JPoint 2021

Building Scalable Microservices for Java

With Helidon and Coherence CE

Dmitry Aleksandrov
Software Developer, Oracle

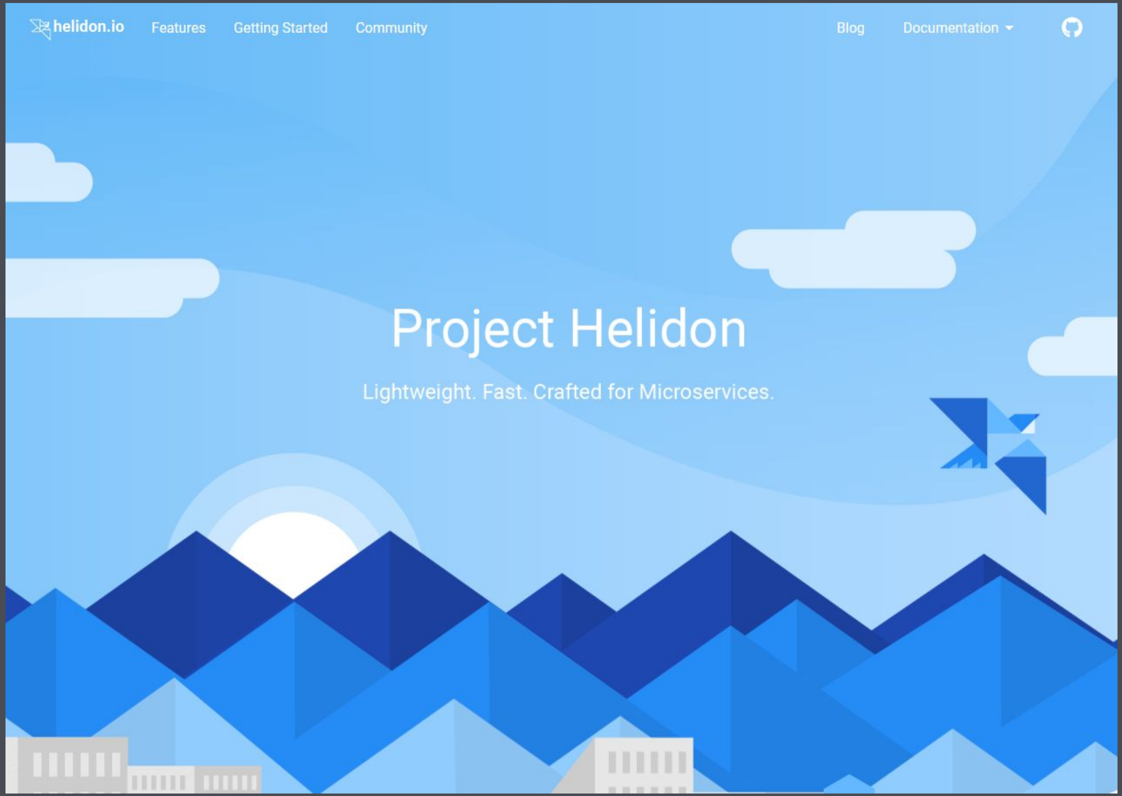
Aleks Seović
Architect, Oracle

#Coherence #Helidon #JPoint

Safe Harbor Statement

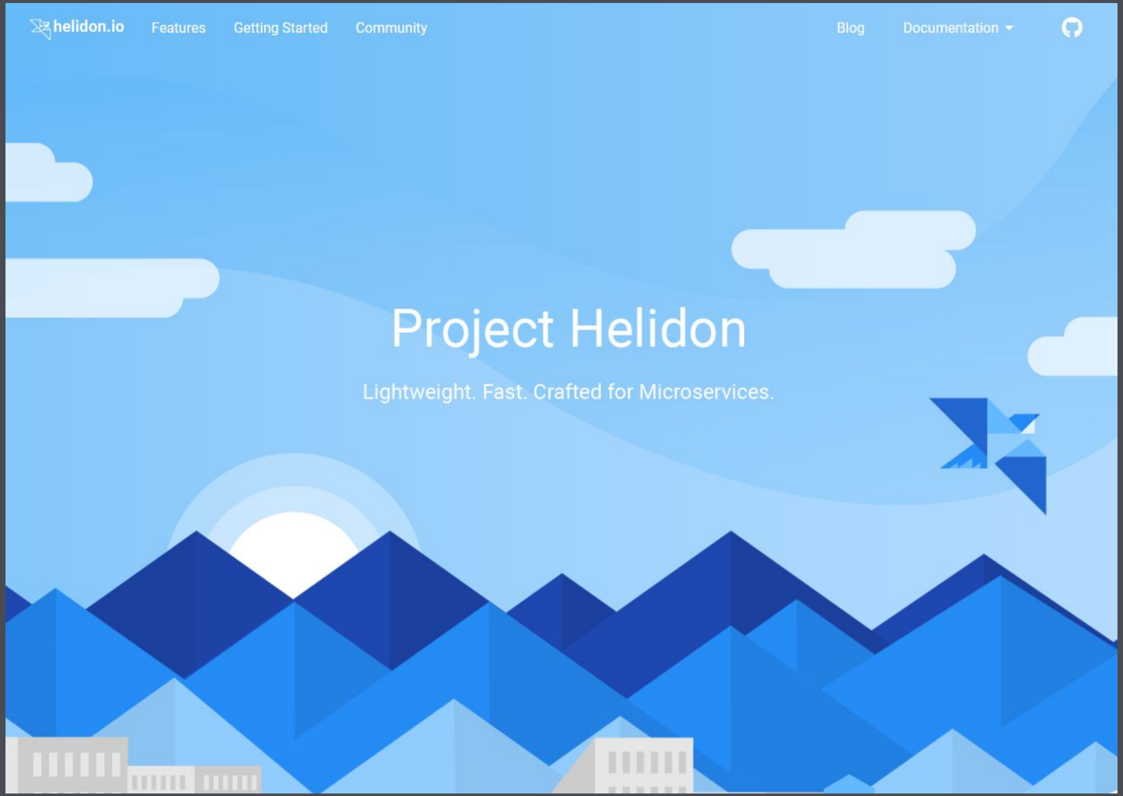
Project Helidon

- A set Java libraries for developing microservices
  - https://helidon.io



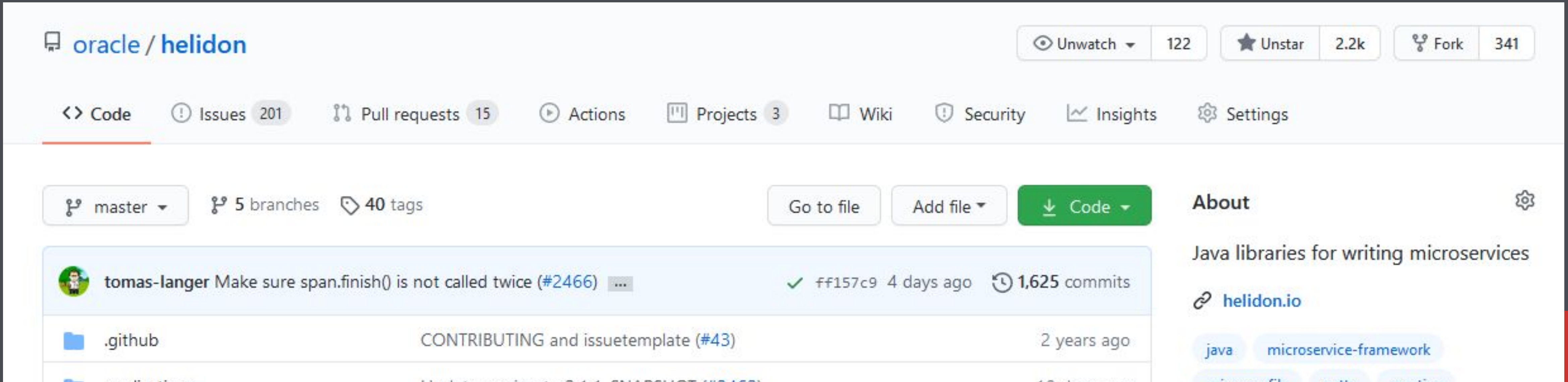
Project Helidon

- A set Java libraries for developing microservices
  - https://helidon.io
- Helidon (Χελιδόνι) = Swallow



Project Helidon

- A set Java libraries for developing microservices
  - https://helidon.io
- Helidon (Χελιδόνι) = Swallow
- Open source under Apache 2.0
  - https://github.com/oracle/helidon



tomas-langer Make sure span.finish() is not called twice (#2466) ... ✓ ff157c9 4 days ago 1,625 commits

.github CONTRIBUTING and issuetemplate (#43) 2 years ago

Java libraries for writing microservices

helidon.io

Project Helidon

- A set Java libraries for developing microservices
  - https://helidon.io
- Helidon (Χελιδόνι) = Swallow
- Open source under Apache 2.0
  - https://github.com/oracle/helidon
- Built on top of Netty

Project Helidon

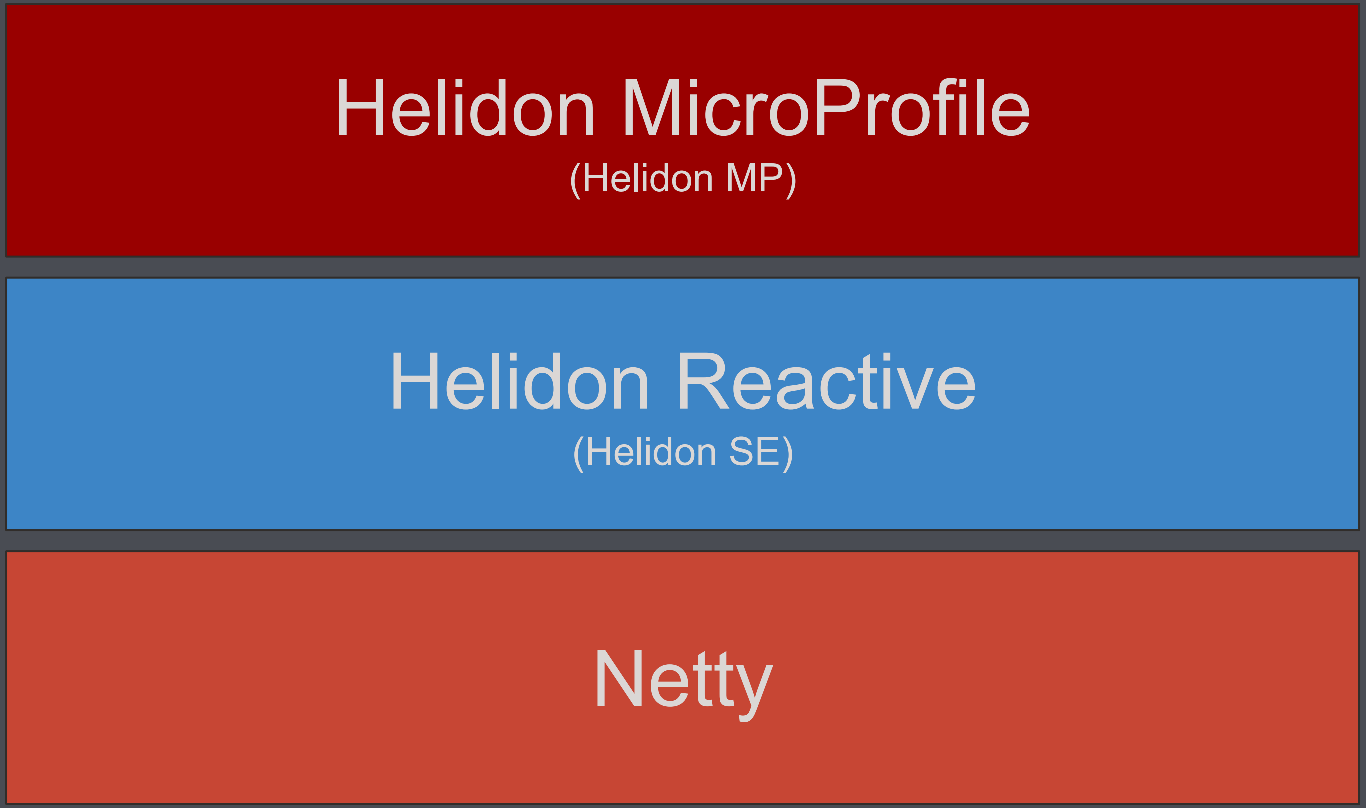
- A set Java libraries for developing microservices
  - https://helidon.io
- Helidon (Χελιδόνι) = Swallow
- Open source under Apache 2.0
  - https://github.com/oracle/helidon
- Built on top of Netty
- Cloud-native ready
  - REST, Health, Metrics, Tracing, …

Project Helidon

- A set Java libraries for developing microservices
  - https://helidon.io
- Helidon (Χελιδόνι) = Swallow
- Open source under Apache 2.0
  - https://github.com/oracle/helidon
- Built on top of Netty
- Cloud-native ready
  - REST, Health, Metrics, Tracing, …
- GraalVM native-image support

Java 11+

Helidon High-Level Architecture



helidon MP

MICROPROFILE™
OPTIMIZING ENTERPRISE JAVA

- Open source community specification for Enterprise Java microservices
- Hosted at Eclipse Foundation
- Participants:
    - Oracle, IBM, Red Hat, Payara, Tomitribe, Microsoft and others
- High release cadence:
    - 4 releases per year
- https://microprofile.io

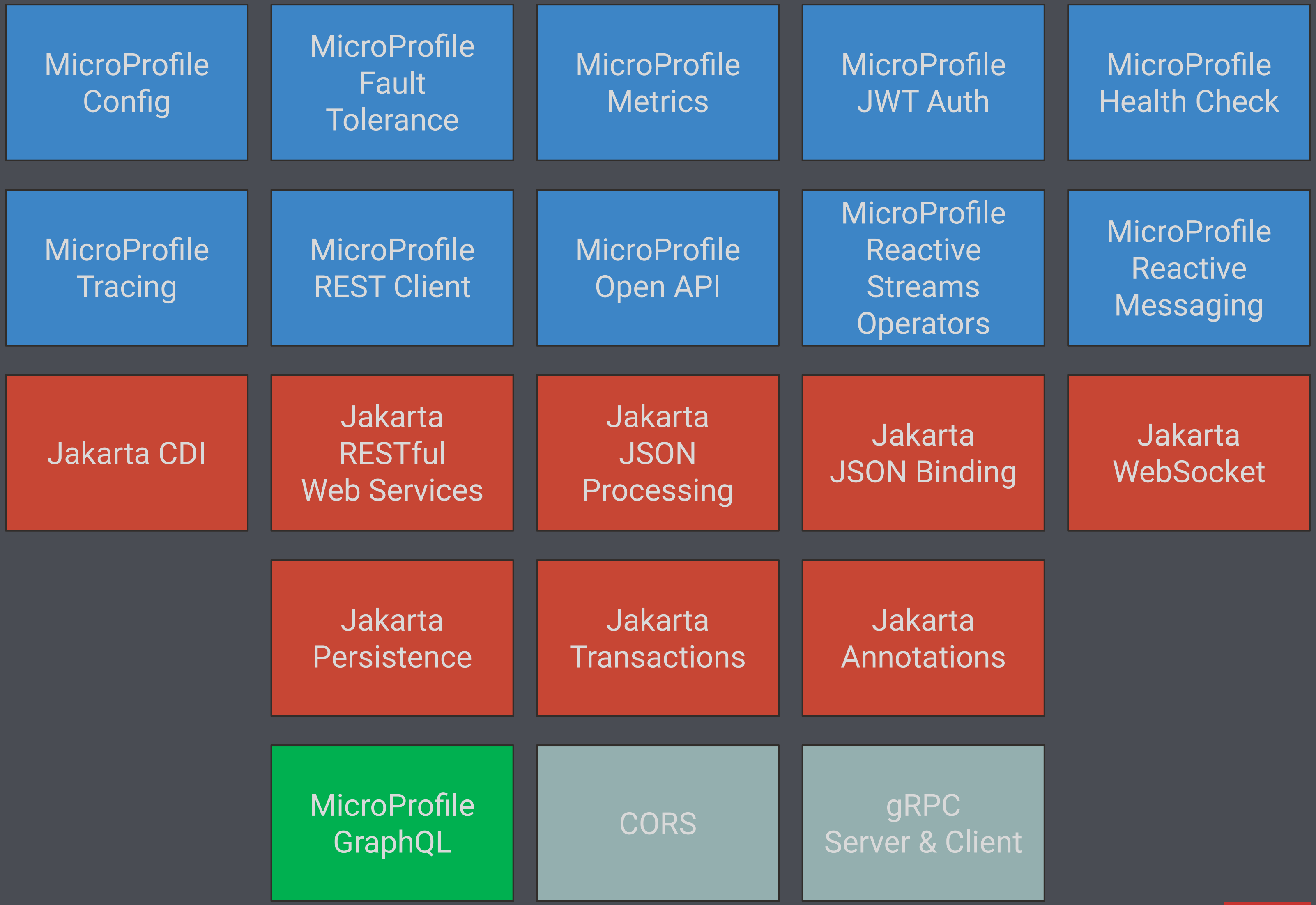
MICROPROFILE™
OPTIMIZING ENTERPRISE JAVA

Open Tracing 2.0	Open API 2.0	Rest Client 2.0	Config 2.0
Fault Tolerance 3.0	Metrics 3.0	JWT Propagation 1.2	Health 3.0
CDI 2.0	JSON-P 1.1	JAX-RS 2.1	JSON-B 1.0

MicroProfile 4.0

Helidon MP Components

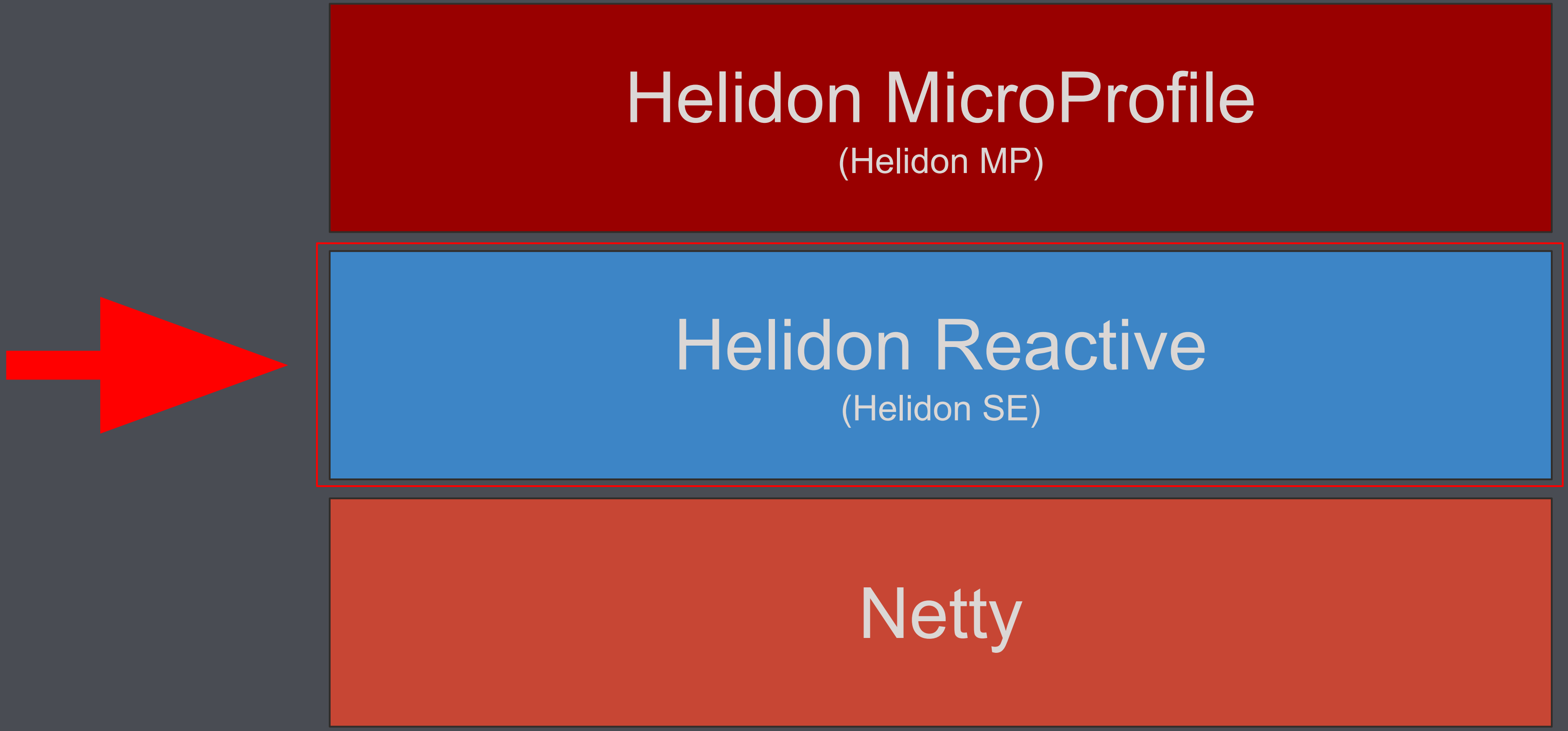
- MicroProfile
- MP Additional
- Jakarta EE
- Helidon Specific



helidon SE

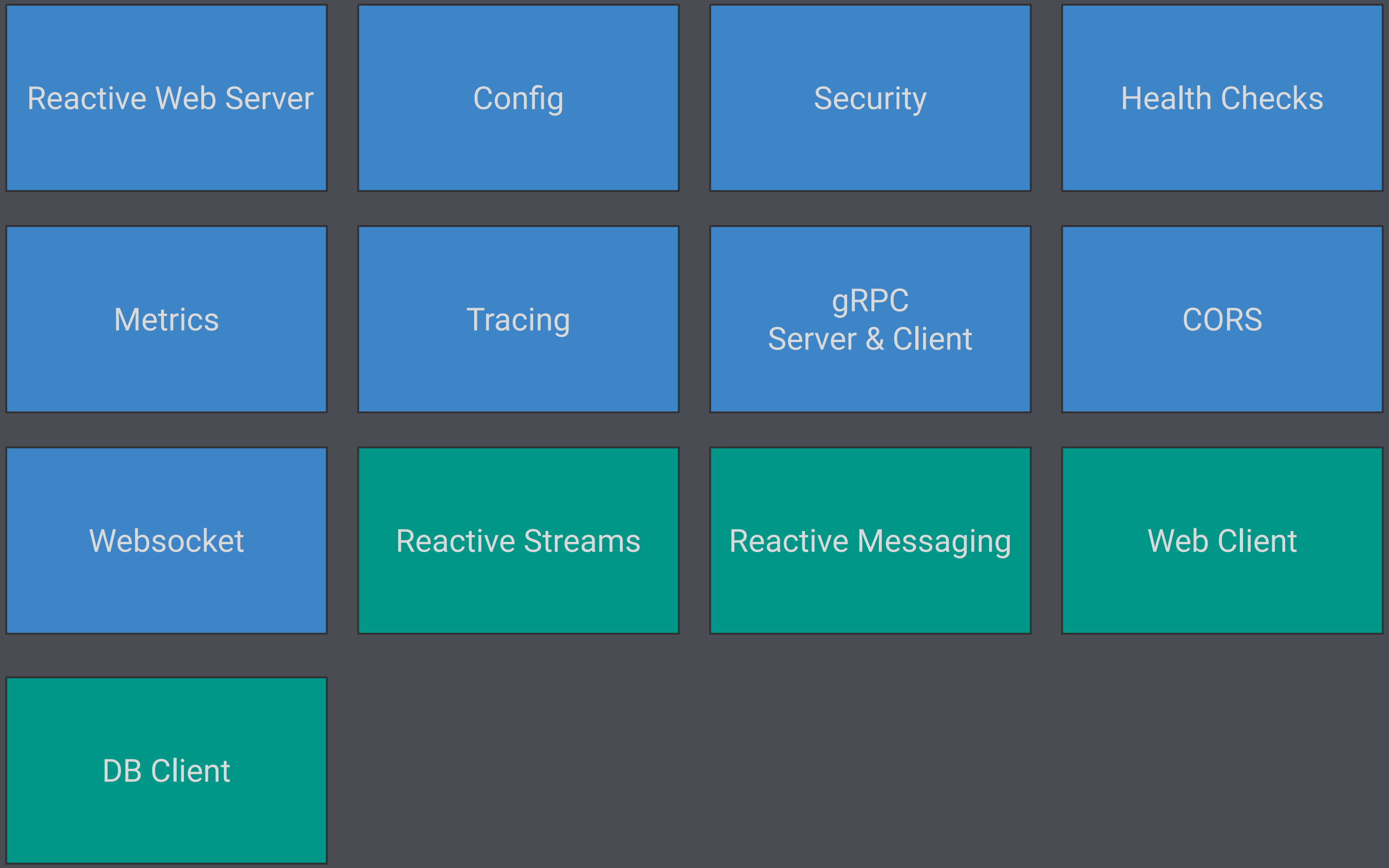
.. or Helidon Reactive APIs

Helidon High-Level Architecture



Helidon SE Components

- Core Components
- Experimental Components



Two flavors

helidon SE

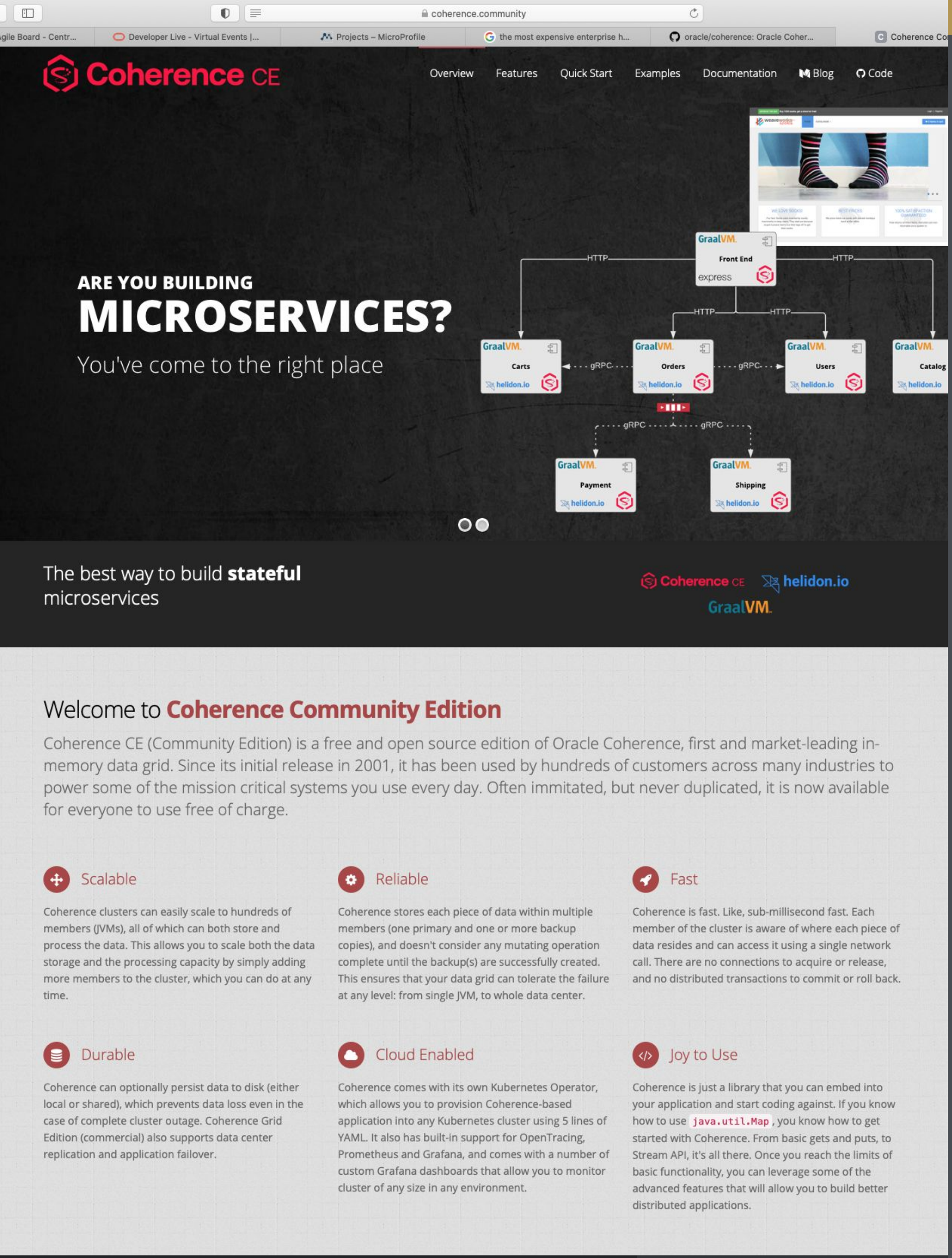
```
Routing routing = Routing.builder()
     .get("/hello", (req, res) ->
     res.send("Hello World"))
    .build();

WebServer.create(routing)
     .start();
```

helidon MP

```
@Path("hello")
public class HelloWorld {
     @GET
     public String hello() {
          return "Hello World";
     }
}
```

That’s all you need to know about Helidon for now!



ARE YOU BUILDING
MICROSERVICES?
You've come to the right place

The best way to build **stateful** microservices

Welcome to **Coherence Community Edition**

Coherence CE (Community Edition) is a free and open source edition of Oracle Coherence, first and market-leading in-memory data grid. Since its initial release in 2001, it has been used by hundreds of customers across many industries to power some of the mission critical systems you use every day. Often immitated, but never duplicated, it is now available for everyone to use free of charge.

Scalable

Coherence clusters can easily scale to hundreds of members (JVMs), all of which can both store and process the data. This allows you to scale both the data storage and the processing capacity by simply adding more members to the cluster, which you can do at any time.

Reliable

Coherence stores each piece of data within multiple members (one primary and one or more backup copies), and doesn't consider any mutating operation complete until the backup(s) are successfully created. This ensures that your data grid can tolerate the failure at any level: from single JVM, to whole data center.

Fast

Coherence is fast. Like, sub-millisecond fast. Each member of the cluster is aware of where each piece of data resides and can access it using a single network call. There are no connections to acquire or release, and no distributed transactions to commit or roll back.

Durable

Coherence can optionally persist data to disk (either local or shared), which prevents data loss even in the case of complete cluster outage. Coherence Grid Edition (commercial) also supports data center replication and application failover.

Cloud Enabled

Coherence comes with its own Kubernetes Operator, which allows you to provision Coherence-based application into any Kubernetes cluster using 5 lines of YAML. It also has built-in support for OpenTracing, Prometheus and Grafana, and comes with a number of custom Grafana dashboards that allow you to monitor cluster of any size in any environment.

Joy to Use

Coherence is just a library that you can embed into your application and start coding against. If you know how to use `java.util.Map`, you know how to get started with Coherence. From basic gets and puts, to Stream API, it's all there. Once you reach the limits of basic functionality, you can leverage some of the advanced features that will allow you to build better distributed applications.

Coherence CE

"The world's most expensive Enterprise HashMap" is now free!

https://coherence.community/

Oracle Coherence

- The world's first In-Memory Data Grid, which created the space
  - Well suited for modern cloud applications
  - A Java library, embeddable into any Java application
  - Easy to deploy and scale in Kubernetes via Coherence Operator
  - Easy to monitor with Prometheus, Grafana and Jaeger
  - Open sourced Community Edition since June 2020

Oracle Coherence: Not just a ”cache”

- Scalable, fast, persistent general-purpose KV data store
  - Stores data in memory for fast access, and optionally persists to disk for durability
  - Partitions data across many cluster members in order to scale both system capacity and throughput
  - Creates back ups as far as possible from the primary data copies for reliability, with completely automatic failover and rebalancing
  - Supports parallel queries, aggregations, and in-place processing

Oracle Coherence: Not just a ”cache”

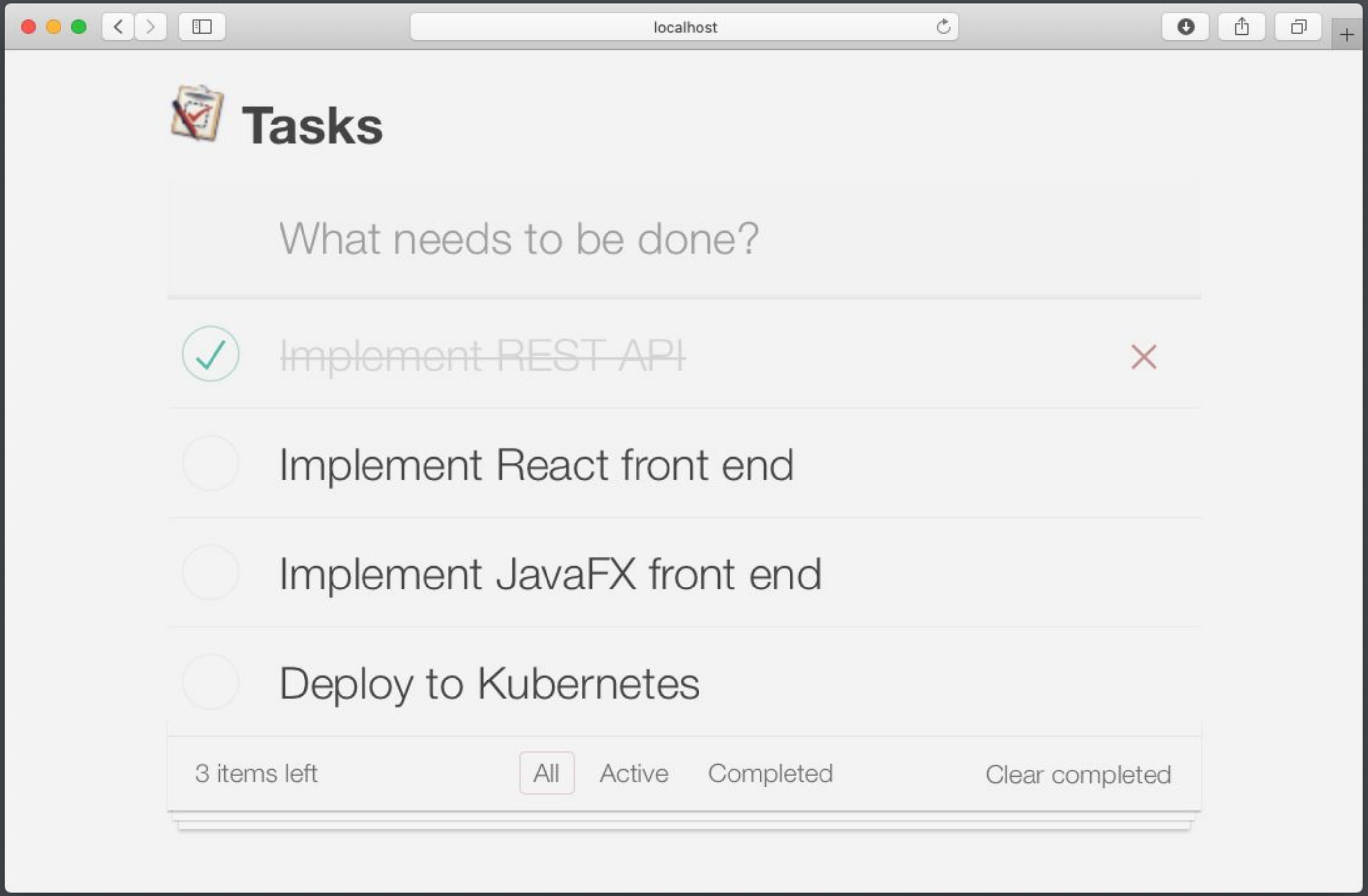
- An event streaming platform
  - Server-side and client-side event listeners, for data processing and integration with other systems
  - Built-in PubSub messaging

Oracle Coherence: Not just a ”cache”

- And yes, a caching solution as well…
  - Partitioned or replicated, time- or size-limited caches
  - Read-through, write-through, write-behind, refresh-ahead for integration with external data stores and/or services
  - Client-side caching with event-based cache invalidation
  - Live materialized views

Let’s get to some code!

Demo Application: A simple (but scalable 😉) To Do List App


localhost

Tasks

What needs to be done?

~~Implement REST API~~

Implement React front end

Implement JavaFX front end

Deploy to Kubernetes

3 items left | All | Active | Completed | Clear completed

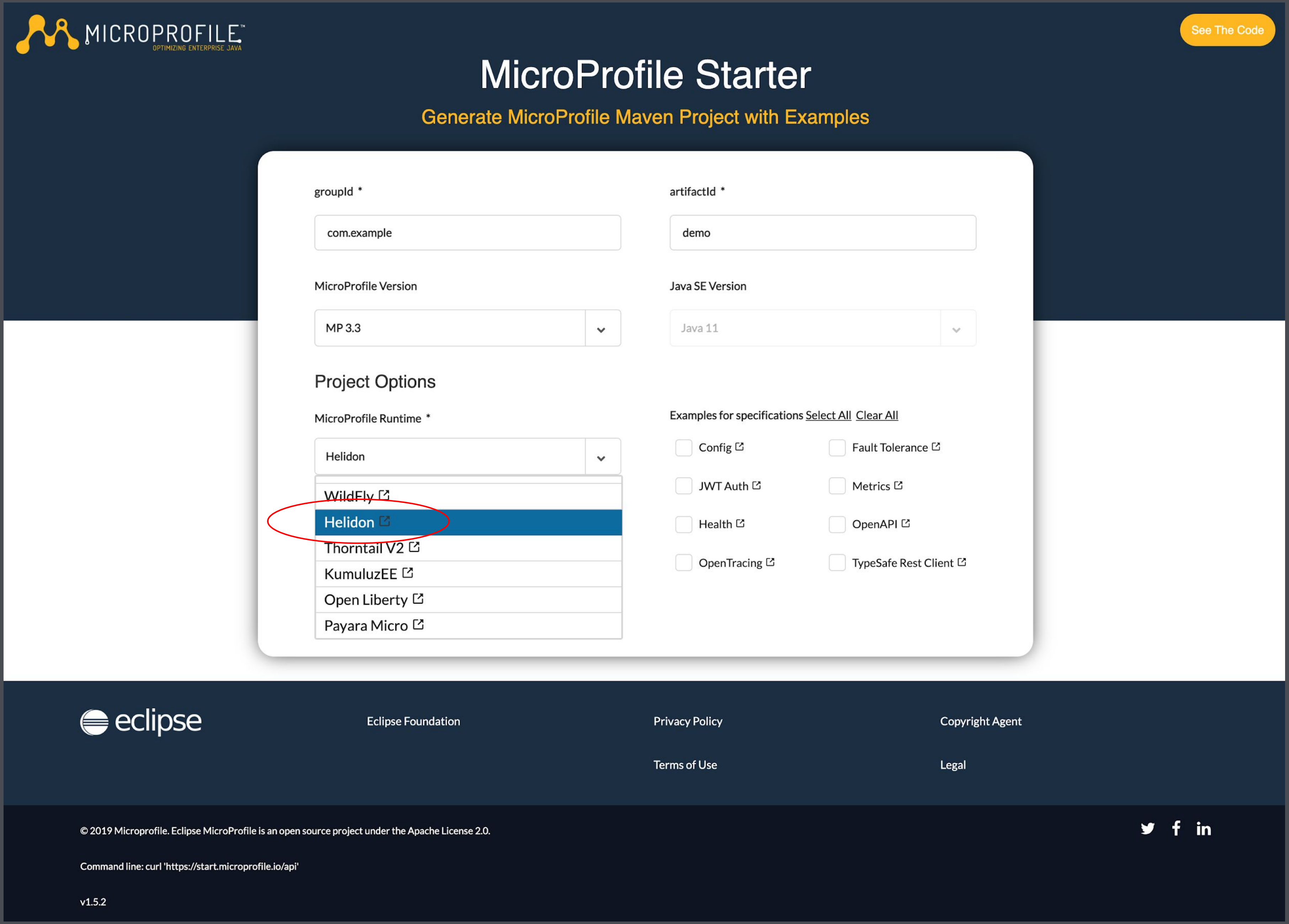
https://github.com/coherence-community/todo-list-example

Create Helidon Project: Maven Archetype

```
mvn -U archetype:generate -DinteractiveMode=false \
   -DarchetypeGroupId=io.helidon.archetypes \
   -DarchetypeArtifactId=helidon-quickstart-mp \
   -DarchetypeVersion=2.2.2 \
   -DgroupId=io.helidon.examples \
   -DartifactId=todo-list \
   -Dpackage=io.helidon.examples.tasks
```

Create Helidon Project: Plain Maven POM

```
<parent>
  <groupId>io.helidon.applications</groupId>
  <artifactId>helidon-mp</artifactId>
  <version>2.2.2</version>
</parent>
...
<dependency>
  <groupId>io.helidon.microprofile.bundles</groupId>
  <artifactId>helidon-microprofile</artifactId>
</dependency>
```



MicroProfile Starter

Generate MicroProfile Maven Project with Examples

groupId * com.example

artifactId * demo

MicroProfile Version: MP 3.3

Java SE Version: Java 11

Project Options

MicroProfile Runtime *: Helidon

- WildFly
- [✓ Helidon]
- Thorntail V2
- KumuluzEE
- Open Liberty
- Payara Micro

Examples for specifications Select All Clear All

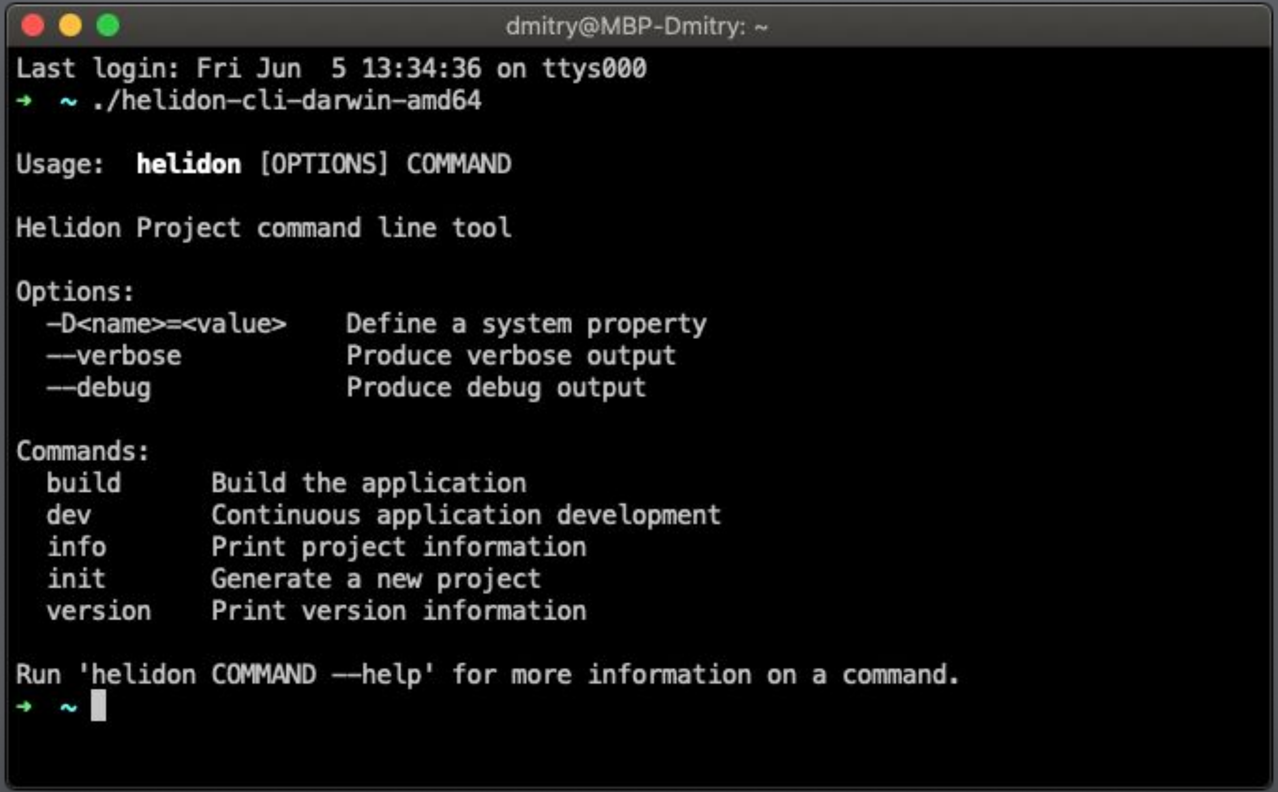
☐ Config	☐ Fault Tolerance
☐ JWT Auth	☐ Metrics
☐ Health	☐ OpenAPI
☐ OpenTracing	☐ TypeSafe Rest Client

© 2019 Microprofile. Eclipse MicroProfile is an open source project under the Apache License 2.0.

Command line: curl 'https://start.microprofile.io/api'

v1.5.2

We are going to do it the modern way!



```
Last login: Fri Jun  5 13:34:36 on ttys000
➜  ~ ./helidon-cli-darwin-amd64

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
  info       Print project information
  init       Generate a new project
  version    Print version information

Run 'helidon COMMAND --help' for more information on a command.
➜  ~ 
```

https://helidon.io/docs/v2/#/about/05_cli

DEMO

Learn More: Helidon

https://helidon.io/

https://medium.com/helidon

https://github.com/oracle/helidon

https://www.youtube.com/channel/UChg00-uTTrCMmPsuzUNaZsA

https://helidon.slack.com

Learn More: Oracle Coherence

https://coherence.community/

https://medium.com/oracle-coherence

https://github.com/oracle/coherence

https://www.youtube.com/user/OracleCoherence

https://oraclecoherence.slack.com

Спасибо за внимание!

Есть вопросы?