KubeCon

CloudNativeCon

Europe 2018

Istio tells me my service has slow response time, now what?

Endre Sara

VP of Advanced Engineering, Turbonomic

Enlin Xu

Sr. Engineering Manger, Advanced Engineering, Turbonomic

Who is Turbonomic?

- **Workload automation for hybrid cloud** assures performance, while minimizing cost and maintaining compliance
- **Software drives continuous state of health** by matching workload demand to infrastructure supply
- **Technology agnostic:** Container Platforms, Virtualization, Cloud, etc.
- Launched in 2010

Please stop by Booth S C-25

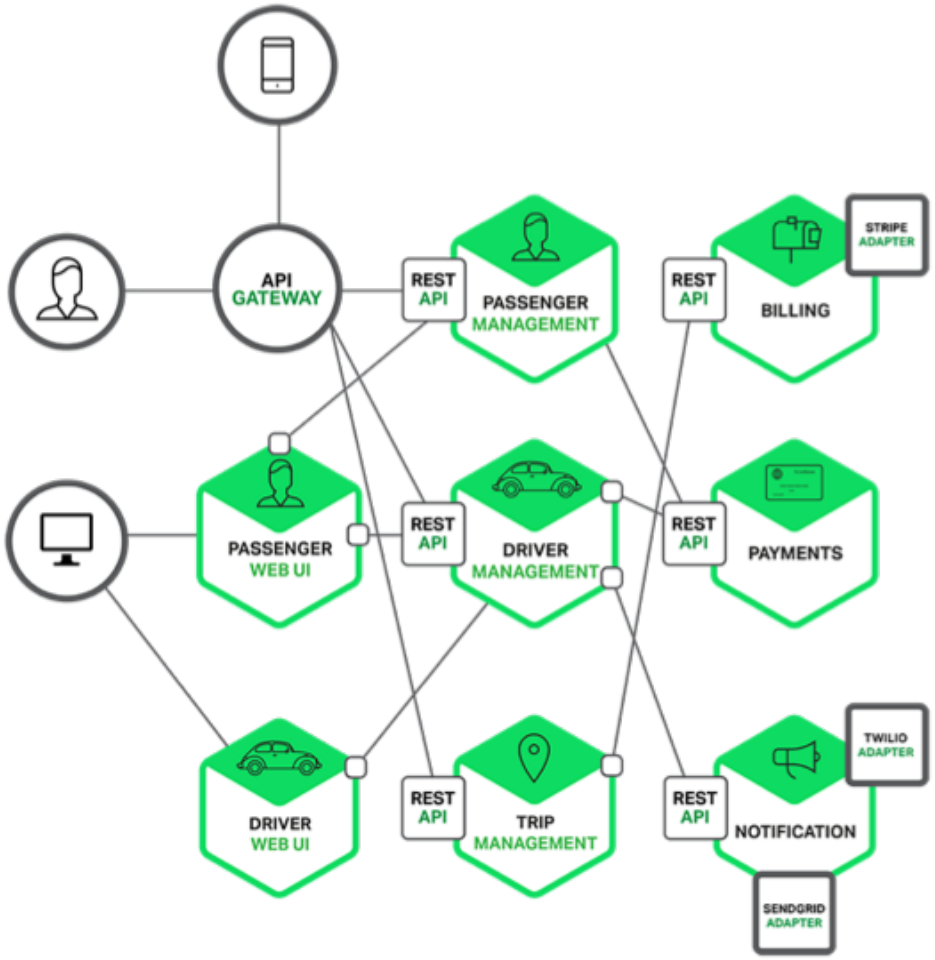
Presented by

- **Endre Sara** is VP of Advanced Engineering at Turbonomic, focused on new technologies. Before joining Turbonomic in 2009, he was VP/Technology Specialist of Enterprise Systems Management at Goldman Sachs. He joined Turbonomic because it's more fun.

- **Enlin Xu** is a proud graduate of Columbia University and has been a software engineer in Turbonomic since 2011. He is now a Senior Engineering Manager that leads the engineering effort for Cloud Native technology design and integration in Turbonomic. Before coming to US, Enlin graduated from Hong Kong University of Science and Technology, obtaining a B.Eng in Electric and Electronic Engineering. During his years at Turbonomic, Enlin has been inventor of three granted patents in resource management space

Why Service Mesh

- Visibility
- Resiliency & Efficiency
- Traffic Control
- Security
- Policy Enforcement

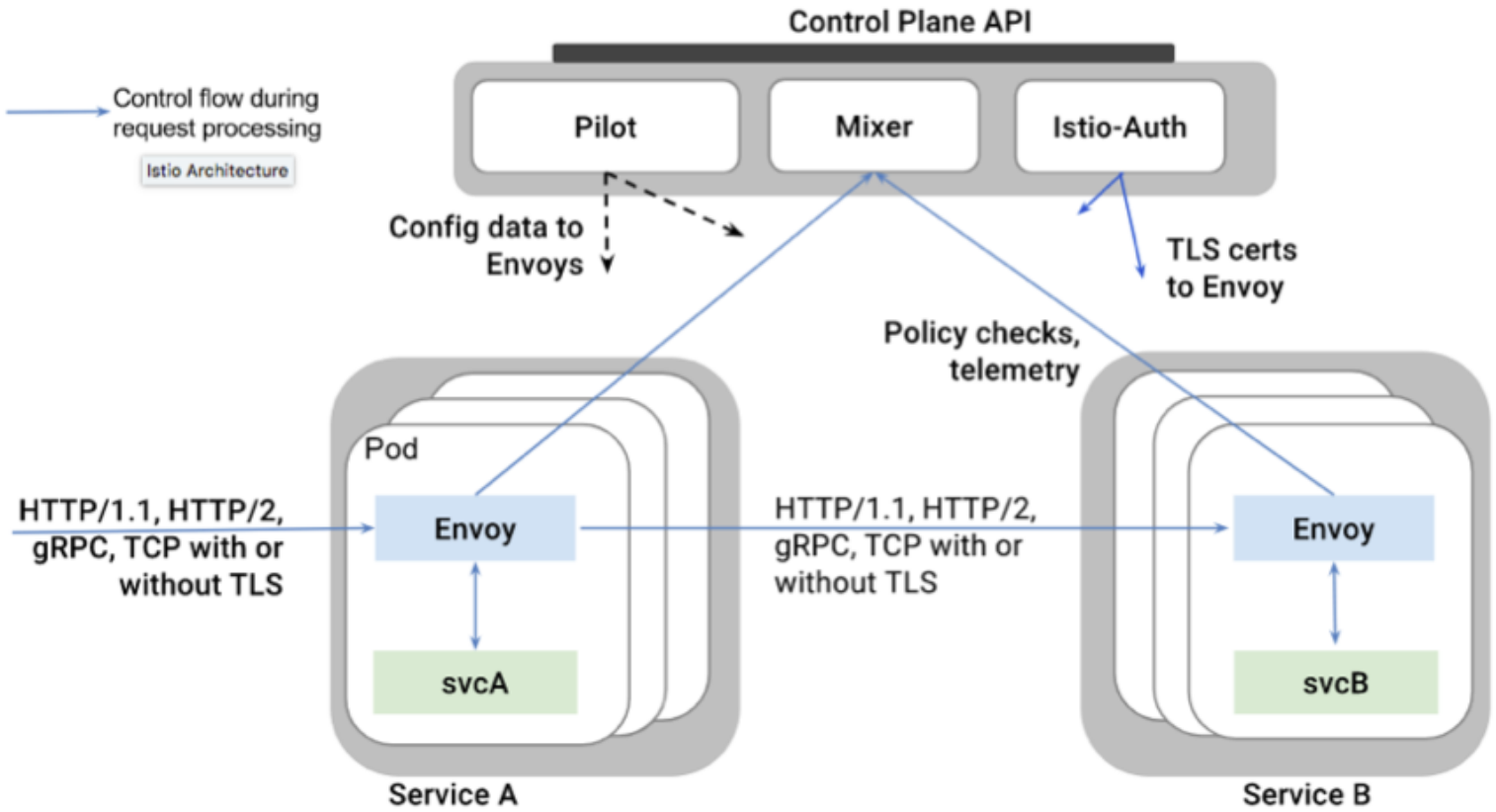


- Authentication?
- Load Balancing?
- Request Routing?
- Failover Policy?
- Security?
- Logs and Metrics?
- Connection Mgmt?
- API Mgmt?
- …

All Hand over to:

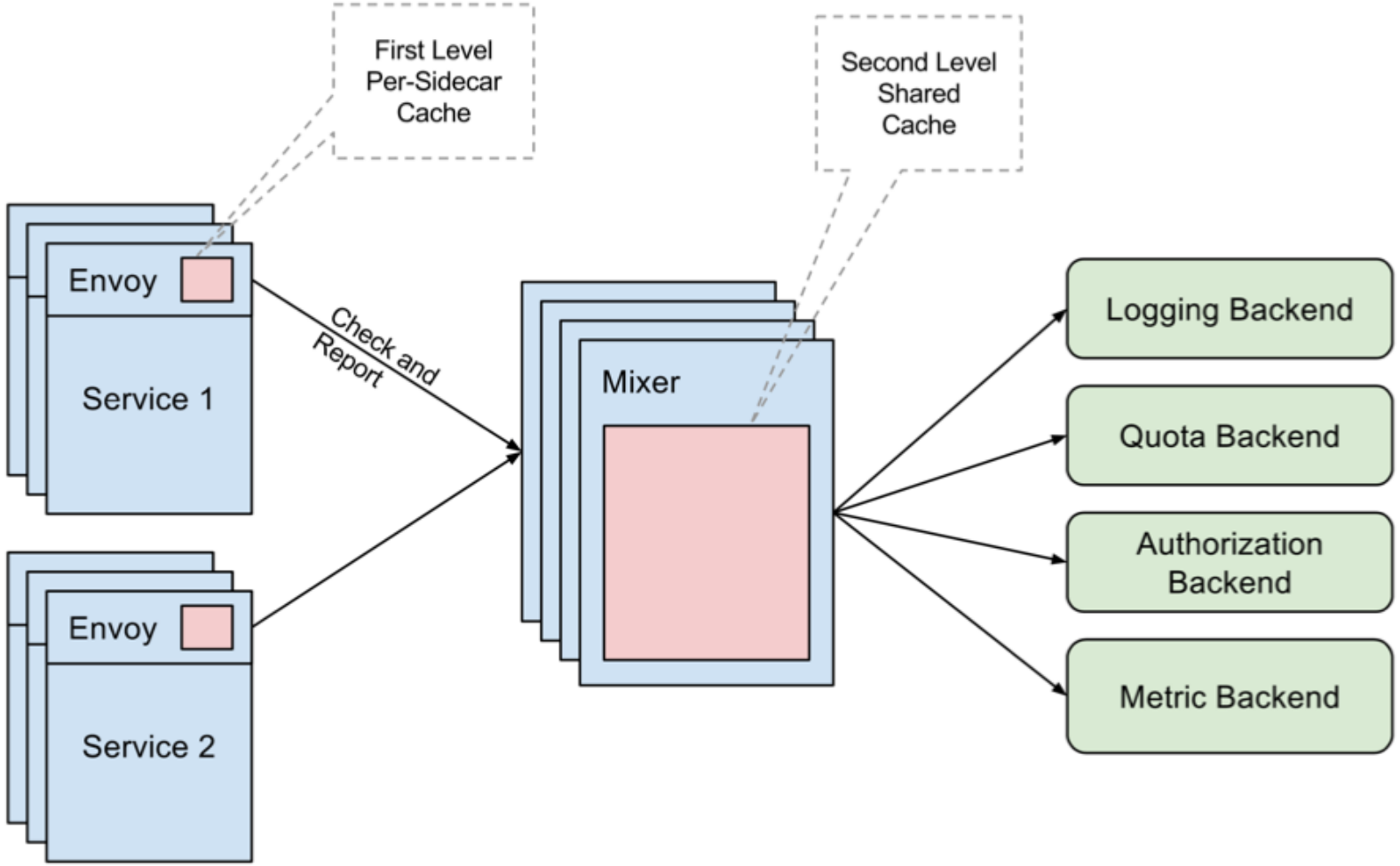
Service Mesh

Istio



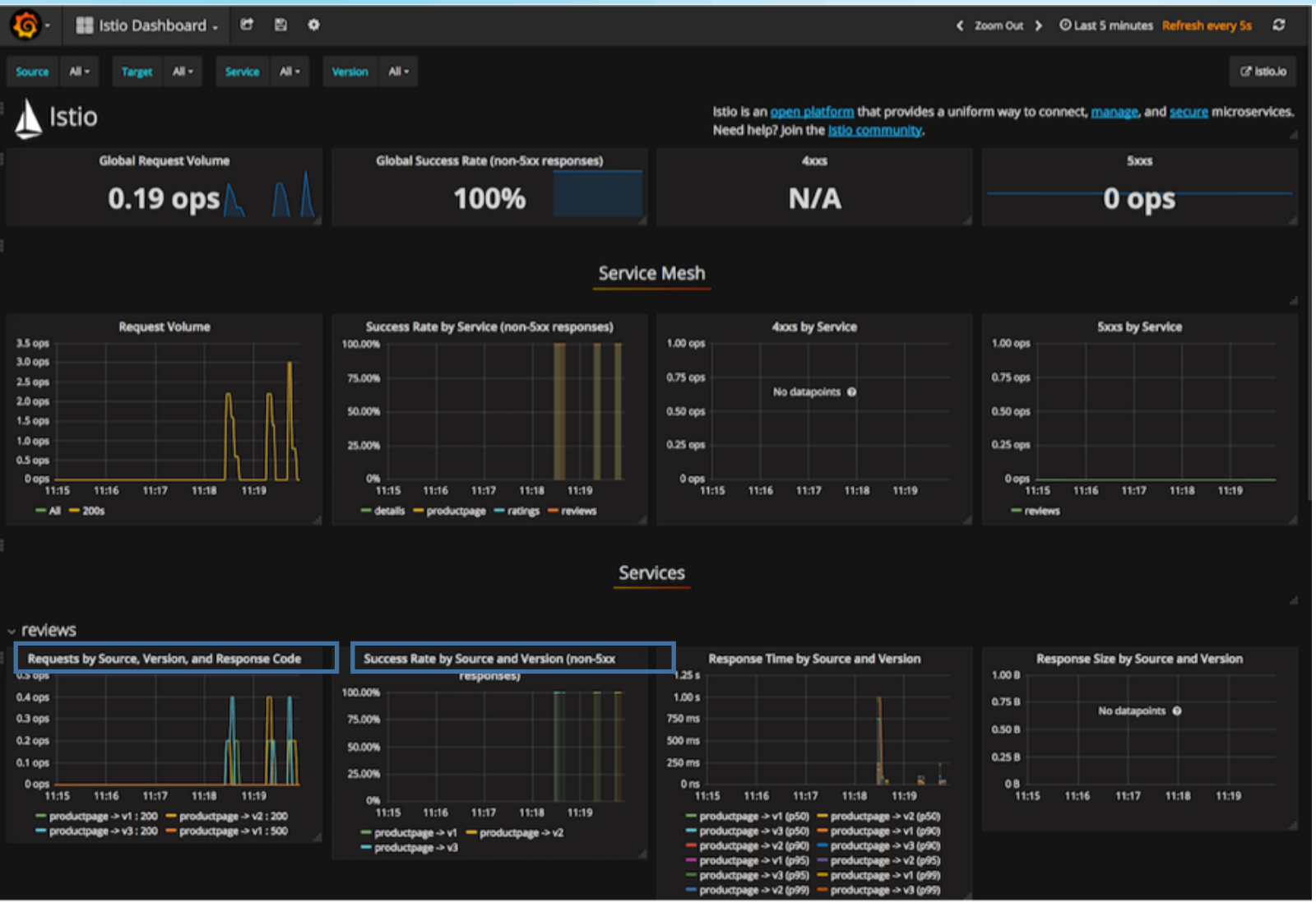
Istio Architecture

Telemetry: Envoy and Mixer

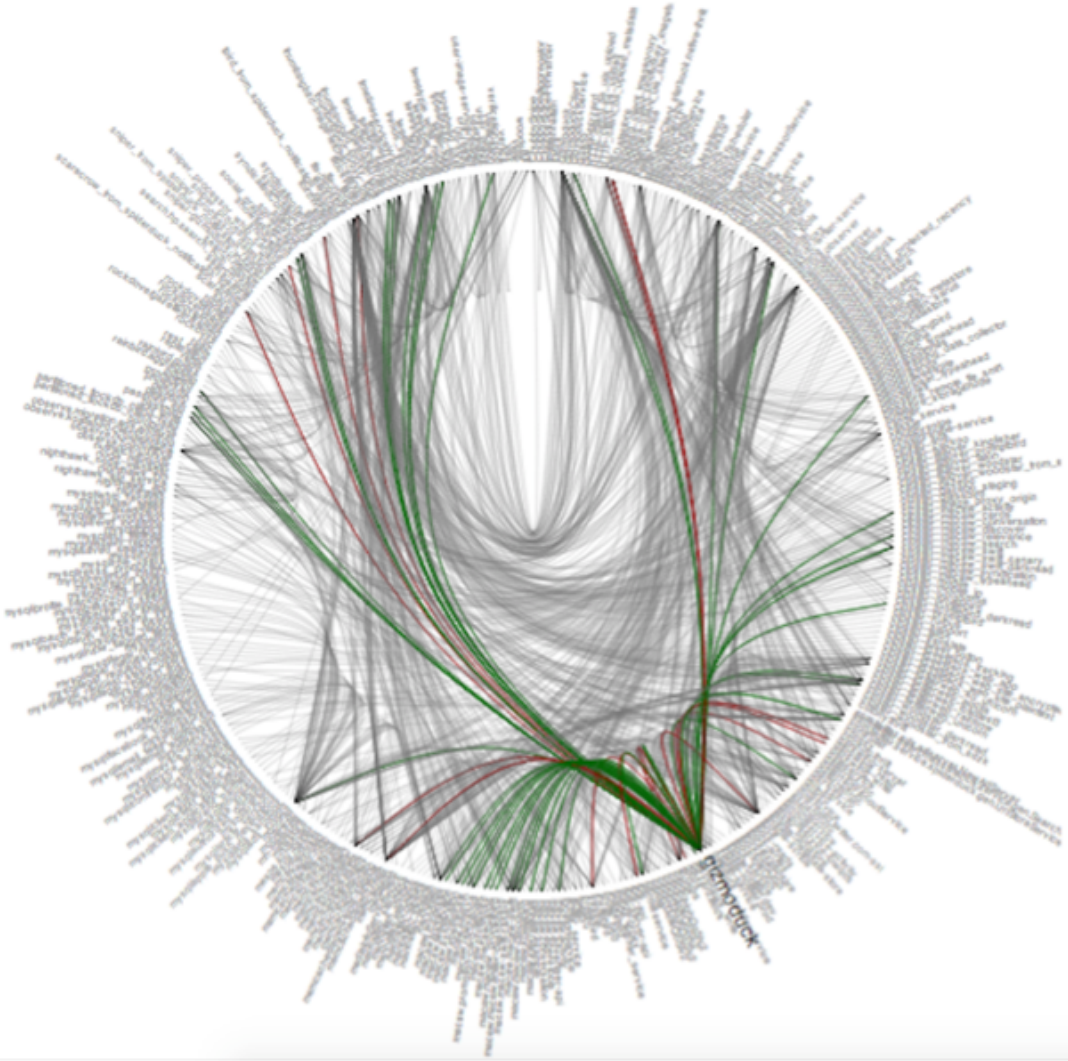


- Envoy calls Mixer before each request to perform precondition checks
- After each request, report the telemetry

Detailed per-service metrics



What about now?



KubeCon

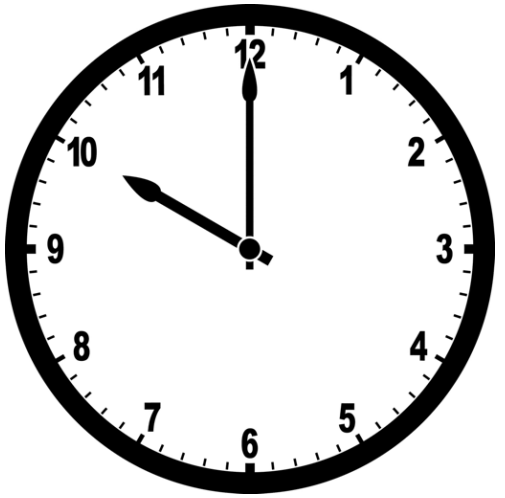
CloudNativeCon

Europe 2018

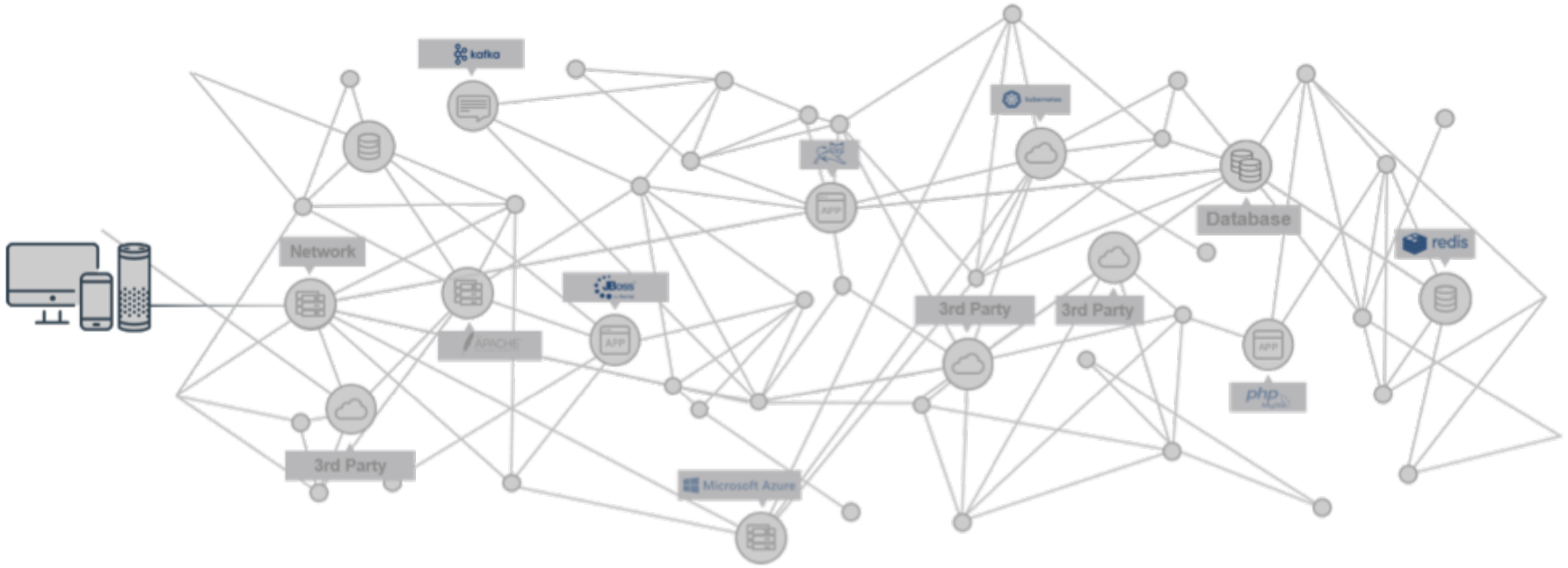
Istio tells me my service has slow response time, now what?

It’s 10 o’clock… Do you know how to self-manage application performance?

- How many instances are required to satisfy the application demand?
- Should a container scale vertically/horizontally up/down?
- Should a node scale vertically/horizontally up/down?
- How many containers can fit in a node?
- How much underline infrastructure is required?
- Where should a pod be placed?
- How close to each other containers should be placed?
- …



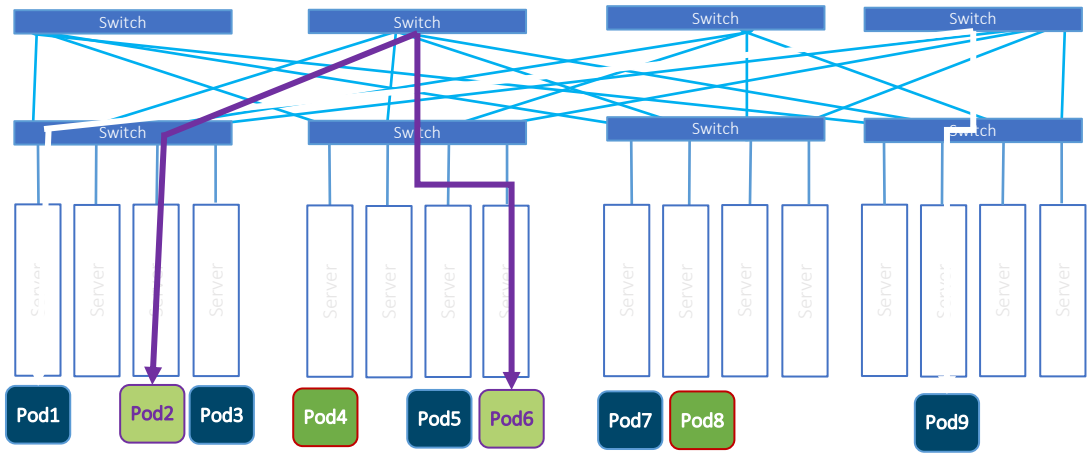
Network impact on Application Performance



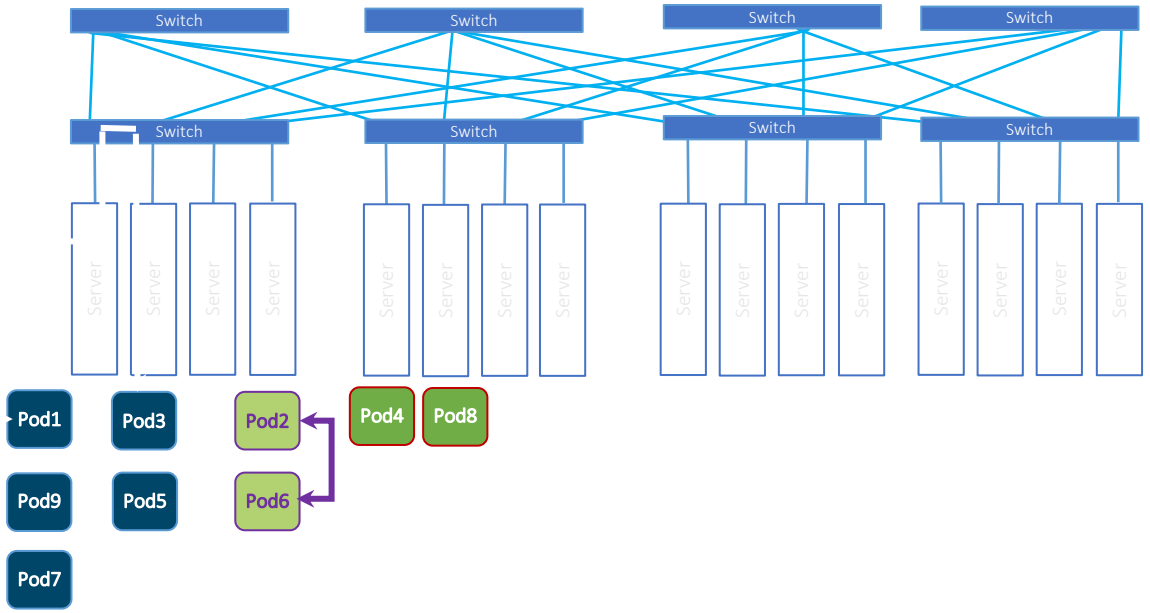
By 2020, 86% of data center traffic will be east-west…

New application architectures increase east-west traffic and risk network congestion.

Service Proximity Factors

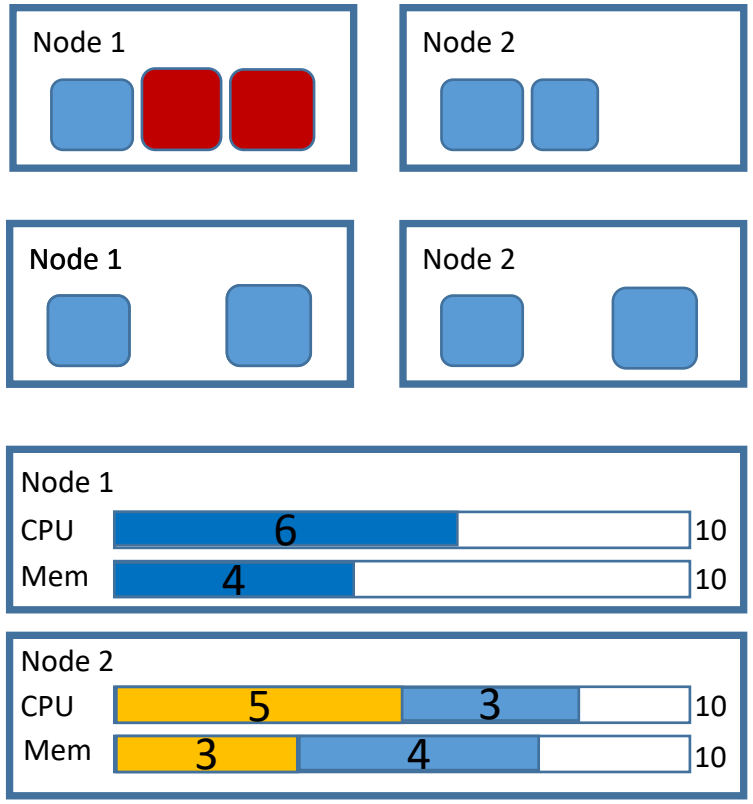


Chatty workloads across distances = latency.



Using telemetry data from Istio, “chatty” workloads can be **localized** to reduce latency.

Continuous Placement Factors



Noisy Neighbor

- Workload that always peaks together

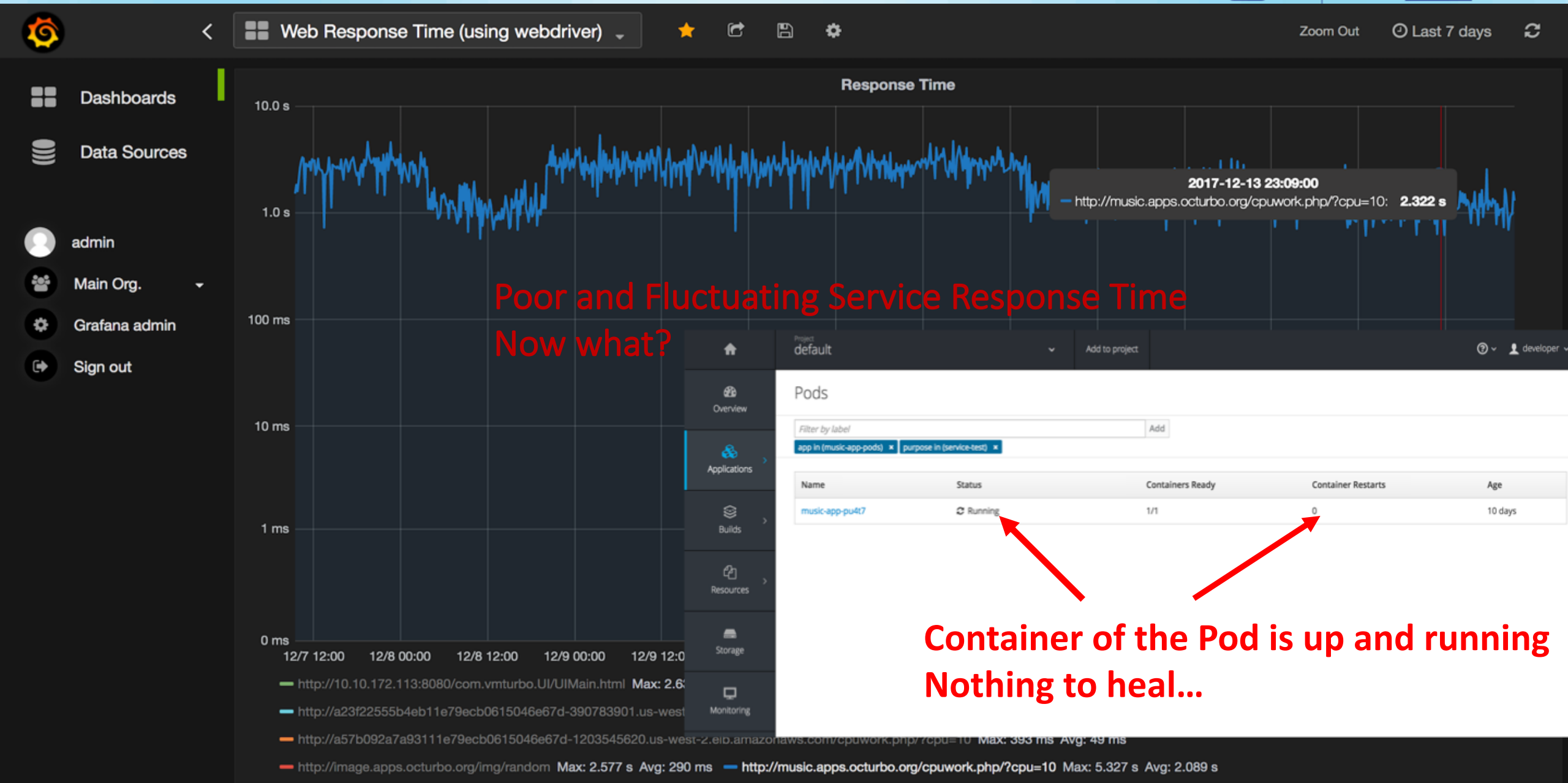
Performance Degradation

- CPU starvation - node cpu congestion

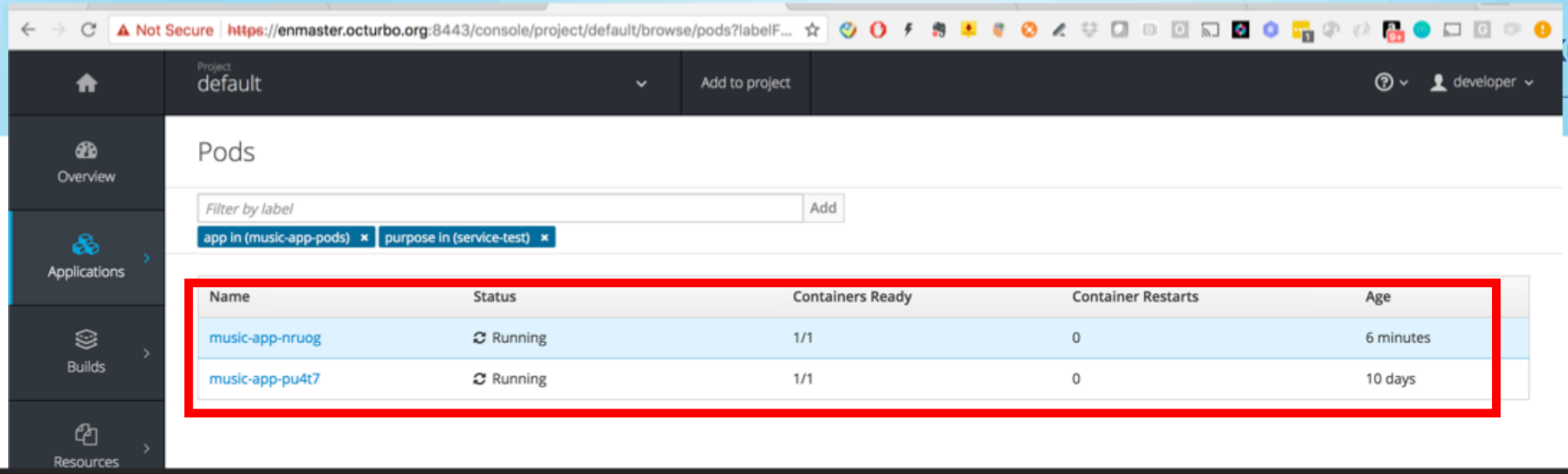
Long pending pod

- Resource Fragmentation

Auto-healing does not assure Application Performance



Auto-scaling does not assure Application Performance



Continuous Scale Factors

- How many replicas does my job need? – Horizontal Scale
- How much CPU/RAM does my job need? – Vertical Scale
- Do I provision for worst-case?
  - Expensive and wasteful
- Do I provision for average case?
  - High failure rate (e.g. OOM)
- What about scaling of underlying infrastructure?

```
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: limit.mem-256-cpu-20
5    labels:
6      purpose: 'test_memory_usage'
7      app: 'memory-load'
8  spec:
9    nodeSelector:
10     env: dev
11   containers:
12   - name: memory-256
13     image: beekman9527/cpumemload:latest
14     resources:
15       requests:
16         memory: "256Mi"
17         cpu: "20m"
18       limits:
19         memory: "512Mi"
20         cpu: "50m"
```

Pre-configured & Allocation based

Application Service Delivery Requirements

Placement

Decide Where to Run the App

Anywhere

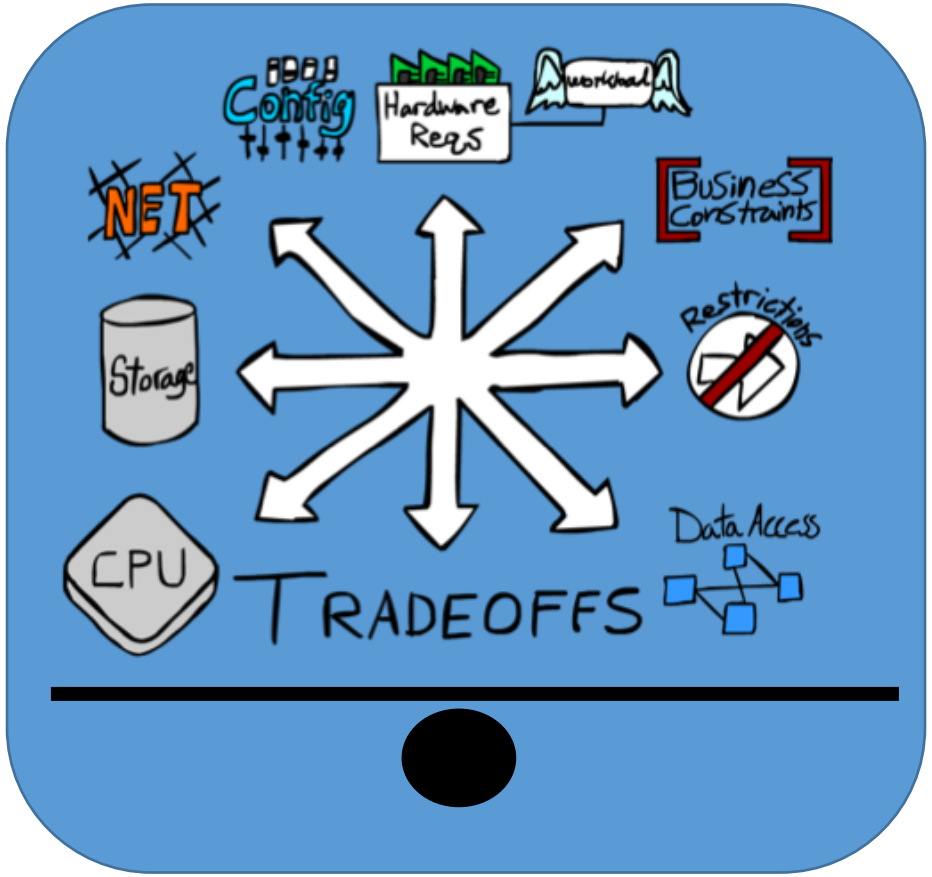
Assure the Performance Of the Apps no Matter Where They Run

Scale

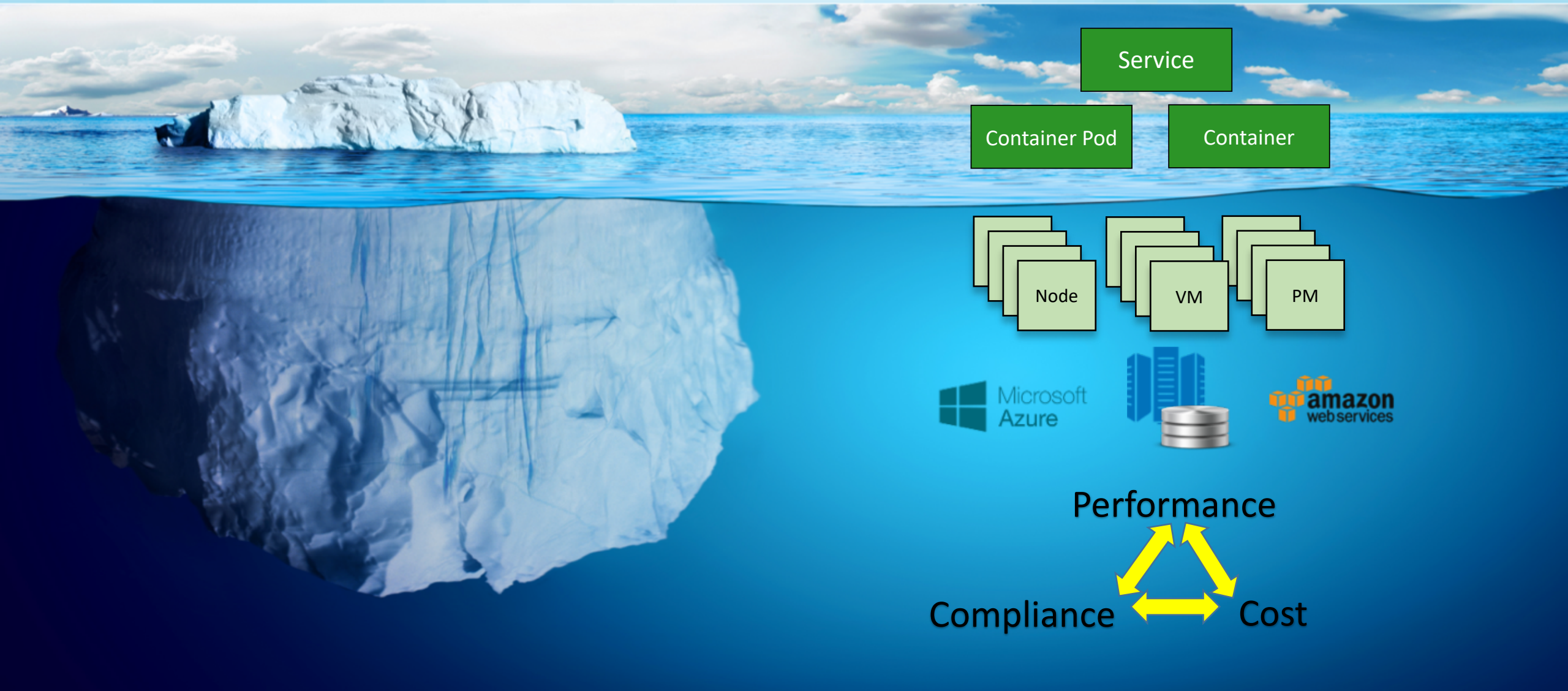
Scale the Apps to Meet the Service Level Agreement

Full Stack

Apps Running on Any Delivery Model PaaS, CloudOS, IaaS



The Need for Full Stack Control



Demo

- Setup: k8s 1.9.2, Istio, Prometheus, Grafana, Turbonomic
  - Multiple dimensions considered simultaneously
  - Telemetry data, affinity/anti-affinity, Compute, etc
- Service experiences response time degradation
- Performance Action(s) defined, executed
- Service performance restored to desired SLA