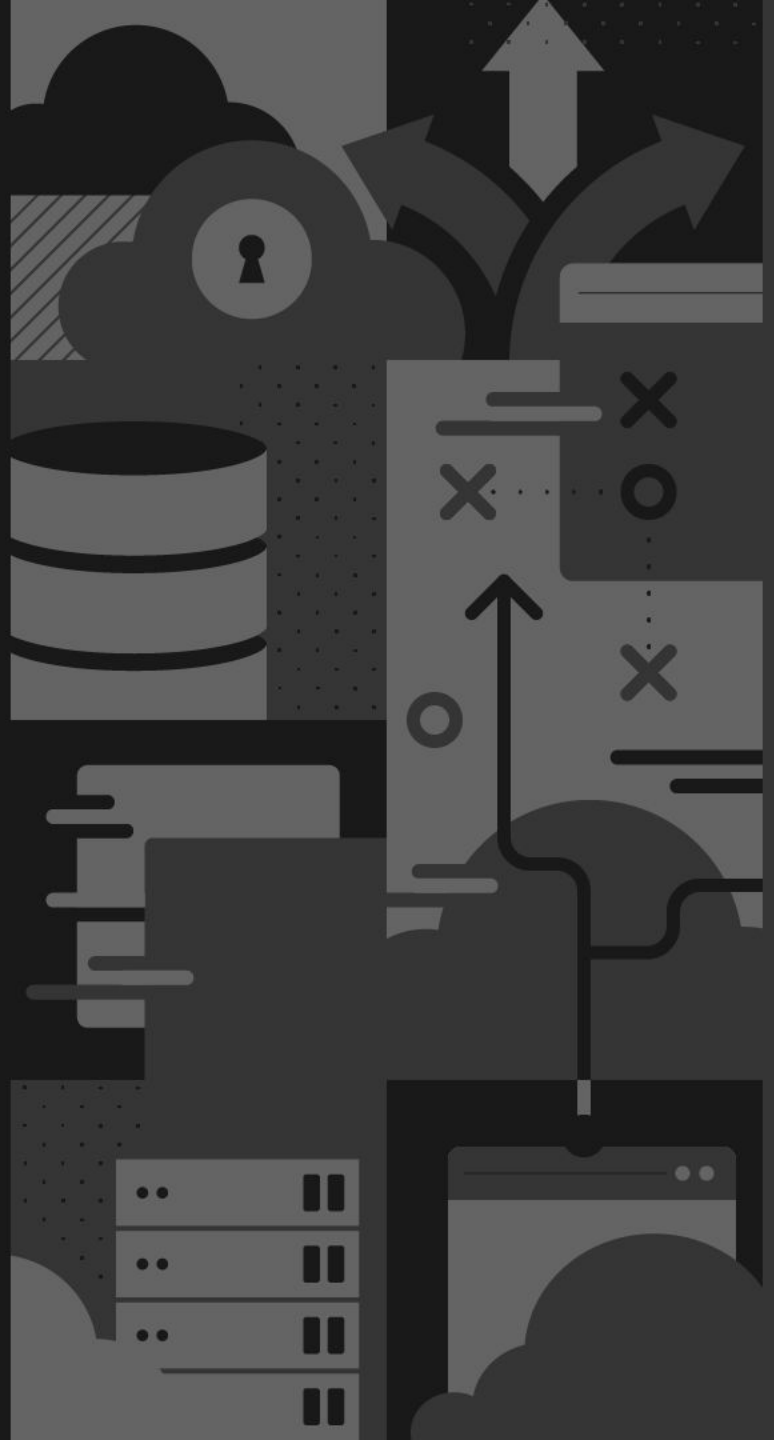


An Introduction to Service Mesh

and a demonstration of it in action

Jason Dudash
Chief Architect / App Platform and Cloud
North America & North America Public Sector

@dudashtweets dudash@redhat.com

Red Hat

Distributed Systems Are Great

There are many “ilities” you can get by distributing your software

- Scalability
- Performance
- Reliability
- Resiliency
- Extensibility
- Availability

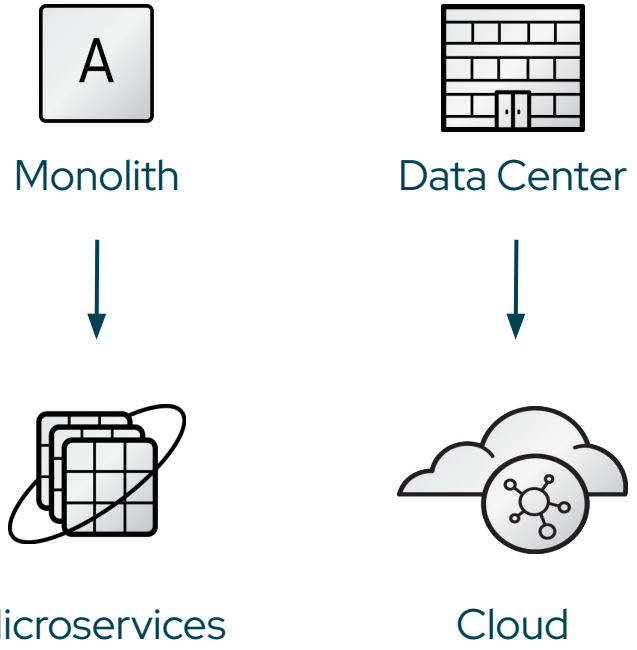
Red Hat

But Distributed Systems Are Hard

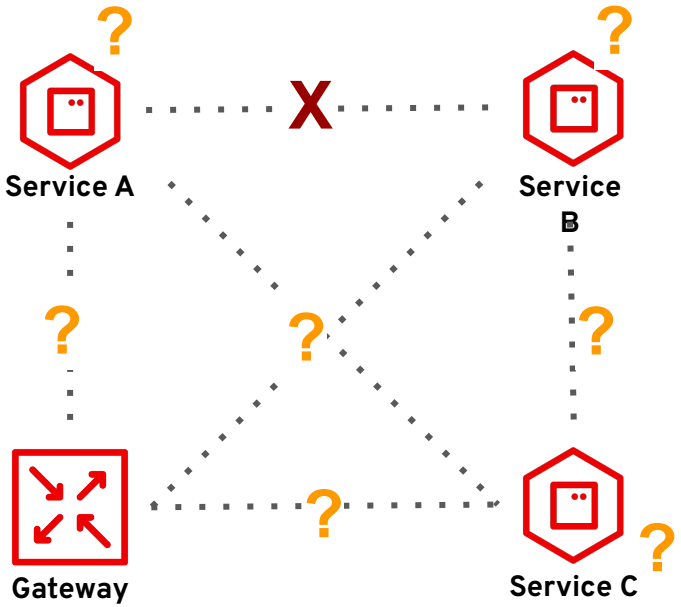
Fallacies of Distributed Computing*

1. The network is reliable.
2. Latency is zero.
3. Bandwidth is infinite.
4. The network is secure.
5. Topology doesn't change.
6. There is one administrator.
7. Transport cost is zero.
8. The network is homogeneous.

Microservices are Trending



Dev/Test/QA != Production



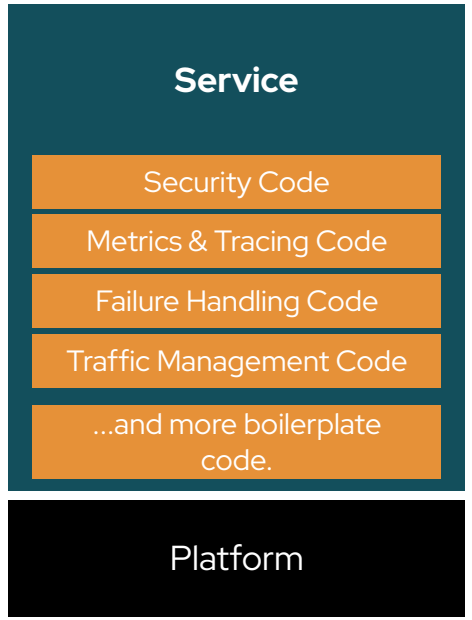
* https://en.wikipedia.org/wiki/Fallacies_of_distributed_computing

Red Hat

Dealing with These Hard Challenges?

- These challenges are often mitigated with:
  - **Code** to handle failures between services
  - Logs, metrics and traces in **source code**
  - **3rd party libraries** for managing deployments, security and more
- A wide range of open source libraries exist to managing these challenges (Netflix are best known)
- This results in:
  - Different solutions in different services
  - Boilerplate code
  - New dependencies to keep up date

Is this a Microservice?

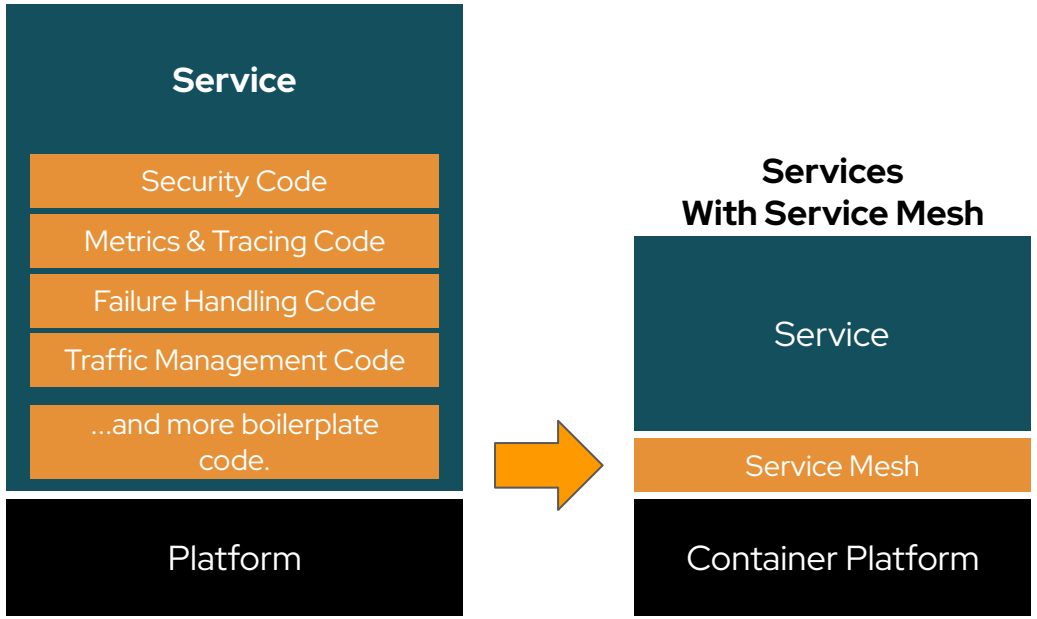


Red Hat

Service Mesh Does This at a Platform Level

which moves the responsibility from the microservices to the infrastructure

- Service Mesh addresses distributed systems challenges at **a common infrastructure layer.**
- This reduces boilerplate code and copy/paste errors across services.
- Enforces common policies across all services.
- Removes the obligation to implement cross cutting concerns from developers.



Red Hat

Analogy:

Roads and Traffic Management

- If the **network represents the roads** between our services, then **service mesh represents traffic control**
- You don’t need traffic controls on closed circuits, private property or a race track
- Most towns and cities do need controls!
- Traditional networking approach:
  - Checkpoints
- Service Mesh approach:
  - Police, speed limits, traffic lights, lanes for cars, bikes, pedestrians

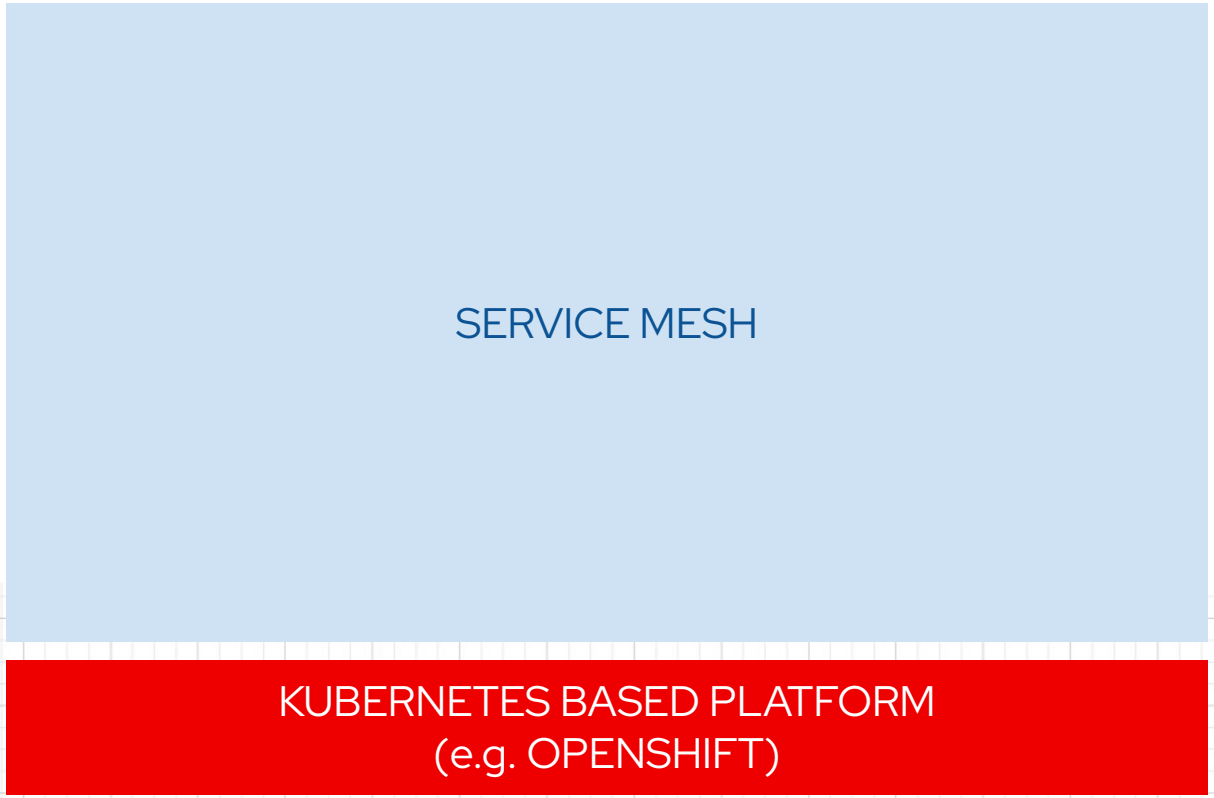
Under the Hood

High Level on the “How” of Service Mesh*

* for Istio based service meshes

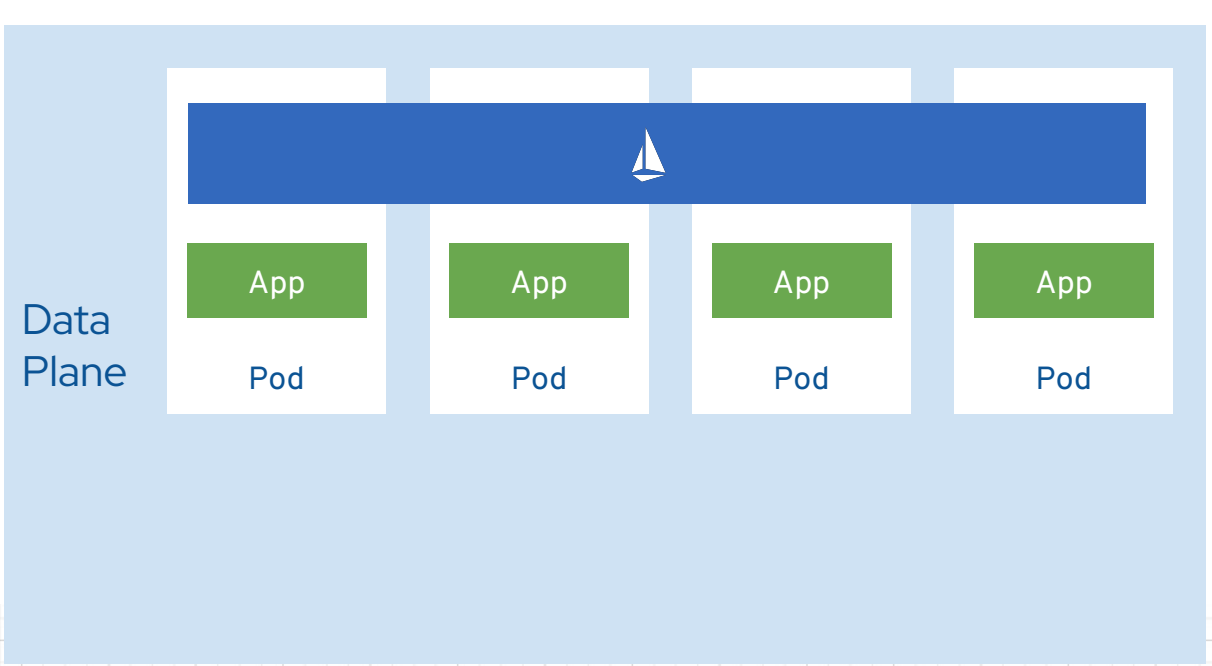
Red Hat

A Platform Capability On Top of Kubernetes



Red Hat

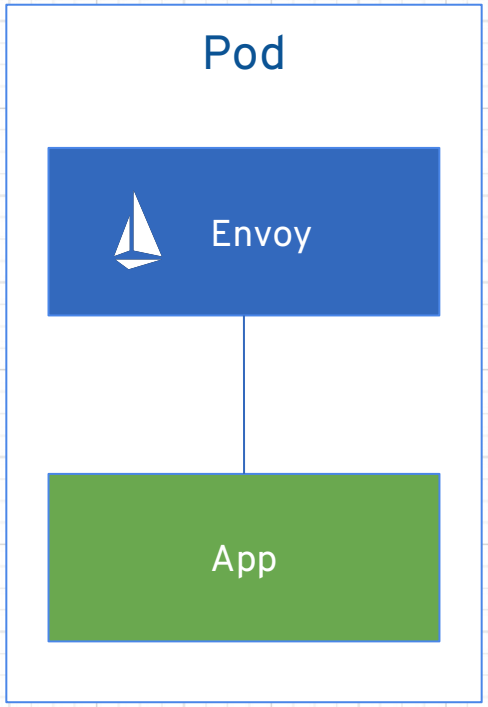
Your Services are in a “Data Plane”



Red Hat

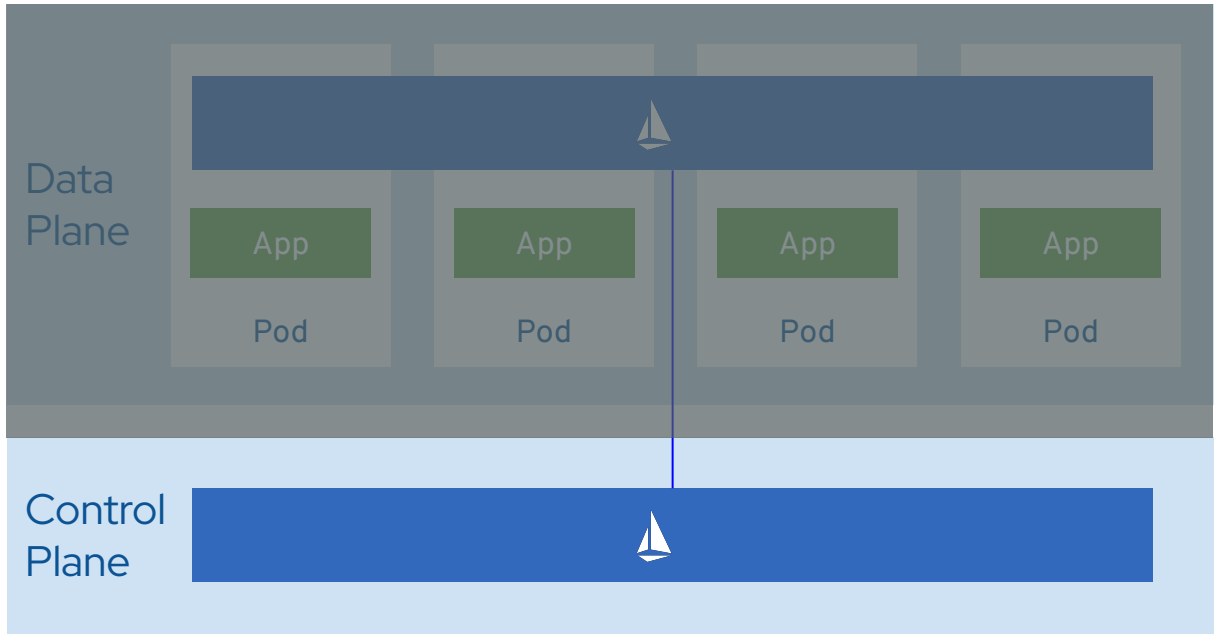
The Dataplane Uses “sidecars”

a platform container co-located with your services



Red Hat

Your Policy is in a “Control Plane”



Red Hat

With this architecture you can dynamically configure services **without code changes** and **without redeploying services**

Demonstrations

Observability

Demo 1

Service Mesh dashboard (Kiali),
Metrics visualizations (Grafana),
Distributed tracing (Jaeger)

Red Hat

Traffic Management

Demo 2

Control traffic routes,
Advanced deployments,
Circuit breakers,
and more

Red Hat

Security

Demo 3

Service to service
encryption via mutual TLS,
Service identity,
Authn / Authz

Red Hat

Service mesh with Istio and Kiali

Everything you need to coordinate microservices in a service mesh with the powerful monitoring and management tools like Istio and Kiali.

Istio: Canaries and Kubernetes

In this session, we will introduce you to cloud native architecture by demonstrating numerous principles and techniques for building and deploying Java microservices

Chief Developer Evangelist

Latest articles on service mesh, Istio, and Kiali

Kubernetes: The evolution of distributed systems

Looking up a hash table library for caching in the

https://developers.redhat.com/topics/service-mesh

Distributed microservices architecture: Istio, managed API gateways and, enterprise integration

Solving the challenges of debugging microservices on a container platform

Observe what your Istio microservices mesh is doing with Kiali

More service mesh articles

Tutorial

Istio on Kubernetes: Enter the Service Mesh

Red Hat

Start tutorial →

Lesson

Introduction to Istio service mesh

Red Hat

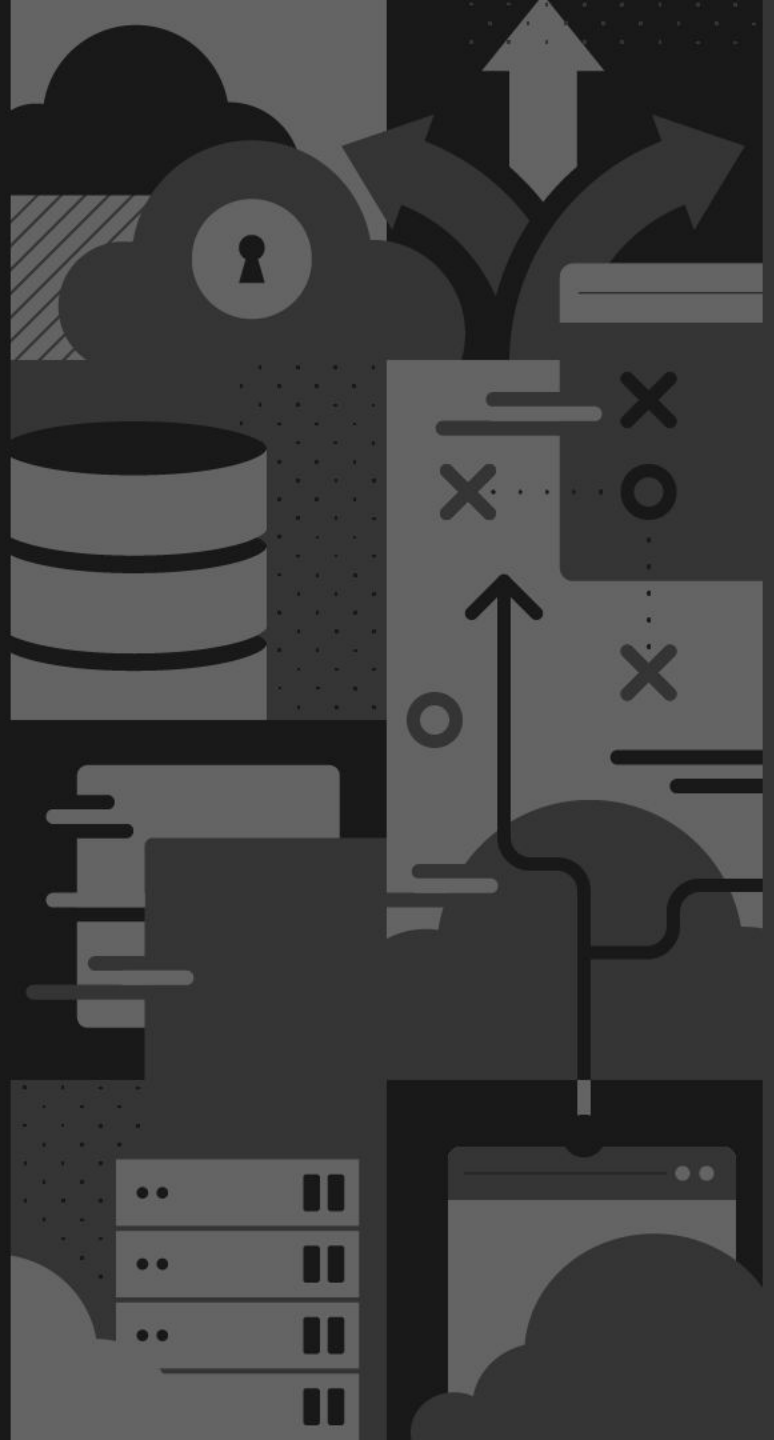
Watch video →

Document

Build sample microservices application with Istio

Istio Authors

Read doc →



Thank you

If you want to continue the discussion, you can find my social accounts using this QR code. And you can find out more about Red Hat using the links below.

linkedin.com/company/red-hat

facebook.com/redhatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat

github.com/dudash

@dudashtweets

Red Hat