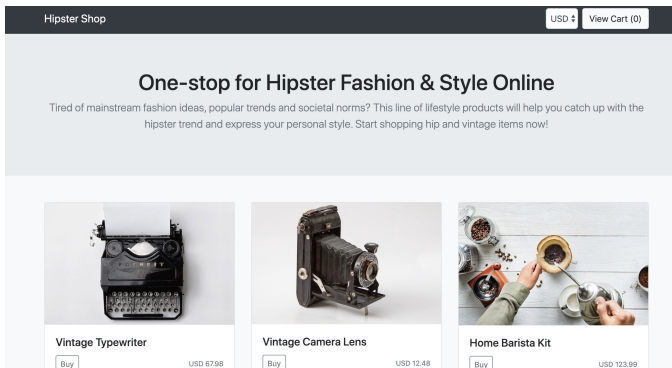
- **Dedicated, infrastructure layer that connects your services together**



Istio and Microservices

Imagine an application

- Hipster Shop
- 12 different microservices
 - With dependencies on other

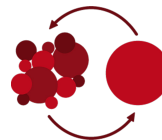
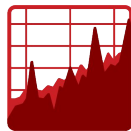
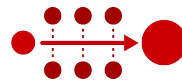


Metrics

monitoring vs observing

Why do we need them?

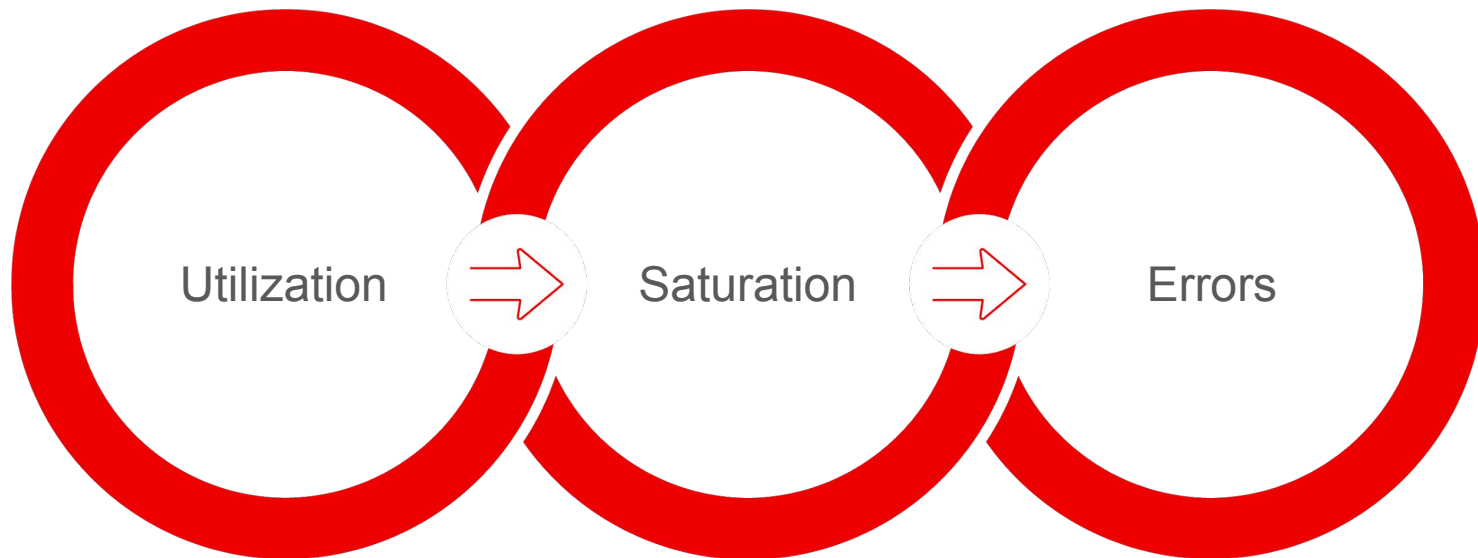
- Real time performance
- Reduce troubleshooting time
- Proactively find issues
- Alerting
- Baselines



Which metrics provide insight into a microservices application?



USE Method



4 Golden Signals



Latency

Time to service requests



Traffic

Payload over time



Saturation

Utilization of resources



Errors

Failure rate

Collecting Metrics without Istio

Pretty simple with small number of services



Pitfalls...

Developers responsibility

Implementation is inconsistent

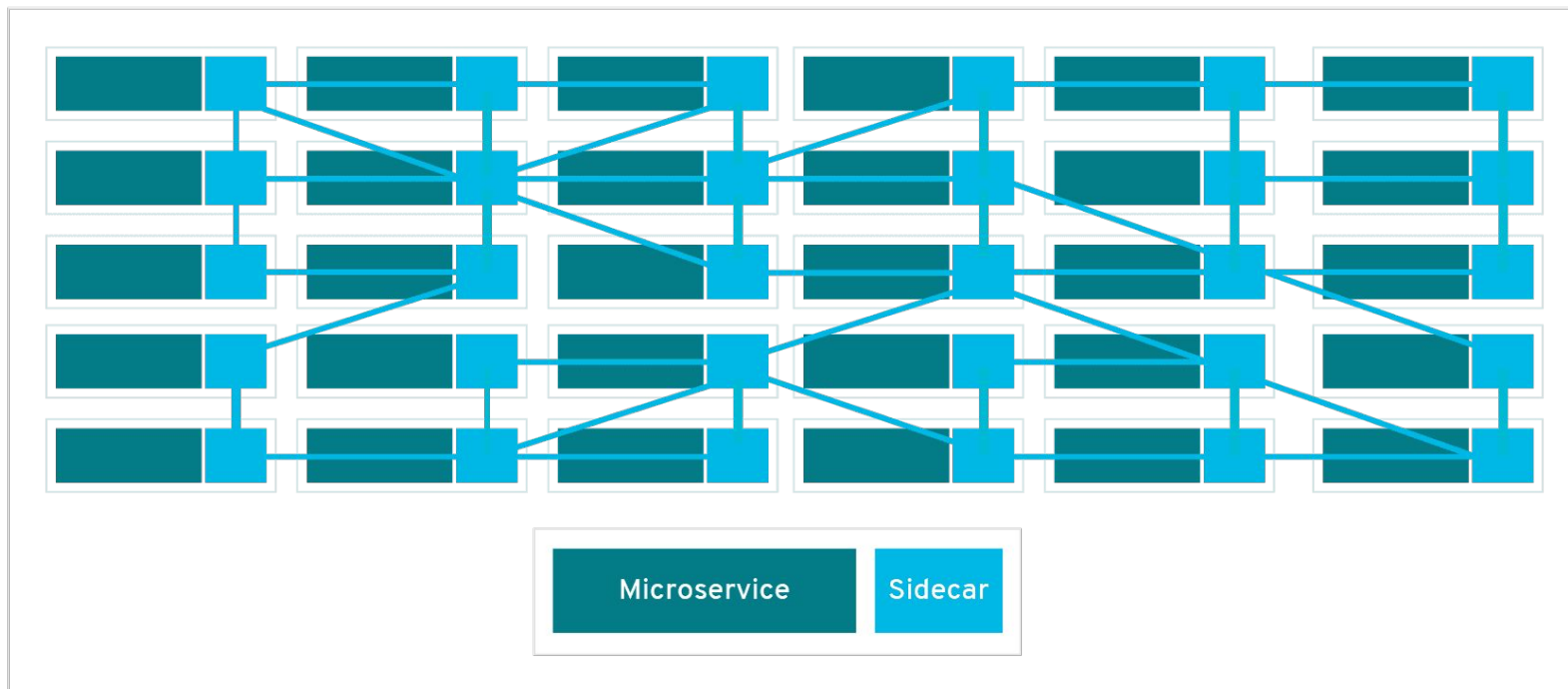
Too many options

Collecting Metrics with Istio

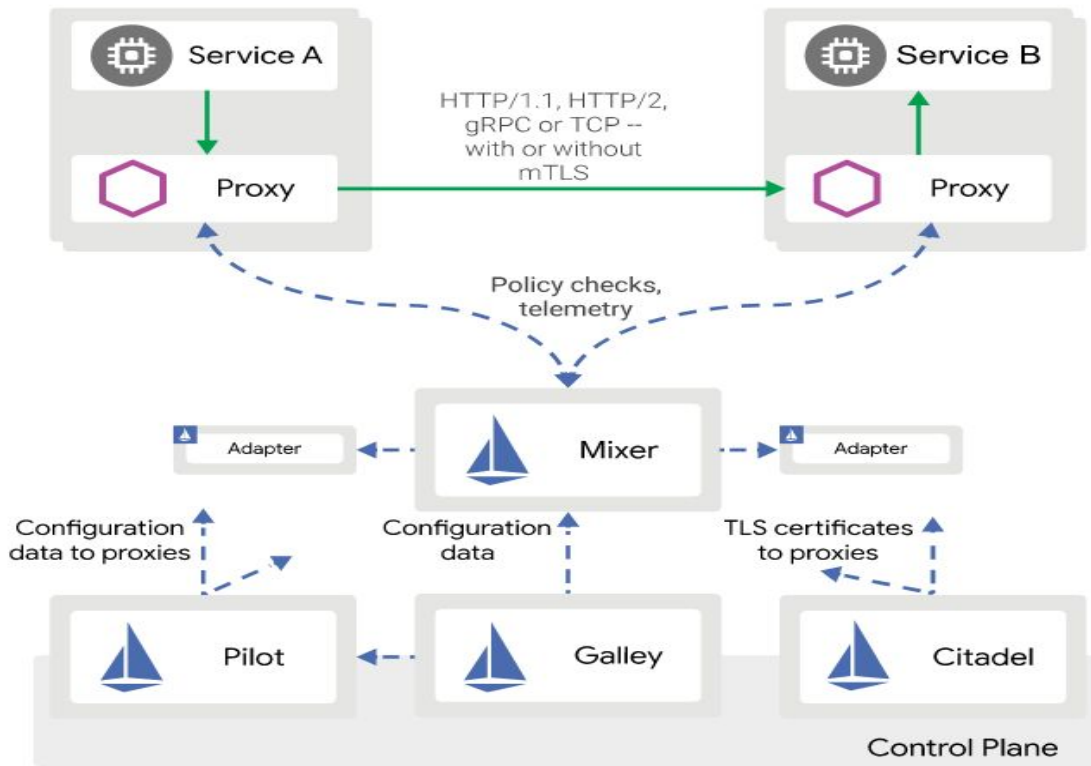


- Requests per second
- Request type
- Response time
- etc.

Bigger Picture



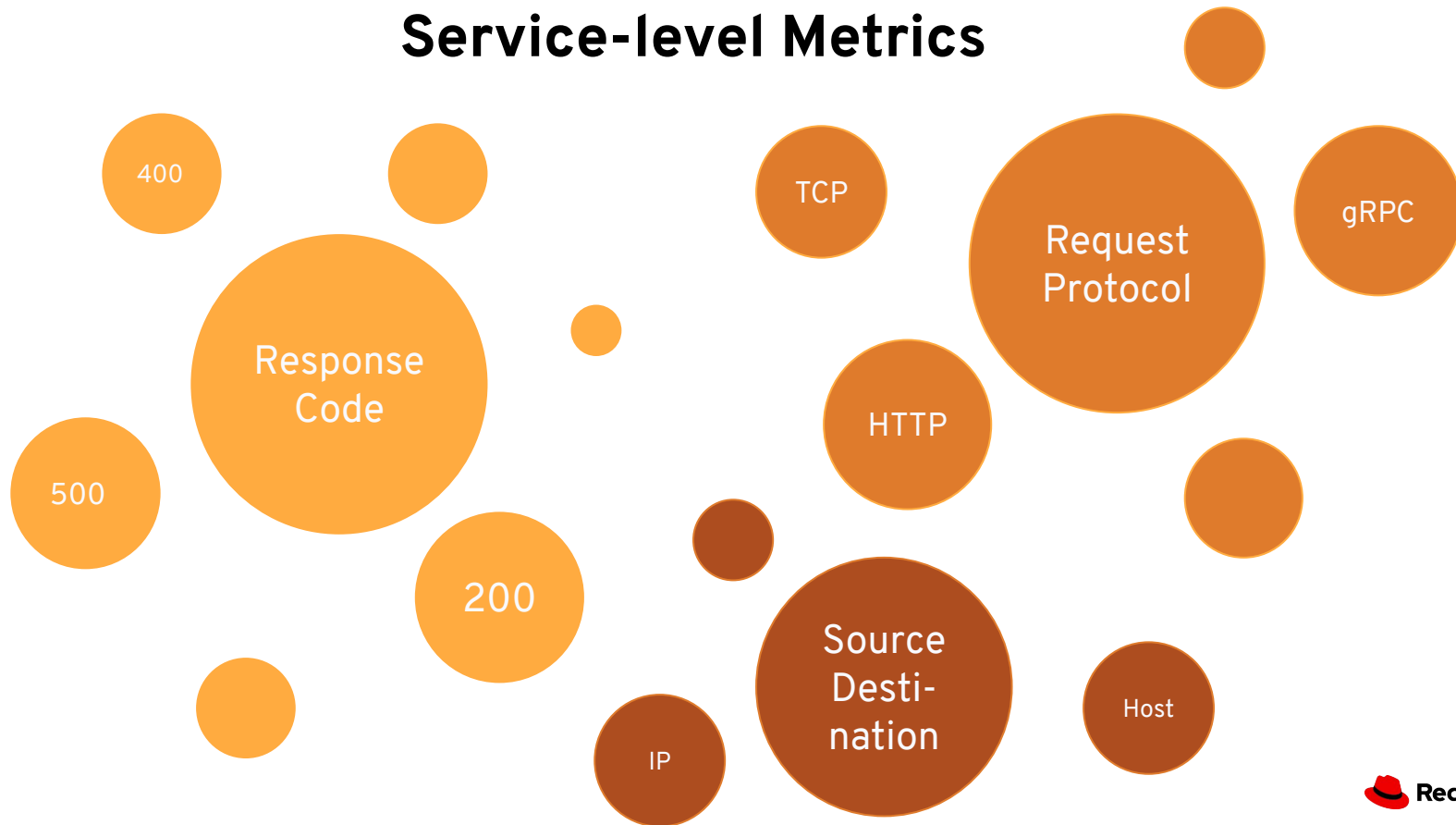
Istio Architecture



Types of Metrics



Service-level Metrics



Custom Metrics

Instance

Identify the metrics

Rule

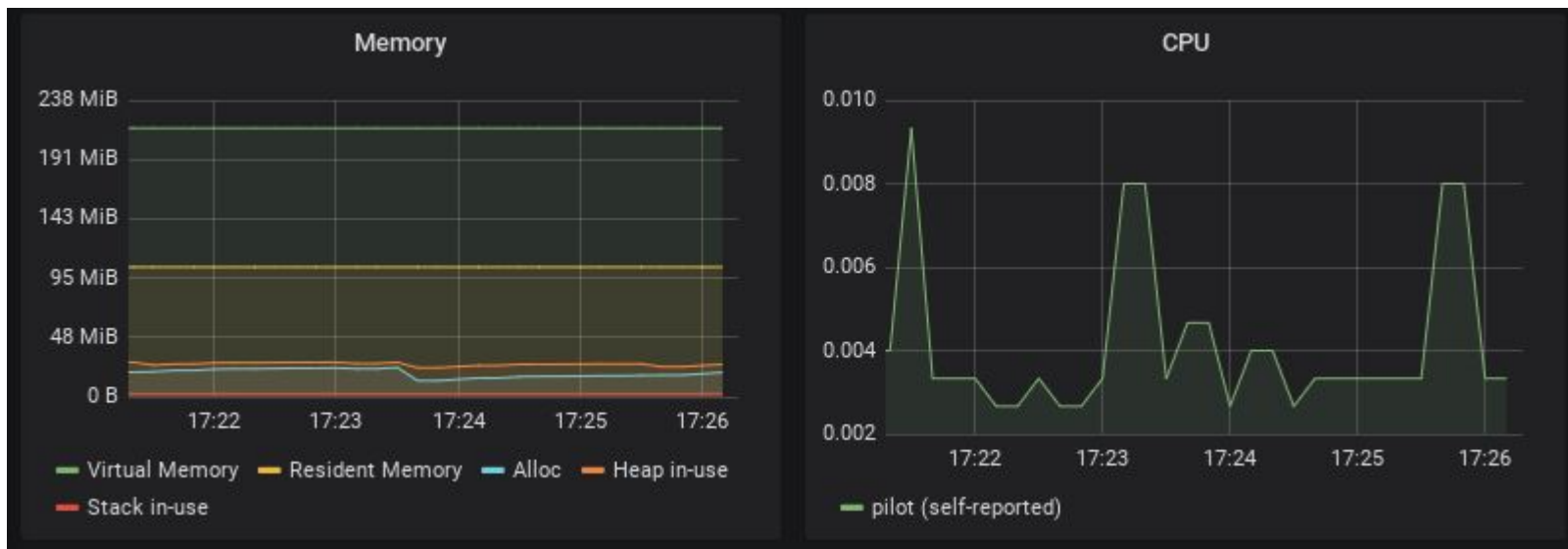
Ties the instance and handler

Handler

Formatter for datastore



Control Plane Metrics



OpenShift Service Mesh vs Istio

Supportability: Install with the OpenShift ServiceMesh Operator	Security: Reduce cluster-scope privileges in control plane
Control: Opt-in model for automatic sidecar injection	Maintainability: Replace BoringSSL with OpenSSL

Demo

- Hipster Shop
- Kiali Mesh Visualization
- Grafana Dashboards
- Prometheus
- OCP Console

Try it yourself

<https://github.com/GoogleCloudPlatform/microservices-demo>


Questions?

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](https://www.linkedin.com/company/red-hat)

 [facebook.com/redhatinc](https://www.facebook.com/redhatinc)

 [youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)

 twitter.com/RedHat

Distributed Traces

supplemental slide

- Follow a request as it moves from one service to another
 - Where is it getting stuck?
 - Where is the latency?
- Would the excessive tracing data slow down everything?
 - Full control over the amount of trace data being generated
- Istio creates the “spans” for you by automatically injecting headers