

Break New Ground

Service Mesh Patterns

Peter Jausovec

Twitter: @pjausovec

Consulting Member of Technical Staff, Oracle

**ORACLE
CODE**

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Microservices?

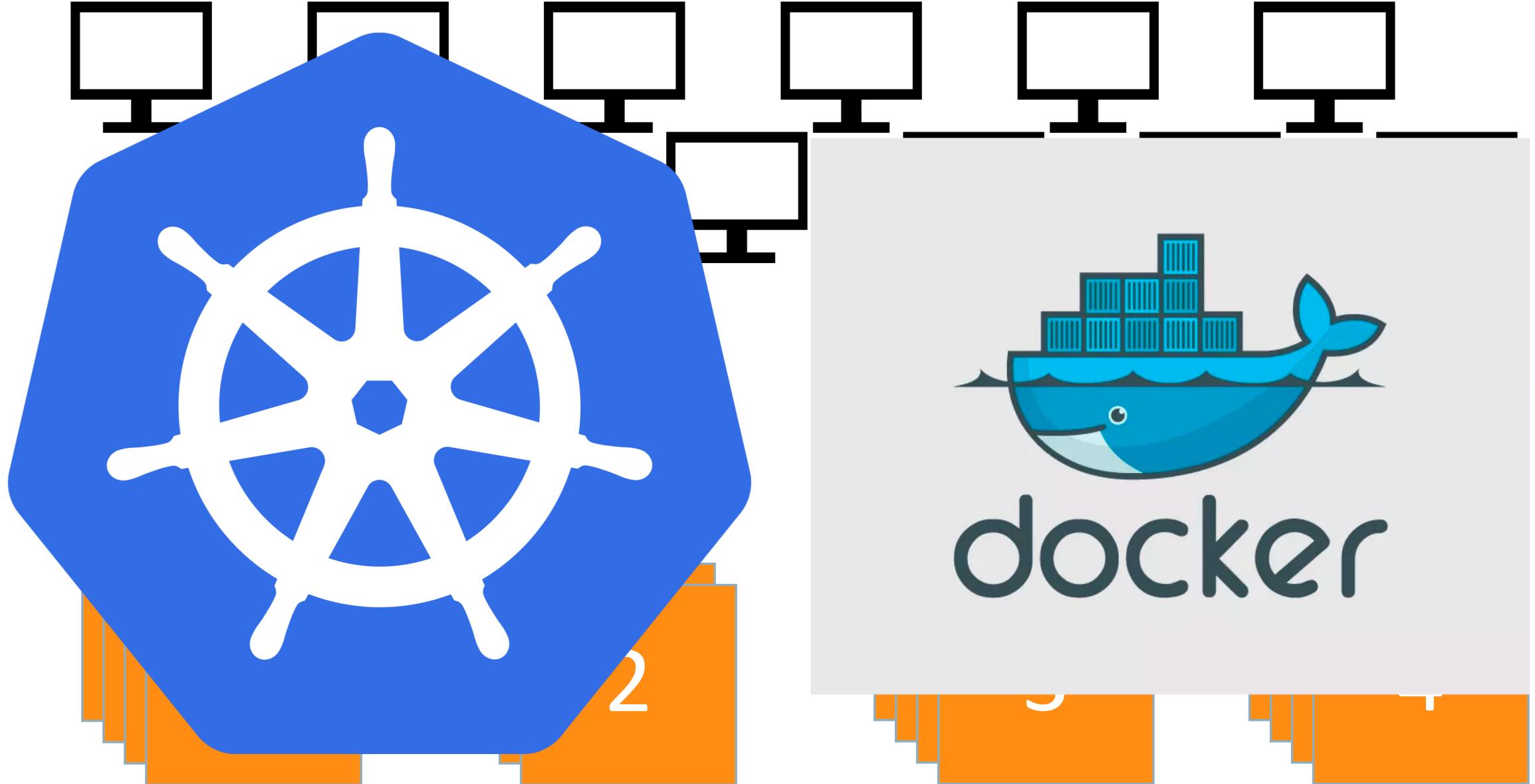
MONOLITH

1

2

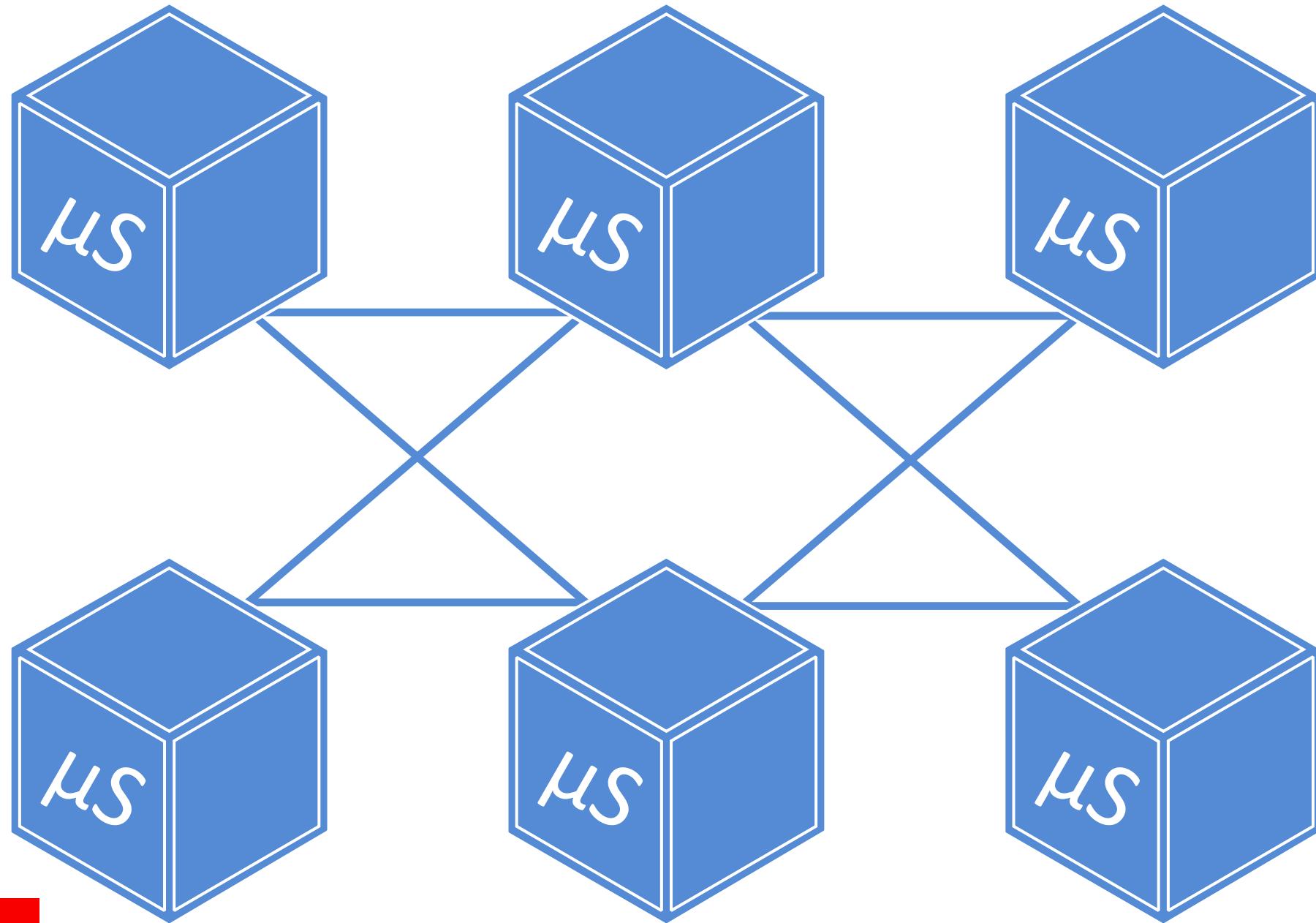
3

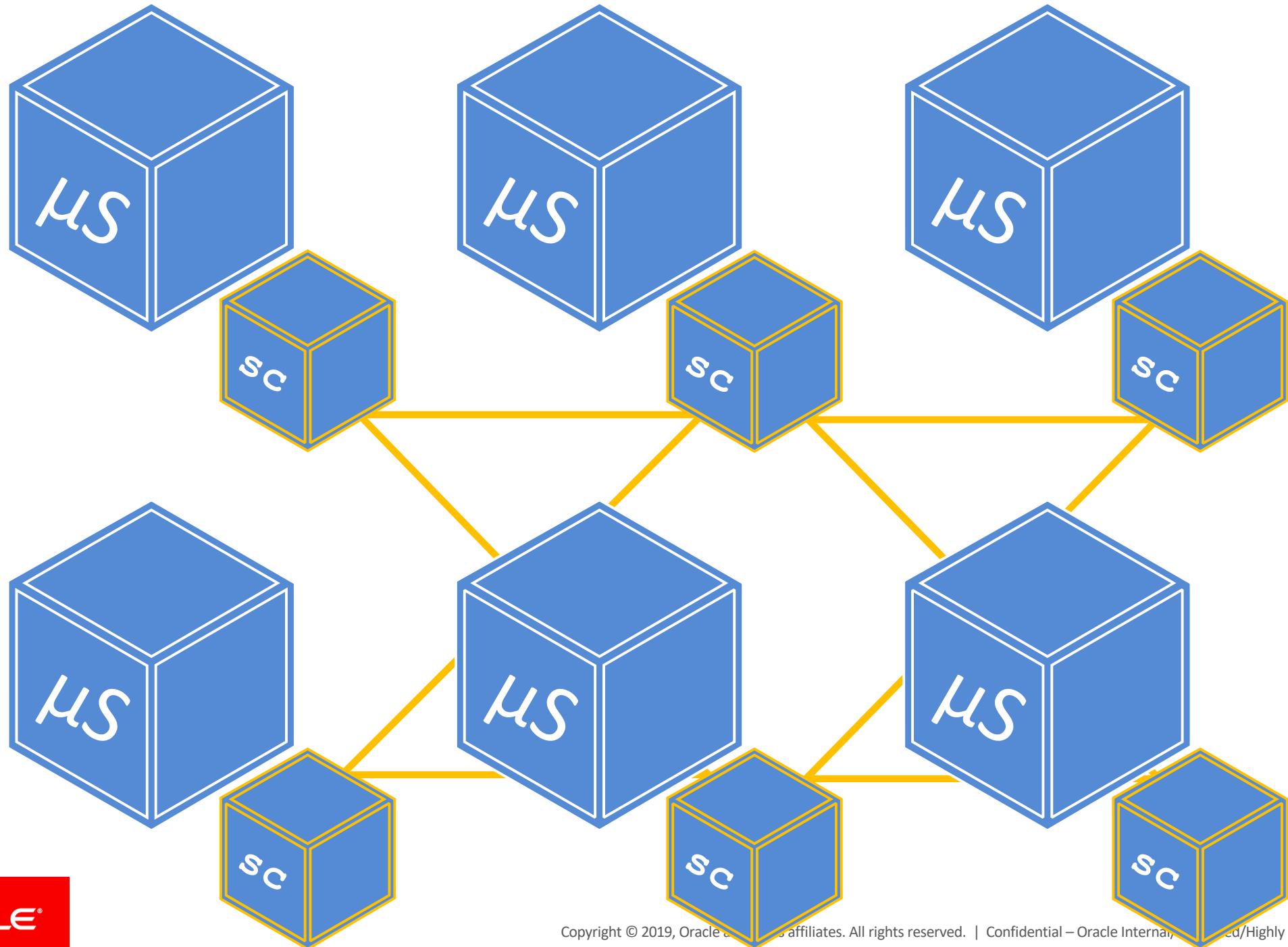
4

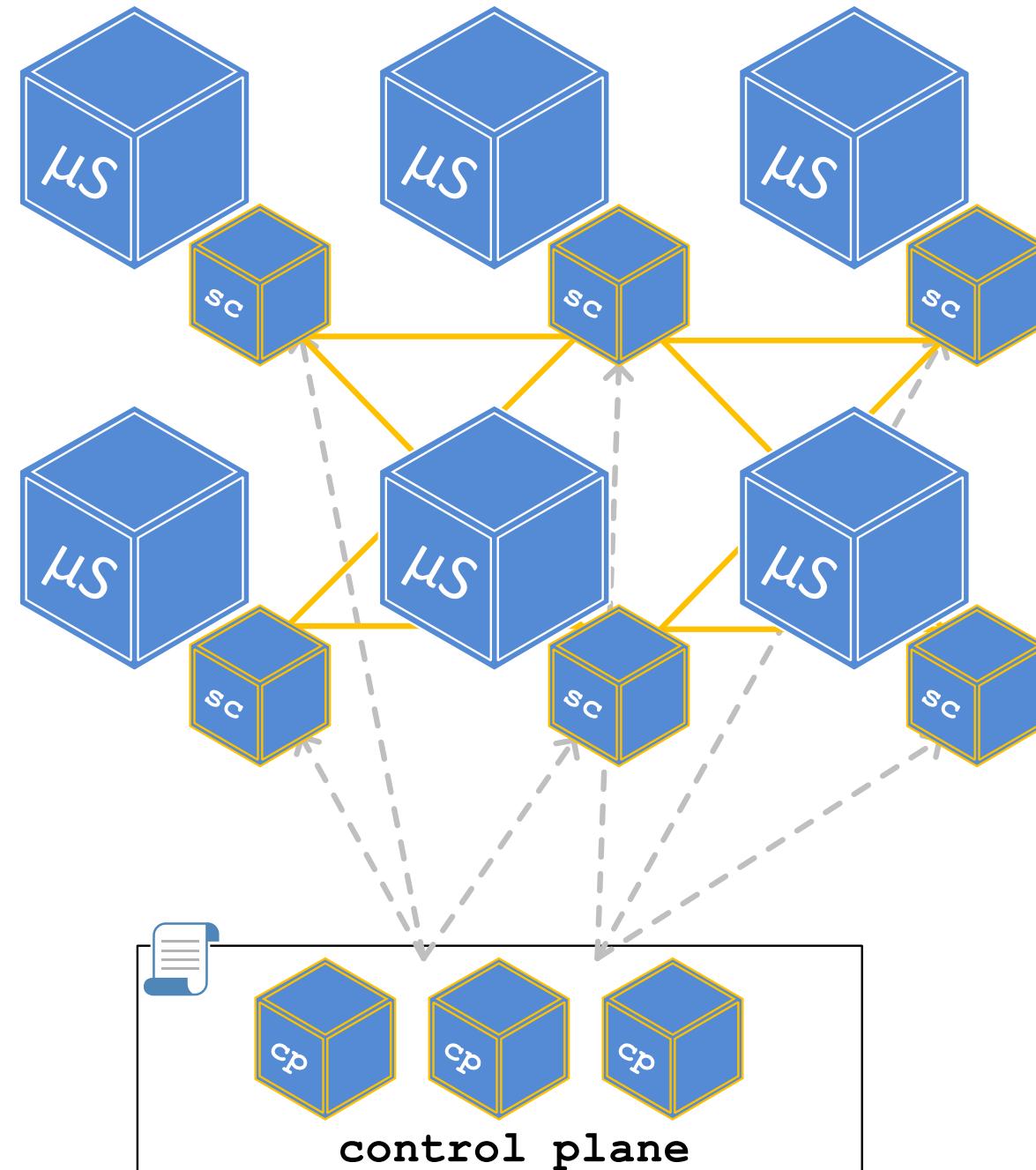




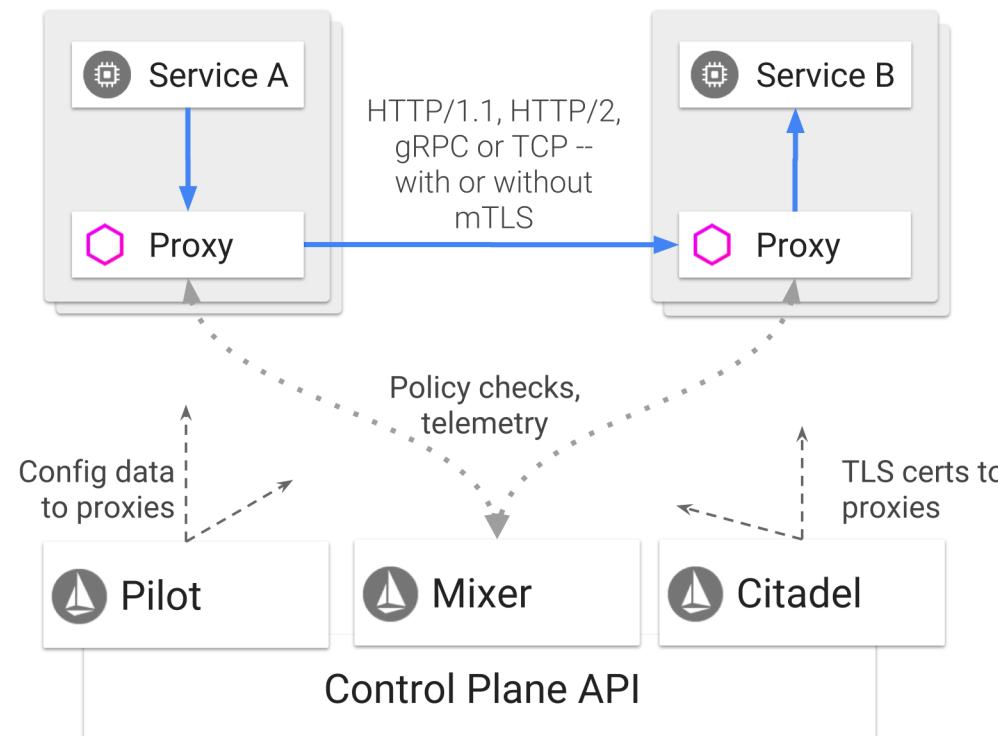
Dedicated **infrastructure layer** for managing service-to-service communication to make it **safe, fast, reliable and configurable**







Istio Service Mesh Architecture



Logically split into a **data plane** and a **control plane**.

The **data plane** is composed of a set of intelligent proxies ([Envoy](#)) deployed as sidecars. These proxies mediate and control all network communication between microservices along with [Mixer](#), a general-purpose policy and telemetry hub.

The **control plane** manages and configures the proxies to route traffic. Additionally, the control plane configures Mixers to enforce policies and collect telemetry.

Service Mesh Architecture - Components

- **Mixer**
 - Enforces access control and usage policies & collects telemetry
- **Pilot**
 - Takes the rules (traffic, resiliency, ...) and sends it to the data plane (proxies)
- **Citadel**
 - Authentication (service-to-service and end-user)
 - Authorization control
 - Policy enforcement

Service Mesh Architecture - Components

- **Envoy** (data plane)
 - Sidecar deployment
 - Sits between the services , extracts signals about traffic behavior
 - Dynamically reconfigures itself
 - Send metrics to other components
 - Injected manually or automatically

Service Mesh Features

- Ingress, Traffic Routing
 - Allows access to the service from the outside of the cluster (gateway resource)
 - Can apply routing rules to traffic entering the cluster





DEMO

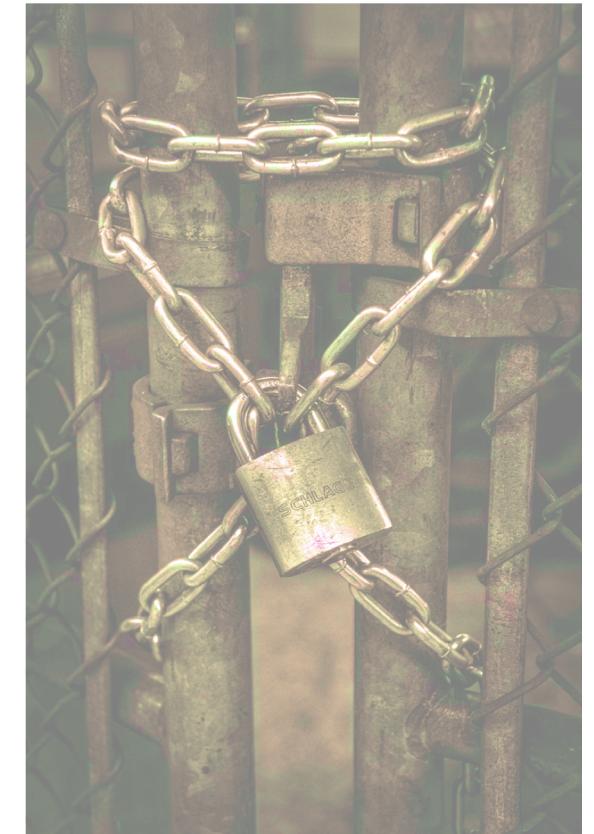
Gateway and traffic routing



**ORACLE
CODE**

Service Mesh Features

- Egress/service entries
 - Controls access to external services (i.e. services outside of the cluster)
- Security
 - Authentication (mTLS, JWT) & Authorization



DEMO

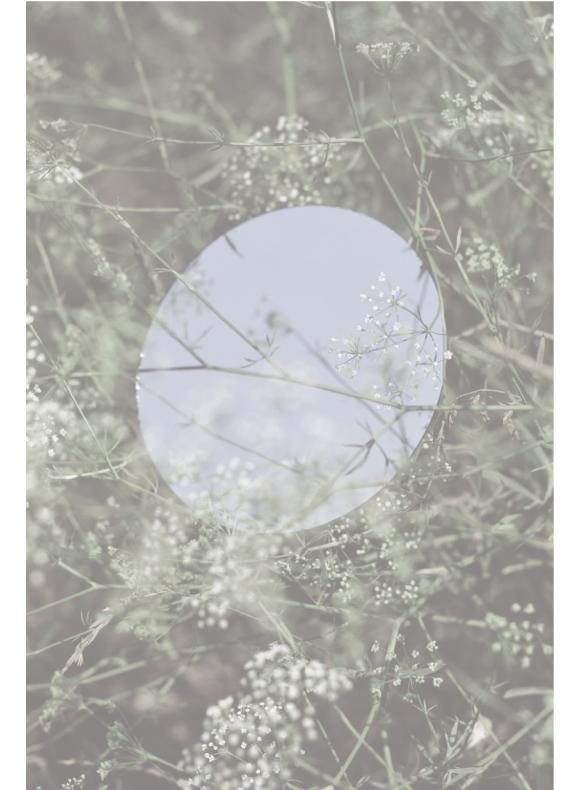
Service entries

ORACLE®

**ORACLE
CODE**

Service Mesh Features

- Resiliency
 - Retries & Timeouts
 - Circuit breakers
- Testing
 - Fault injection: delays and faults
- Traffic Mirroring
 - Duplicates production traffic and sends it to another version of the service
 - Requests are “fire and forget”, all responses are discarded



DEMO

Resiliency and testing

ORACLE®

**ORACLE
CODE**

Thank you!

<http://bit.ly/ocode-BLR>

Email: peter.jausovec@oracle.com

Twitter: @pjausovec