KubeCon | CloudNativeCon

Europe 2018

Performance and Scale @Istio Service Mesh

Surya V Duggirala, IBM

Laurent Demailly, Google

Fawad Khaliq, VMWare

About the speakers

Surya V Duggirala

suryadu@us.ibm.com / @Duggirala1

STSM, IBM Watson and Cloud Architecture & Performance Engineering

IBM

Laurent Demailly

ldemailly@google.com / @LaurentDemailly

Staff Engineer

Google

Fawad Khaliq

fkhaliq@vmware.com / @fawadkhaliq

Senior Software Engineer

VMWare

Agenda

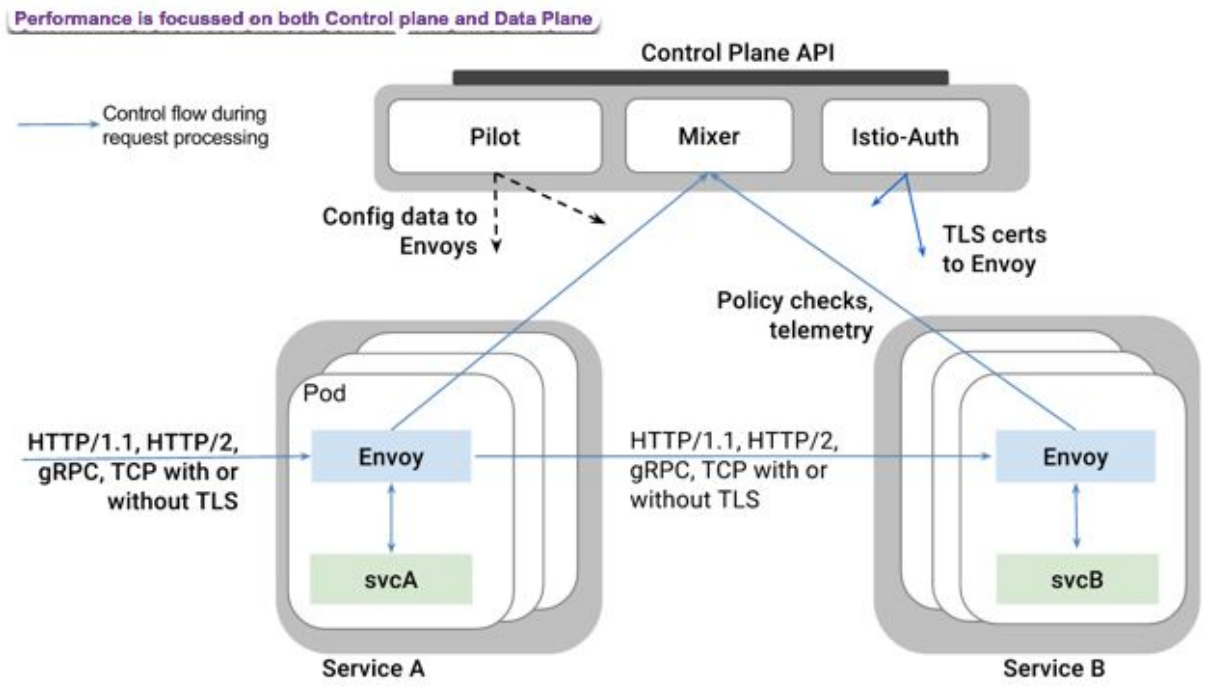
- Introduction of Istio Performance and Scalability working group
- Our multi-pronged approach
- Performance Environments and Scenarios
- Performance Characterizations, Issues fixed and Results
- Performance across Multiple Industry Use Cases
- Istio performance/scalability next steps
- Q&A

Istio Performance Workgroup

What is Istio Service Mesh?

- Istio is an open platform that provides an uniform way to connect, manage and secure microservices
- Istio Service Mesh offers the following key features for Microservices
  - Intelligent Routing and Load Balancing
  - Resilience across Languages and Platforms
  - Fleet Wide Policy Enforcement
  - In-depth Telemetry and Reporting
  - Secure service-to-service authentication with strong identity assertions between services in a cluster
- Istio can be deployed on multiple Cloud Platforms
  - Kubernetes
  - Nomad with Console
  - Cloud Foundry (Future)
  - Apache Mesos (Future)

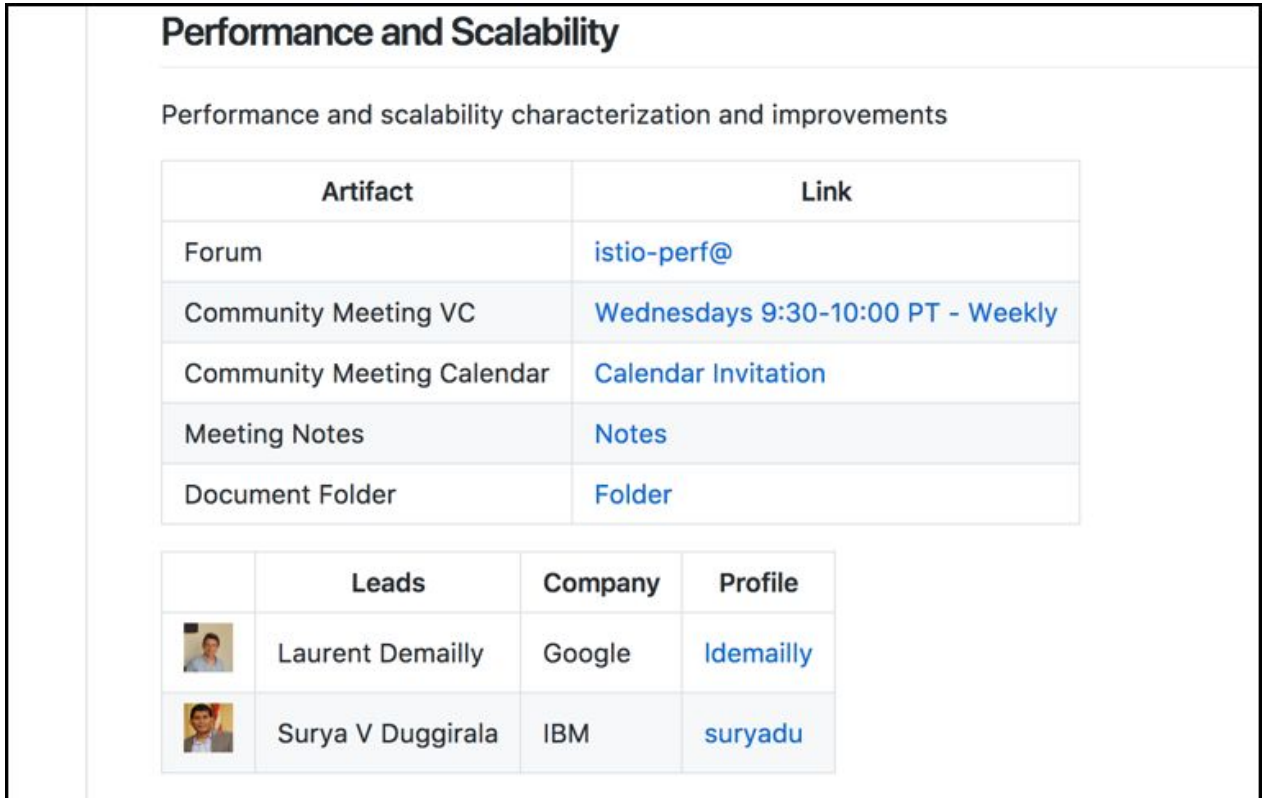
Istio Service Mesh High Level Architecture



Istio Architecture

- Istio Data Plane consists of intelligent proxies (Envoy) deployed as sidecars that mediate and control all network communication between microservices
- Istio Control Plane is responsible for managing and configuring proxies to route traffic, as well as enforcing policies at runtime
- Istio has a main design goal of making the system capable of dealing with services at scale with high performance

Istio Service Mesh Performance Workgroup



Performance and Scalability

Performance and scalability characterization and improvements

Artifact	Link
Forum	istio-perf@
Community Meeting VC	Wednesdays 9:30-10:00 PT - Weekly
Community Meeting Calendar	Calendar Invitation
Meeting Notes	Notes
Document Folder	Folder

	Leads	Company	Profile
	Laurent Demailly	Google	ldemailly
	Surya V Duggirala	IBM	suryadu

https://github.com/istio/community/blob/master/WORKING-GROUPS.md#performance-and-scalability

Our approach

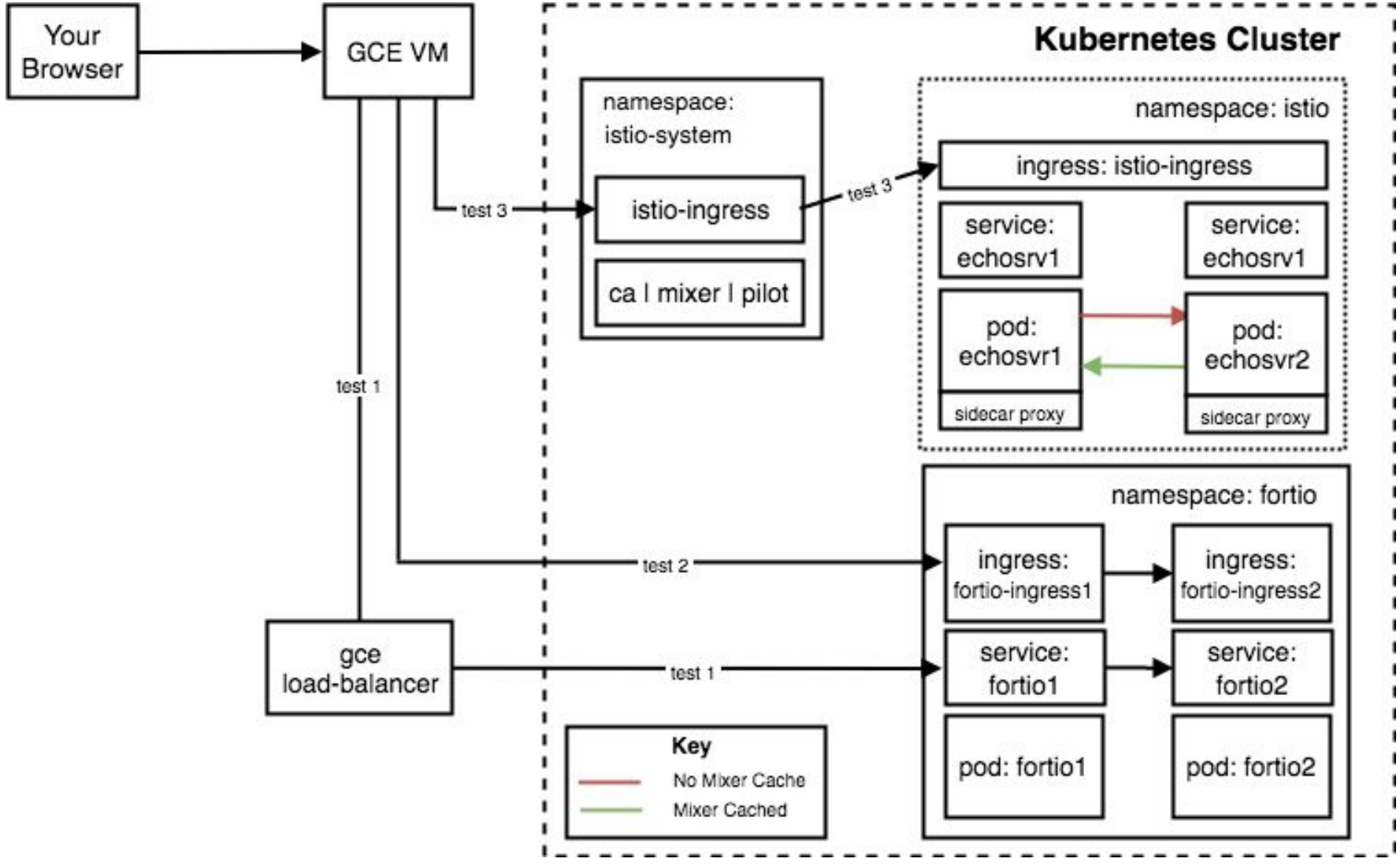
Istio Perf WG: multi pronged approach

- Code level micro benchmarks (go benchmarks,...)
- Synthetic benchmark
  - Fortio
- Industry representative benchmark
  - Blueperf
- “Scaling” performance characterization through the community
  - Reusable tooling
  - Multiple scenarios
  - Multiple dimensions
- Automation
  - Regpatrol
  - Fortio

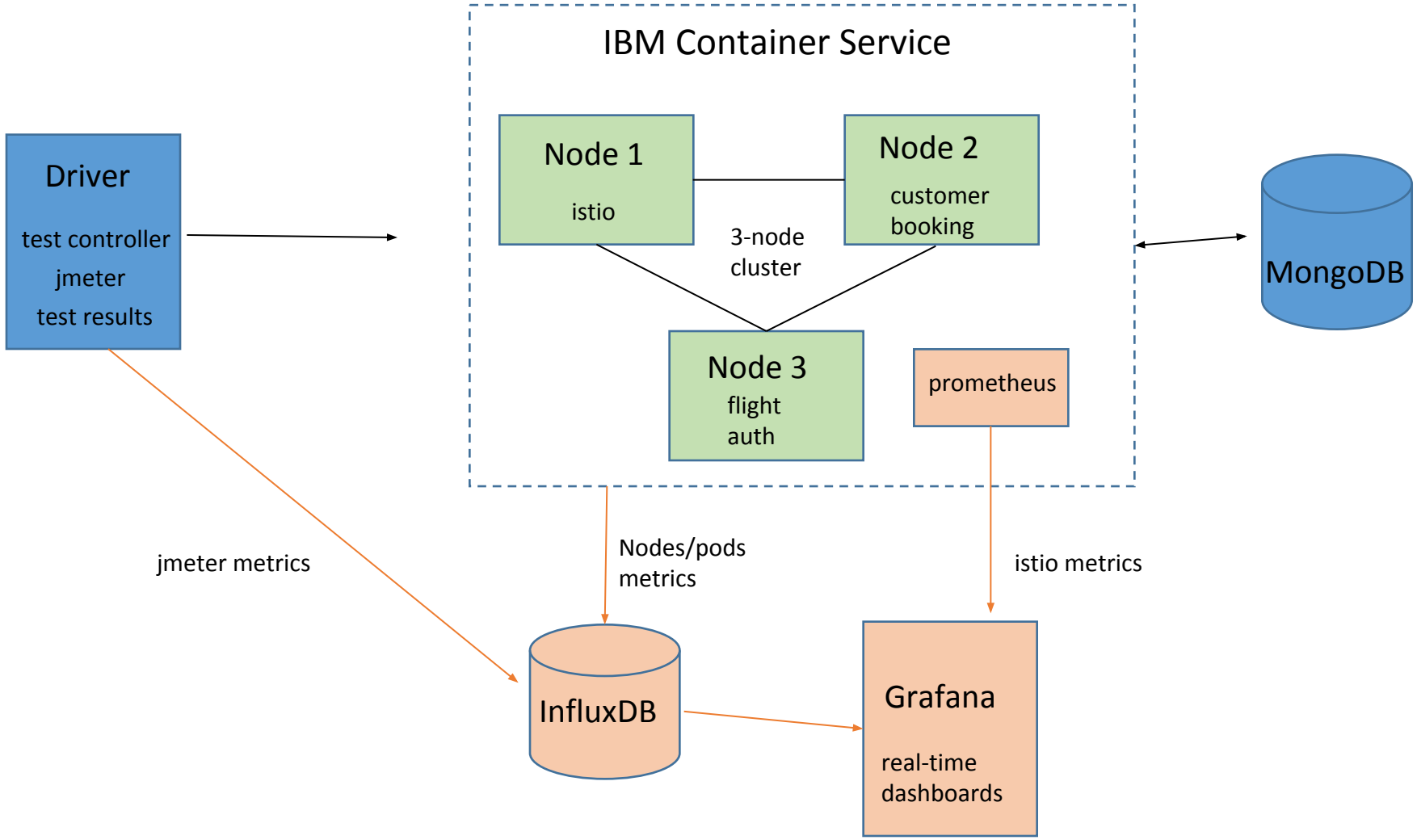
Performance environments & scenarios

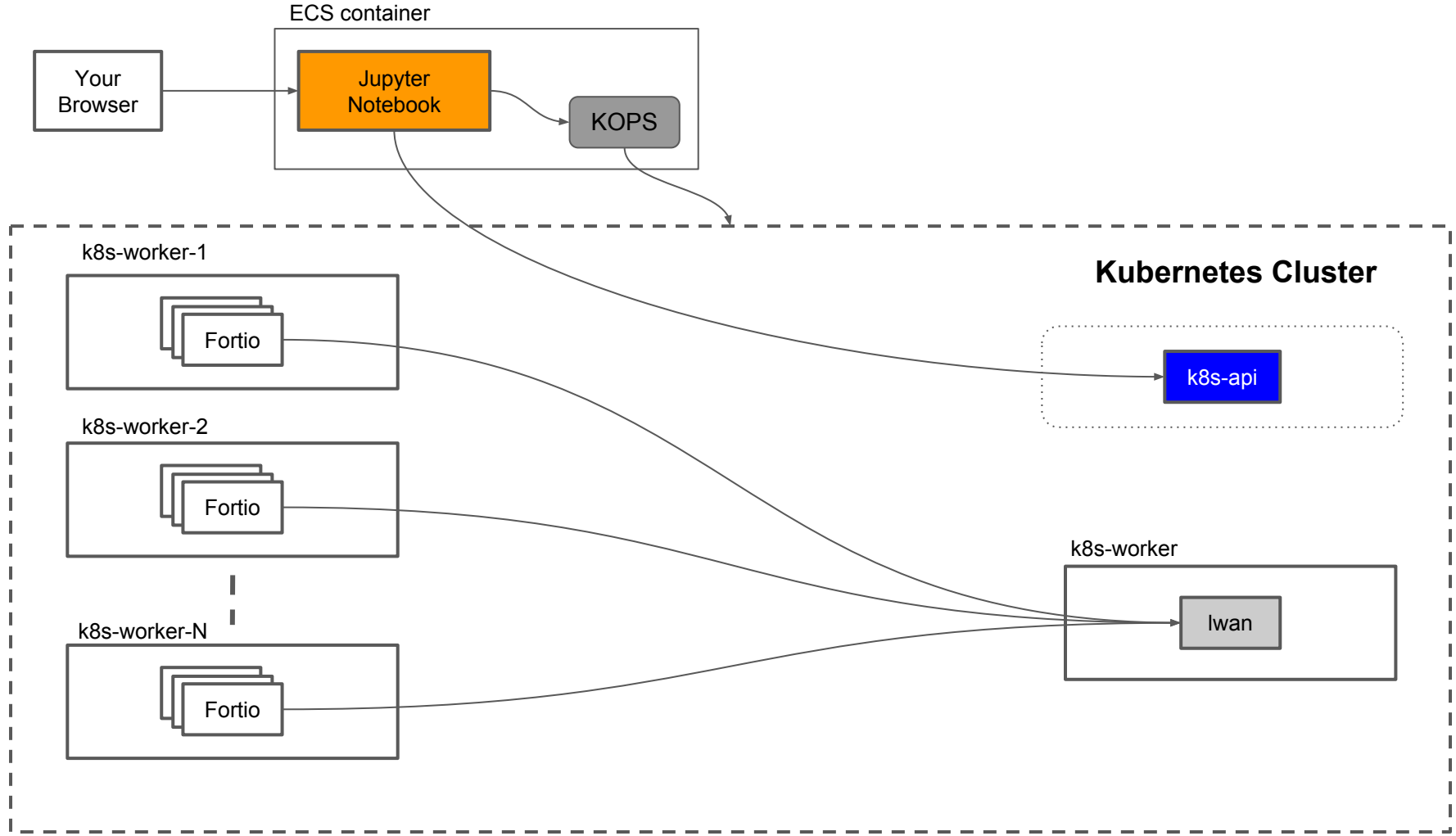
GKE, IBM Cloud, AWS, Azure and on-prem

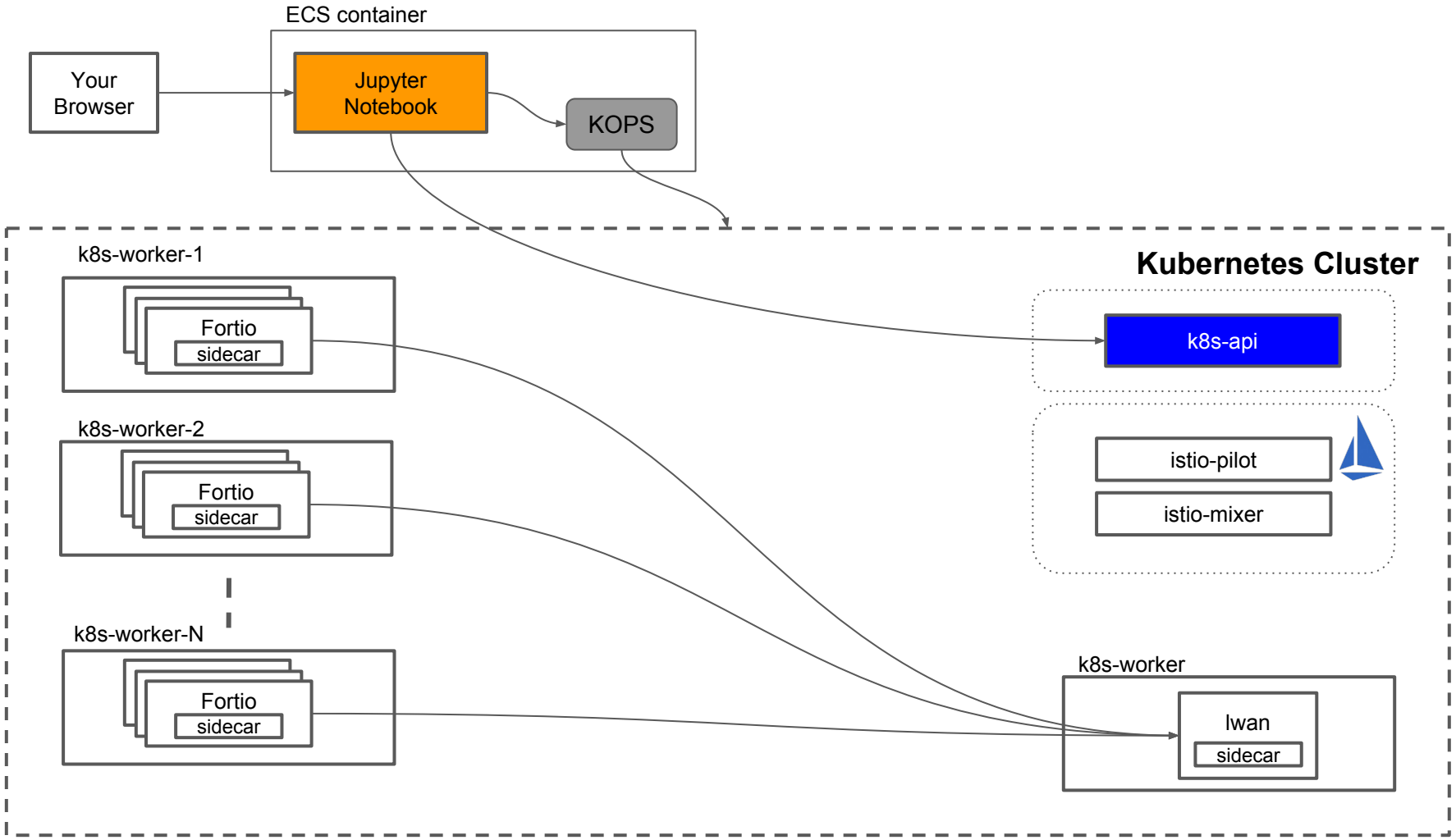
GCE/GKE

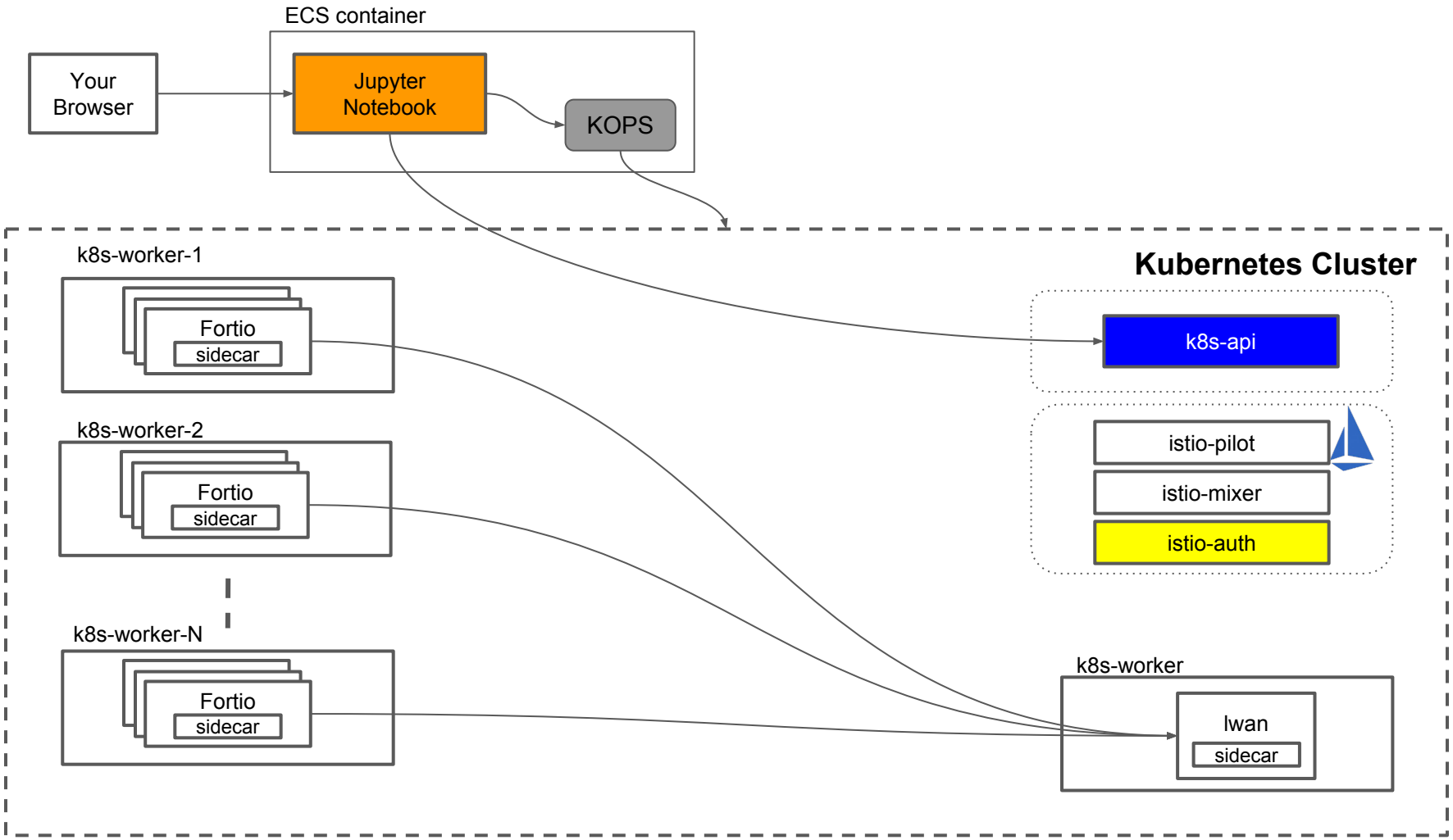

Your Browser
GCE VM
Kubernetes Cluster
namespace: istio-system
istio-ingress
ca | mixer | pilot
test 3
test 3
namespace: istio
ingress: istio-ingress
service: echosrv1
service: echosrv1
pod: echosvr1
pod: echosvr2
sidecar proxy
sidecar proxy
test 1
test 2
namespace: fortio
ingress: fortio-ingress1
ingress: fortio-ingress2
service: fortio1
service: fortio2
pod: fortio1
pod: fortio2
gce load-balancer
test 1
Key
No Mixer Cache
Mixer Cached

IBM Cloud


IBM Container Service
Driver
test controller
jmeter
test results
Node 1
istio
Node 2
customer
booking
3-node
cluster
Node 3
flight
auth
prometheus
MongoDB
jmeter metrics
Nodes/pods
metrics
istio metrics
InfluxDB
Grafana
real-time
dashboards

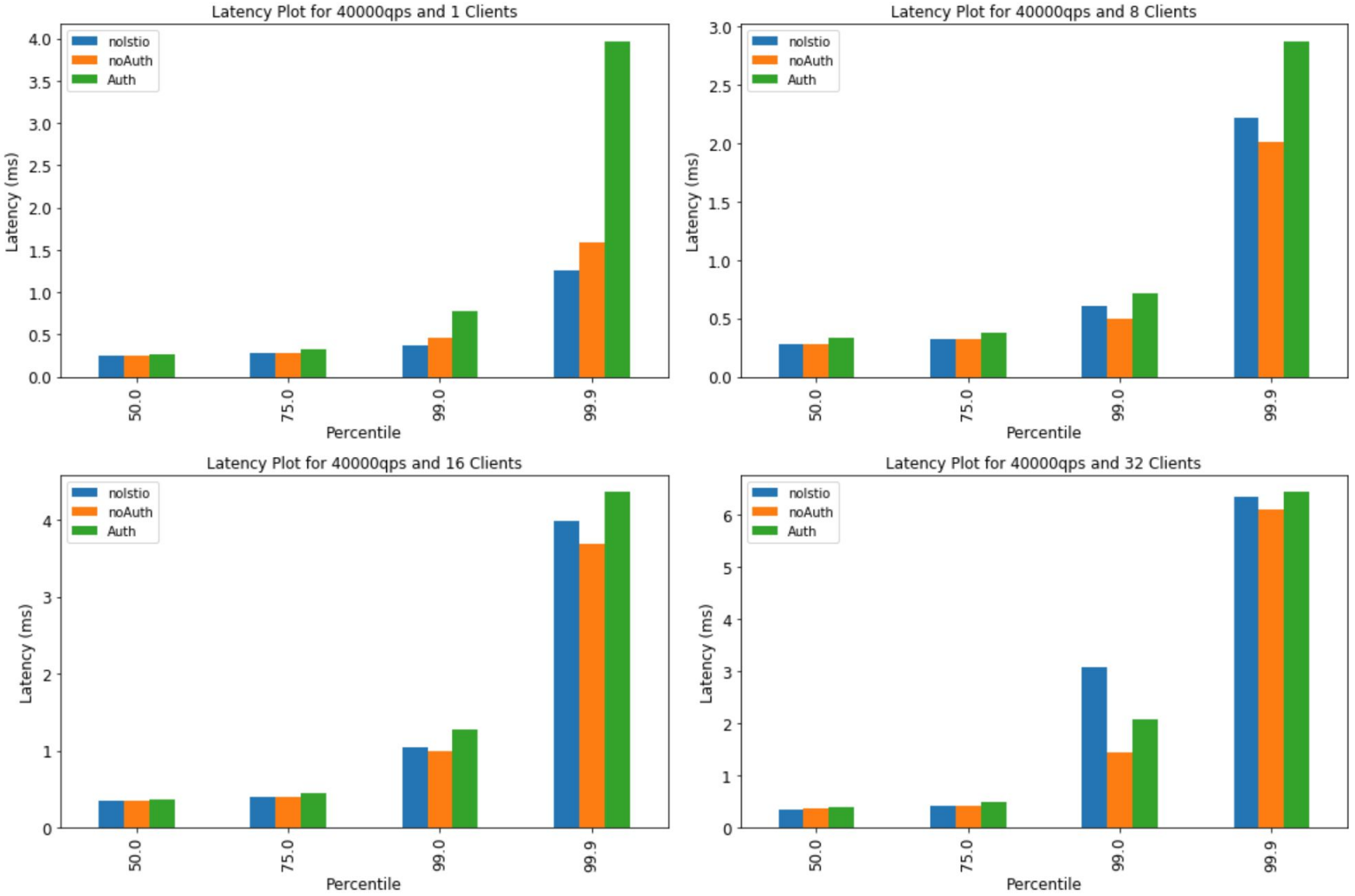

ECS container
Your Browser
Jupyter Notebook
KOPS
Kubernetes Cluster
k8s-worker-1
Fortio
k8s-worker-2
Fortio
k8s-worker-N
Fortio
k8s-api
k8s-worker
Iwan

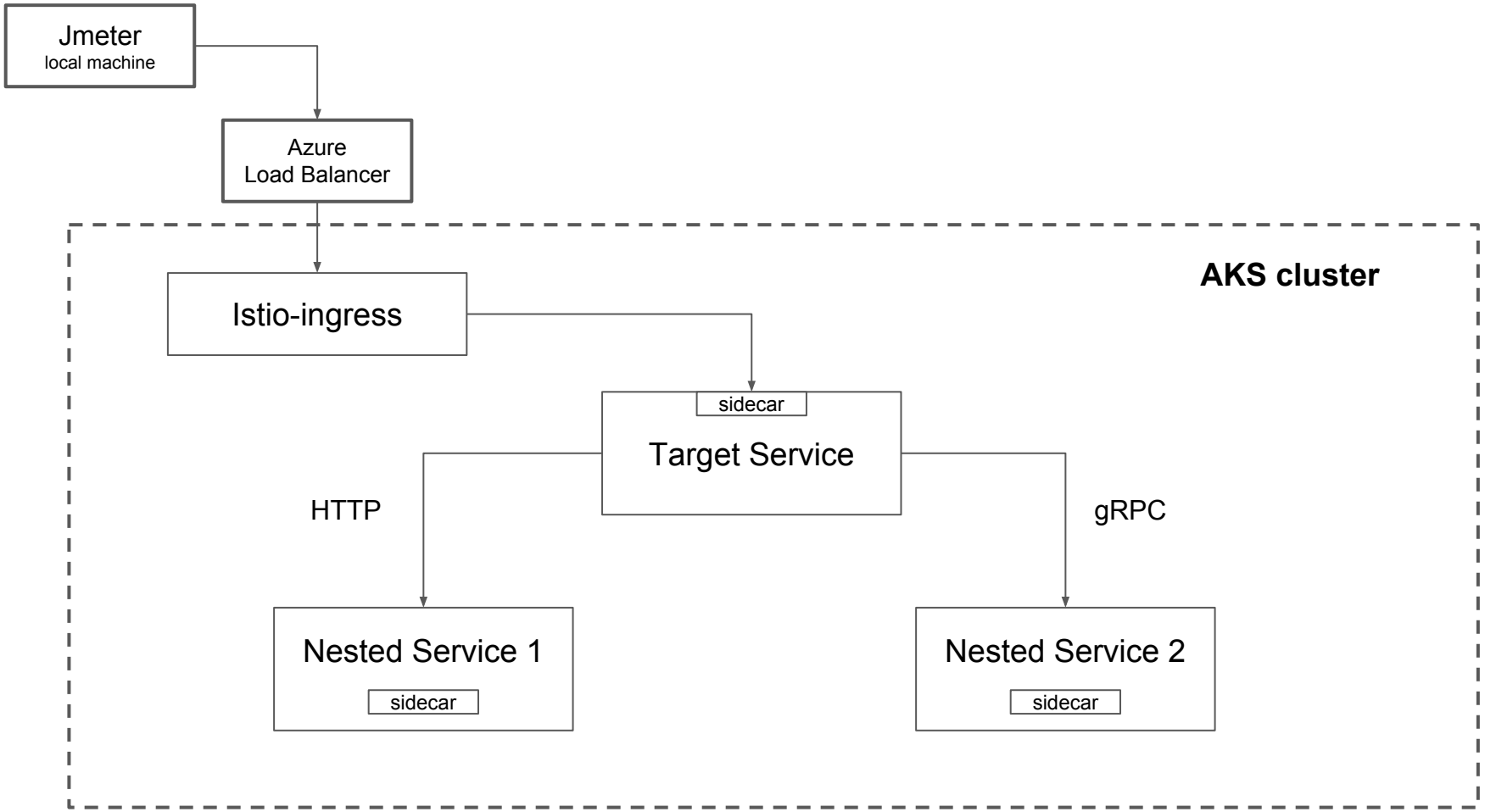

ECS container
Your Browser
Jupyter Notebook
KOPS
Kubernetes Cluster
k8s-worker-1
Fortio
sidecar
k8s-worker-2
Fortio
sidecar
k8s-worker-N
Fortio
sidecar
k8s-api
istio-pilot
istio-mixer
k8s-worker
Iwan
sidecar

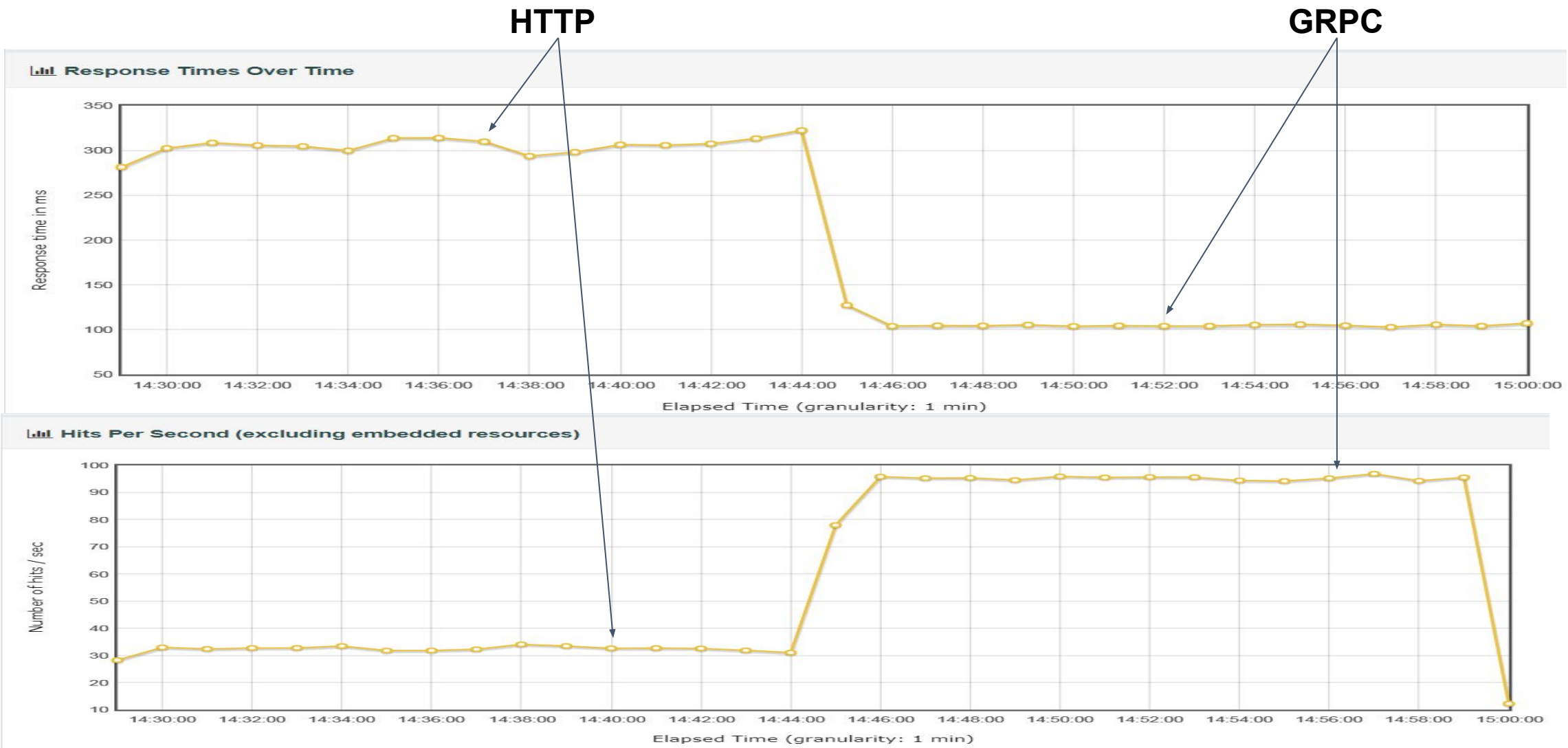

ECS container
Your Browser
Jupyter Notebook
KOPS
Kubernetes Cluster
k8s-worker-1
Fortio
sidecar
k8s-worker-2
Fortio
sidecar
k8s-worker-N
Fortio
sidecar
k8s-api
istio-pilot
istio-mixer
istio-auth
k8s-worker
Iwan
sidecar

AWS

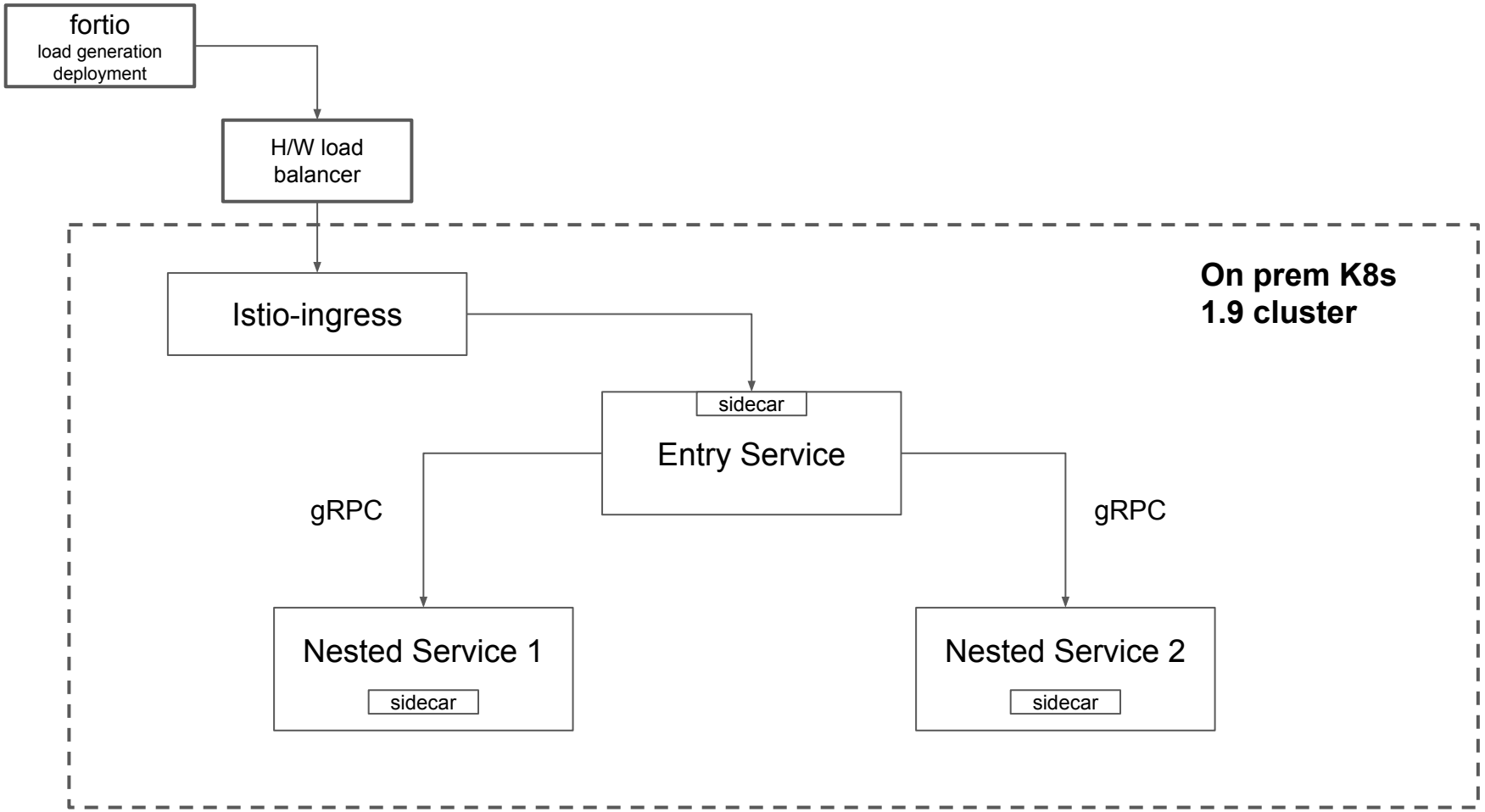
- fortio(s)->lwan
- pod to pod communication



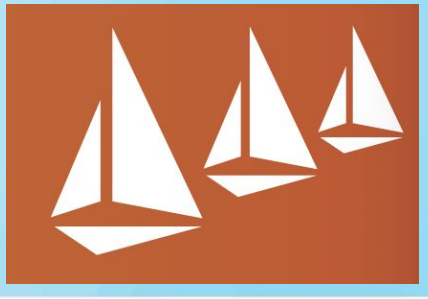

Jmeter
local machine
Azure
Load Balancer
AKS cluster
Istio-ingress
sidecar
Target Service
HTTP
gRPC
Nested Service 1
sidecar
Nested Service 2
sidecar


HTTP
GRPC
Response Times Over Time
Response time in ms
350
300
250
200
150
100
50
14:30:00
14:32:00
14:34:00
14:36:00
14:38:00
14:40:00
14:42:00
14:44:00
14:46:00
14:48:00
14:50:00
14:52:00
14:54:00
14:56:00
14:58:00
15:00:00
Elapsed Time (granularity: 1 min)
Hits Per Second (excluding embedded resources)
Number of hits / sec
100
90
80
70
60
50
40
30
20
10
14:30:00
14:32:00
14:34:00
14:36:00
14:38:00
14:40:00
14:42:00
14:44:00
14:46:00
14:48:00
14:50:00
14:52:00
14:54:00
14:56:00
14:58:00
15:00:00
Elapsed Time (granularity: 1 min)

On-prem


fortio
load generation
deployment
H/W load
balancer
Istio-ingress
On prem K8s
1.9 cluster
sidecar
Entry Service
gRPC
gRPC
Nested Service 1
sidecar
Nested Service 2
sidecar

Istio Performance Characterization

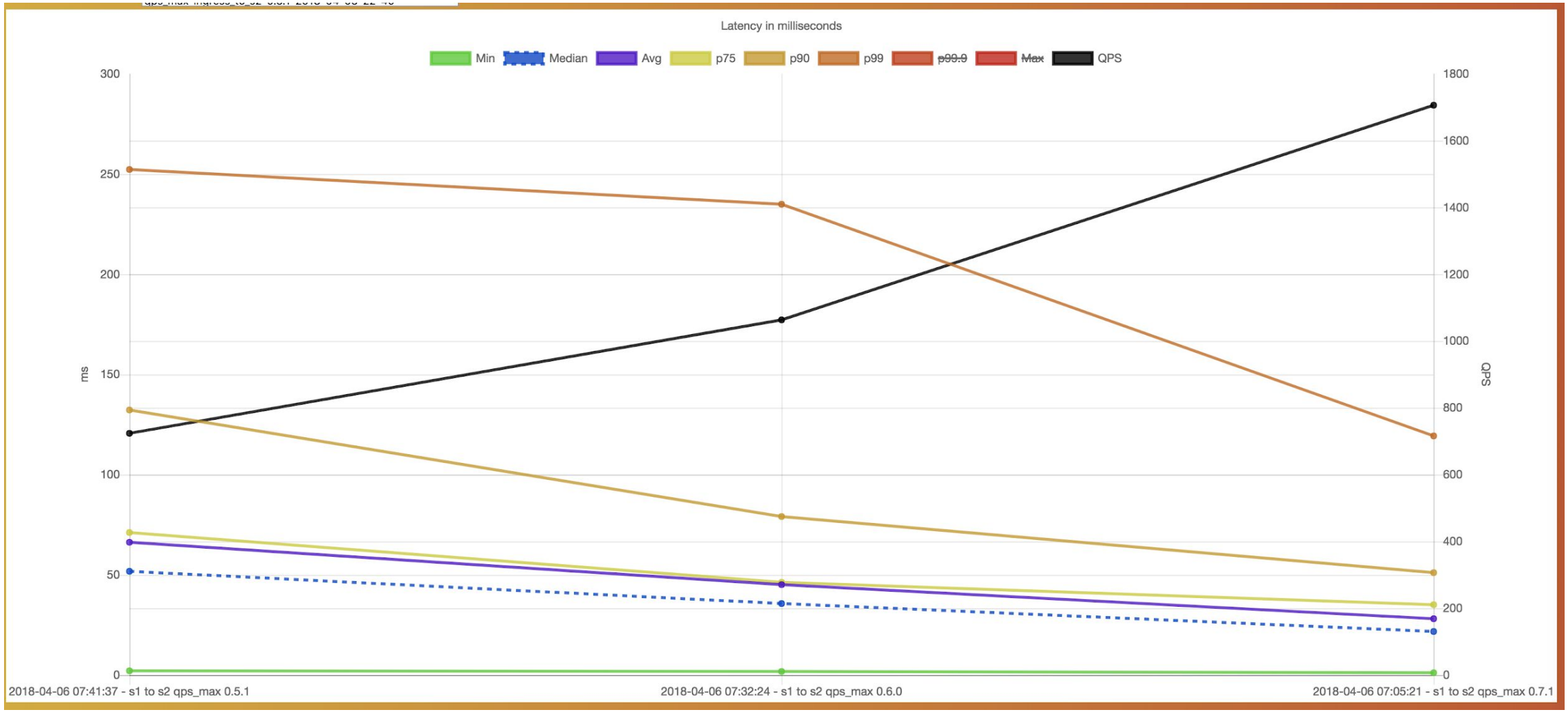


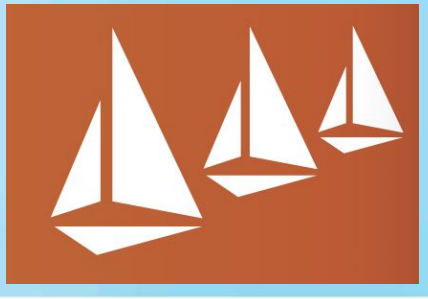
Fortio

- Lightweight (no external dependencies beside golang and optional gRPC) - 3 Mbytes image.
- Go library (used in e2e tests for functional checks: run N requests, check the result codes, metrics)
- Command line and docker image (istio/fortio) advanced echo server (similar to httpbin features and more)
- Runs at a set (lower than max) QPS for meaningful latency data.
- Simple graphing/data visualization/exploration
- High (enough) performance: >400K qps single node self test

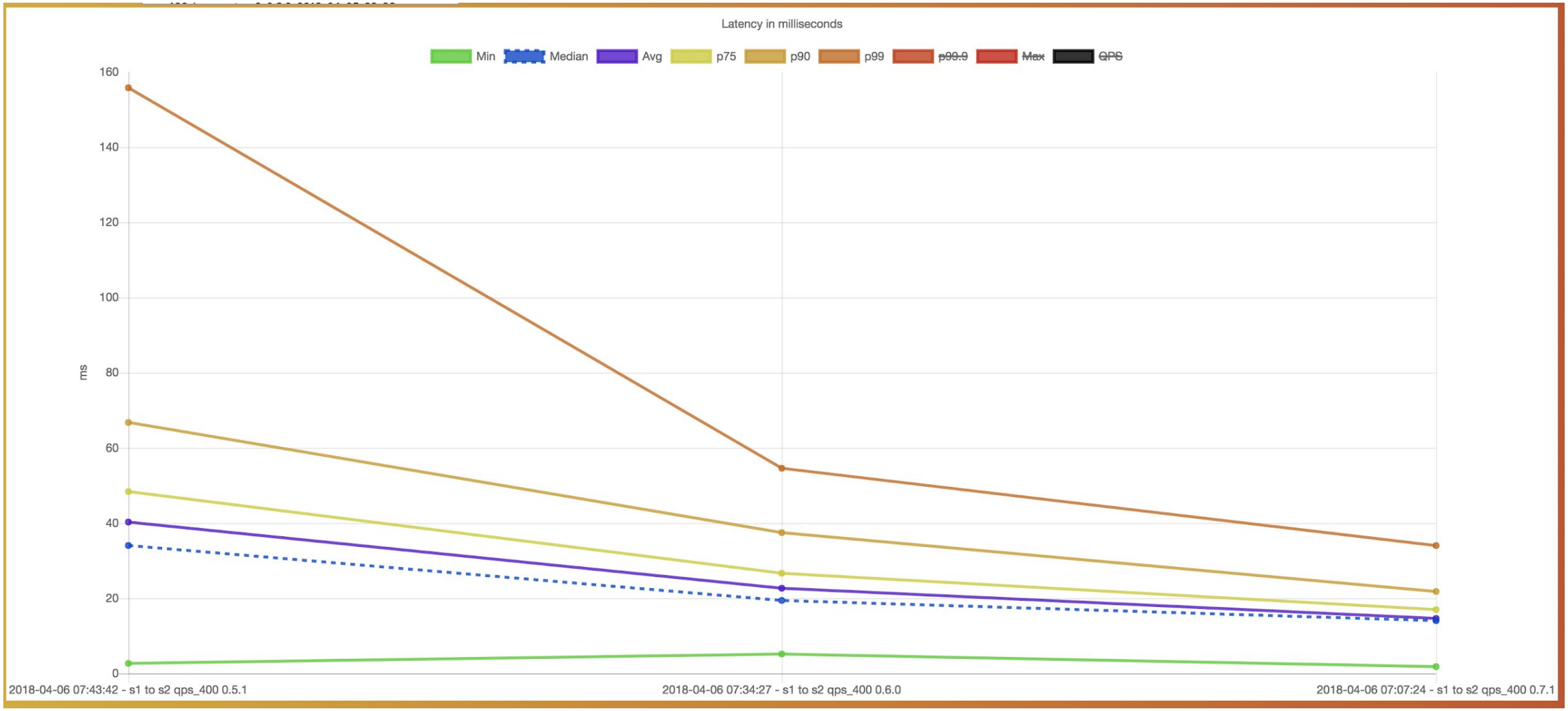
Fortio: max qps improvements

KubeCon CloudNativeCon Europe 2018

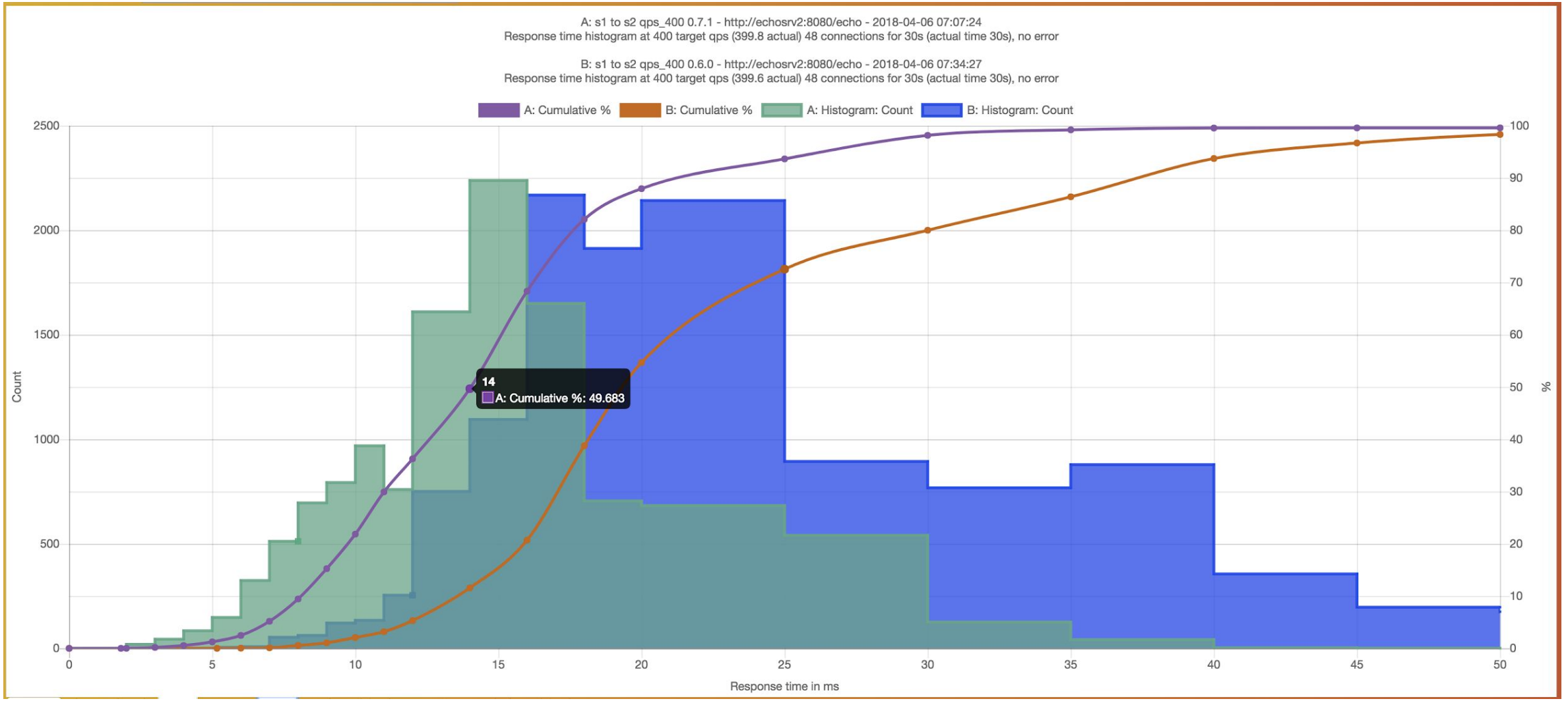




Fortio: 400 qps latencies



Fortio: 400 qps latencies



Issues we found and fixed

- Excessive logging (IOs + serialization)
- Memory leak / go routine contention in zipkin library
- Double mixer calls due to proxy filter misconfiguration
- Envoy (m)TLS perf issues
- TCP buffering and half close issues
- Mixer rule short circuiting
- Mixer cache key issue at ingress
- Pilot scale with hundreds of services (excessive memory and cpu usage)
- Mixer client lock contention. (still in progress)
- Mixer cache parameters. (not yet fixed)

Performance Summary

Over the last 3 monthly Istio releases:

(two of many scenarios using fortio)

- p50 (median) latencies at fixed, moderate 400 qps, dropping 34ms (0.5.1) → 19ms (0.6.0) → **14ms** (0.7.1)

- max qps (2 vCPU) is increasing from 700 (0.5.1) to 1000 (0.6.0) to **1700** (0.7.1)

Absolute numbers are not where we want yet (single digit ms EOY goal, lower overhead) but the trend is in the right direction.

Istio latency improvements

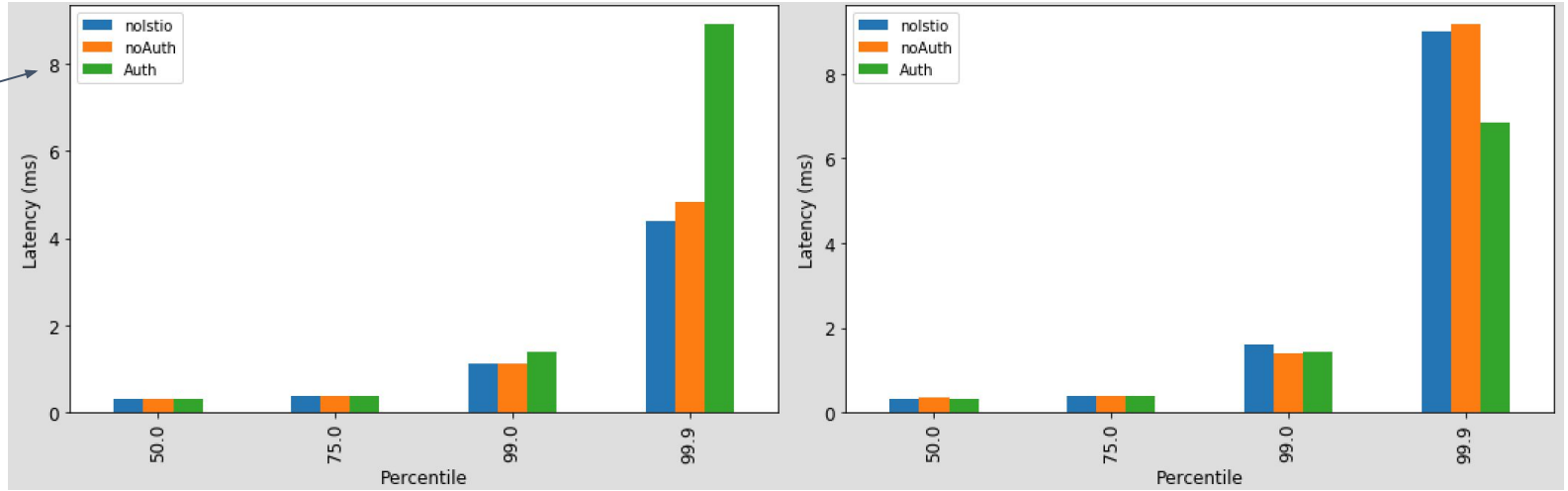
Scenario: 20000 qps with 16 and 32 Fortio instances on AWS **(no Mixer)**

Istio 0.4

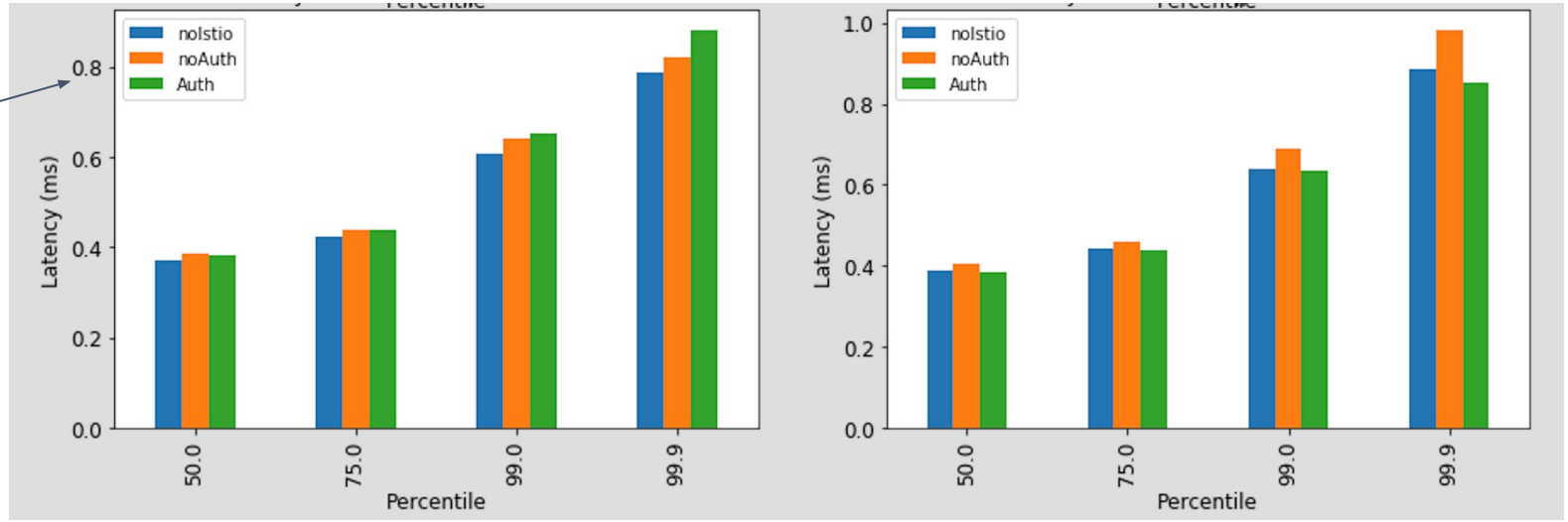
p99 and p99.9 improved by a factor of 10

Istio 0.7.1

8ms

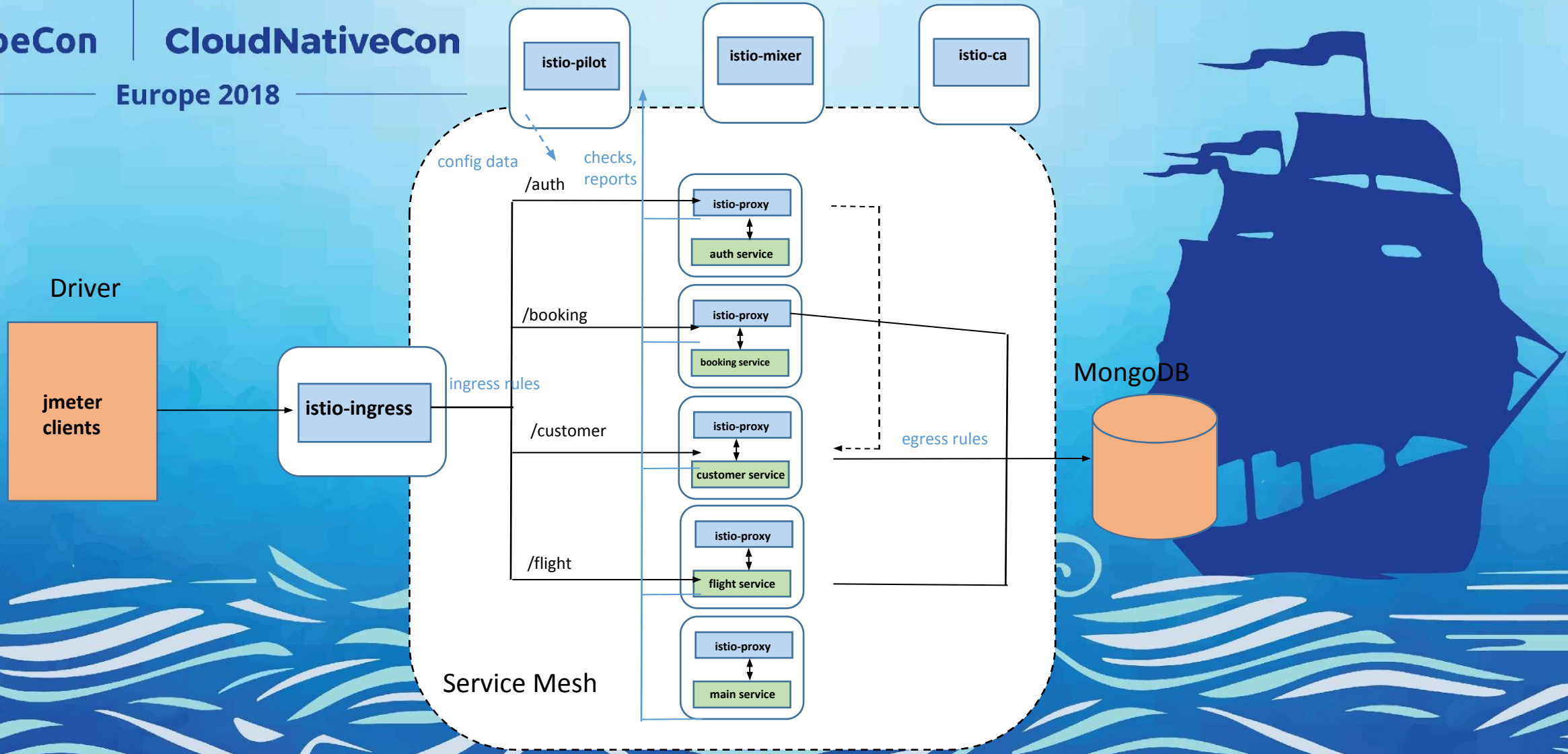


0.8ms

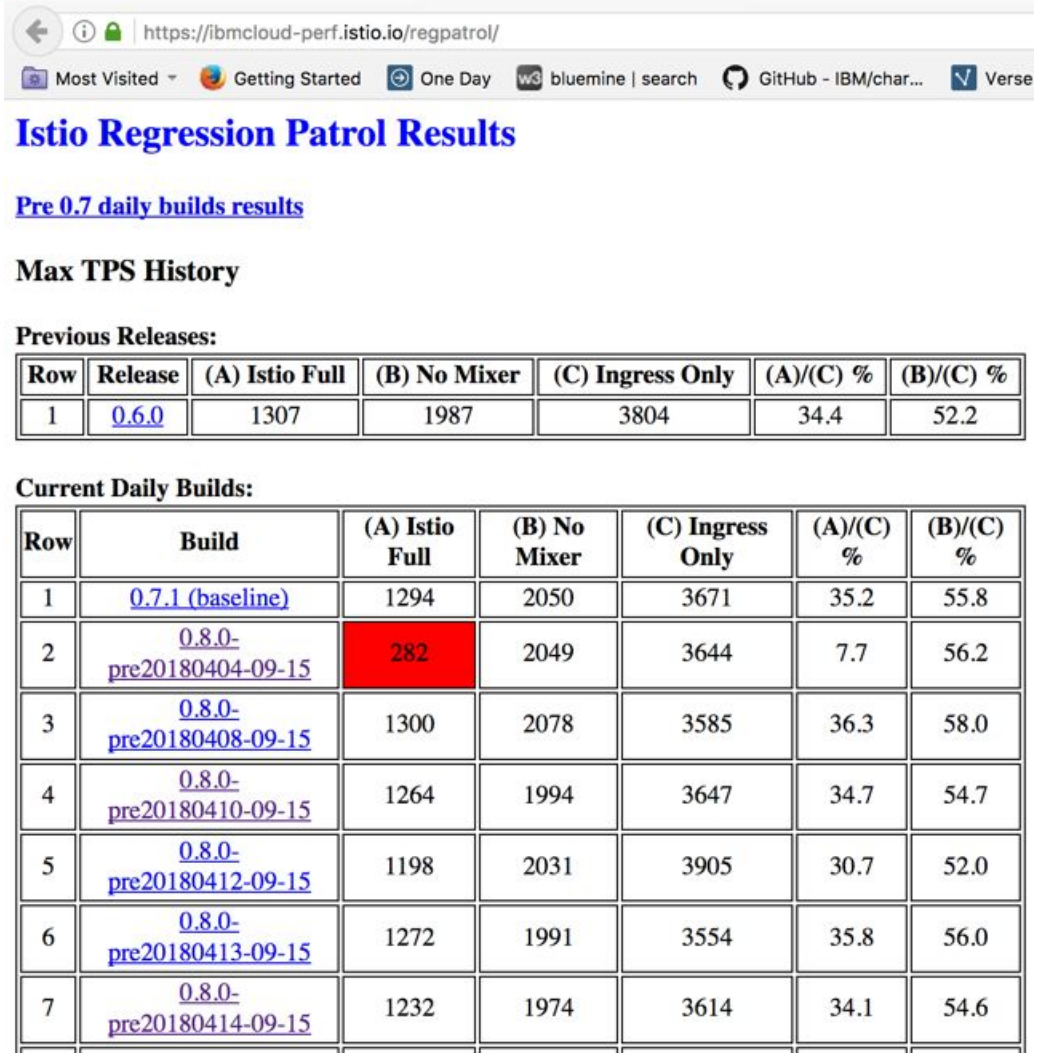


KubeCon CloudNativeCon Europe 2018

Acmeair Microservices and Istio



Istio Regression Patrol Master Dashboard



https://ibmcloud-perf.istio.io/regpatrol/

Most Visited | Getting Started | One Day | bluemine | search | GitHub - IBM/char... | Verse

Istio Regression Patrol Results

Pre 0.7 daily builds results

Max TPS History

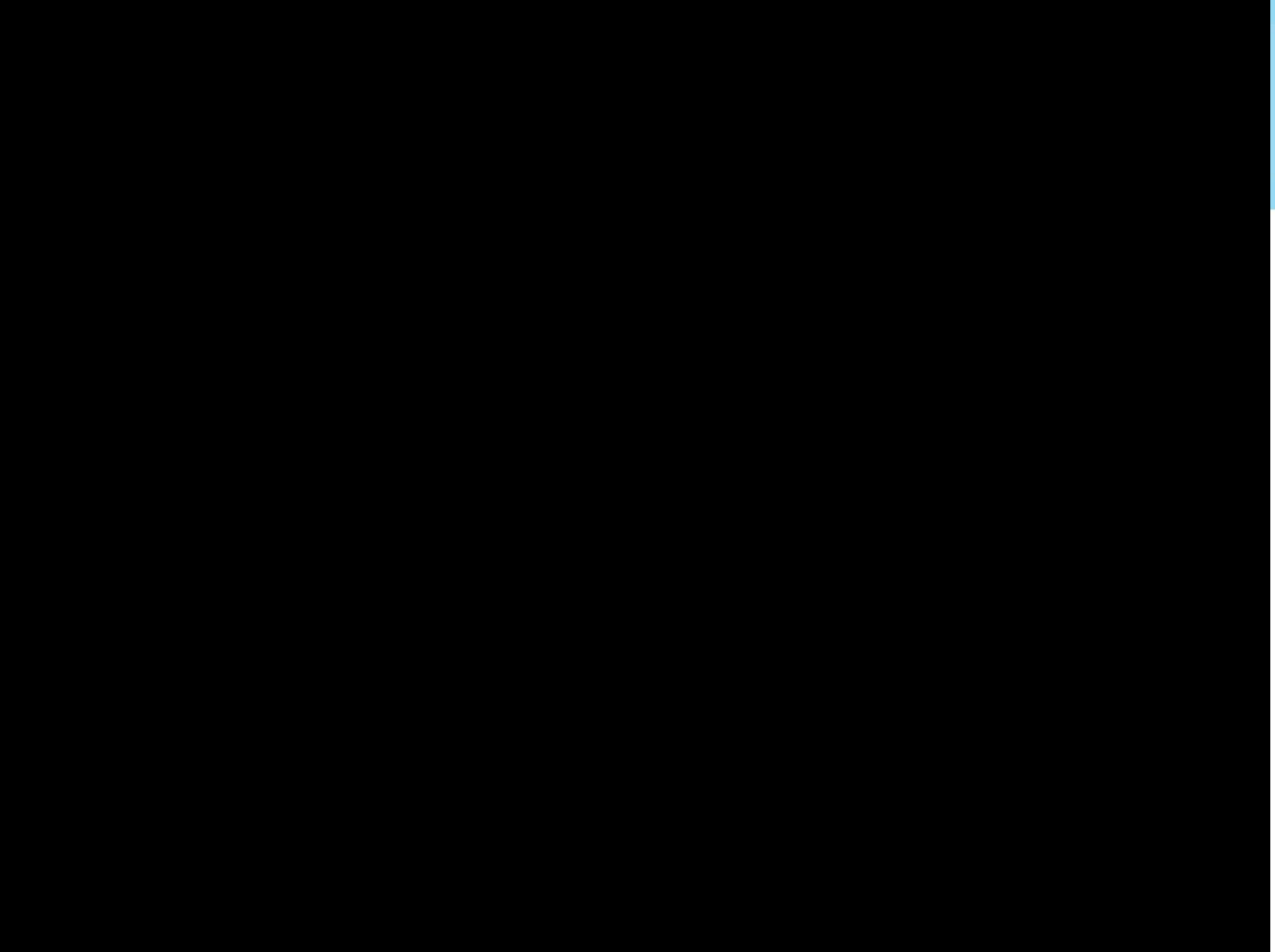
Previous Releases:

Row	Release	(A) Istio Full	(B) No Mixer	(C) Ingress Only	(A)/(C) %	(B)/(C) %
1	0.6.0	1307	1987	3804	34.4	52.2

Current Daily Builds:

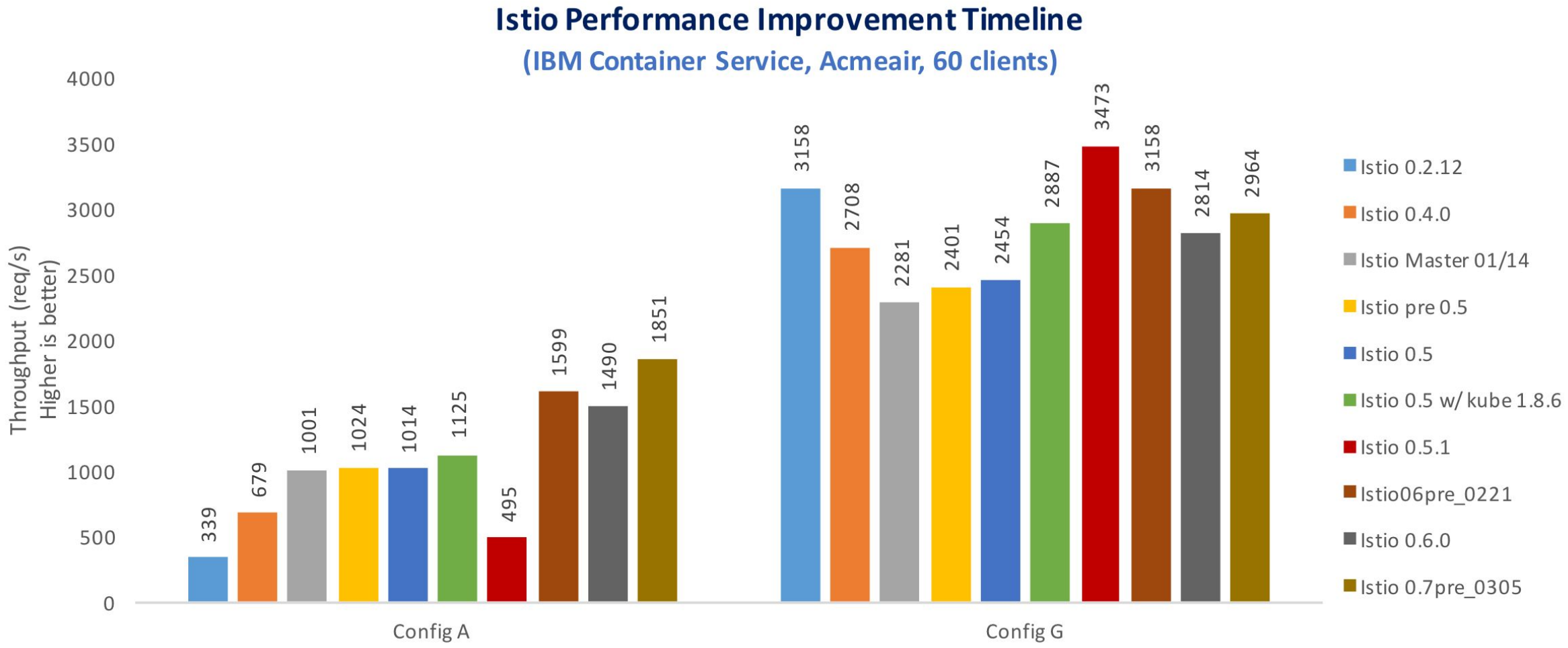
Row	Build	(A) Istio Full	(B) No Mixer	(C) Ingress Only	(A)/(C) %	(B)/(C) %
1	0.7.1 (baseline)	1294	2050	3671	35.2	55.8
2	0.8.0-pre20180404-09-15	282	2049	3644	7.7	56.2
3	0.8.0-pre20180408-09-15	1300	2078	3585	36.3	58.0
4	0.8.0-pre20180410-09-15	1264	1994	3647	34.7	54.7
5	0.8.0-pre20180412-09-15	1198	2031	3905	30.7	52.0
6	0.8.0-pre20180413-09-15	1272	1991	3554	35.8	56.0
7	0.8.0-pre20180414-09-15	1232	1974	3614	34.1	54.6

- A Regression Petrol automation framework is developed for Istio performance analysis used by Istio development community for daily builds



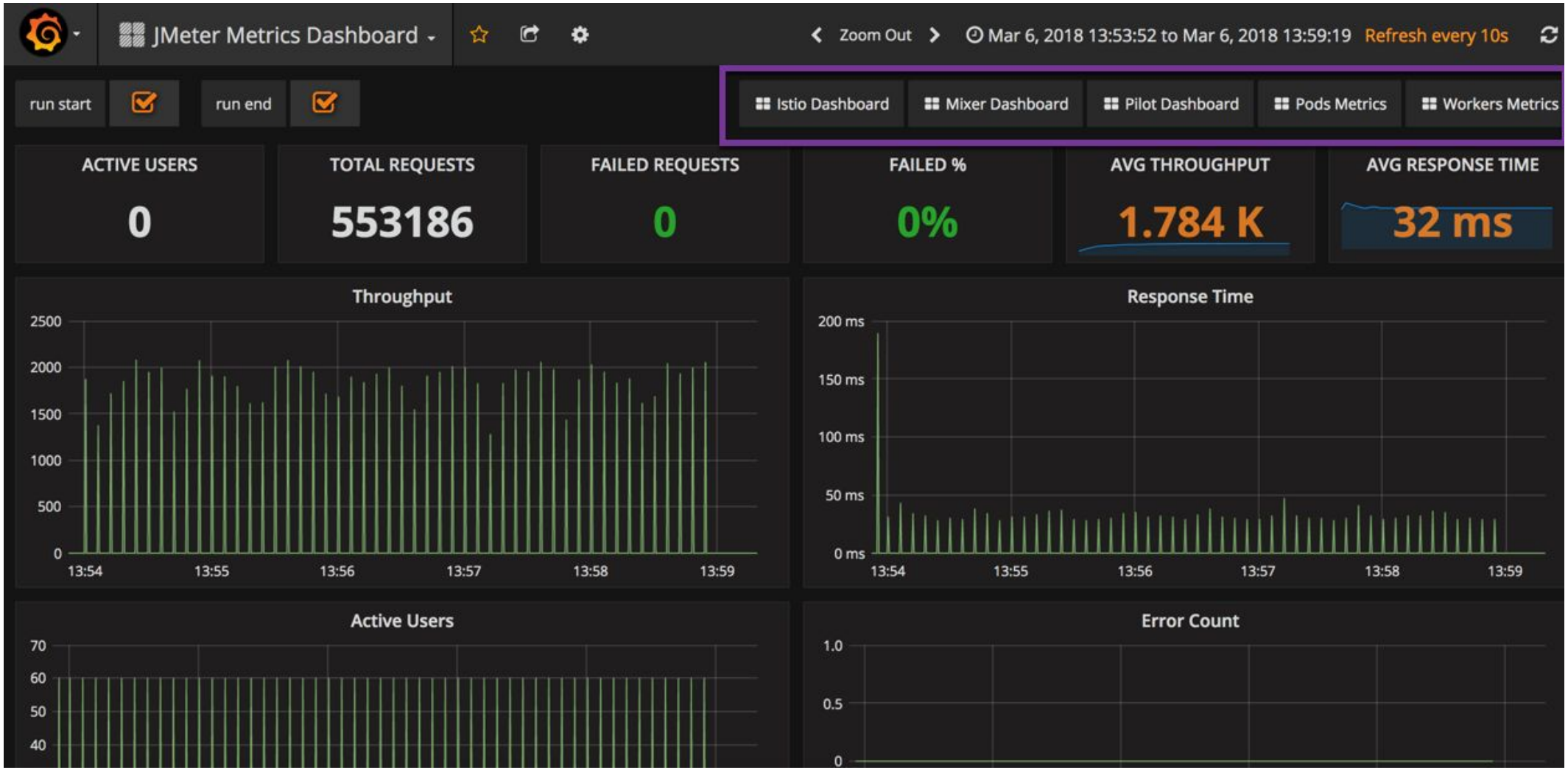

KubeCon
CloudNativeCon
Europe 2018

Istio Regression Patrol Results



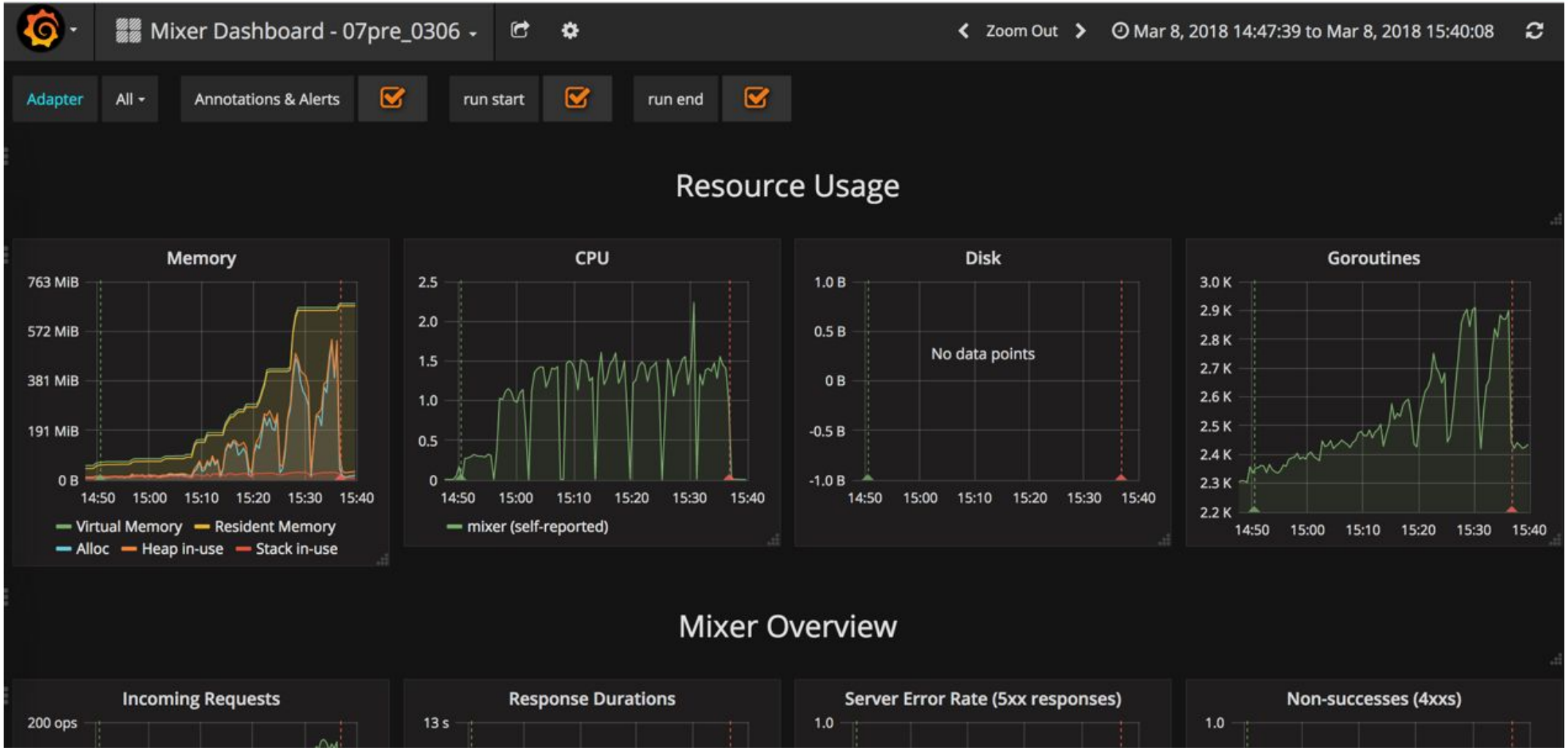
✓6x Improvement overtime for Istio

Istio Performance Master Dashboard (Regression Patrol)



➢ An integrated Performance Dashboard is developed for Istio performance analysis used by Istio development community

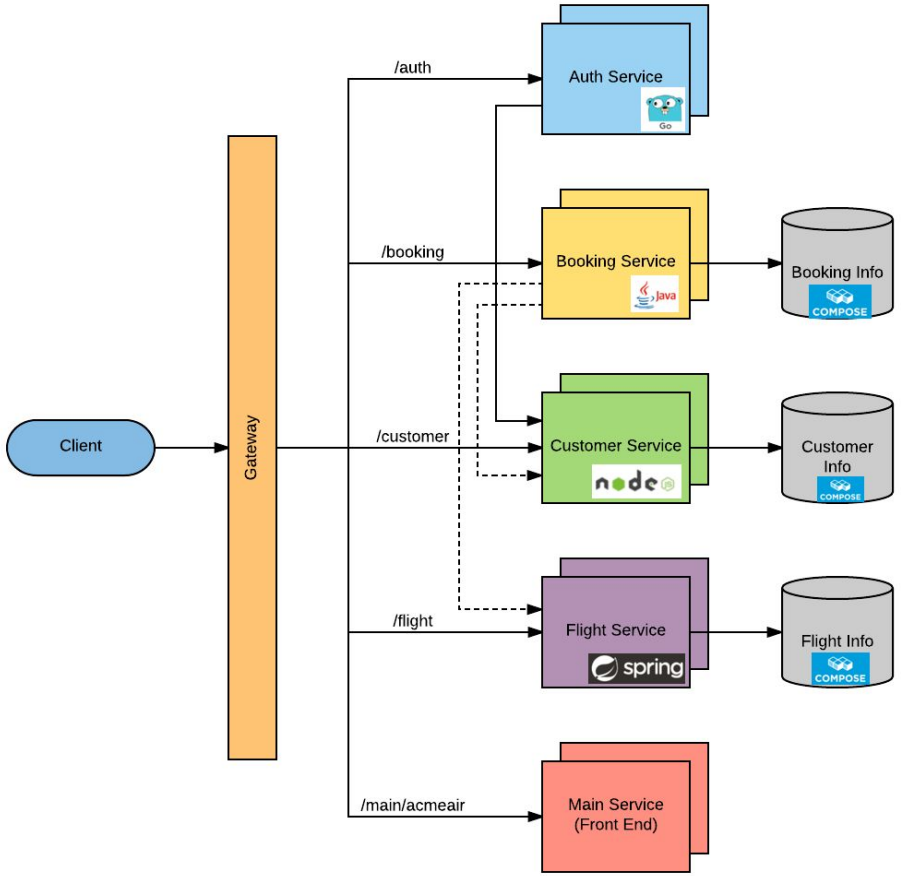
Mixer Dashboard (Regression Patrol)



➢ Mixer dashboard gives comprehensive information about Mixer resource usage, individual adapter configuration and usage information

Istio Performance across Multiple Industry Use Cases

Acme Air Polyglot Microservices Benchmark

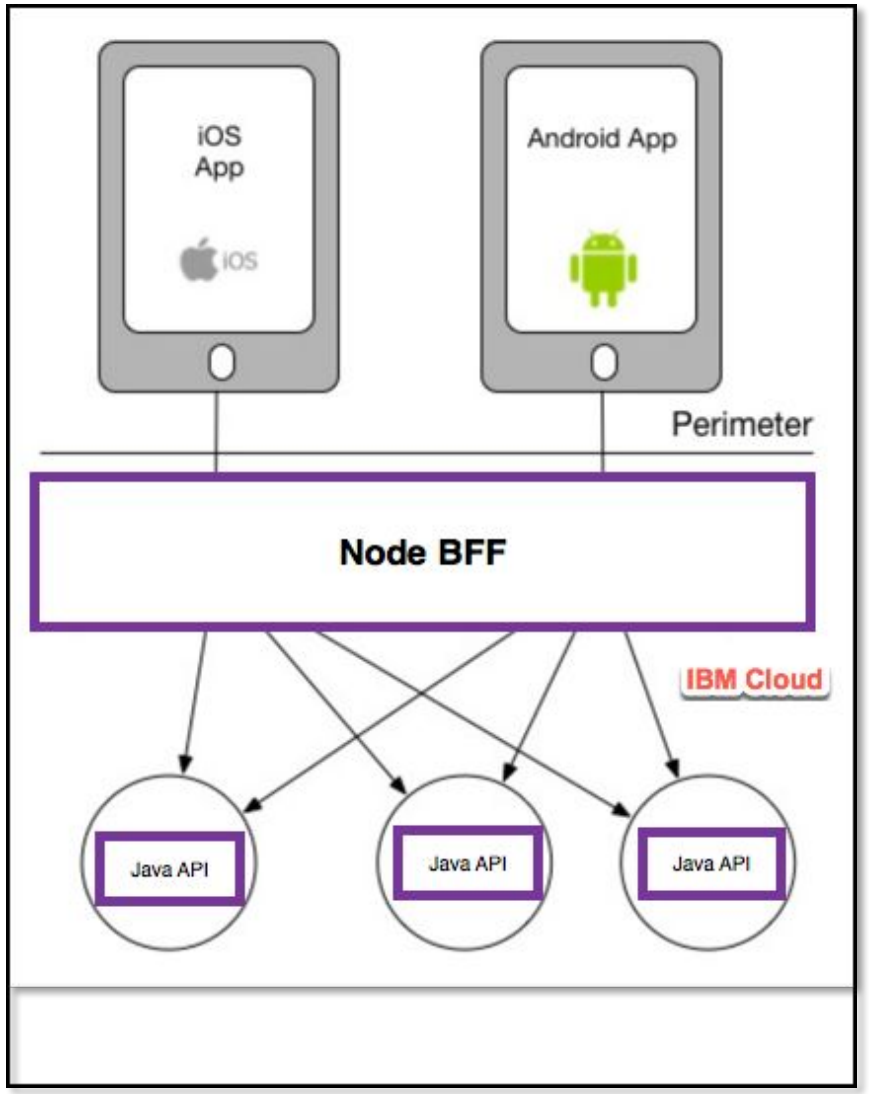


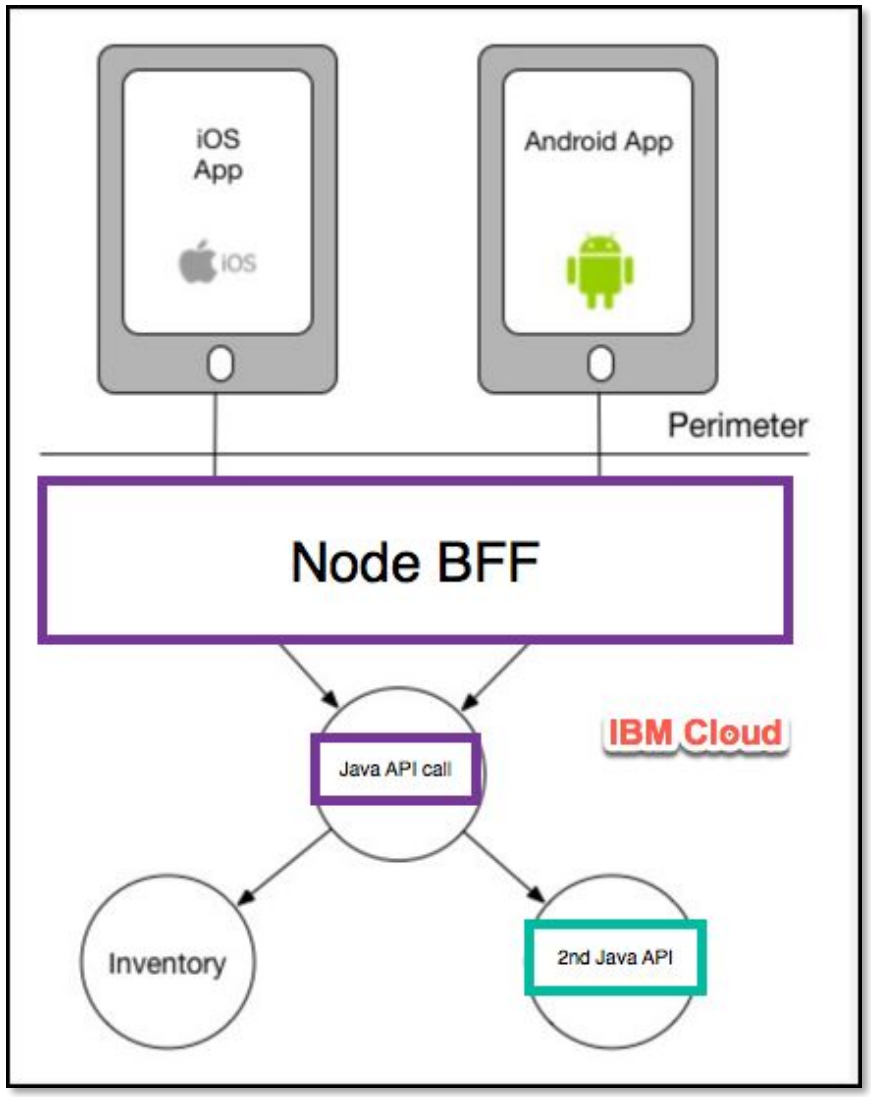
➢ Acme air polyglot microservices Benchmark is used to evaluate performance and scalability of Istio service mesh (https://github.com/blueperf/)

Online Banking Microservices Application

- Online Banking workload simulates a typical retail Banking functionality
- This application is based a major North American Bank application in production
- Online Banking Application does the following
  - Simulation of Retail on-line Banking scenario (UI, Security and Services)
  - Traffic is encrypted across the board and users are authenticated (using TAI for Liberty)
  - Backend Services interaction is simulated with Stub application
  - Account Summary page is developed using Angular JS with corresponding services layer simulation

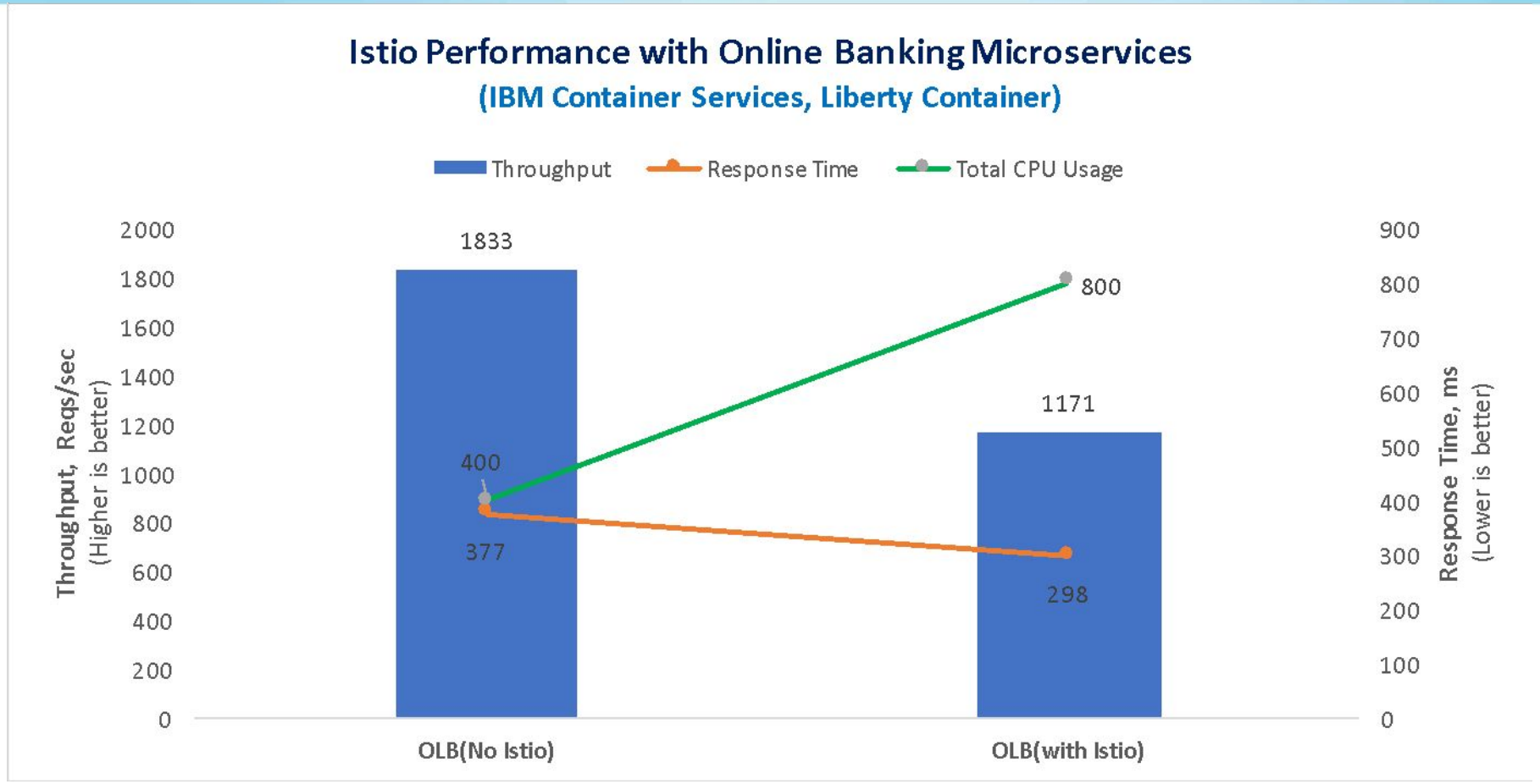
Healthcare Microservices Application (BFF Pattern)





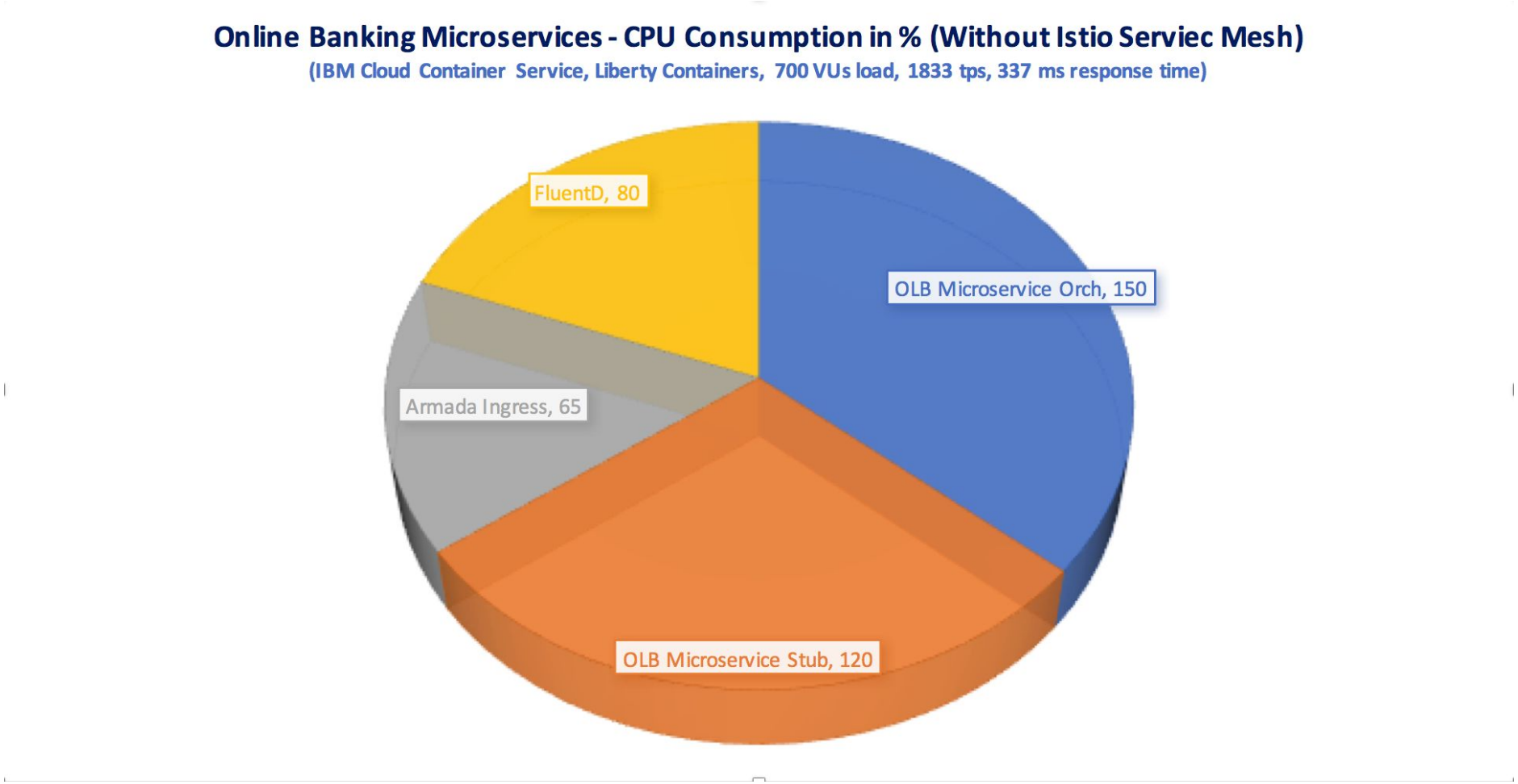
- Healthcare microservices workloads based on Rx and other Lines of Businesses of healthcare industry to evaluate performance and scalability Istio Service Mesh

Online Banking Microservices with Istio Service Mesh



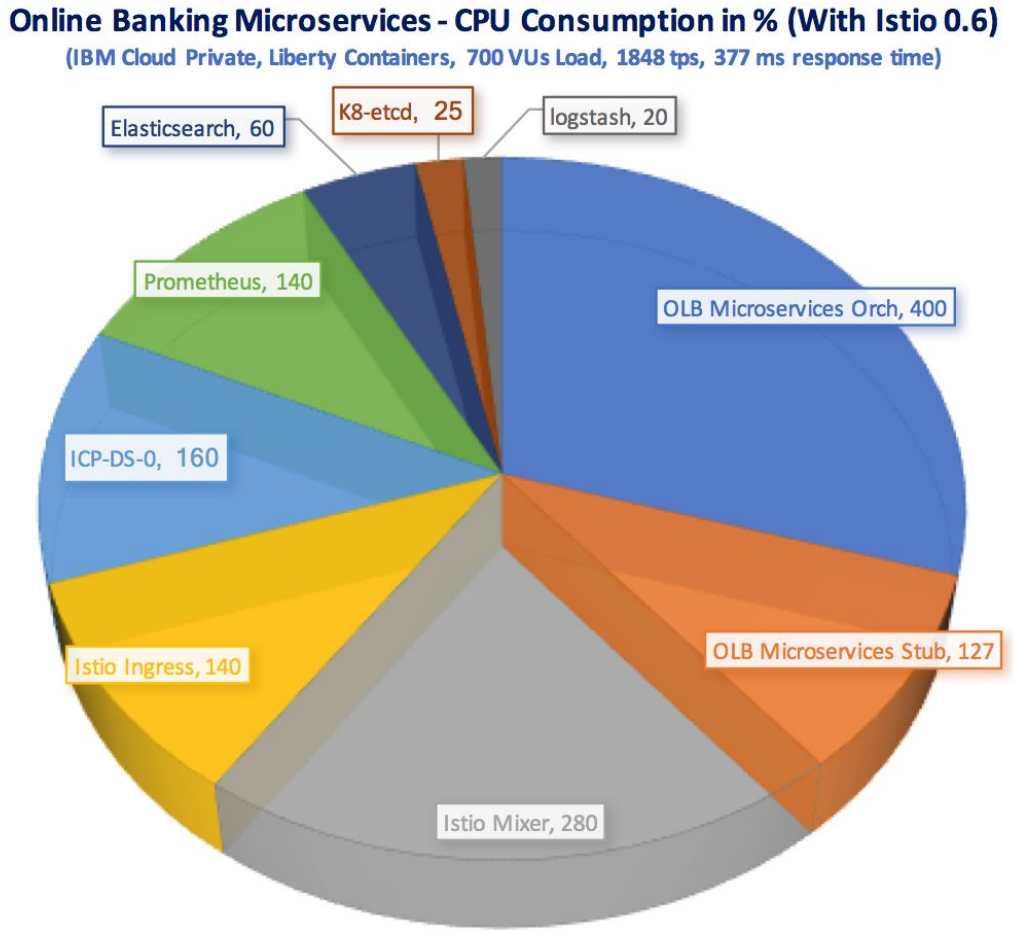
- Online Banking Microservices Applications can exploit Istio Service Mesh availing of all modern Service Mesh features
- To enable linear scaling of OLB Services, it is recommended to deploy Istio components to a separate set of Dedicated Nodes in the cluster
- While Istio Open Community is working hard to optimize the service mesh, at this point, one needs to allocate more Compute power to Istio components
- In the chart above, there is a 56% overhead to OLB with Istio mainly because of resource constraints on the worker node where the OLB Services are deployed

Online Banking Microservices CPU Consumption (Without Istio Service Mesh)



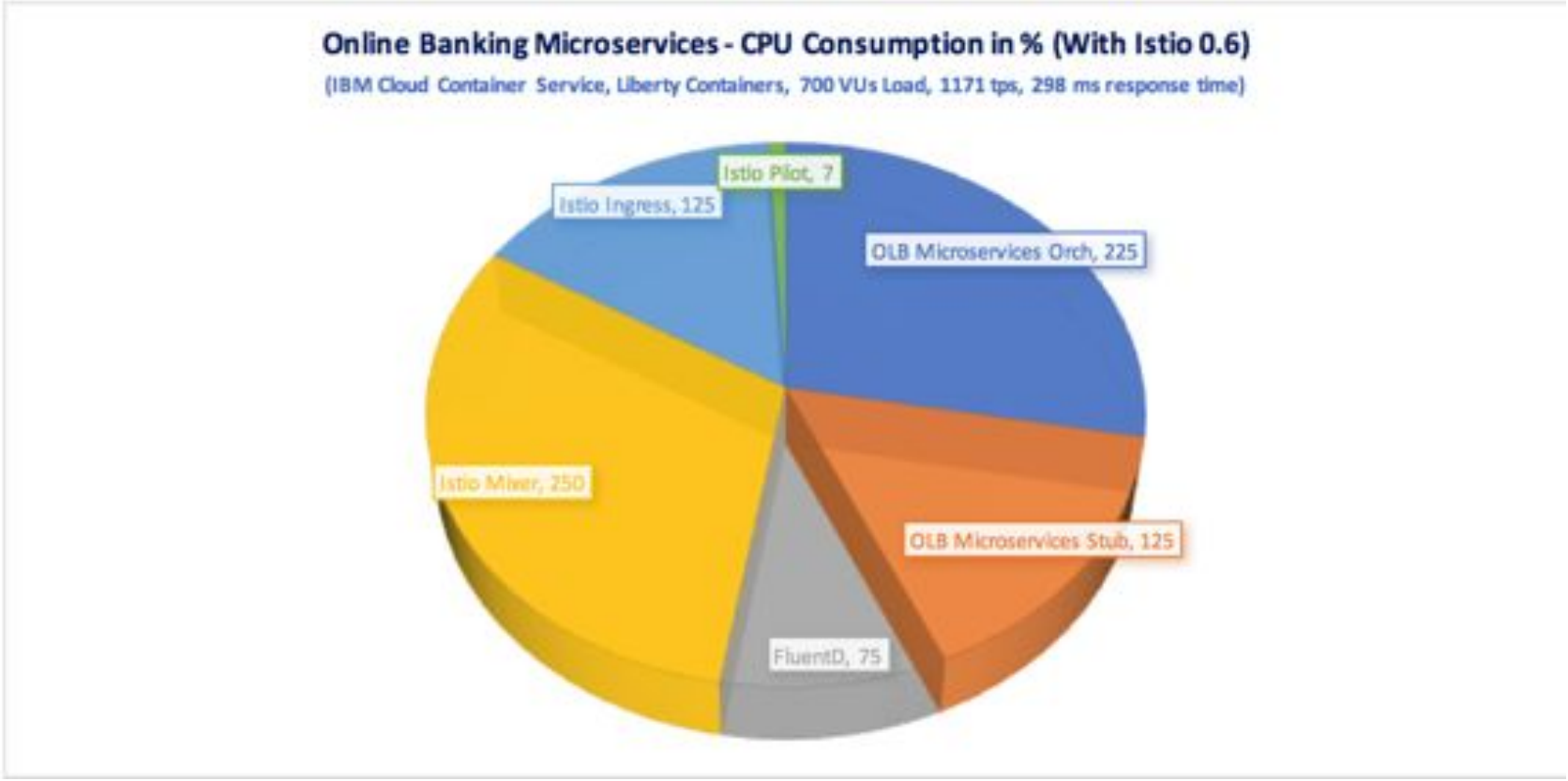
- Online Banking Microservices Application deployed on Kubernetes Service is able to support 158 Million API calls/day @337 ms response time using 4 vCPU compute power

Online Banking Microservices CPU Consumption (With Istio v0.6)



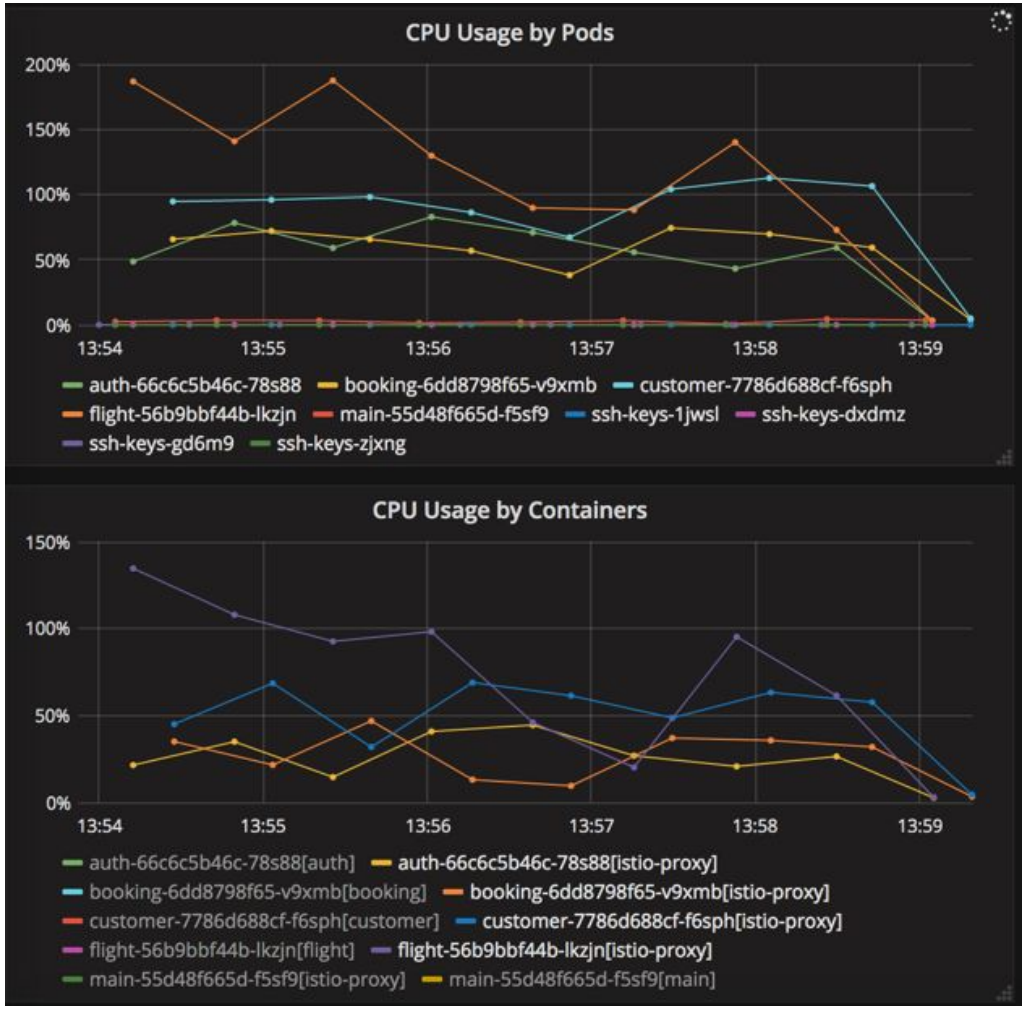
- Online Banking Microservices Application deployed on Kubernetes Service with Istio service mesh is able to support 101 Million API calls/day @298 ms response time using 8 vCPU compute power

Online Banking Microservices with Istio Service Mesh



- Online Banking Microservices Application with Istio service mesh performs and scales almost linearly on ICP Cloud Platform provided there are enough resources allocated
- OLB Microservices with and without Istio service mesh can get to 1848 tps @377 ms response time, but Istio requires about 4 vCPUs additional compute power

Acme air Microservices with Istio Service Mesh (Impact of Sidecar)



- For microservices connected to external services like Compose MongoDB etc., there will be additional CPU pressure on sidecars as shown above
- For microservices not connected to external services, this impact is much smaller

Next Steps / How can you participate ?

KubeCon

CloudNativeCon

Europe 2018

- Join the WG Meetings (9:30a Wed PST; 18:30 Europe time)
- Add and explore more dimensions to our existing standard tests: payload sizes, gRPC, number of rules, number of service/endpoints, mTLS on/off, mixer on/off, node placement, horizontal pod scaling, ...
- Add environments and cloud providers
- Analyze pprof, flamegraphs, cpu and memory profiles per component
- Contribute to (and fork/star) fortio (github.com/istio/fortio) and blueperf (github.com/blueperf)

KubeCon

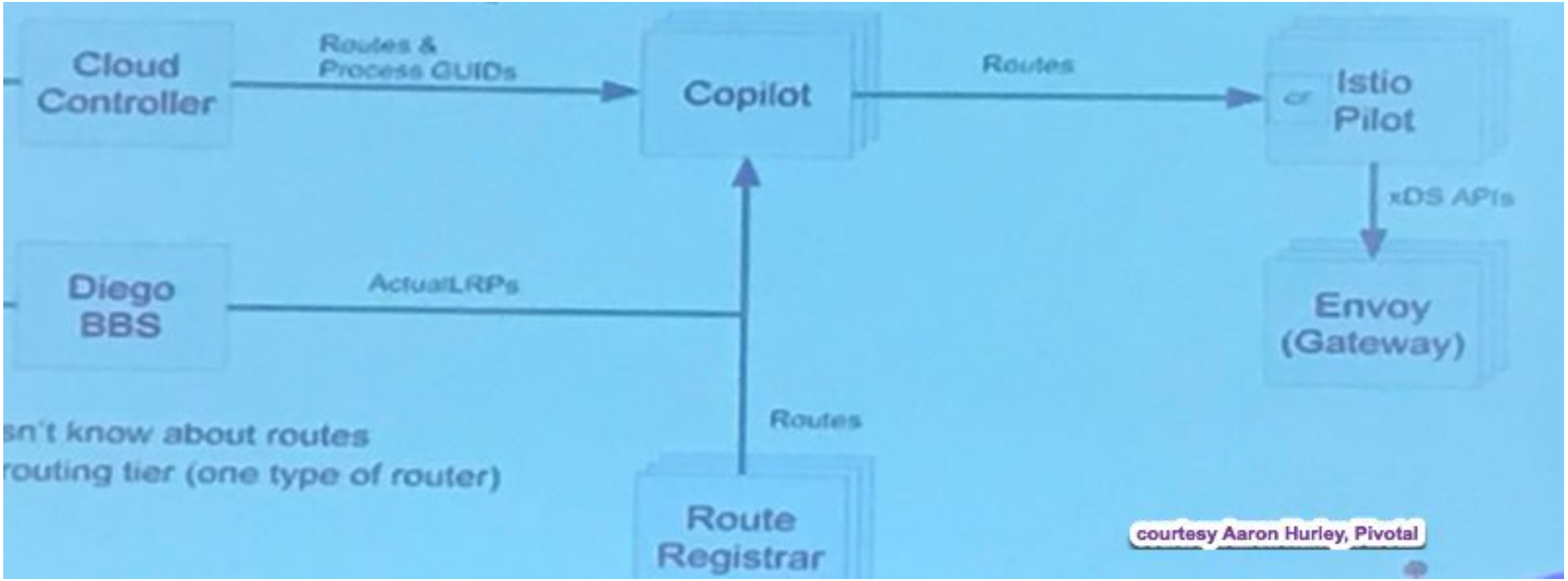
CloudNativeCon

Europe 2018

Q&A

Additional slides

Istio Support for Cloud Foundry



- CF Istio control plane component called ‘Copilot’ will work with Istio Pilot
- North –South traffic of Cloud Foundry will be handled by Copilot and Envoy replacing GO Router
- East-West traffic of Cloud Foundry can be handled by support of sidecar with OPI or Diego Garden redesign ???

Istio Service Mesh Support on Cloud Foundry

KubeCon | CloudNativeCon Europe 2018

- There is a Cloud Foundry Project which is a BOSH release that packages Istio and Envoy for Cloud Foundry
- This project is not complete and it is in active development
- Istio-release requires bosh-cli >= 2.0.45
- Design changes to Fabric to support sidecar pattern
- For more details, visit -> https://github.com/cloudfoundry/istio-release