Coherence Community Edition and Helidon 2.0

Will Lyons, Randy Stafford, Dmitry Kornilov, Aleks Seovic, Dave Cabelus

Oracle Enterprise Cloud Native Java

June 25, 2020

Safe Harbor

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.

Statements in this presentation relating to Oracle's future plans, expectations, beliefs, intentions and prospects are "forward-looking statements" and are subject to material risks and uncertainties. A detailed discussion of these factors and other risks that affect our business is contained in Oracle's Securities and Exchange Commission (SEC) filings, including our most recent reports on Form 10-K and Form 10-Q under the heading "Risk Factors." These filings are available on the SEC's website or on Oracle's website at http://www.oracle.com/investor. All information in this presentation is current as of September 2019 and Oracle undertakes no duty to update any statement in light of new information or future events.

Coherence Community Edition and Helidon 2.0

- Java microservices
- Open source
- Innovative
- Polyglot architectures

Coherence CE

helidon

Today's Speakers

Randy Stafford

Dmitry Kornilov

Aleks Seovic

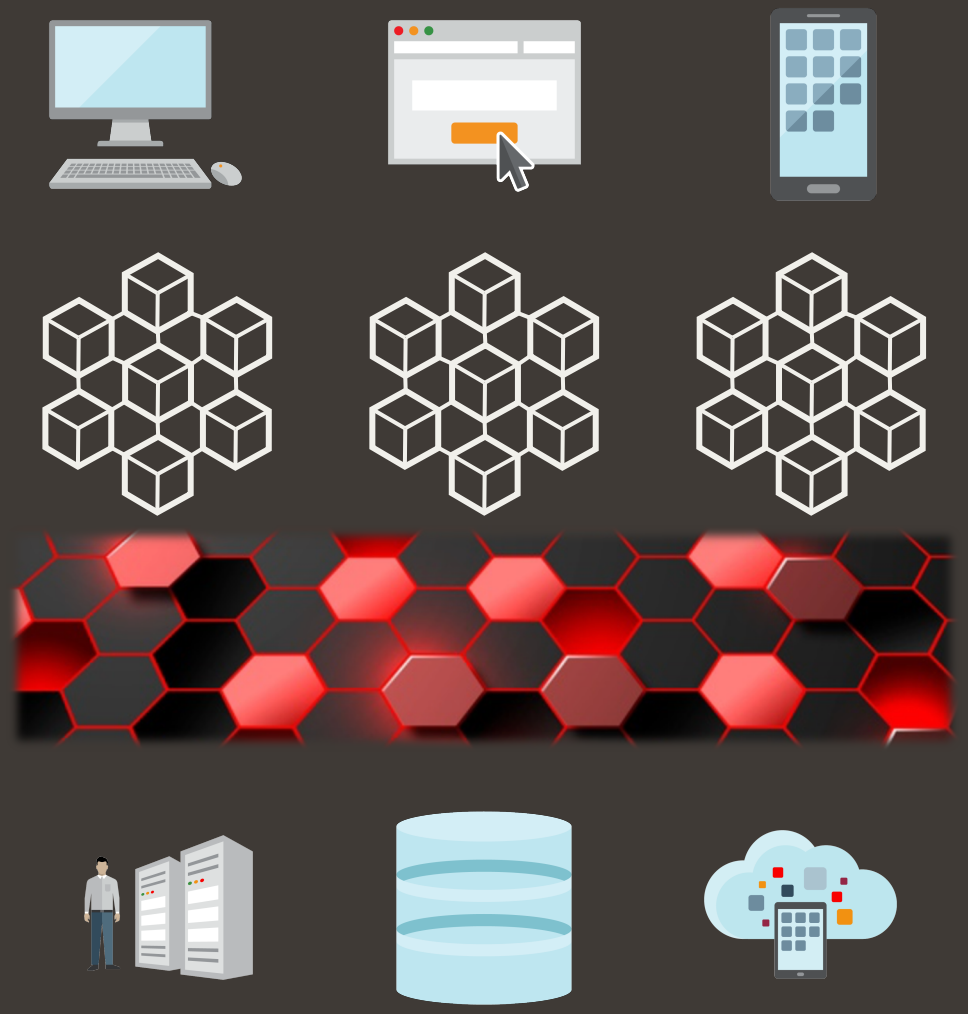
Dave Cabelus

Announcing Coherence Community Edition

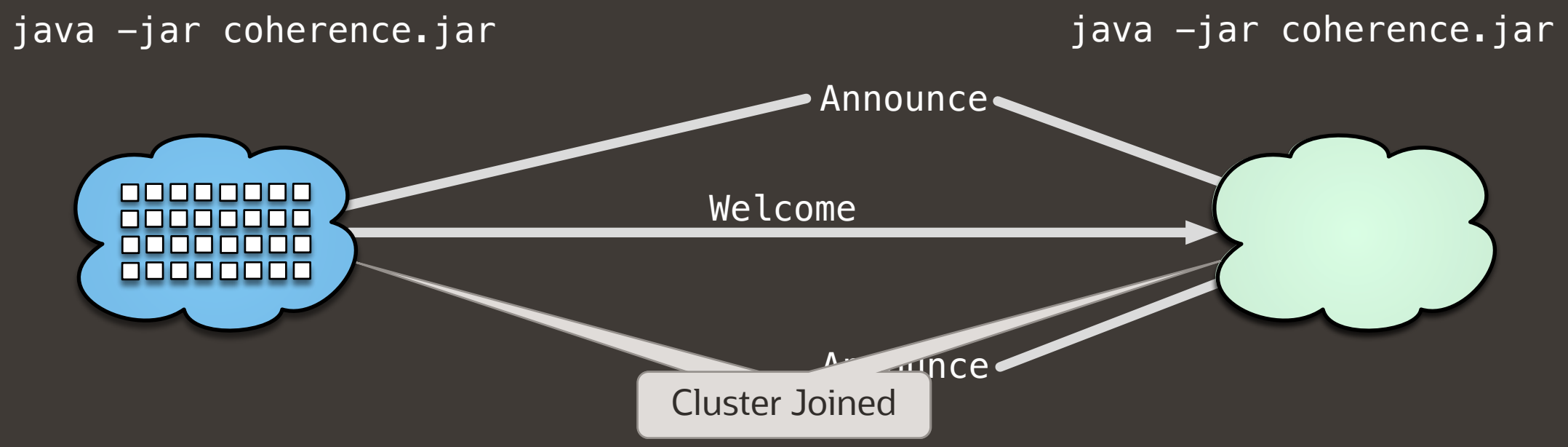
Randy Stafford
Senior Manager
Oracle Coherence Product Management
June 25, 2020

What is Coherence?

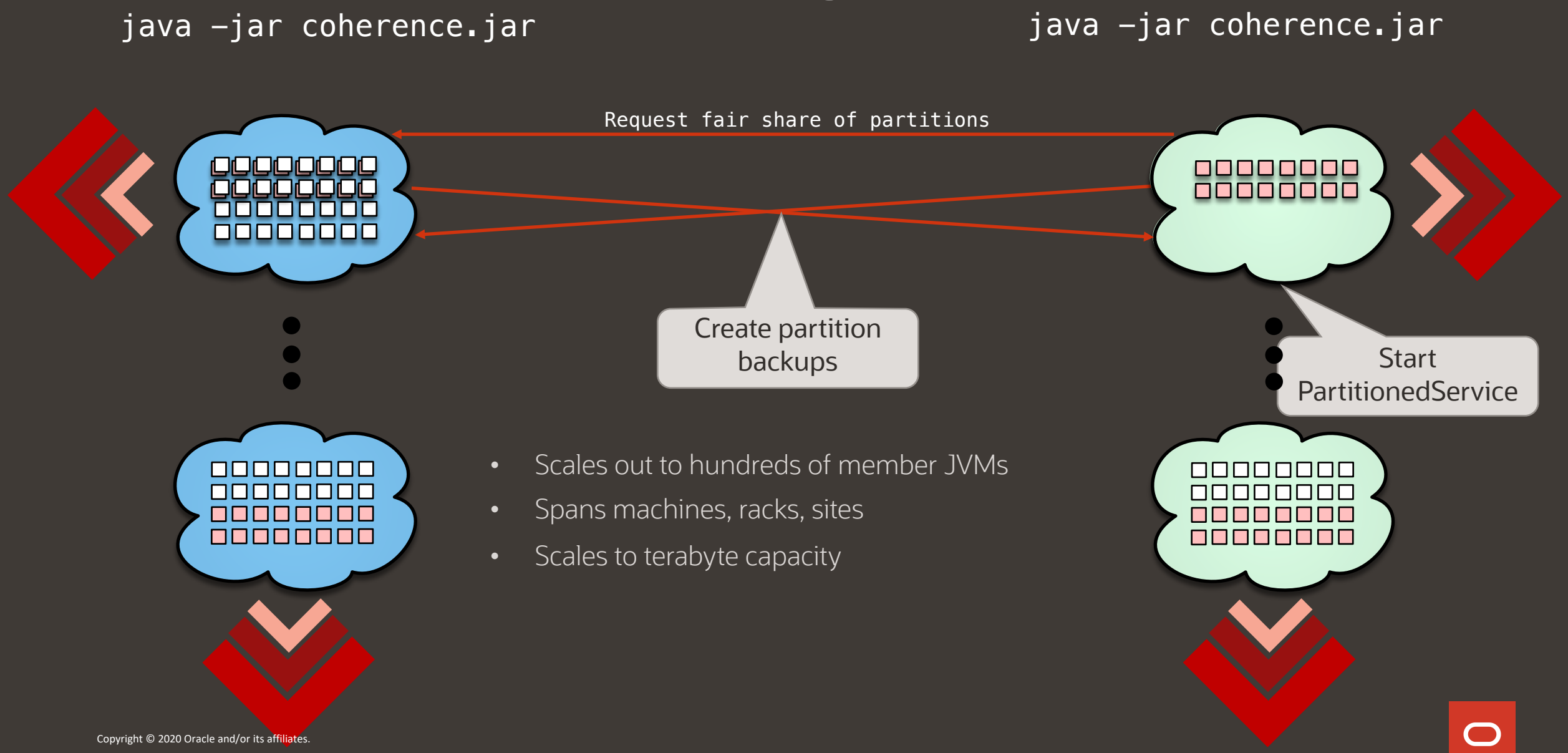
- The first and leading In-Memory Data Grid
- Clustered data management and grid computing software
- Scaling mission-critical systems since 2001



Clustering

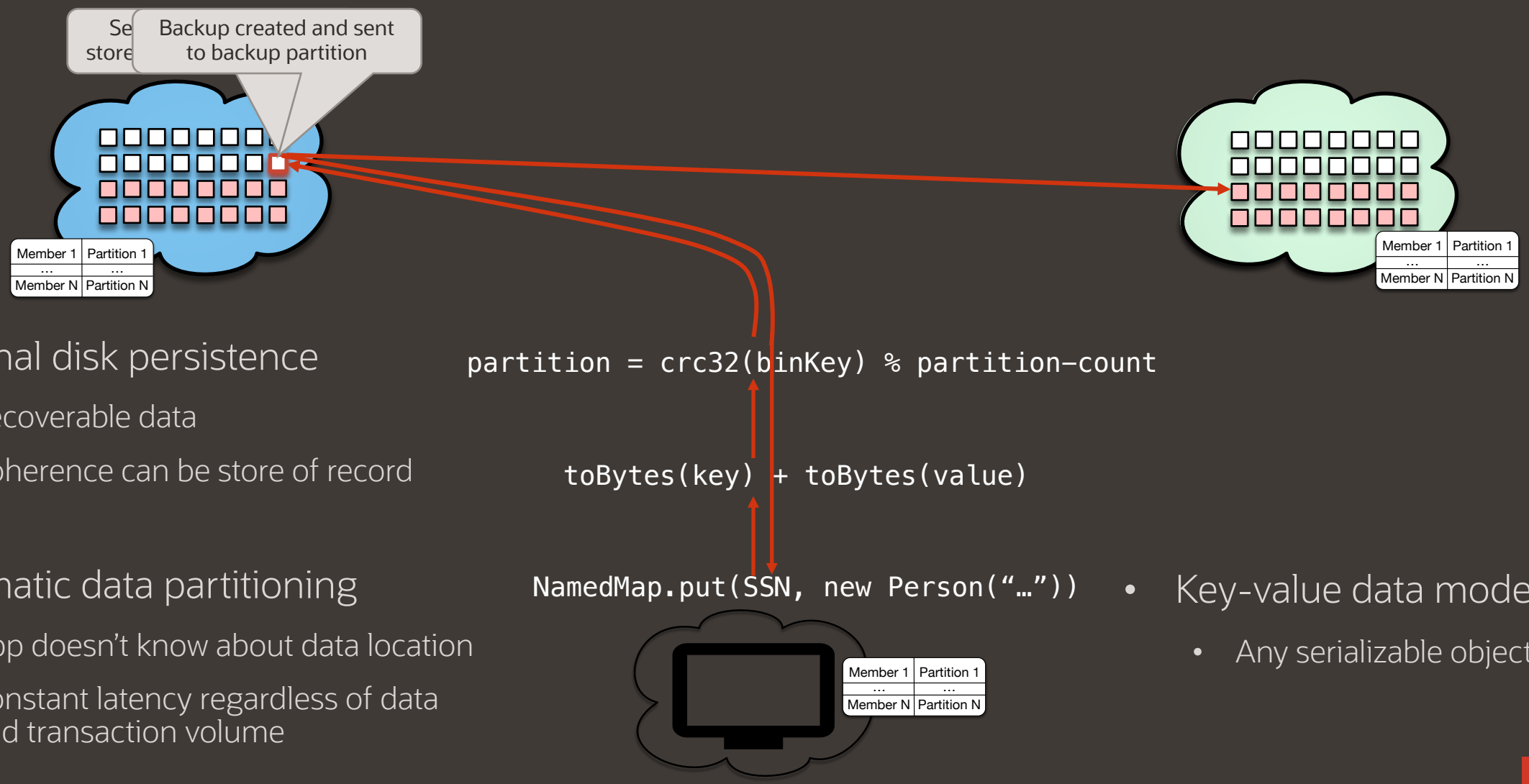


Partitioned Data Storage



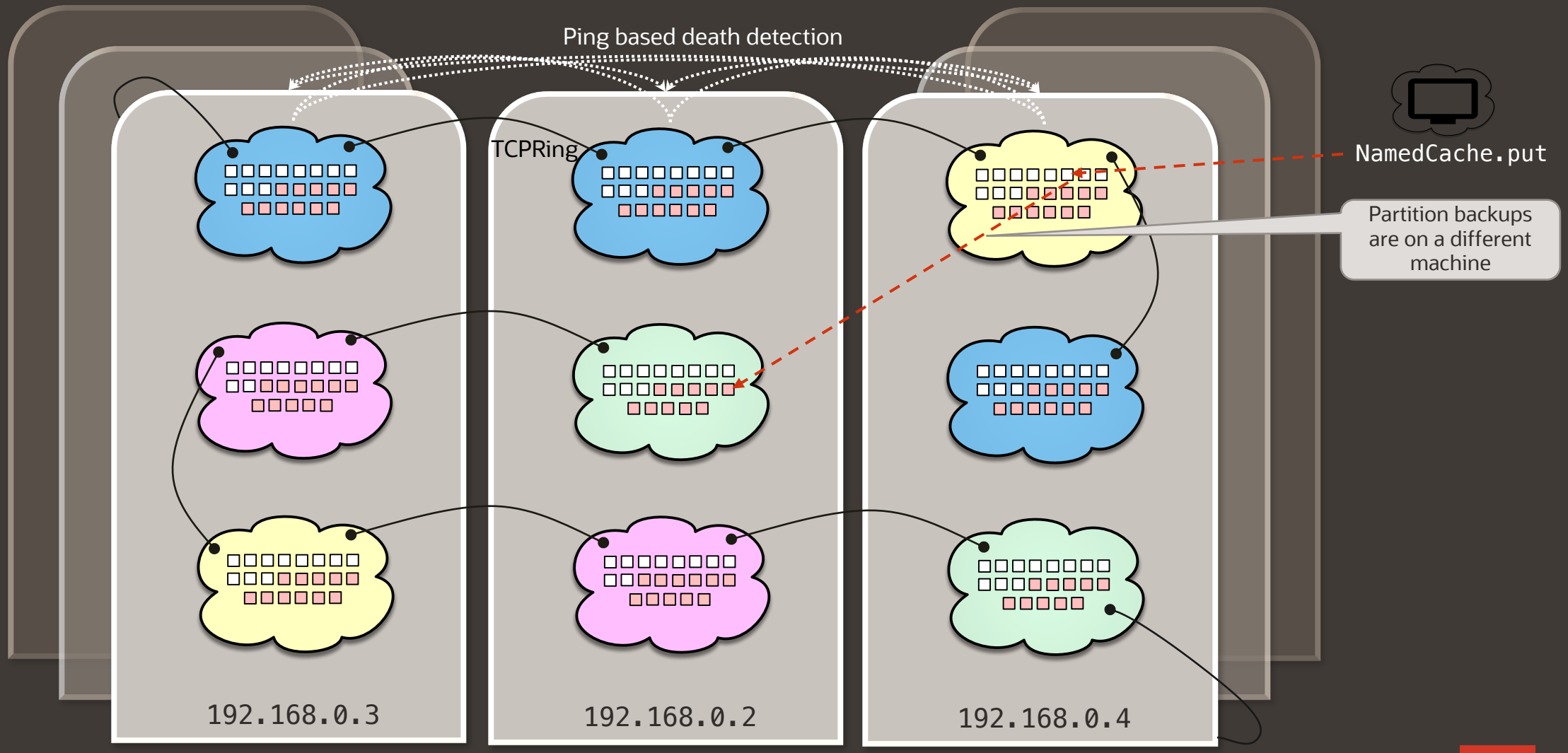
- Scales out to hundreds of member JVMs
- Spans machines, racks, sites
- Scales to terabyte capacity

Data Management: Put

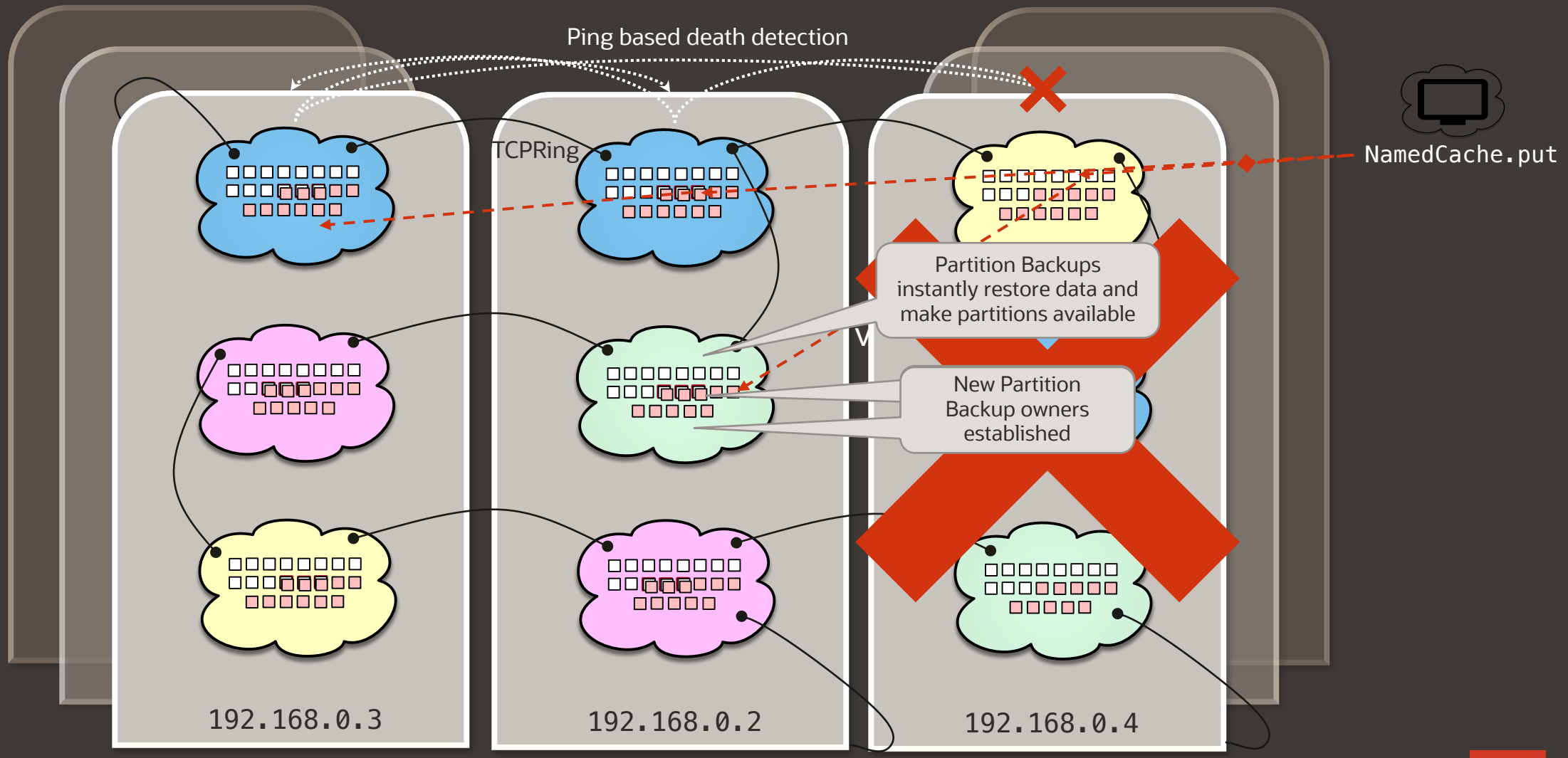


- Optional disk persistence
  - Recoverable data
  - Coherence can be store of record
- Automatic data partitioning
  - App doesn't know about data location
  - Constant latency regardless of data and transaction volume
- Key-value data model
  - Any serializable object graph

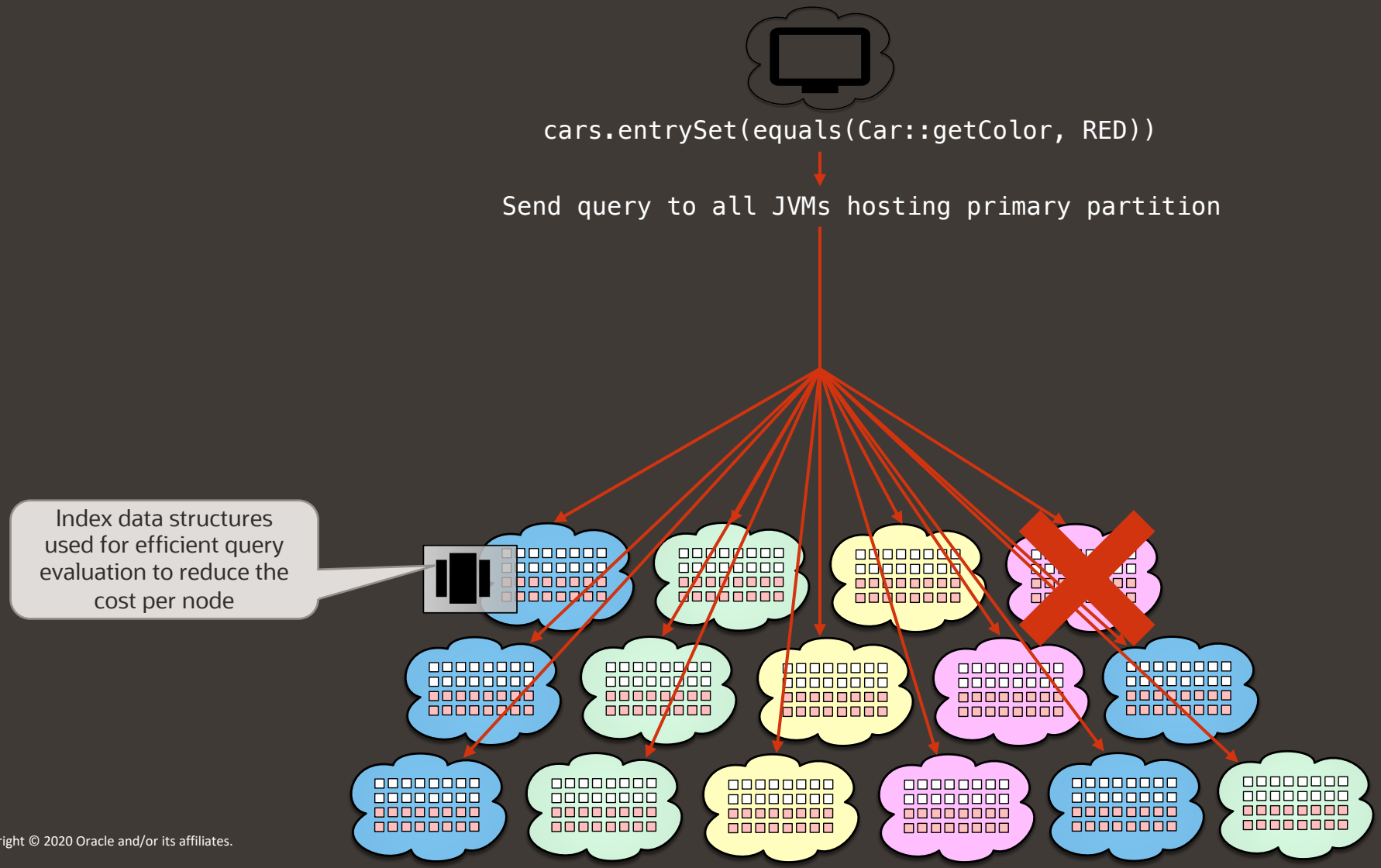
Fault Tolerance



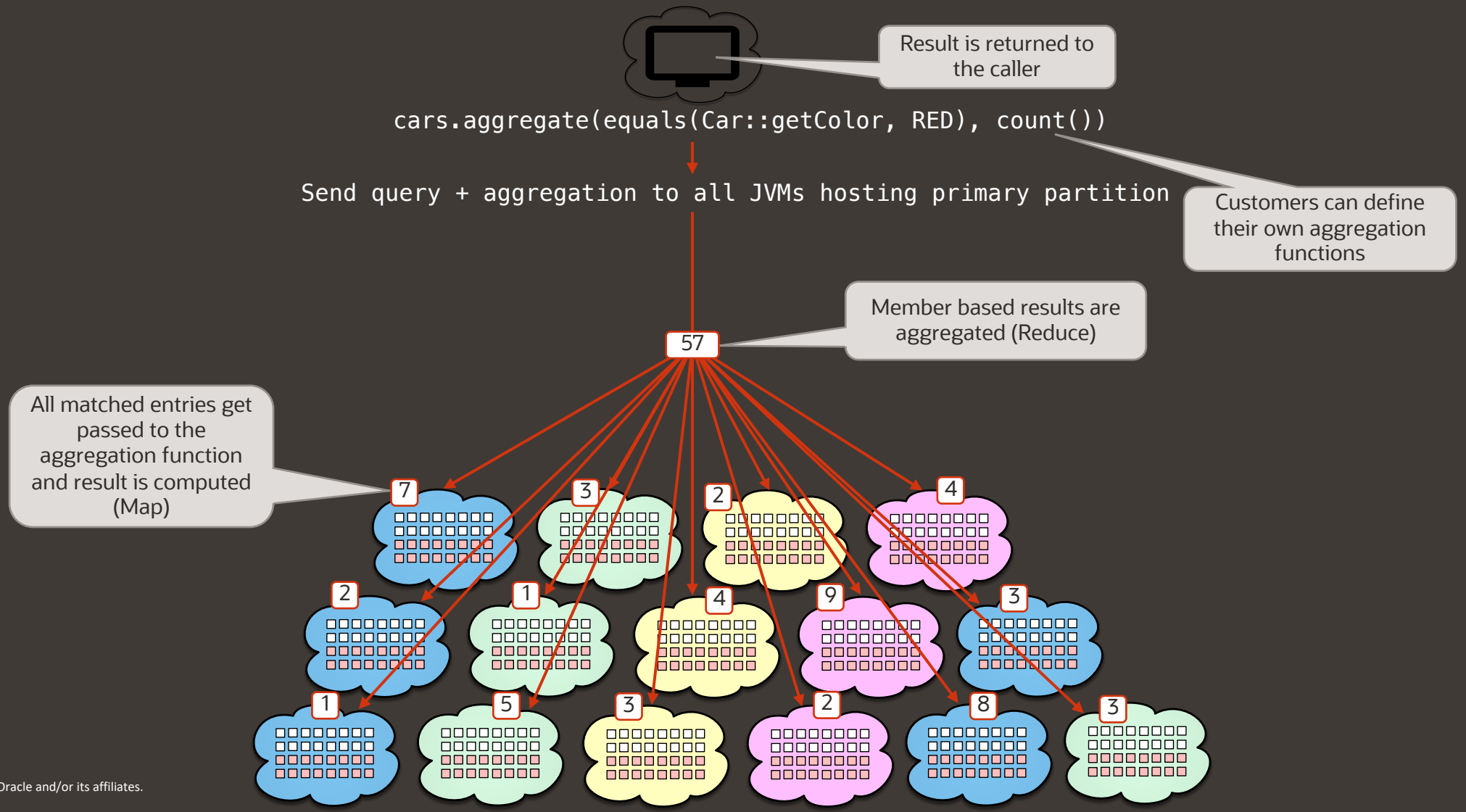
Fault Tolerance



Data Management: Distributed Queries



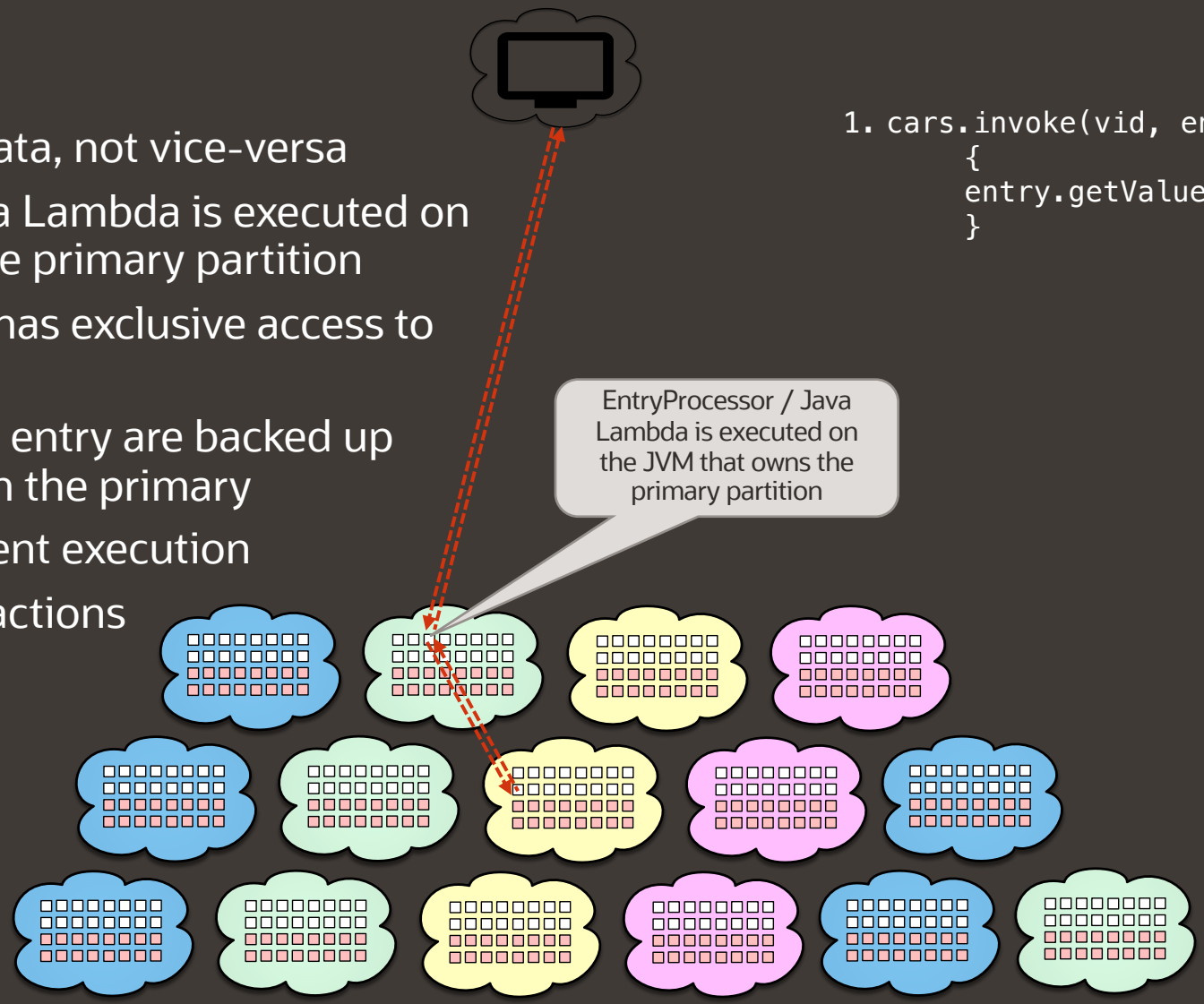
Data Management: Aggregation (Map/Reduce)



Grid Computing: In-Place Processing

- Send processing to data, not vice-versa
- EntryProcessor / Java Lambda is executed on the JVM that owns the primary partition
- During execution EP has exclusive access to the partition
- Changes made to the entry are backed up once EP completes on the primary
- Guaranteed Idempotent execution
- Partition Local Transactions

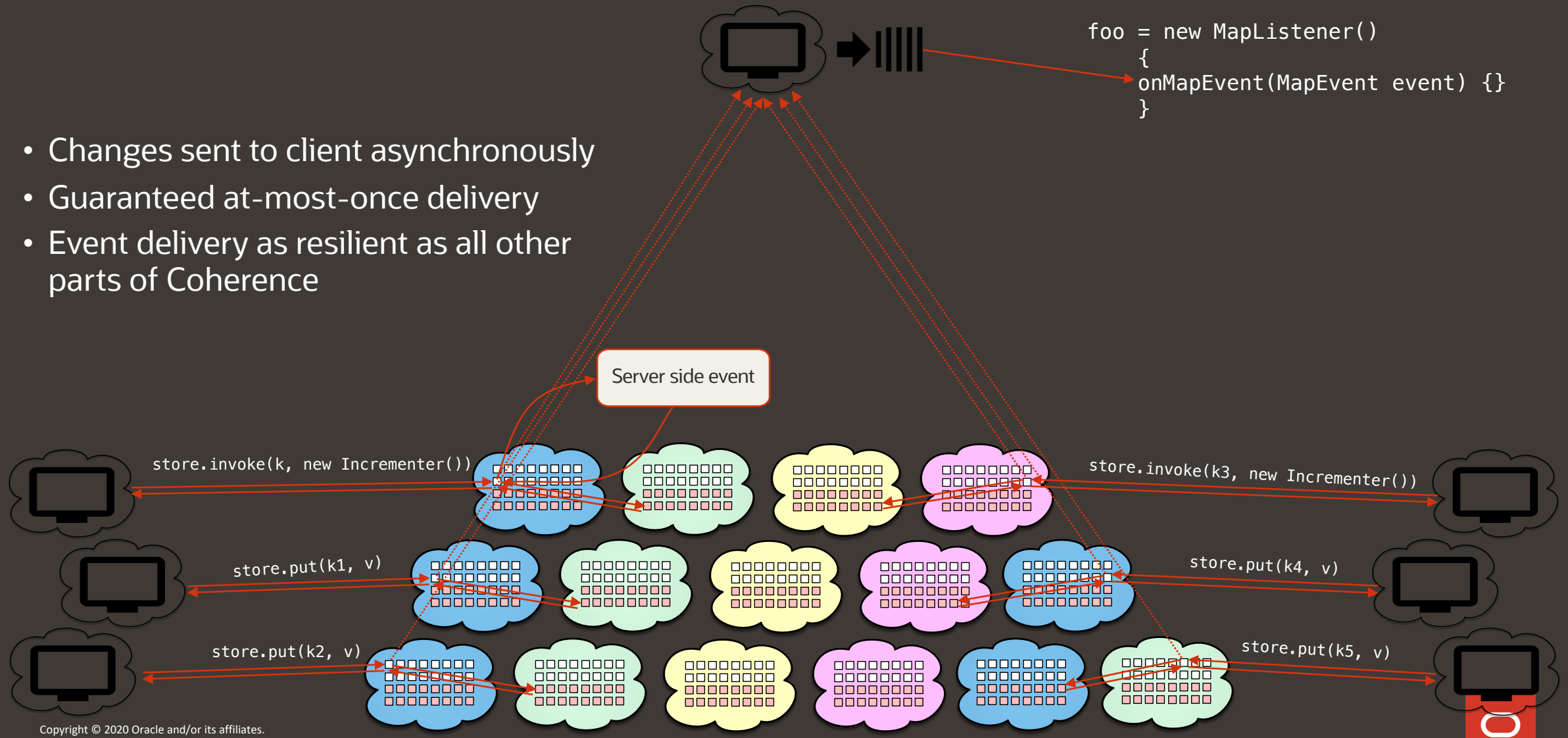
```
1. cars.invoke(vid, entry ->
        {
        entry.getValue().setPrice(…)
        }
```



Grid Computing: Eventing

```
foo = new MapListener()
    {
    onMapEvent(MapEvent event) {}
    }
```

- Changes sent to client asynchronously
- Guaranteed at-most-once delivery
- Event delivery as resilient as all other parts of Coherence



Proven Coherence Use Cases

Motive: improve performance, scalability, reliability, availability, cost of systems

Fast data access, backend offload at omni-channel scale

Application Services

Enterprise systems of record

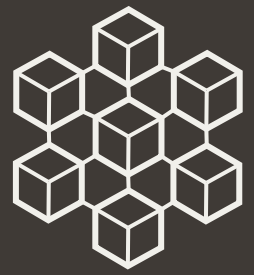
Grid computing platform for analytics and execution

Event processing and Event-driven architecture

Event stream

Event responses

System-of-Record For Microservices



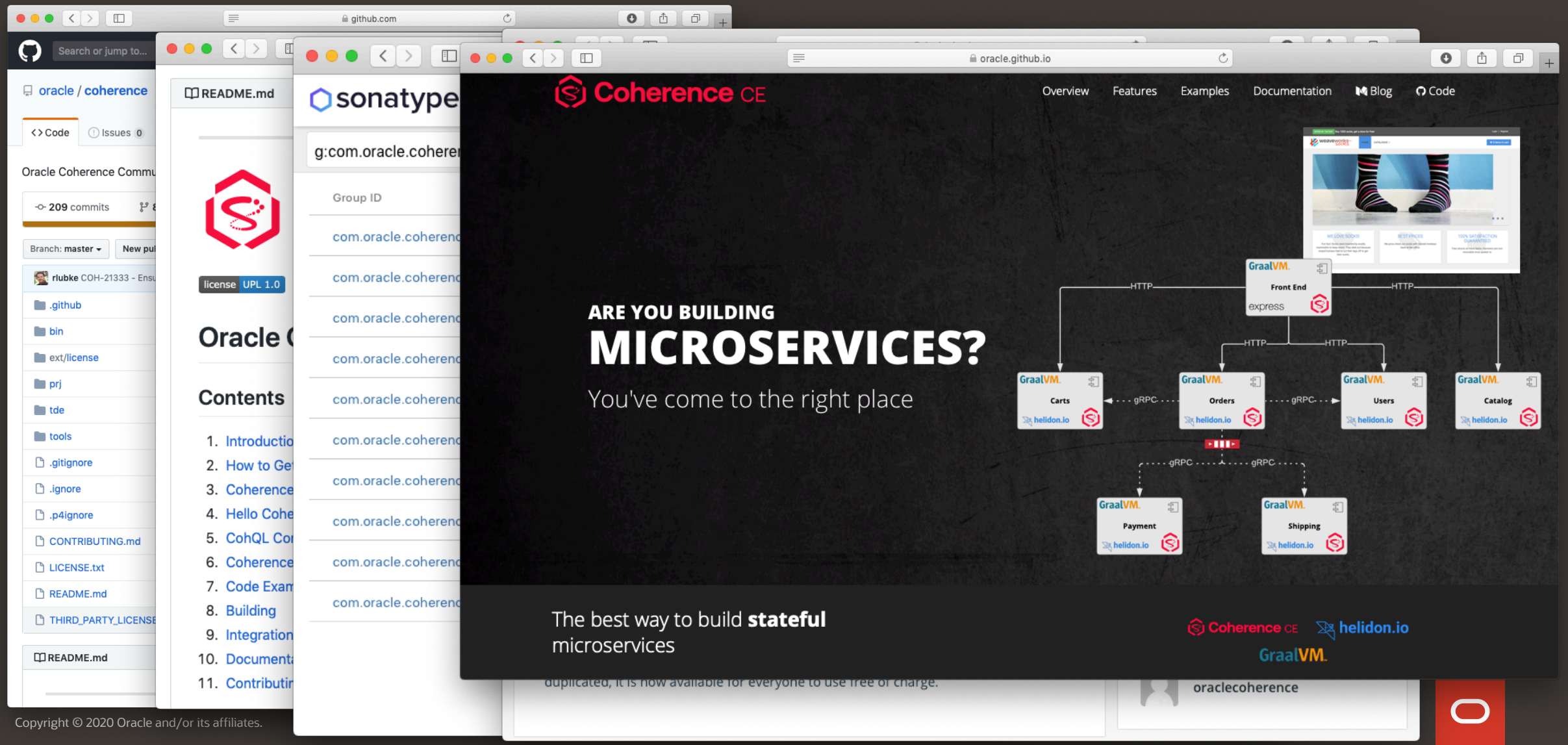
Why Coherence?

- Proven reliability at scale
  - Powering hundreds of high-scale mission critical systems across industries around the world for nearly two decades
- Track record of innovation, from creating to leading the IMDG space
  - Often imitated, but never duplicated
- Powerful foundation for polyglot microservices applications
- Free and Open-Source
- Elegant simplicity: a joy to use

Arun Giri, Union Pacific Railroad

- Associate VP, Lead Engineer for IT
- Platform Engineering Group Lead
- Enterprise Architecture Group Lead
- NetControl Architecture Lead
- 1st UP Distinguished Technologist

Hello, World!



Coherence Community Edition

- A free and open-source edition of Oracle Coherence
- Hosted on GitHub, under the Universal Permissive License (UPL)
- Artifacts published to Maven Central; Docker images to Docker Hub

- Coherence CE entitles a subset of Coherence EE features
  - Includes everything necessary to write modern cloud microservices applications
- No impact on existing licenses; a new option for new projects

- Part of a platform for cloud-native microservices applications

Coherence Editions and Entitlements

Feature	Comm Edition	Ent Edition	Grid Edition
Fault-tolerant data caching	Y	Y	Y
Data management - write-behind, partition-level transactions, analytics and events	Y	Y	Y
Local cache, Near cache, continuous query cache, real-time events	Y	Y	Y
Fully replicated data management	Y	Y	Y
Partitioned data management	Y	Y	Y
Data source integration - read-through/write-through/write-behind caching	Y	Y	Y
Persistent, Recoverable Caching	Y	Y	Y
Hibernate integration	Y	Y	Y
REST, memcached, JCache Clients	Y	Y	Y
Java Real Time Data Clients	Y	Y	Y
Scale-out Caching, Query, Aggregation, and Processing	Y	Y	Y
.NET and C++ Real-Time Data Clients	Y	Y	Y
Unlimited Real-Time Compute Clients	Y	Y	Y
JTA transactions	N	Y	Y
Oracle WebLogic Management Framework	N	Y	Y
Grid Archive (GAR) deployment	N	Y	Y
HTTP session management for application servers	N	Y	Y
TopLink-based CacheLoaders/CacheStores	N	Y	Y
WorkManager	N	Y	Y
Elastic Data	N	N	Y
GoldenGate HotCache	N	N	Y
Multitenancy	N	N	Y
Federated Caching / WAN support	N	N	Y

Releases and Support

- Oracle-numbered releases e.g. 14.1.x per Oracle schedule, supported according to Oracle Lifetime Support Policy
- YY.MM-numbered interim releases e.g. 20.06 every six months, with features in development, supported until next release
- Features from interim releases rolled into next appropriate Oracle-numbered release
- Community Edition launch includes two releases: 14.1.1.0.1 and 20.06
- Coherence CE users have GitHub tools, Coherence Slack channels for support
- Oracle Coherence support customers also have My Oracle Support for CE

Coherence CE 14.1.1 and 20.06 Features

Coherence CE 14.1.1 Features
All core Coherence features, and new in 14.1.1:
+Coherence Topics
+Distributed tracing
+GraalVM support
+Asynchronous persistence
+JDK 8 and 11 certification

Coherence CE 20.06 Features
All 14.1.1 features, and new in 20.06:
+gRPC proxy
+Java client for gRPC proxy
+CDI support
+Helidon MP integration

Platform for Cloud-Native Microservices

- Helidon
  - Standards-based application/microservices framework
  - Support for Eclipse MicroProfile specifications
- Coherence
  - Scalable, reliable, fast KV data store
  - Stateful services that are as easy to scale as the stateless services
- GraalVM
  - Polyglot runtime
  - Faster JVM and native image support
- Verrazzano
  - Application Lifecycle Management (ALM)
  - Hybrid cloud and PCA support

helidon.io

Coherence CE

GraalVM™

verrazzano

Helidon 2.0

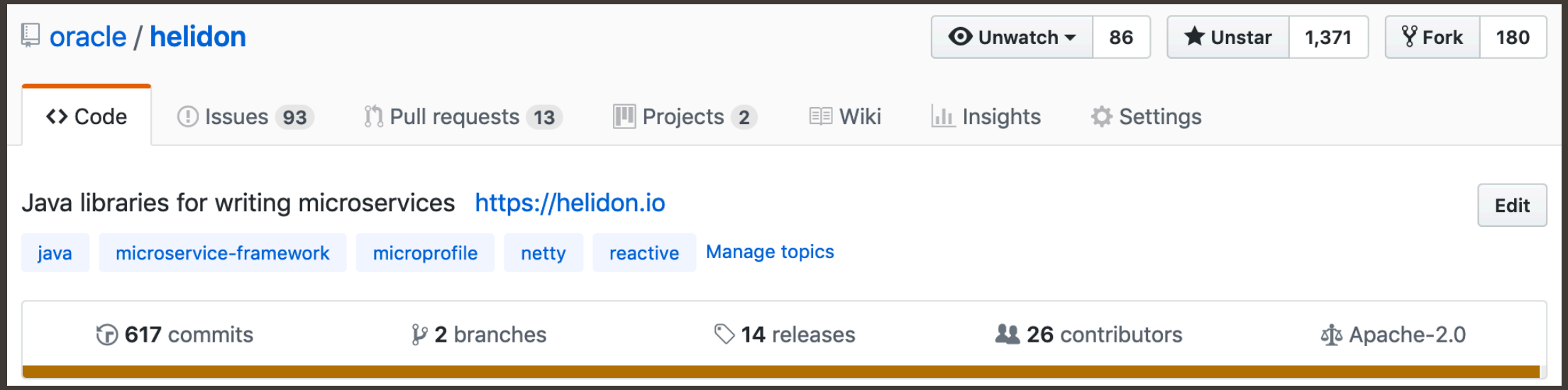
Dmitry Kornilov

Oracle
June 25, 2020

A set of Java libraries for developing microservices

Open Source

- Hosted on GitHub
  - https://github.com/oracle/helidon
- Apache 2.0 License



Landscape

Bigger

Full-stack

springBoot by Pivotal

Dropwizard

MicroProfile based

Built On Open Liberty

THORNTAIL

Apache TomEE

kumuluzEE

payara MICRO

QUARKUS

Micro frameworks

Launcher

Hammock CDI Based MicroServices

SMALLRYE

MICRONAUT

Restlet

Spark

VERTX

javalin

Smaller

Landscape

Bigger

Smaller

Full-stack

spring Boot by Pivotal

Dropwizard

MicroProfile based

Built On Open Liberty

THORNTAIL

Apache TomEE

kumuluzEE

payara MICRO

Helidon MP

Micro frameworks

Launcher

Hammock CDI Based MicroServices

QUARKUS

SMALLRYE

MICRONAUT

Restlet

Spark

VERTX

javalin

Helidon SE

helidon SE

- Reactive, non-blocking microframework
- Very Fast
- Tiny Footprint
- Functional style APIs
- Transparent, no “magic”

helidon MP

- MicroProfile + some Jakarta EE components
- Fast
- Small Footprint
- Java EE style APIs
- Annotations & Dependency Injection

Oracle Enterprise Cloud Native Java

- Oracle WebLogic Server
  - Jakarta EE application server
- Oracle Coherence
  - In-memory data grid
- Helidon
  - Java libraries for microservices
- Verrazzano
  - Hybrid application management

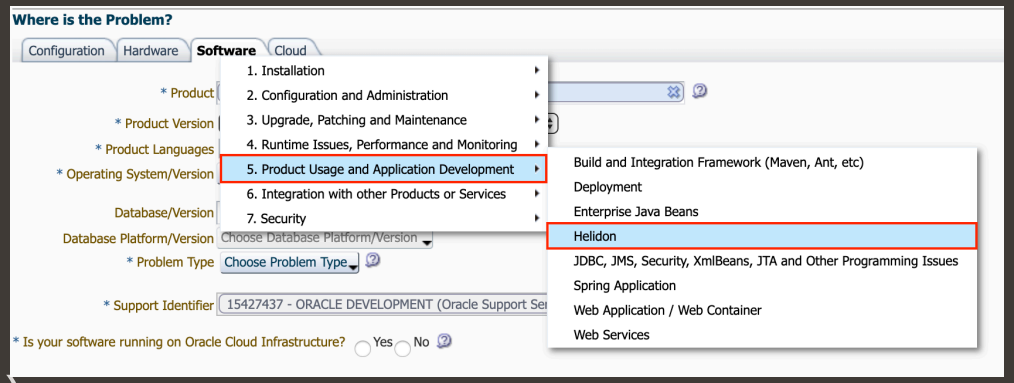
Supported on Premises and in the Cloud

New Releases and Innovation

Current and Future Application Needs

Enterprise Support

- Included in Oracle Support contracts for the following Oracle products with the same support level:
  - Oracle WebLogic Server Standard Edition
  - Oracle WebLogic Server Enterprise Edition
  - Oracle WebLogic Suite
  - Oracle Coherence Enterprise Edition
  - Oracle Coherence Grid Edition



- Access to My Oracle Support (MOS)
  - https://www.oracle.com/support/
- Helidon users with Oracle Support for Helidon are encouraged to use releases in *Active* or *Maintenance* status.

Helidon SE 2.0

Reactive Web Server	Health Check	gRPC Server and Client	CORS	Reactive Streams
Config	Metrics	Security	DB Client	Reactive Messaging
Tracing			Web Client	WebSocket

Existing Components | Added in Helidon 2.0 | Reworked | Experimental

Helidon MP 2.0

MicroProfile Config	MicroProfile Fault Tolerance	Jakarta Restful Web Services	Jakarta Persistence	MicroProfile Reactive Streams Operators
MicroProfile Metrics	MicroProfile JWT Auth	Jakarta JSON Processing	Jakarta Transactions	MicroProfile Reactive Messaging
MicroProfile Health Check	MicroProfile REST Client	Jakarta JSON Binding		Jakarta WebSocket
MicroProfile Tracing	MicroProfile Open API	Jakarta CDI		CORS

MicroProfile Components | Jakarta EE Component | Added in Helidon 2.0

Build & Tools

Executable jar

GraalVM native-image

Jlink

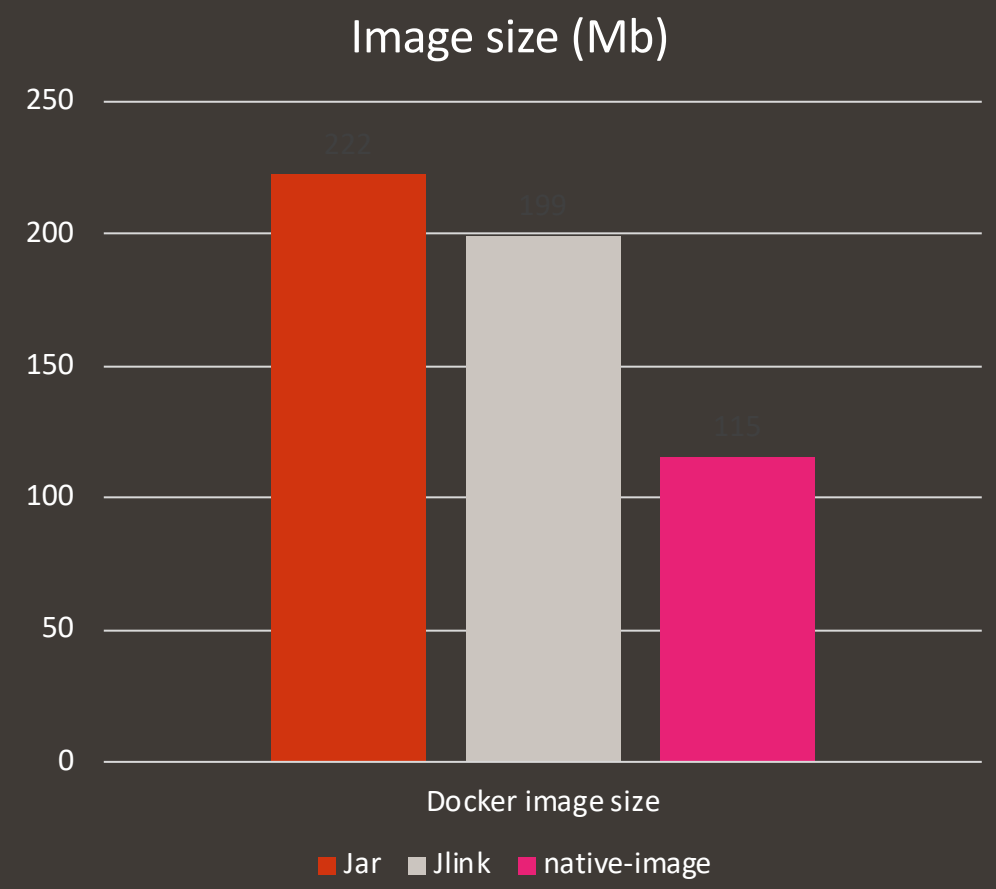
Helidon CLI

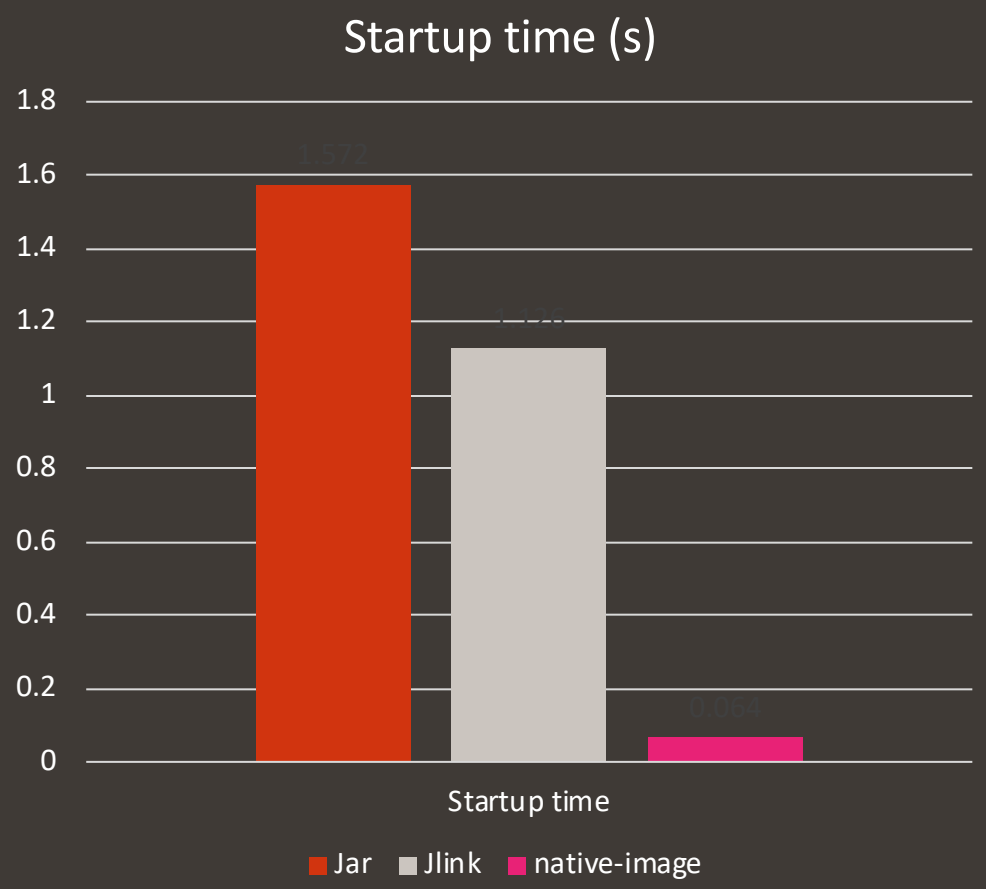
Added in Helidon 2.0

Build Profiles

- Jar
  - Executable hollow jar
  - All third-party dependencies are stored separately to take advantage of Docker layering
- Jlink image
  - Jlink optimized JRE + your application
  - Faster startup time and smaller image size with no code restrictions
- GraalVM native-image
  - Fastest startup time and smallest memory consumption
  - Introduces some code restrictions related to usage of runtime operations

Build Profiles Comparison





Helidon CLI

```
>> helidon
Usage: helidon [OPTIONS] COMMAND
Helidon Project command line tool
Options:
  -D<name>=<value>    Define a system property
  --verbose           Produce verbose output
  --debug             Produce debug output
Commands:
  build       Build the application
  dev         Continuous application development
  features    List or add features to the project
  info        Print project information
  init        Generate a new project
  version     Print version information
Run 'helidon COMMAND --help' for more information on a command.
```

Helidon DB Client

- Reactive non-blocking database client
- Supports existing blocking JDBC drivers
- Supports relational and non-relational databases
- Supports all Helidon SE observability features
  - Metrics, tracing, health checks
- Externalized data access statements
  - Utilizes Helidon Config
- Extensible

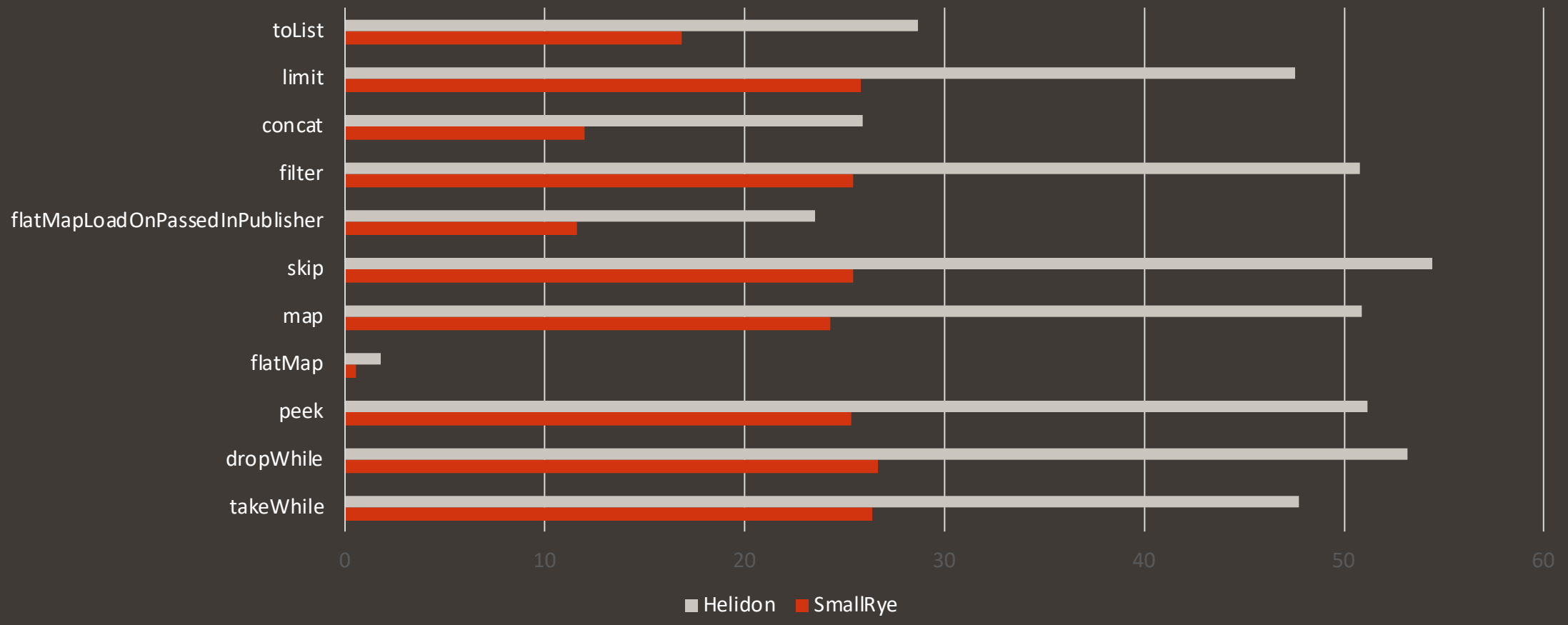
Helidon WebClient

- Reactive non-blocking HTTP client
- Designed in Helidon SE API flavor
- Support all Helidon SE observability features
- Follows Redirects
- Extensible

Reactive Streams/Messaging

- Helidon SE
  - Makes Helidon SE feature complete reactive framework
  - Implementation respects back-pressure
- Helidon MP
  - MicroProfile Reactive Streams Operators spec implementation
  - MicroProfile Reactive Messaging spec implementation
- Reactive Streams implementation is contributed by David Karnok

Reactive Streams Operators



Higher number == faster

Helidon CORS

- Same Origin Policy
  - Browser app can access resources only from same origin that provided the app
- Cross-Origin Resource Sharing
  - Controls access by web app from one origin to resources from another
  - Uses additional HTTP headers
    - Client (browser) seeks access to a host server resource on behalf of app
    - Server grants or refuses
- Helidon CORS
  - Developers can add CORS support to their SE and MP applications
  - Automatically supported in Helidon health, metrics, OpenAPI services

Learn More

https://helidon.io → Getting Started

https://medium.com/helidon

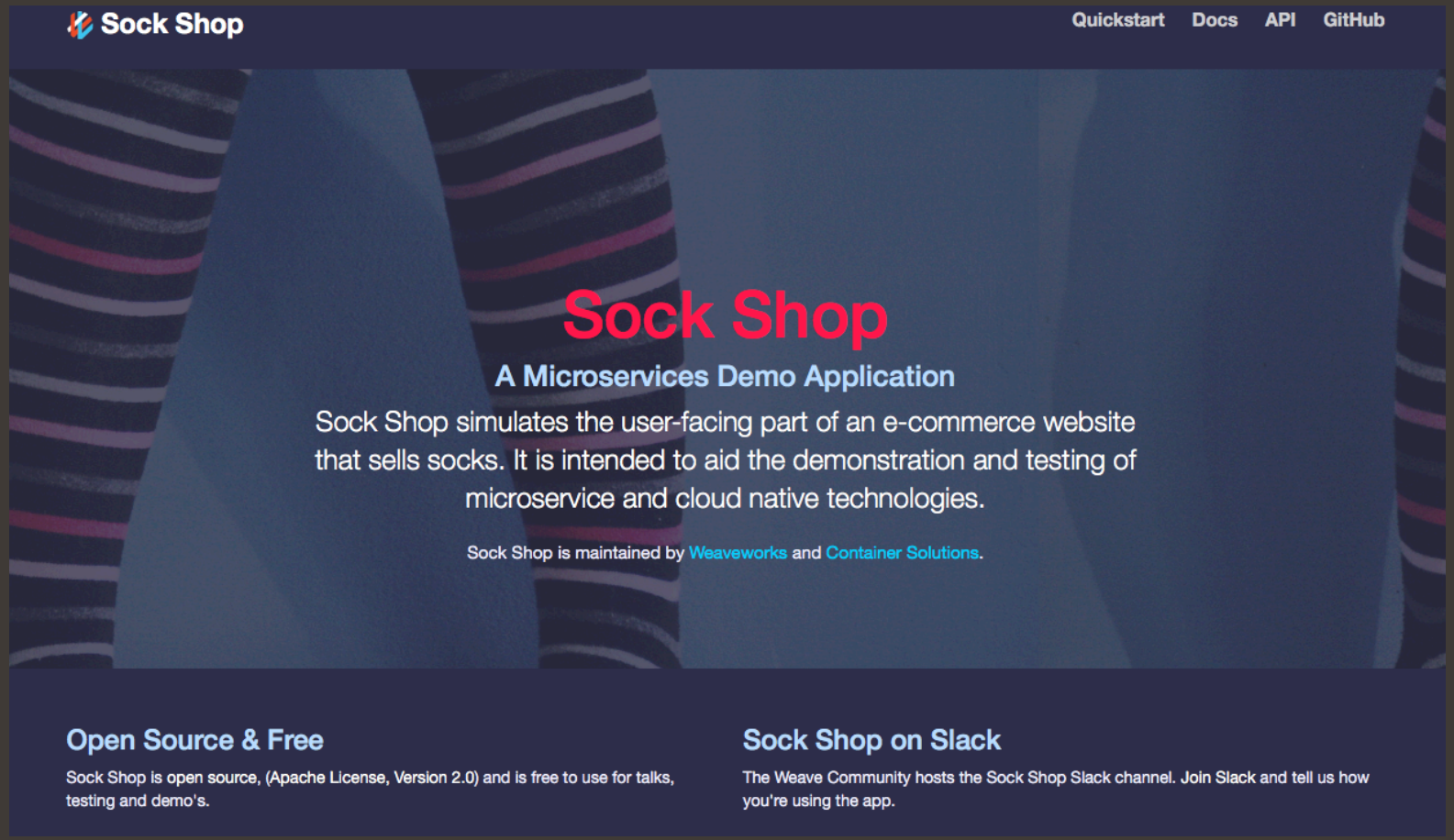
Helidon YouTube Channel - http://youtube.helidon.io/

Demo: Alex Seovic

helidon.io

Coherence CE

Introducing Sock Shop



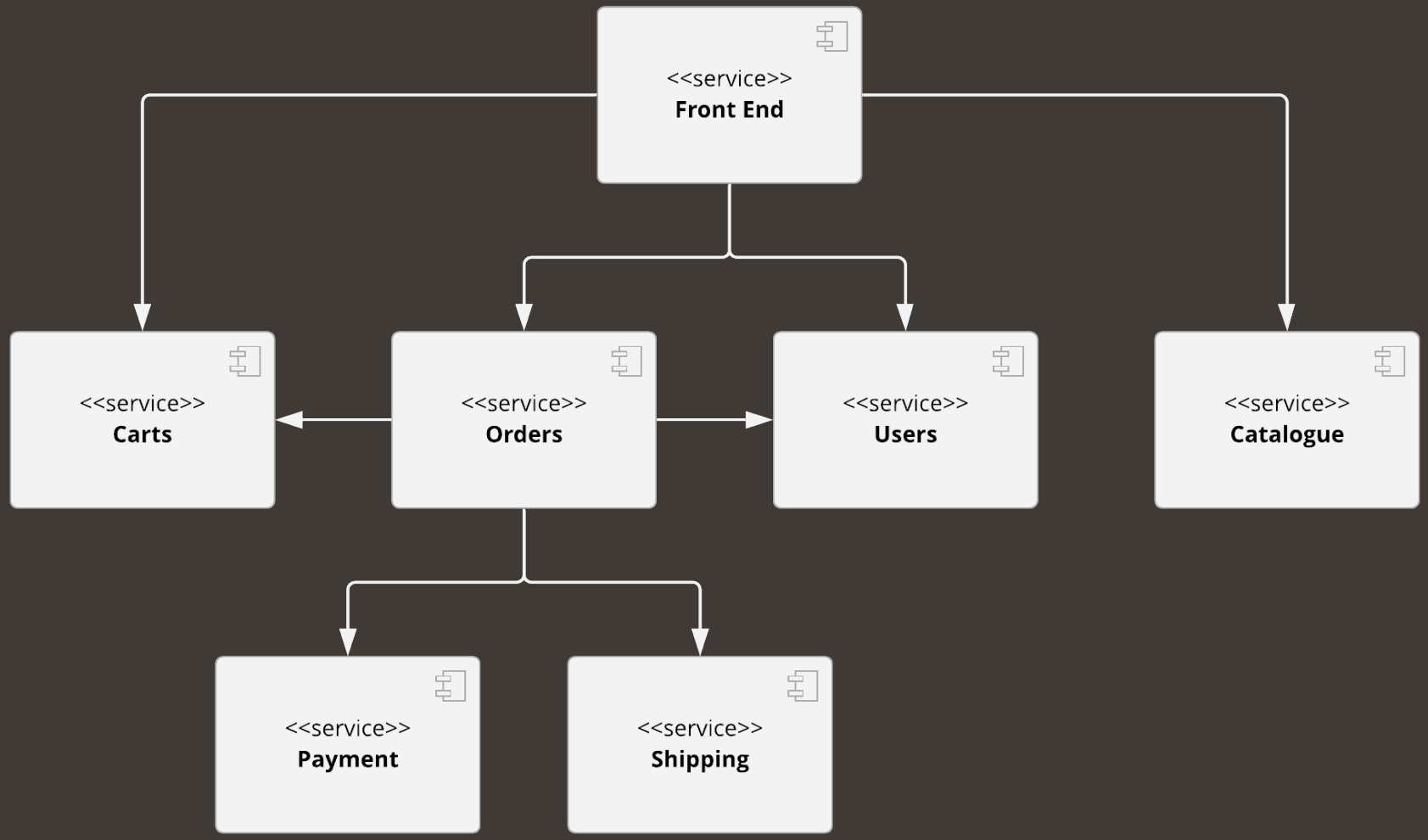
https://microservices-demo.github.io

Sock Shop Demo

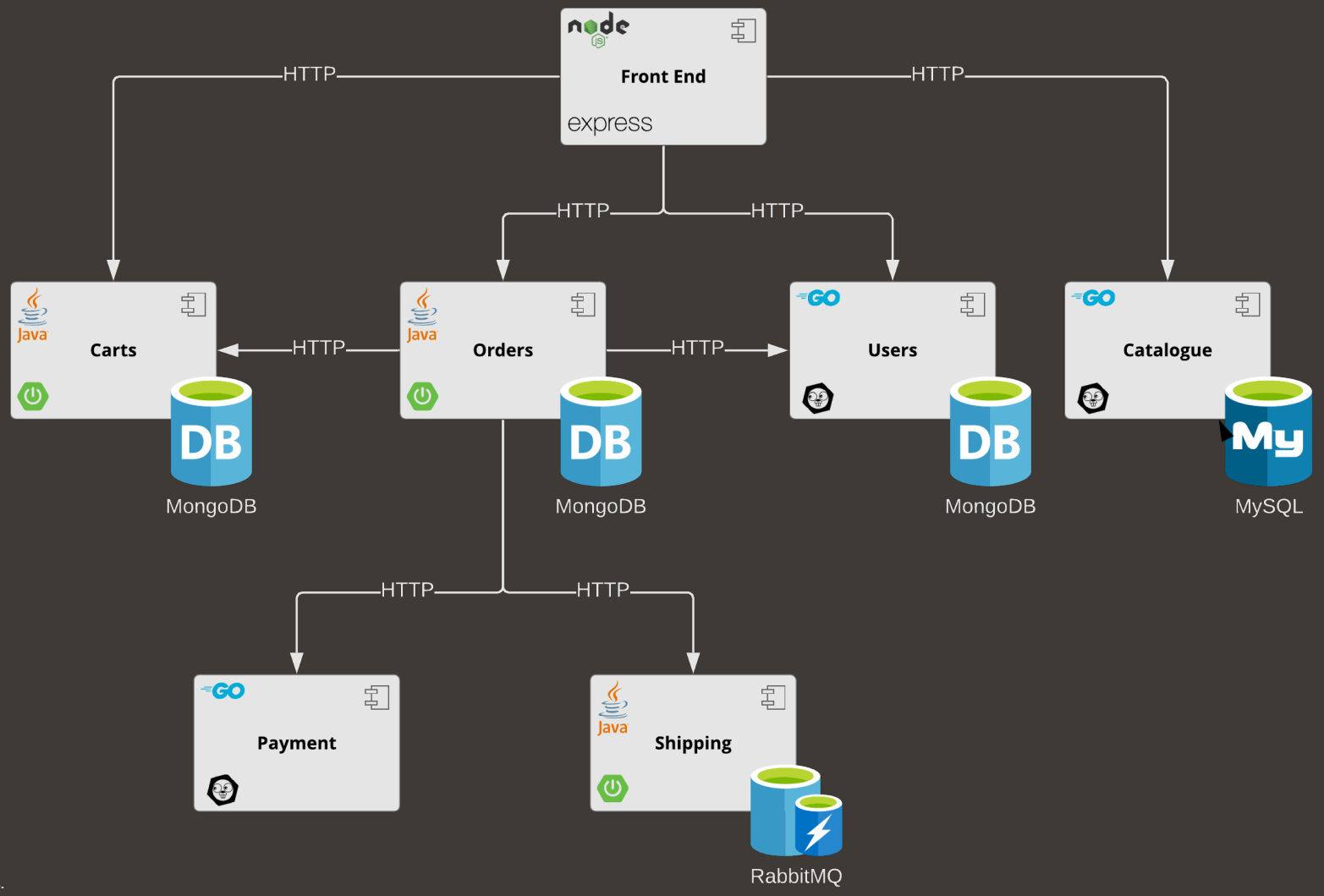
Aleks Seovic

Oracle
June 25, 2020

Sock Shop Architecture



Original Sock Shop Implementation



Here be Dragons…

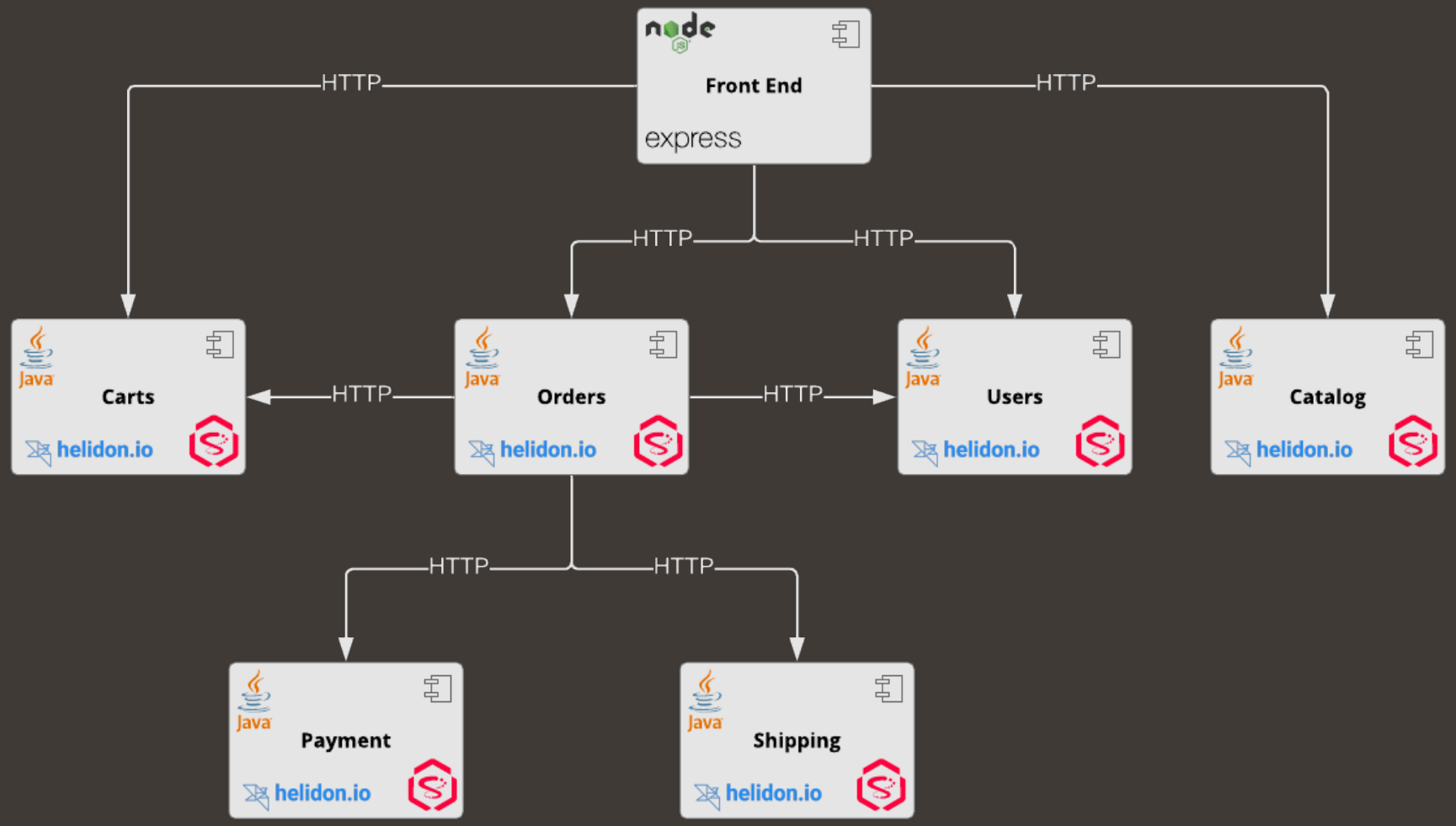
- How easy it is to scale each of these services?

  You need to be able to scale Mongo, MySQL and RabbitMQ as well, not just the “stateless” service pods

- How easy it is to monitor each of these services?

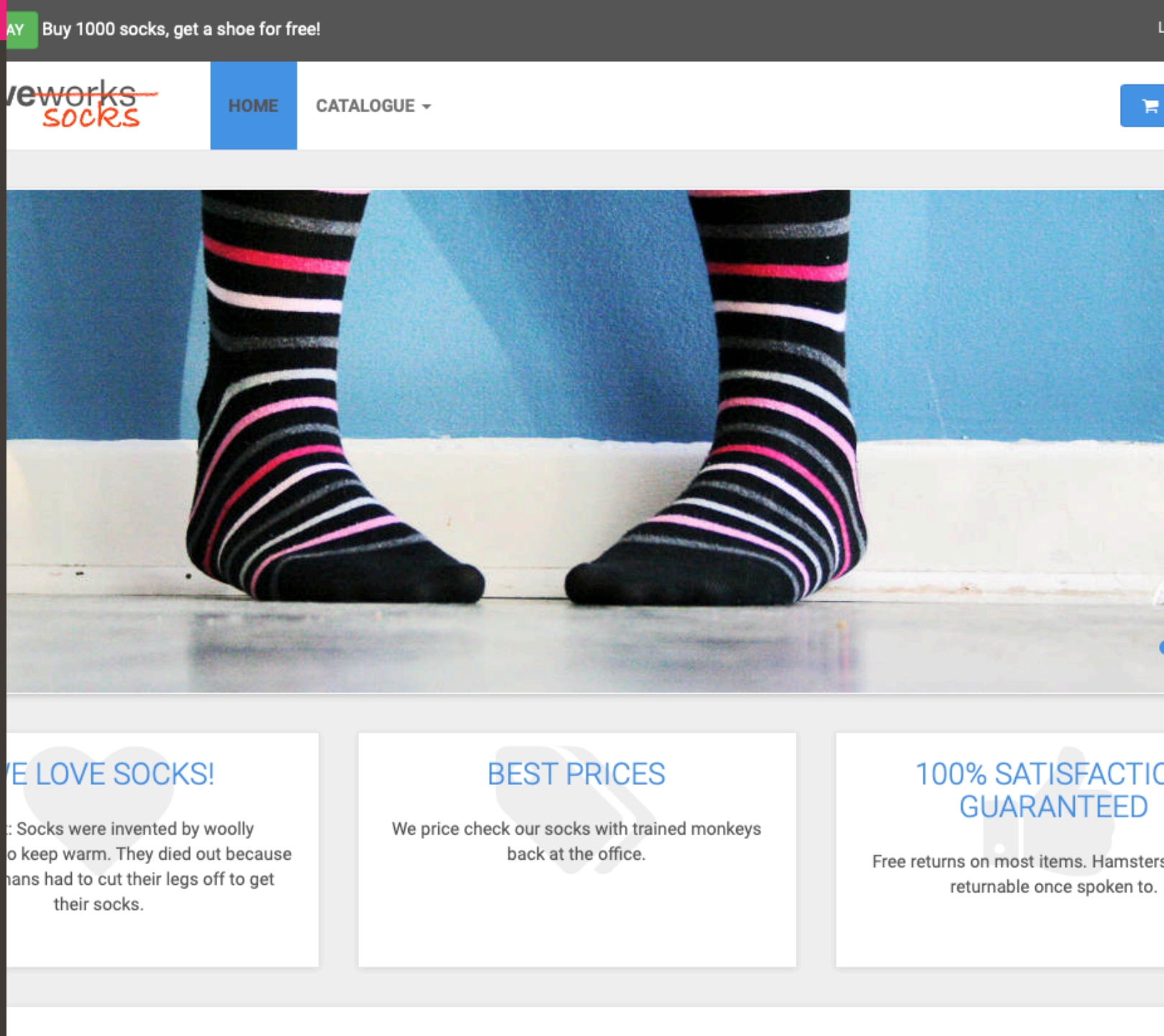
  You need to monitor both the services and their data stores, across different runtime platforms (Node, Java and Go)

Helidon + Coherence Sock Shop



Demo

Helidon Sock Shop



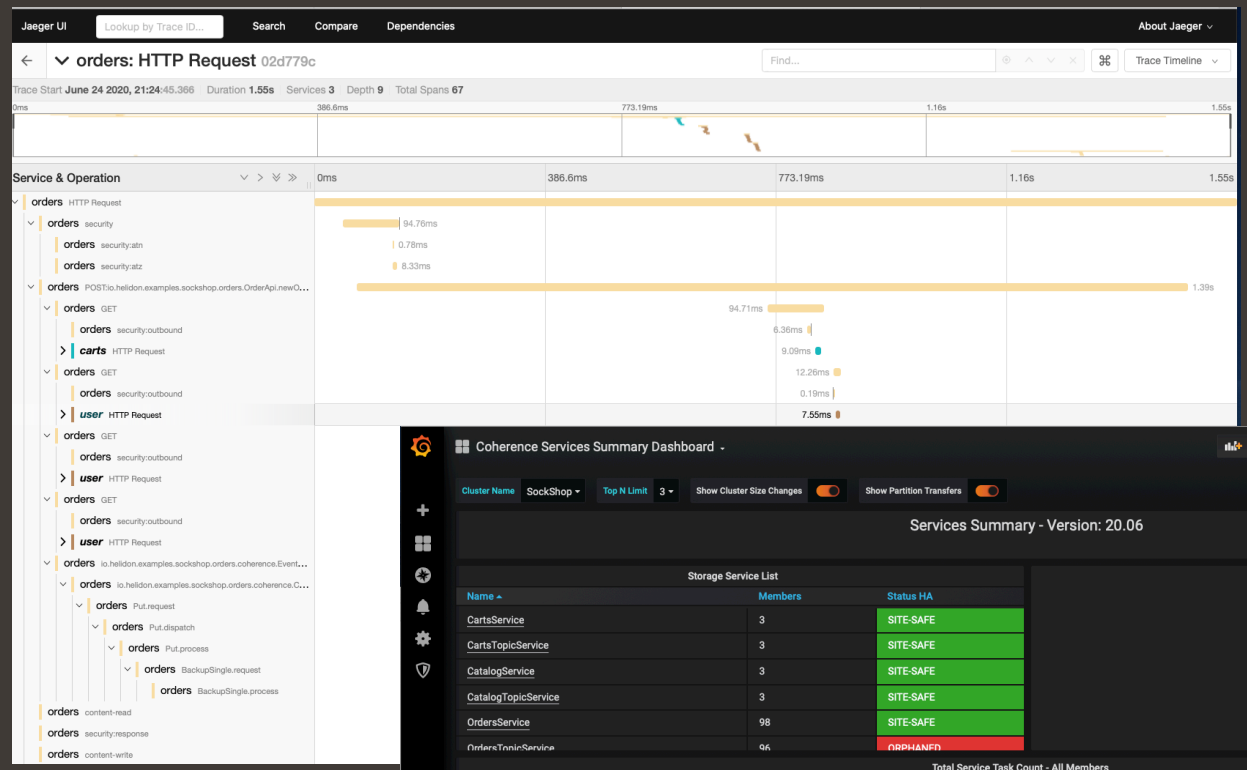
No More Dragons!

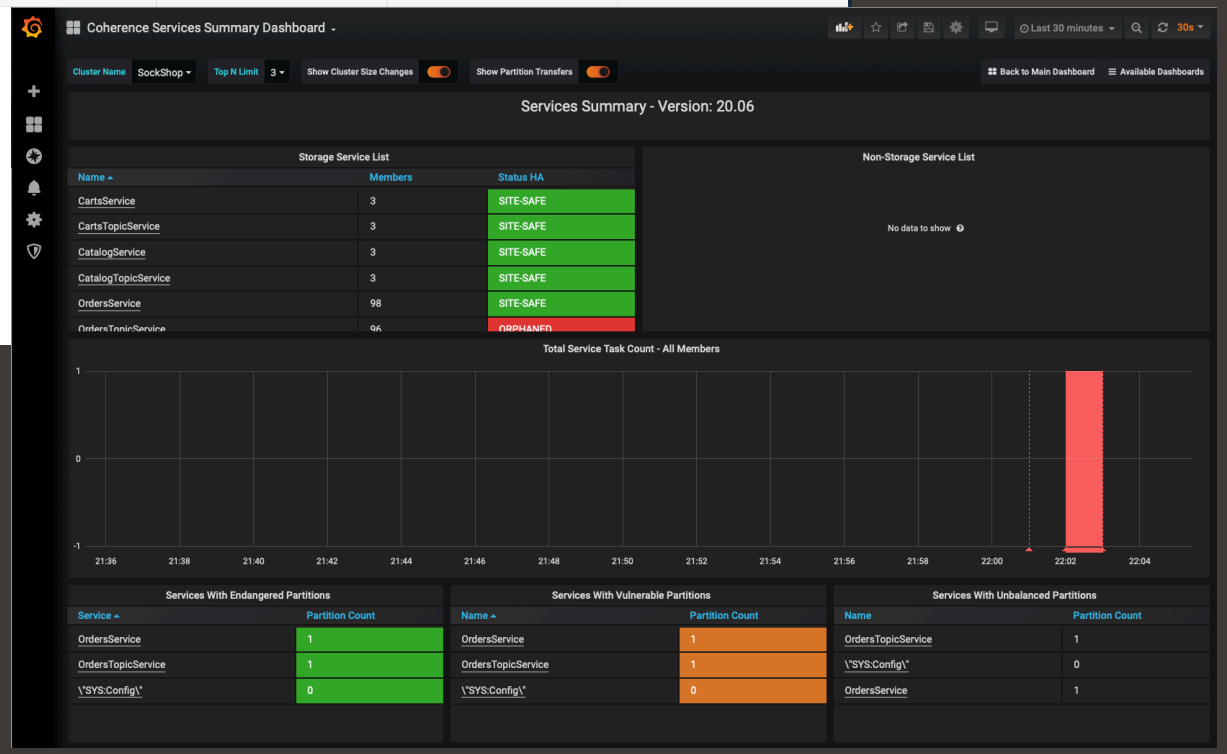
- How easy are these services to scale?
- How easy are these services to monitor?
- Is the implementation simpler?

Let's find out…

Demo

DevOps Features





Coherence CE and Helidon MP

- Bootstrap Coherence via CDI within Helidon MP apps
  - Ensures that the REST and gRPC services can access their data as soon as they are up
  - Helidon controls the main method, Coherence is just a library
- CDI support
  - Inject Coherence resources e.g. NamedMap into Helidon services
  - Inject CDI-managed objects into Coherence (event interceptors, cache stores, etc.)
  - Use CDI observers to handle Coherence server- and client-side events
- Metrics
  - Coherence metrics available via standard Helidon MP /metrics endpoint
- Configuration
  - Configure Coherence using MP Config
  - Use Coherence as a mutable, observable MP Config Source
- Tracing
  - Coherence tracing spans automatically included into Helidon traces

Open Tracing 1.3	Open API 1.1	Rest Client 1.4	Config 1.4
Fault Tolerance 2.1	Metrics 2.3	JWT Propagation 1.1	Health 2.2
CDI 2.0	JSON-P 1.1	JAX-RS 2.1	JSON-B 1.0

Implementation: No Impedance Mismatch

- Coherence stores Java objects
- Supports rich domain models and DDD

```
@Inject
protected final NamedMap<String, Cart> carts;

@Override
public Item addItem(String cartId, Item item) {
    return carts.invoke(cartId, entry -> {
        Cart cart = entry.getValue();
        Item newItem = cart.add(item);
        entry.setValue(cart);
        return newItem;
    });
}
```

Implementation: Rich Event Model

- Supports reactive and event-driven processing

```
void onOrderCreated(@ObservesAsync @MapName("orders")
                    @Inserted @Updated EntryEvent<String, Order> event) {
    Order order = event.getValue();

    switch (order.getStatus()) {
    case CREATED:
        processPayment(order);
        break;
    case PAID:
        shipOrder(order);
        break;
    default:        // do nothing, order is in a terminal state already
}
```

Implementation: JAX-RS for gRPC (server)

```
@ApplicationScoped
@Grpc
@GrpcMarshaller("jsonb")
public class PaymentGrpc {
    @Unary
    public Collection<Authorization> getAuthorizations(String orderId) {
        // implementation omitted
    }

    @Unary
    @Metered
    public Authorization authorize(PaymentRequest paymentRequest) {
        // implementation omitted
    }
}
```

Implementation: JAX-RS for gRPC (client)

```
@Grpc(name = "PaymentGrpc")
@GrpcChannel(name = "payment")
@GrpcMarshaller("jsonb")
public interface PaymentClient {
   @Unary
   Payment authorize(PaymentRequest request);
}


// usage
@Inject
@GrpcProxy
protected PaymentClient paymentService;
```

Better Together

Verrazzano

Dave Cabelus

Oracle
June 25, 2020

Coming Soon....

verrazzano Enterprise Container Platform

Traditional Applications

Java Microservices

Polyglot microservices

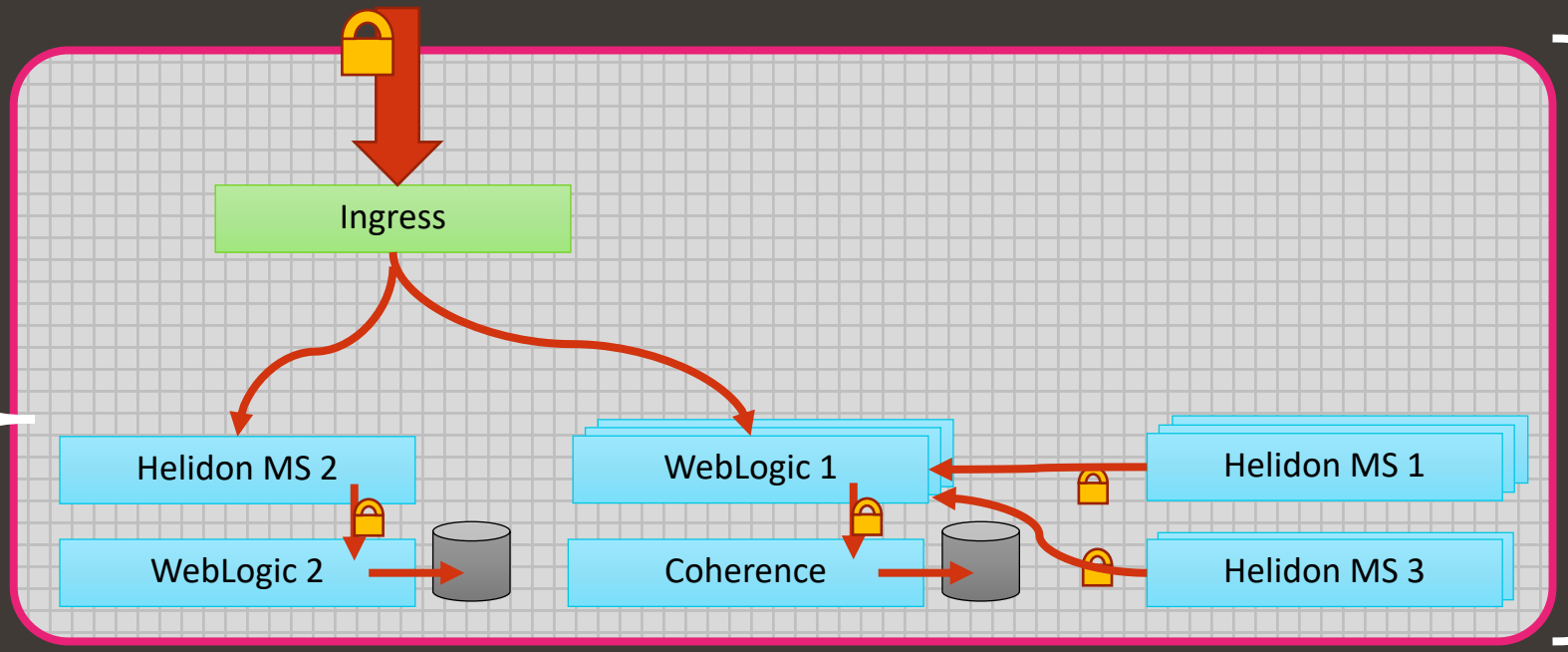
Kubernetes | Kubernetes | Kubernetes

Public Cloud | Private Cloud | Multi-Cloud

Verrazzano Application Model and Binding

App Model
- WebLogic 1
- WebLogic 2
- Coherence
- Helidon MS 1
- Helidon MS 2
- Helidon MS 3
- Connections
- Etc.

App Binding
- Placements
- Connection details
- Databases
- Ingress details
- Secrets
- Etc.



Kubernetes | Kubernetes | Kubernetes

Public Cloud | Private Cloud | Multi-Cloud

Monitoring Stack
- Grafana
- Prometheus
- Kibana
- Elasticsearch
- ...

Management Stack
- Operators
- Keycloak
- Istio
- Rancher
- ...

verrazzano

Resources

- github.com/oracle/coherence
- coherence.community/
- hub.docker.com/r/oraclecoherence/coherence-ce
- github.com/coherence-community/coherence-demo
- oraclecoherence.slack.com
- medium.com/oracle-coherence
- twitter.com/oraclecoherence

- github.com/oracle/helidon
- helidon.io
- helidon.slack.com
- medium.com/helidon
- http://youtube.helidon.io/
- stackoverflow.com/tags/helidon
- github.com/helidon-sockshop
- twitter.com/helidon_project

Safe Harbor

The preceding 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.

Statements in this presentation relating to Oracle's future plans, expectations, beliefs, intentions and prospects are "forward-looking statements" and are subject to material risks and uncertainties. A detailed discussion of these factors and other risks that affect our business is contained in Oracle's Securities and Exchange Commission (SEC) filings, including our most recent reports on Form 10-K and Form 10-Q under the heading "Risk Factors." These filings are available on the SEC's website or on Oracle's website at http://www.oracle.com/investor. All information in this presentation is current as of September 2019 and Oracle undertakes no duty to update any statement in light of new information or future events.

Thank You