Break New Ground

Path to Resilient and Observable Microservices

Peter Jausovec
Twitter: @pjausovec
Consulting Member of Technical Staff, Oracle

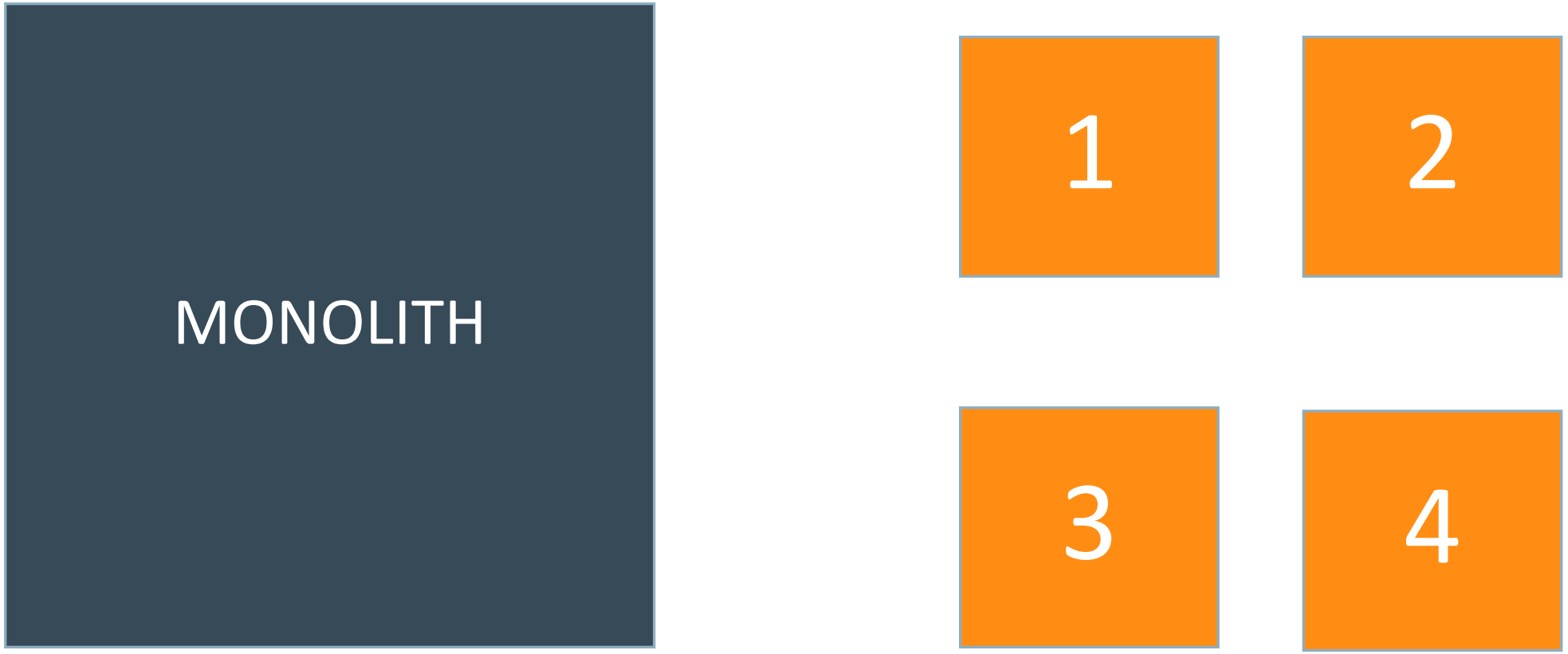
ORACLE CODE

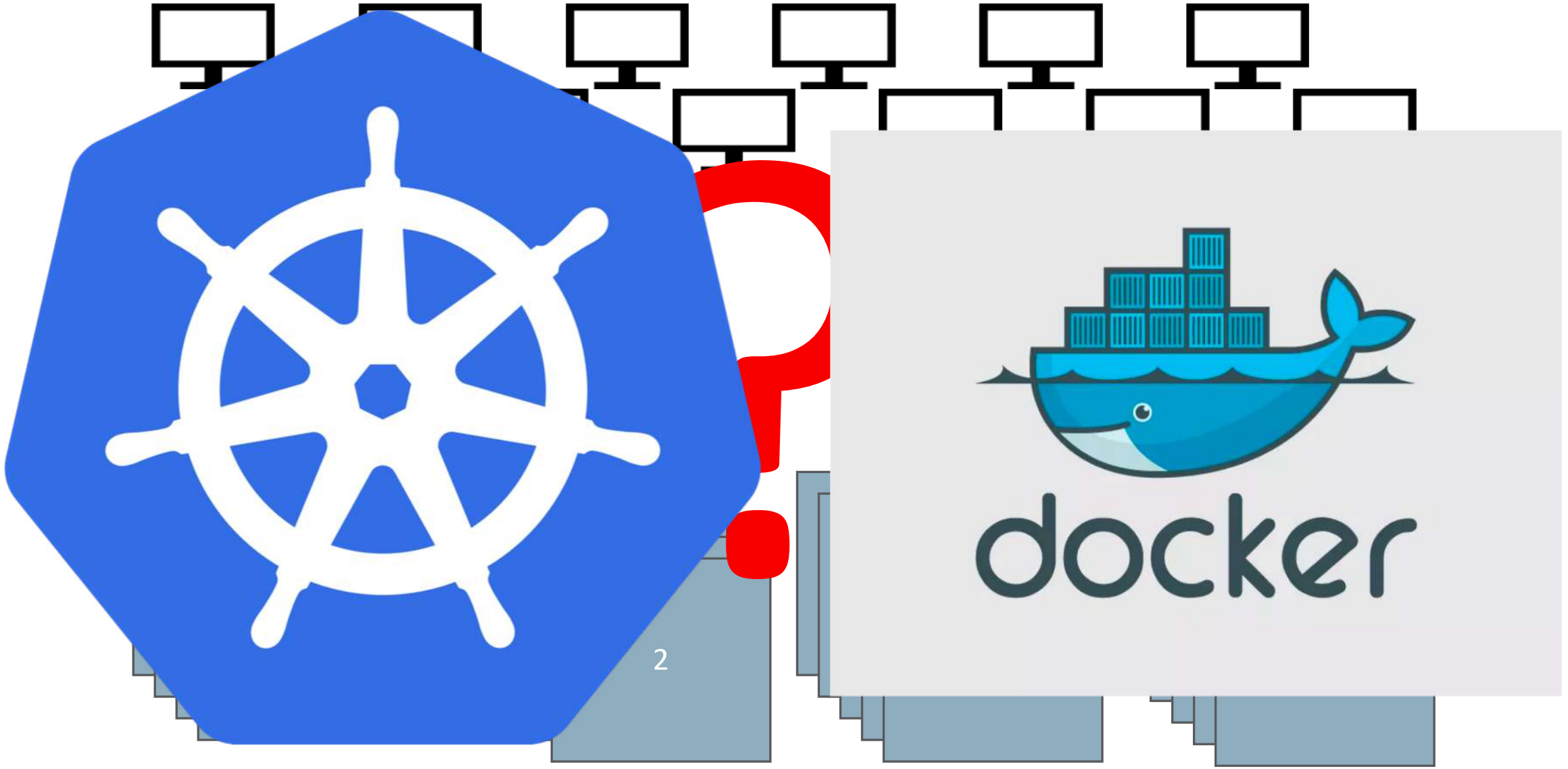
ORACLE

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?





Ability to **provide and maintain** an **acceptable level** of service in the face of **faults and challenges** to normal operation

Ability to **recover from failures** and **continue to function**

After failure occurs, return the service to a fully functioning state

High Availability

Healthy

No (significant) downtime

Responsive

Meeting SLAs

Disaster Recovery

HA design can’t handle impact of faults

- Data backup
- Archiving

Path to resiliency

Understand the requirements

Define service availability

Design for resiliency

Strategies for detection & recovery

Testing, testing, testing

Monitoring

Resiliency Strategies

Load balancing

Timeouts & retries

Circuit breakers & bulkhead pattern

Data replication

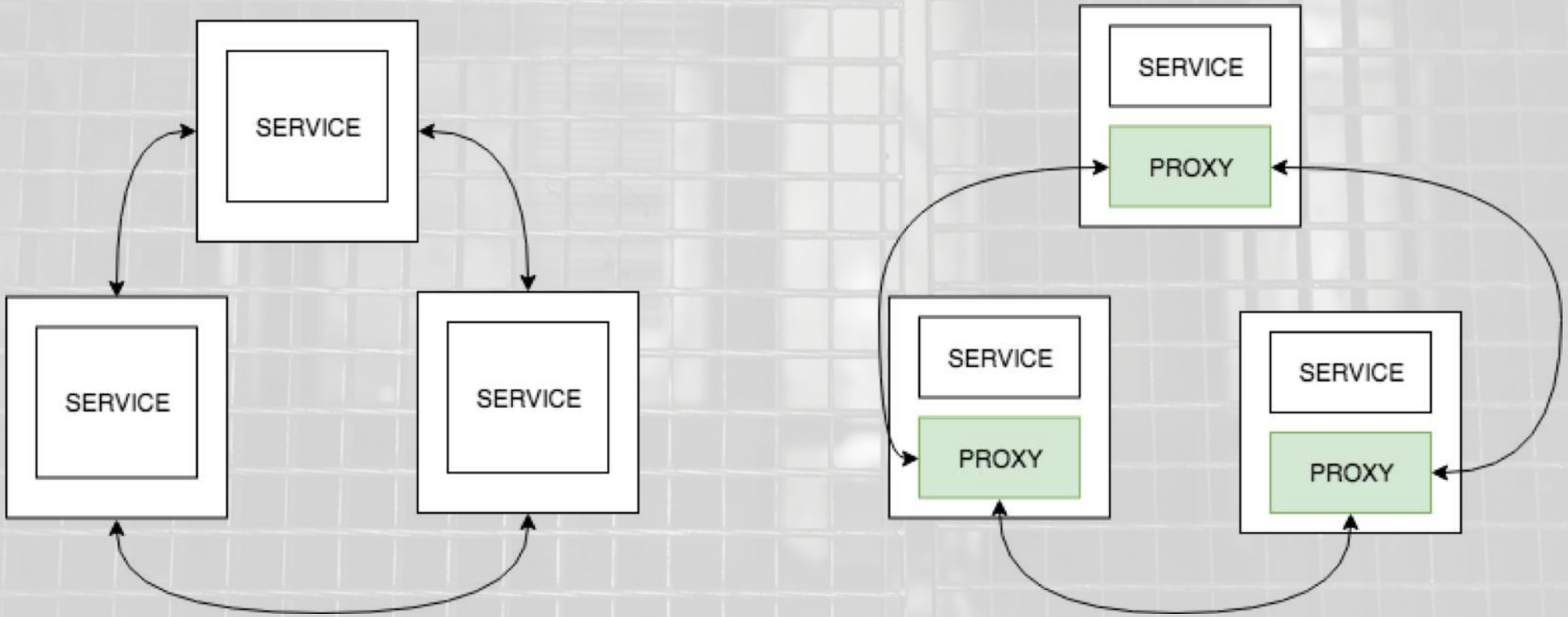
Graceful degradation

Rate limiting

Resiliency Strategies

How to do it?

A dedicated infrastructure layer for managing service-to-service communication to make it manageable, visible and controlled



Resiliency Strategies with Service Mesh

Load balancing

Timeouts & retries

Circuit breakers & bulkhead pattern

Data replication

Graceful degradation

Rate limiting

Testing for Resiliency

Test

Measure

Analyze (fix the issues)

Testing for Resiliency with Service Mesh

Injecting failures:

- Delays
- Faults

Observability is the act of **measuring, collecting and analyzing** metrics, traces, logs, events, … from services

instrument → collect & store → analyze, visualize and alert

Observability – collecting metrics

Use Correlation IDs
Structured logging
No private information!
Unique metrics names
Categorize
Common format
Context matters

Observability – collecting metrics

Prometheus

Counters, gauges, histograms, summaries

Observability – visualizing

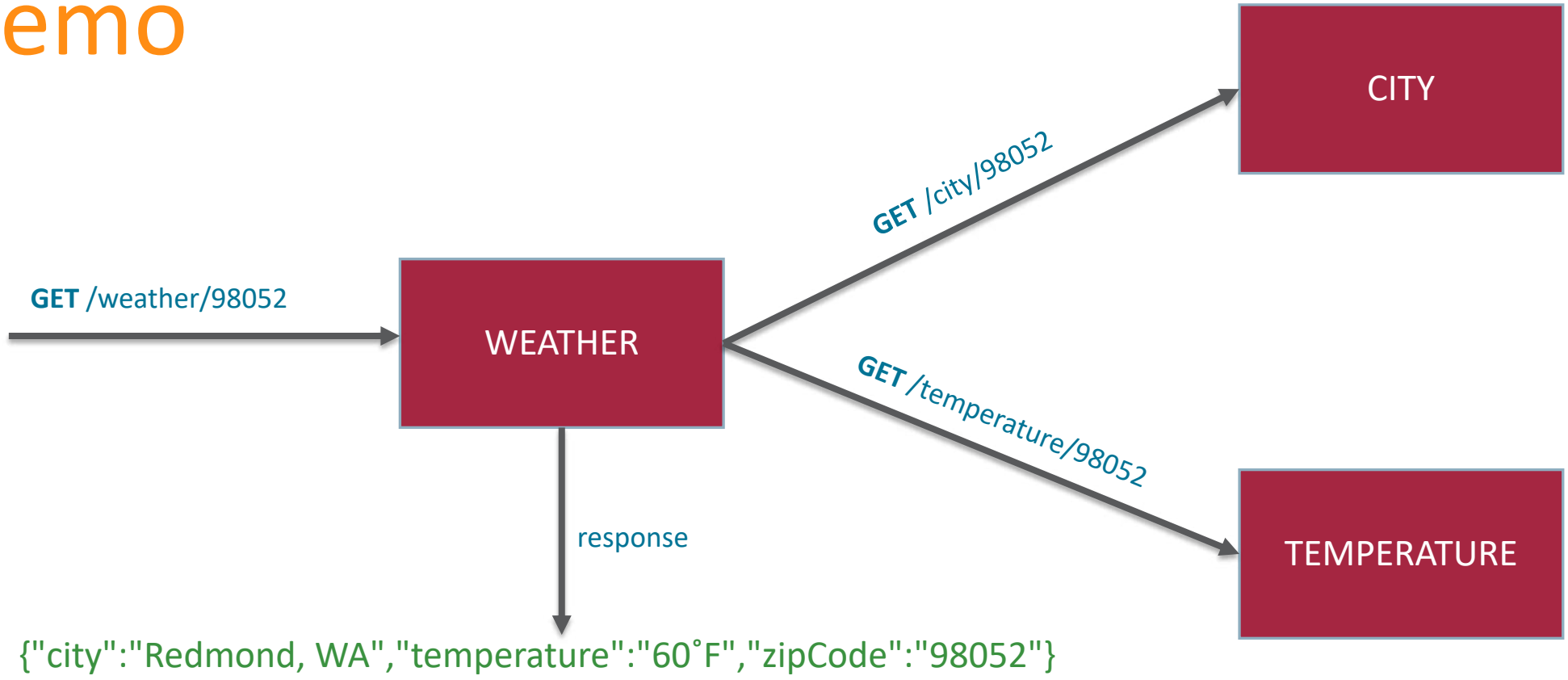
Grafana

Jaeger

EFK (Elasticsearch + Fluentd + Kibana)

PagerDuty

Demo



DEMO

Service Resiliency, Observability and Testing

ORACLE CODE

ORACLE

Thank you!

http://bit.ly/oracle-code-BLR

Email: peter.jausovec@oracle.com

Twitter: @pjausovec