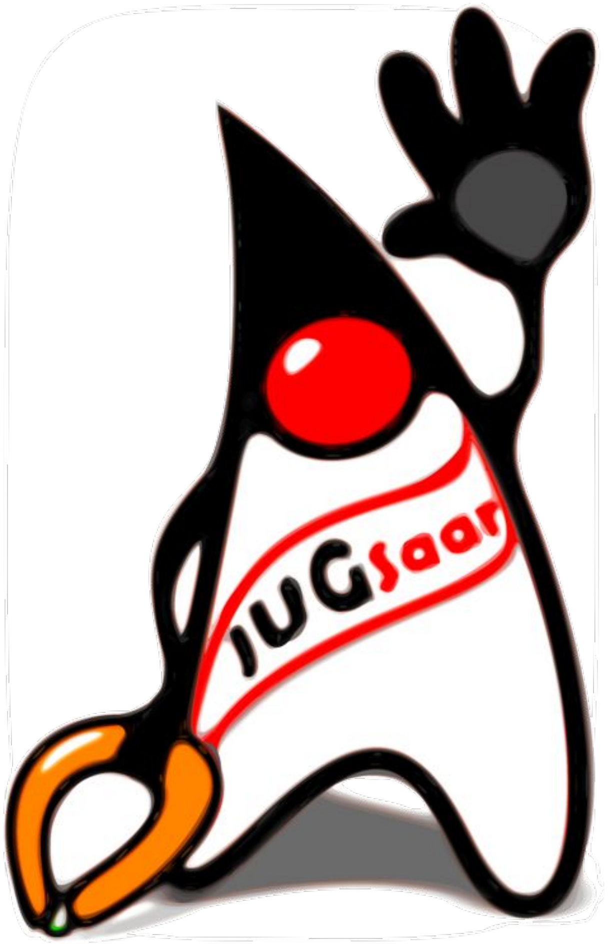
Java User Group Saarland



Quarkus Jumpstart

Übersicht und Best Practices

Thomas Darimont

55. Meeting

09. März 2021

Sponsored by

codecentric

>eurodata

INFOSERVE
>eurodata-Gruppe

The Battlestation

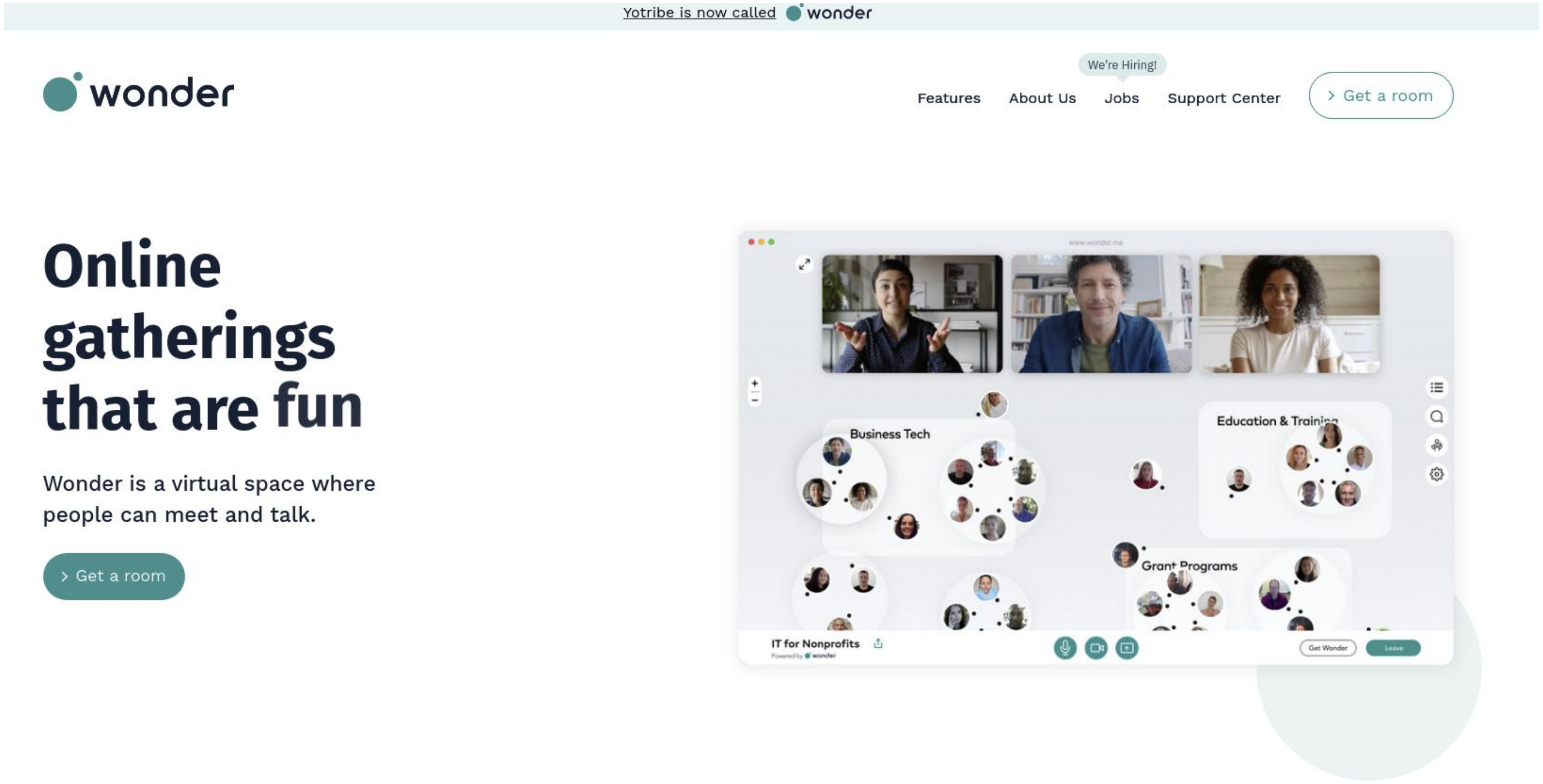
```
tom@neumann ~/dev/repos/gh/thomasdarimont/quarkus-dev/quarkus-workshops/quarkus-workshop-super-heroes/super-heroes/infrastructure (master=)
$ ll
total 28K
drwxr-xr-x.  5 tom tom 4,0K  8. Mär 20:46 ./
drwxr-xr-x. 11 tom tom 4,0K  8. Mär 20:47 ../
drwxr-xr-x.  2 tom tom 4,0K  8. Mär 20:34 db-init/
-rw-r--r--.  1 tom tom 1,5K  8. Mär 20:34 docker-compose-linux.yaml
-rw-r--r--.  1 tom tom 1,8K  8. Mär 21:02 docker-compose.yaml
drwxr-xr-x.  2 tom tom 4,0K  8. Mär 20:46 .history/
drwxr-xr-x.  2 tom tom 4,0K  8. Mär 20:34 monitoring/

tom@neumann ~/dev/repos/gh/thomasdarimont/quarkus-dev/quarkus-workshops/quarkus-workshop-super-heroes/super-heroes/infrastructure (master=)
$

$ ll
total 44K
drwxr-xr-x. 11 tom tom 4,0K  8. Mär 20:47 ./
drwxr-xr-x.  8 tom tom 4,0K  9. Mär 2021  ../
drwxr-xr-x.  4 tom tom 4,0K  9. Mär 2021  event-statistics/
drwxr-xr-x.  6 tom tom 4,0K  8. Mär 20:48 .history/
drwxr-xr-x.  5 tom tom 4,0K  8. Mär 20:46 infrastructure/
drwxr-xr-x.  2 tom tom 4,0K  8. Mär 20:34 kubernetes/
drwxr-xr-x.  5 tom tom 4,0K  9. Mär 2021  load-super-heroes/
drwxr-xr-x.  5 tom tom 4,0K  9. Mär 2021  rest-fight/
drwxr-xr-x.  5 tom tom 4,0K  9. Mär 2021  rest-hero/
drwxr-xr-x.  5 tom tom 4,0K  9. Mär 2021  rest-villain/
drwxr-xr-x.  5 tom tom 4,0K  9. Mär 2021  ui-super-heroes/

tom@neumann ~/dev/repos/gh/thomasdarimont/quarkus-dev/quarkus-workshops/quarkus-workshop-super-heroes/super-heroes (master=)
$
```

Thomas Darimont

- Fellow codecentric
- Pivotal Spring Team Alumni
- Open Source Enthusiast
- Java User Group Saarland
- Keycloak Contributor for over 5 years

@thomasdarimont
@jugsaar

Wonder.me



Yotribe is now called wonder

wonder

We're Hiring!

Features About Us Jobs Support Center > Get a room

Online gatherings that are fun

Wonder is a virtual space where people can meet and talk.

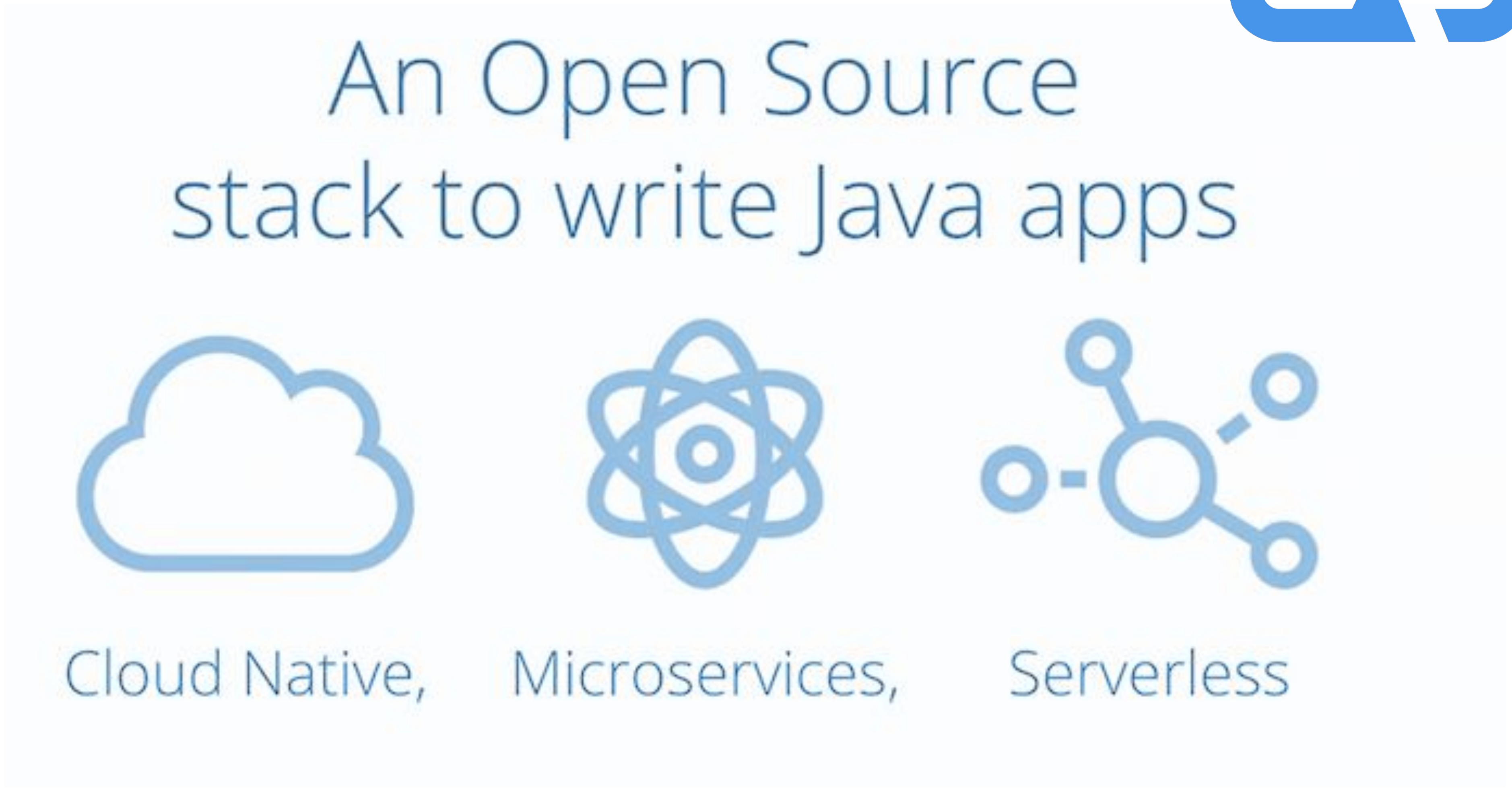
> Get a room

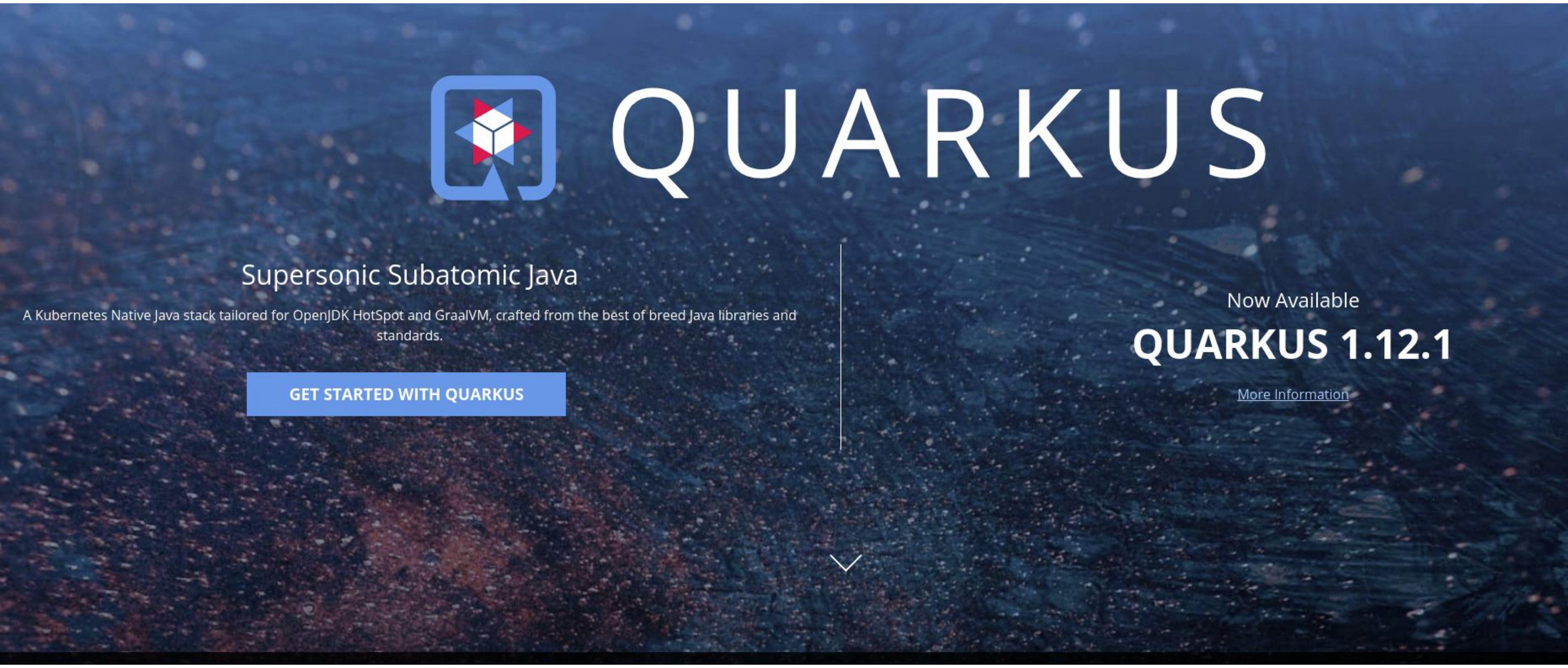
Nächste Veranstaltungen

- ✓ 09 Mar **Quarkus Jumpstart** **Thomas Darimont**
- 06 Apr **Keycloak Extension Development** **Thomas Darimont**
- 20 Mai **Blockchain 101 for Java Developers** **Kevin Witteck**
- XX Jun **Services Meshes for Java Developers** **Thomas Darimont**
- XX Juli **Awesome Talk TBA**
- XX Aug **Awesome Talk TBA**
- XX Sep **Awesome Talk TBA**
- XX Okt **Awesome Talk TBA**
- XX Nov **Awesome Talk TBA**
- XX Dez **Awesome Talk TBA**

http://www.meetup.com/de-DE/Java-User-Group-Saarland-jugsaar/#upcoming

Meet Quarkus

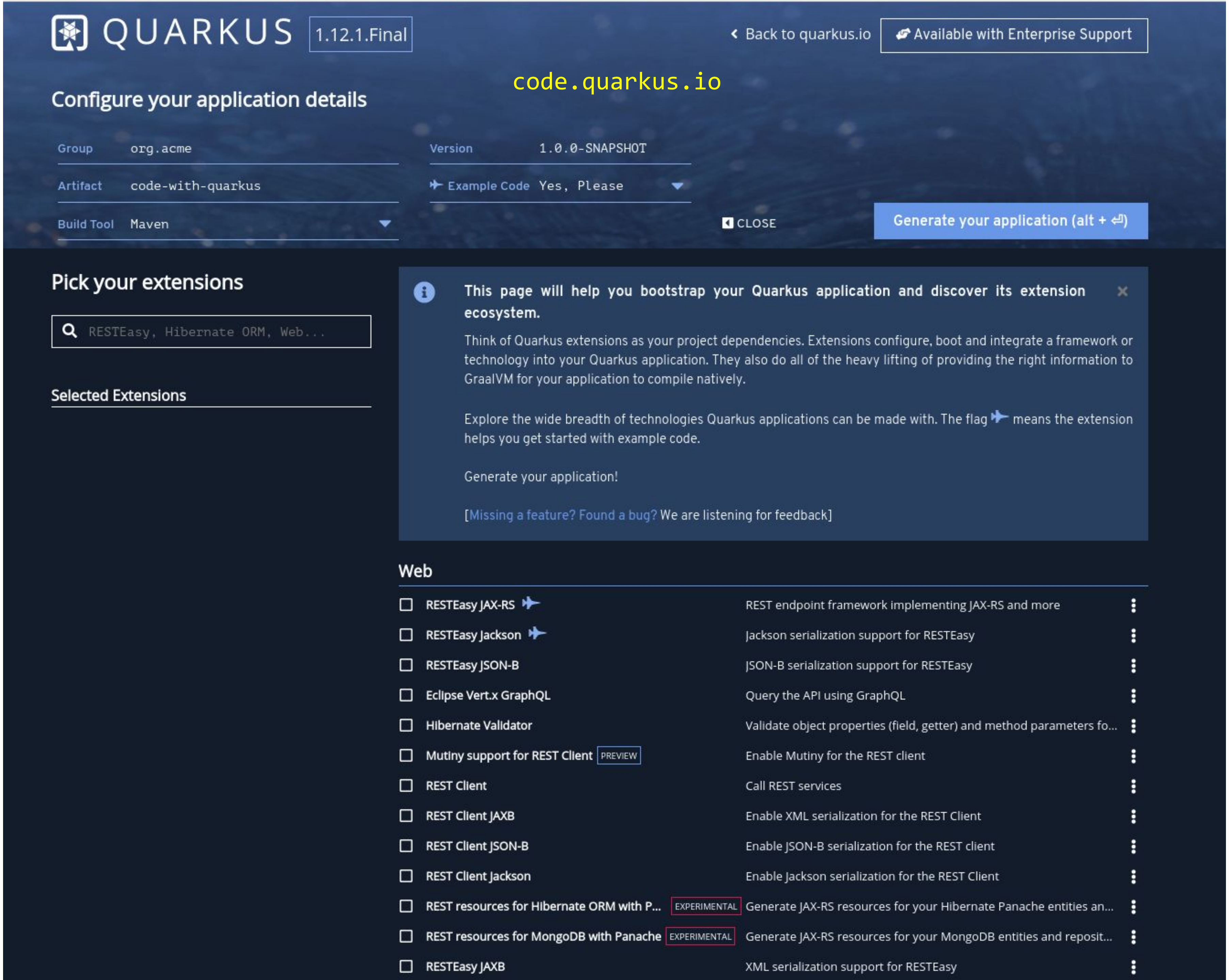




QUARKUS

Supersonic Subatomic Java

A Kubernetes Native Java stack tailored for OpenJDK HotSpot and GraalVM, crafted from the best of breed Java libraries and standards.


QUARKUS 1.12.1.Final
Back to quarkus.io
Available with Enterprise Support

code.quarkus.io

Configure your application details

Group	org.acme	Version	1.0.0-SNAPSHOT
Artifact	code-with-quarkus	Example Code	Yes, Please
Build Tool	Maven		

CLOSE
Generate your application (alt + ⏎)

Pick your extensions

RESTEasy, Hibernate ORM, Web...

Selected Extensions

This page will help you bootstrap your Quarkus application and discover its extension ecosystem.

Think of Quarkus extensions as your project dependencies. Extensions configure, boot and integrate a framework or technology into your Quarkus application. They also do all of the heavy lifting of providing the right information to GraalVM for your application to compile natively.

Explore the wide breadth of technologies Quarkus applications can be made with. The flag means the extension helps you get started with example code.

Generate your application!

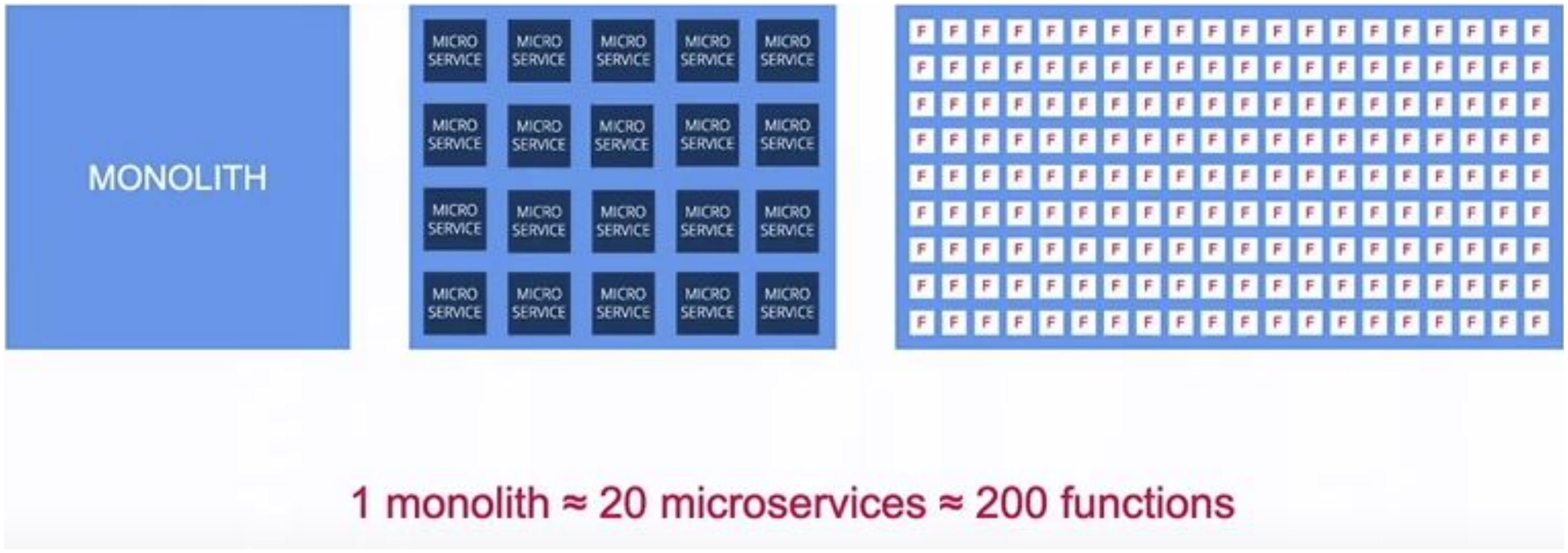
[Missing a feature? Found a bug? We are listening for feedback]

Web

☐ RESTEasy JAX-RS	REST endpoint framework implementing JAX-RS and more
☐ RESTEasy Jackson	Jackson serialization support for RESTEasy
☐ RESTEasy JSON-B	JSON-B serialization support for RESTEasy
☐ Eclipse Vert.x GraphQL	Query the API using GraphQL
☐ Hibernate Validator	Validate object properties (field, getter) and method parameters fo...
☐ Mutiny support for REST Client PREVIEW	Enable Mutiny for the REST client
☐ REST Client	Call REST services
☐ REST Client JAXB	Enable XML serialization for the REST Client
☐ REST Client JSON-B	Enable JSON-B serialization for the REST client
☐ REST Client Jackson	Enable Jackson serialization for the REST Client
☐ REST resources for Hibernate ORM with P... EXPERIMENTAL	Generate JAX-RS resources for your Hibernate Panache entities an...
☐ REST resources for MongoDB with Panache EXPERIMENTAL	Generate JAX-RS resources for your MongoDB entities and reposit...
☐ RESTEasy JAXB	XML serialization support for RESTEasy

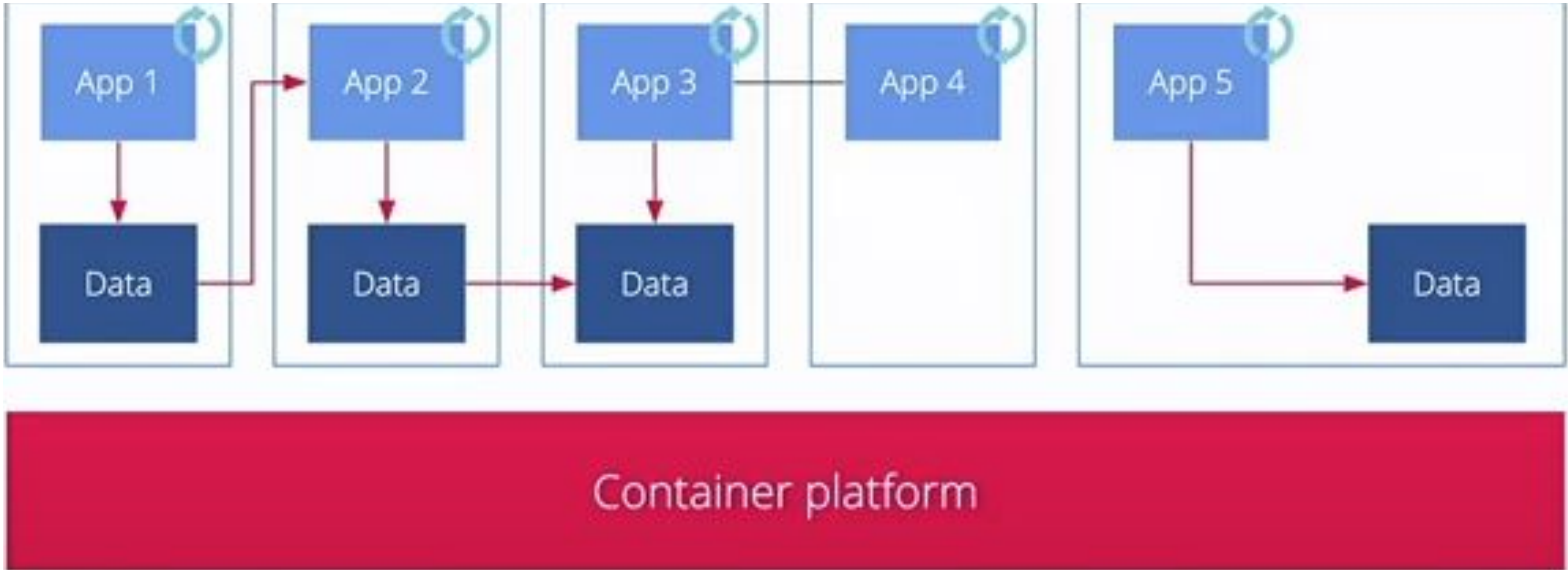
Quarkus Demo 1

From Monolith towards Serverless



1 monolith ≈ 20 microservices ≈ 200 functions

Modern Microservice Architectures

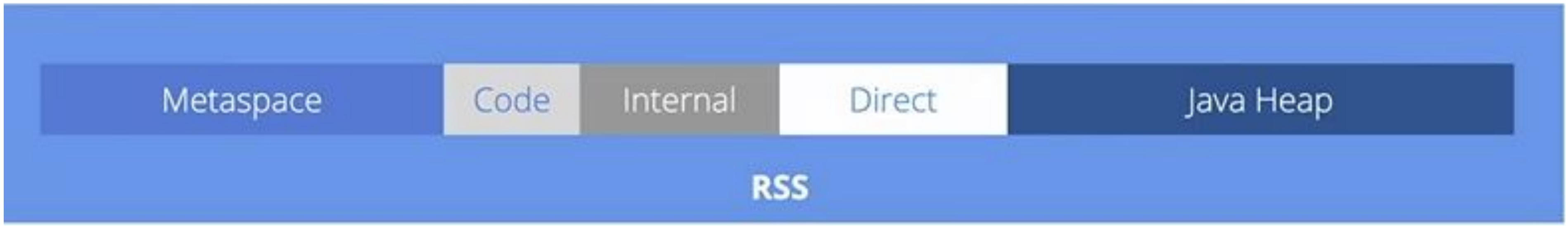


Java Resources in Containers

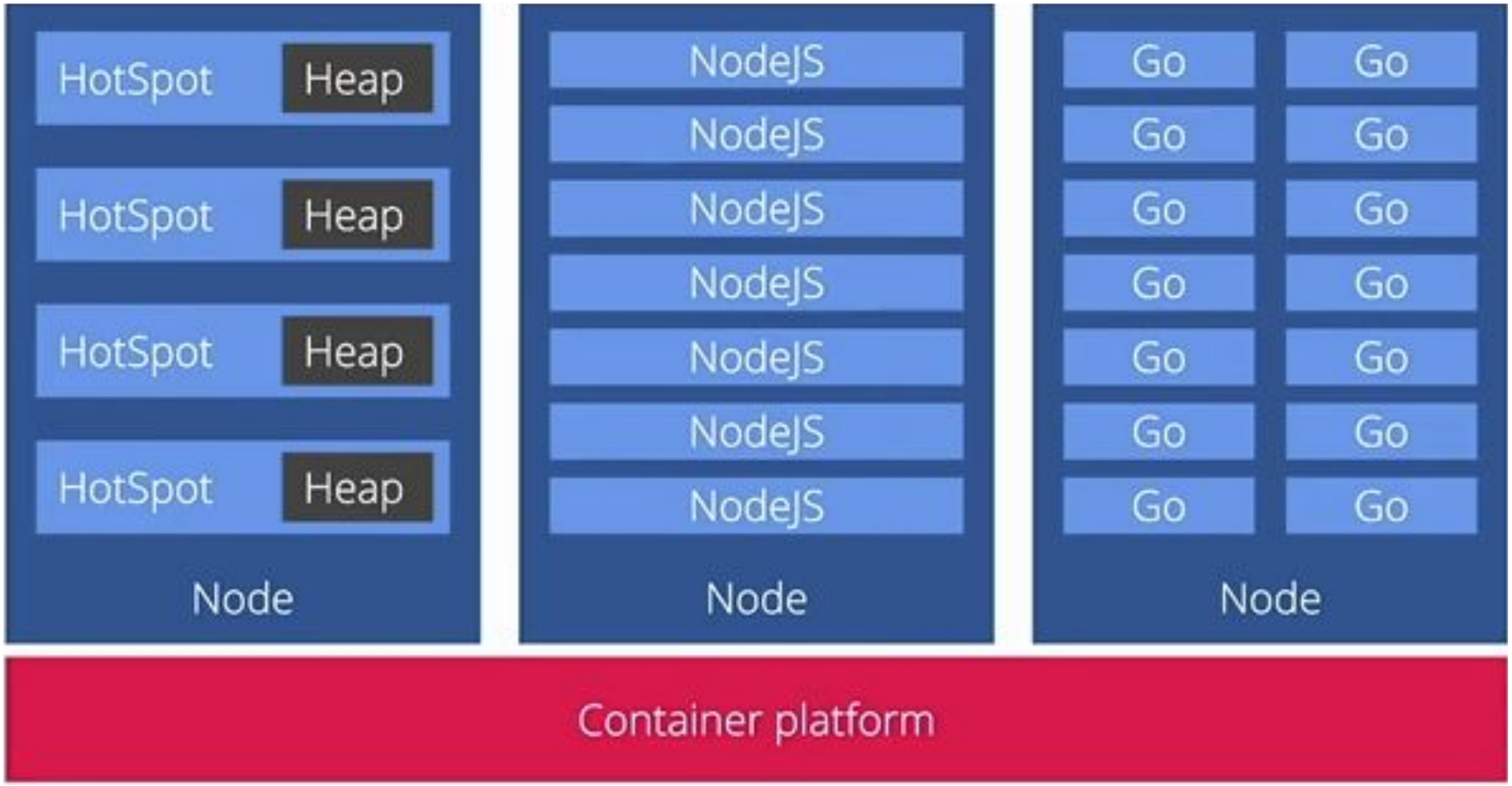
- JVM Optimized for throughputs (Requests/s)
- Startup overhead

  # Classes, Bytecode, JIT
- Memory Overhead

  Classes, Metadata, Compiled Code

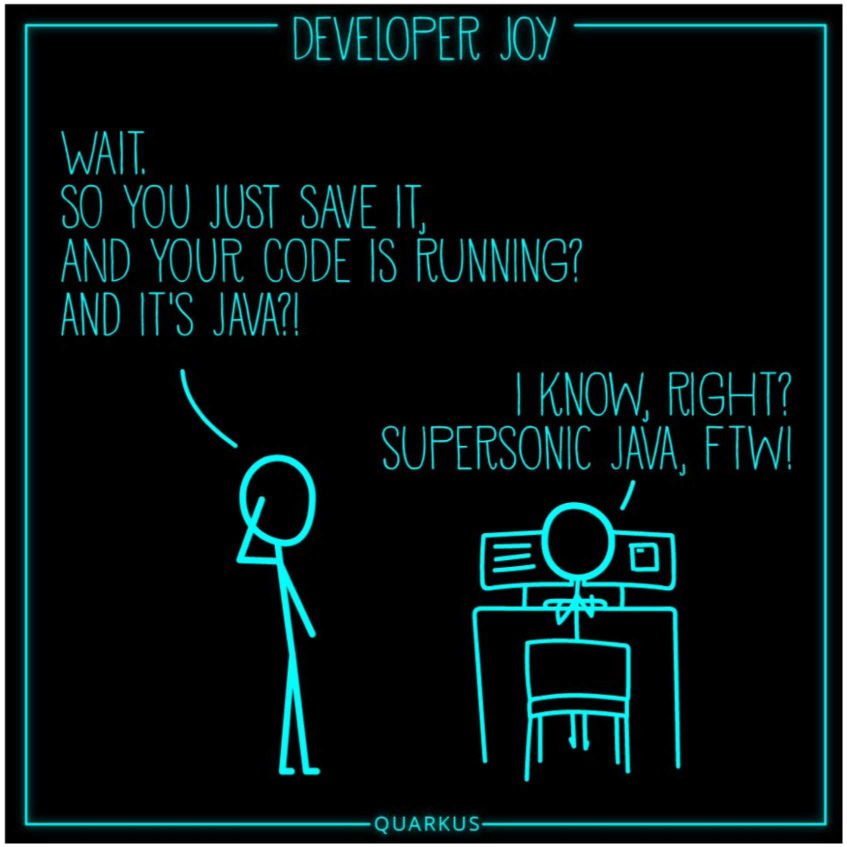


Java Resource Requirements

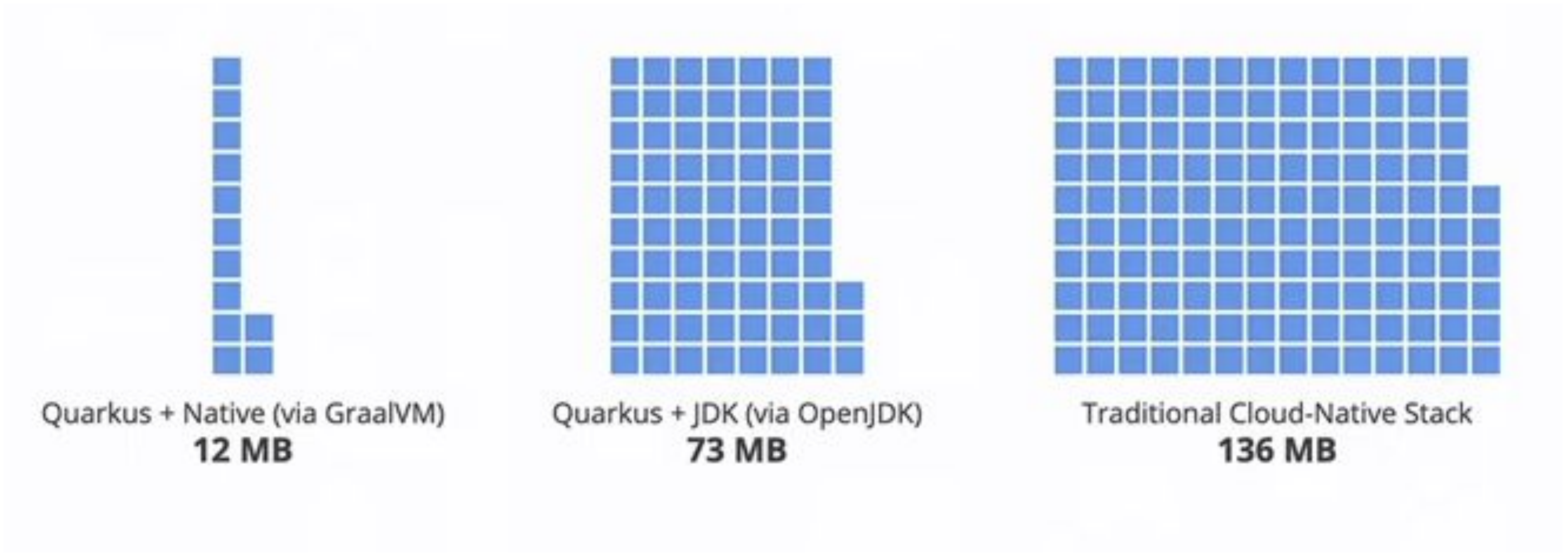


Developer Joy

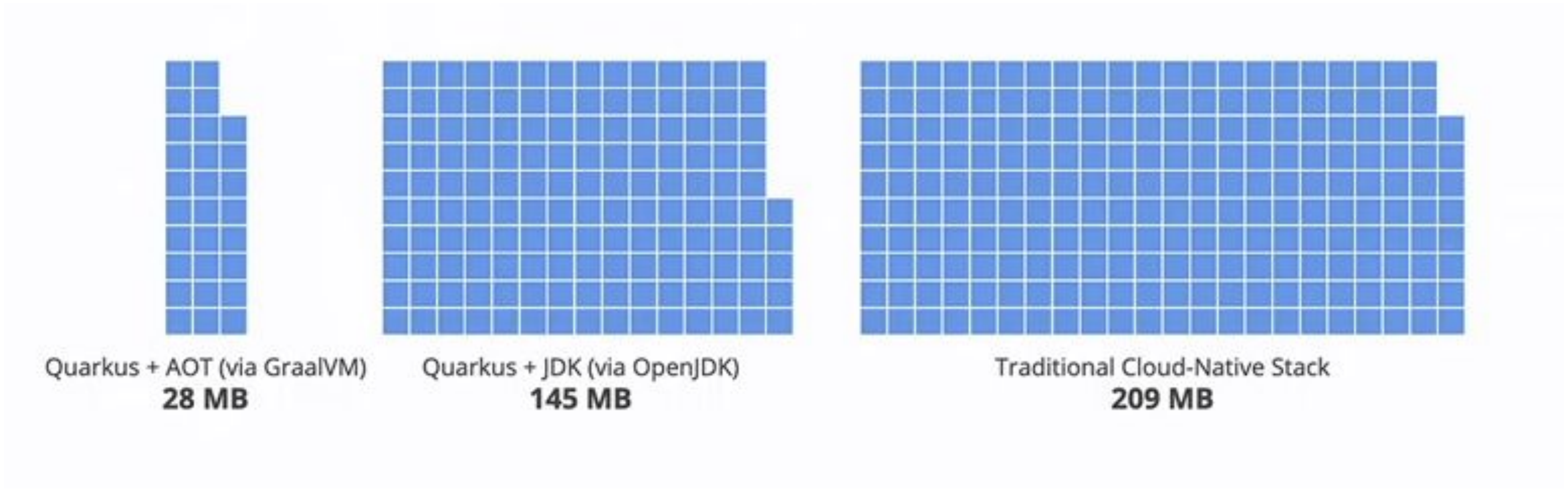
- Unified Configuration
- Live Coding
- Works on all the things
- Opinionated


DEVELOPER JOY
WAIT. SO YOU JUST SAVE IT, AND YOUR CODE IS RUNNING? AND IT'S JAVA?!
I KNOW, RIGHT? SUPERSONIC JAVA, FTW!
QUARKUS

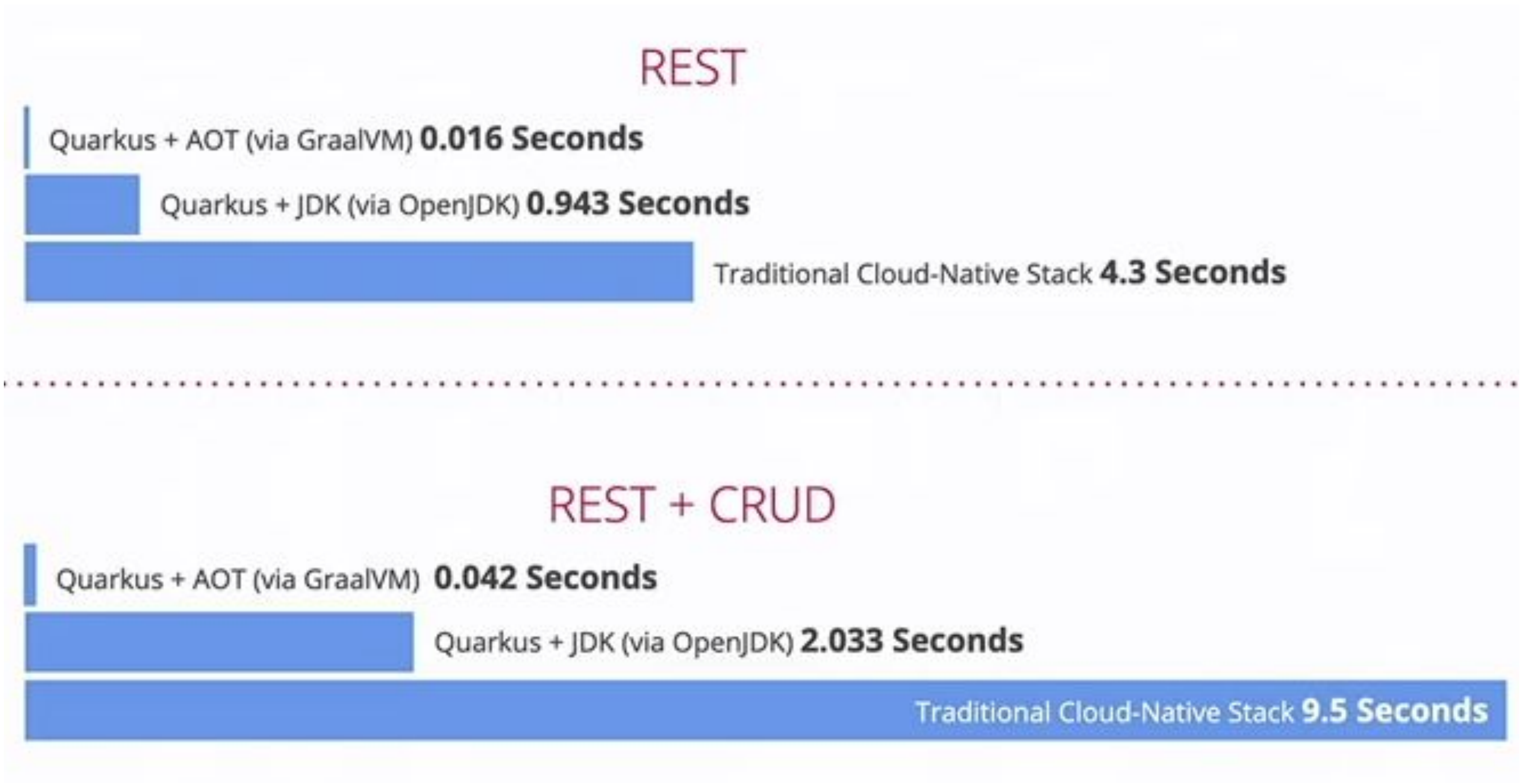
Memory Footprint REST-Service



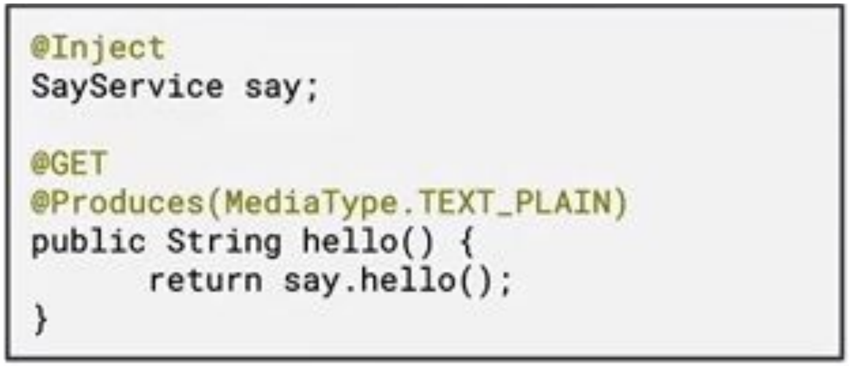
Memory Footprint REST+CRUD



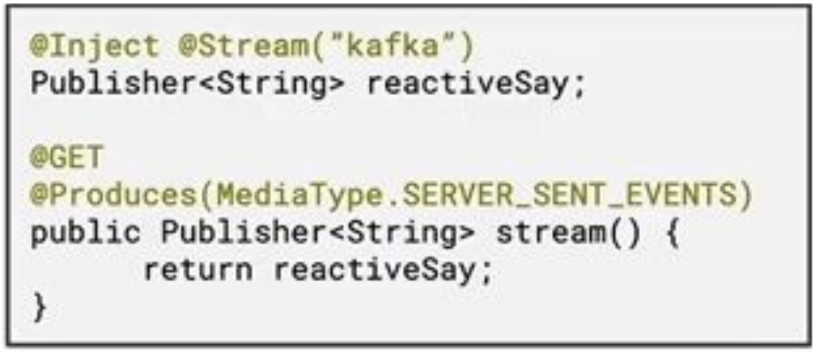
Faster Startup Times



Imperative and Reactive Model



```
@Inject
SayService say;

@GET
@Produces(MediaType.TEXT_PLAIN)
public String hello() {
      return say.hello();
}
```



```
@Inject @Stream("kafka")
Publisher<String> reactiveSay;

@GET
@Produces(MediaType.SERVER_SENT_EVENTS)
public Publisher<String> stream() {
      return reactiveSay;
}
```

- Reactive and Imperative style can be used
- Can use what fits your needs
- Enabler for reactive event-driven Systems

Plays well with others

Eclipse Vert.x	Eclipse MicroProfile	Spring Compat	Hibernate	RESTEasy	Apache Camel
Kubernetes	OpenShift	Jaeger	Prometheus	Apache Kafka	Netty

Trade Build-time for Startup-time

Typical framework tasks during startup time:

- Parse configuration files
- Scan Classpath and Inspect classes

  Annotations, Properties, Metadata
- Build framework metamodel
- Prepare Reflection and generate Proxies
- Start and Open I/O, Threads etc.
- … handle Requests

Trade Build-time for Startup-time

Typical framework tasks during startup time:

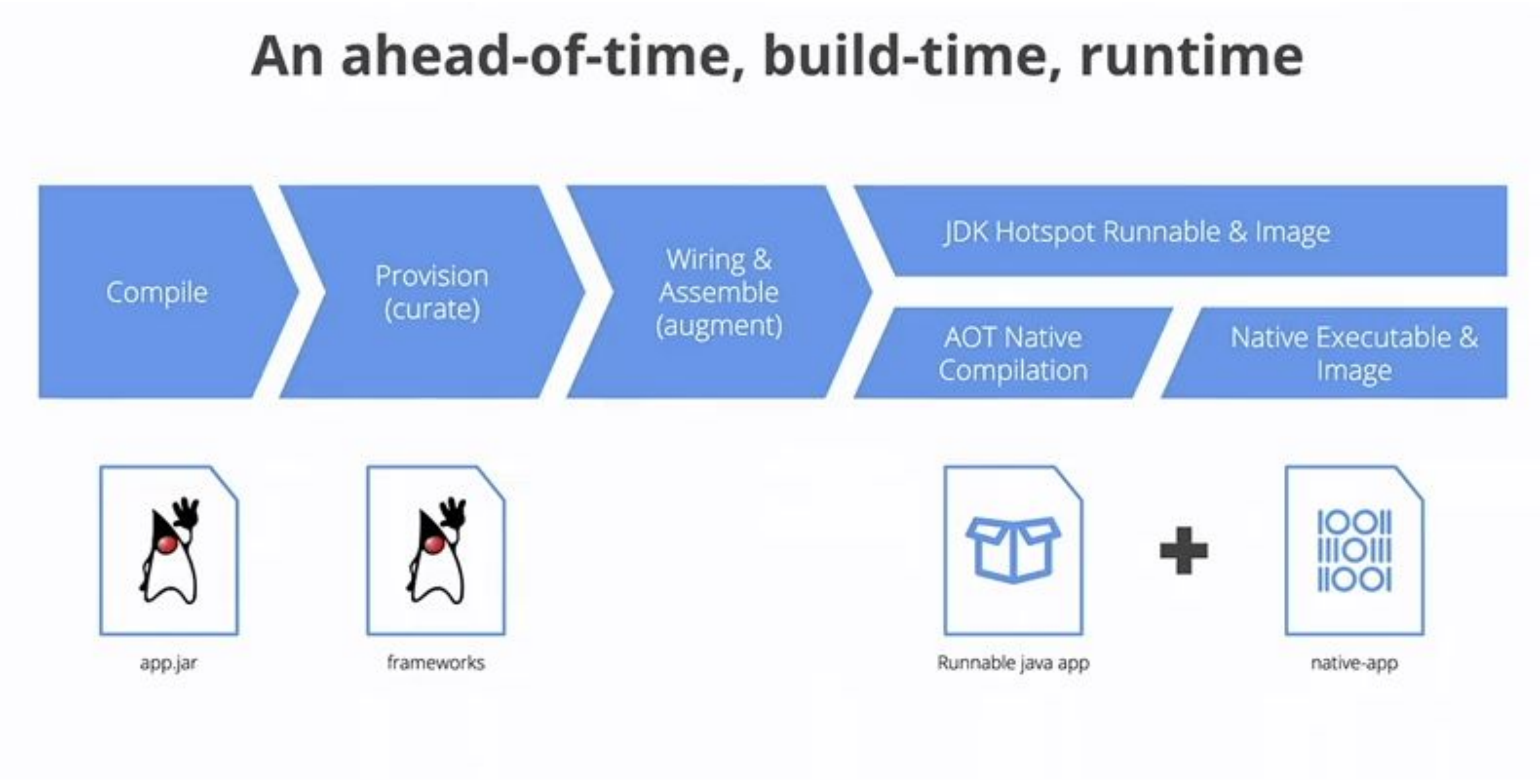
- Parse configuration files
- Scan Classpath and Inspect classes
  Annotations, Properties, Metadata
- Build framework metamodel
- Prepare Reflection and generate Proxies
- Start and Open I/O, Threads etc.
- … handle Requests

Move to Build-time!

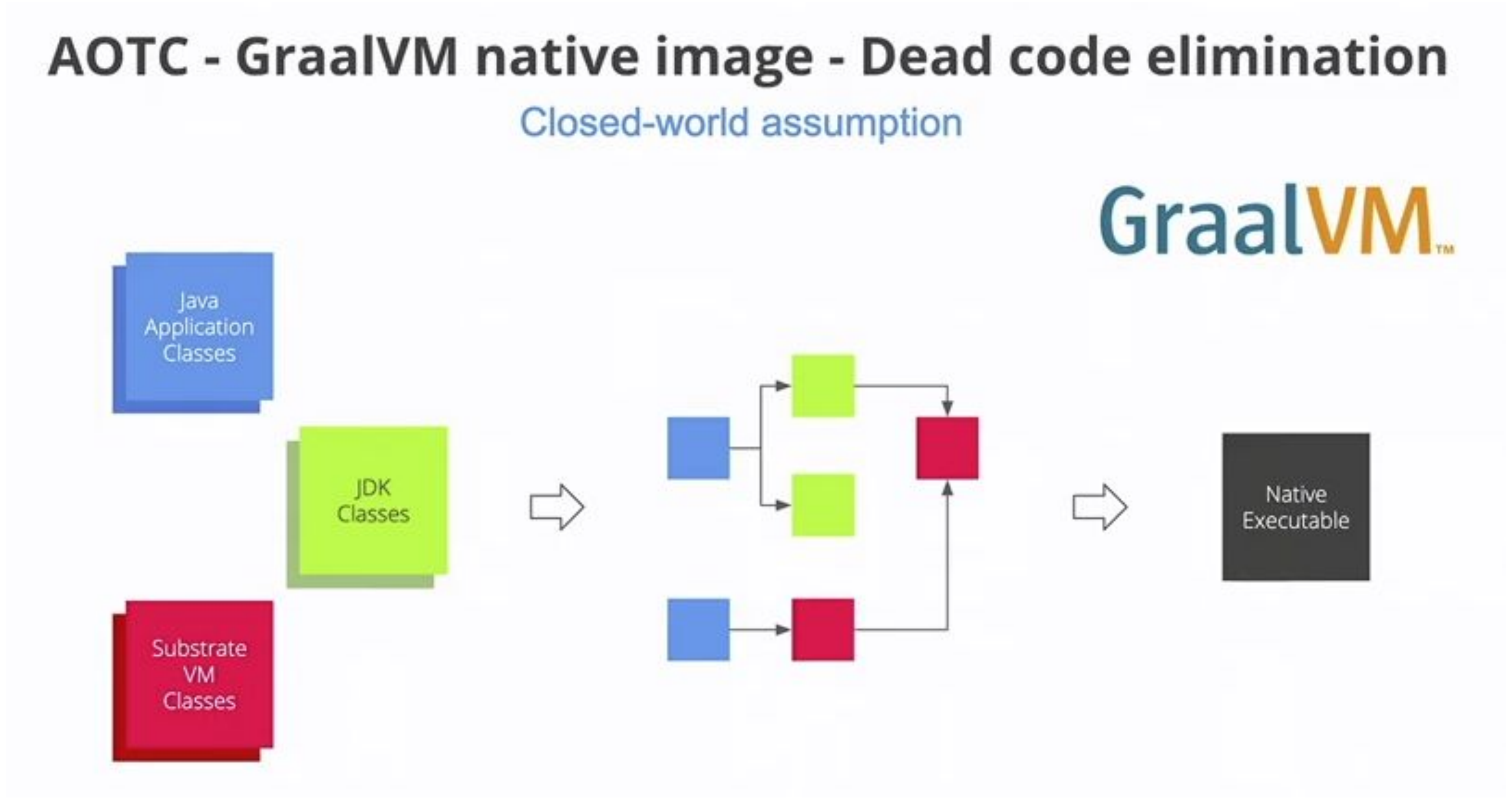
Benefits of Build-time Instrumentation

- Work is done once, instead of for every start
- Bootstrap classes are not needed at runtime
- Less time to start, less memory used
- Little to no reflection, nor dynamic proxy

Quarkus runs at Build-time


An ahead-of-time, build-time, runtime
Compile
Provision (curate)
Wiring & Assemble (augment)
JDK Hotspot Runnable & Image
AOT Native Compilation
Native Executable & Image
app.jar
frameworks
Runnable java app
native-app

GraalVM Native Image



Quarkus Benefits for GraalVM

- 100% of the ecosystem supported by GraalVM

  (if it doesn’t work it’s a bug)
- Orchestrates metadata needed by GraalVM
  - Take advantage of framework knowledge
  - Classes using reflection, resources, etc
  - No need for agent, pre-run, long JSON metadata or command lines
- Minimized Dependencies
- Help dead-code elimination

VM Selection Guide for Quarkus

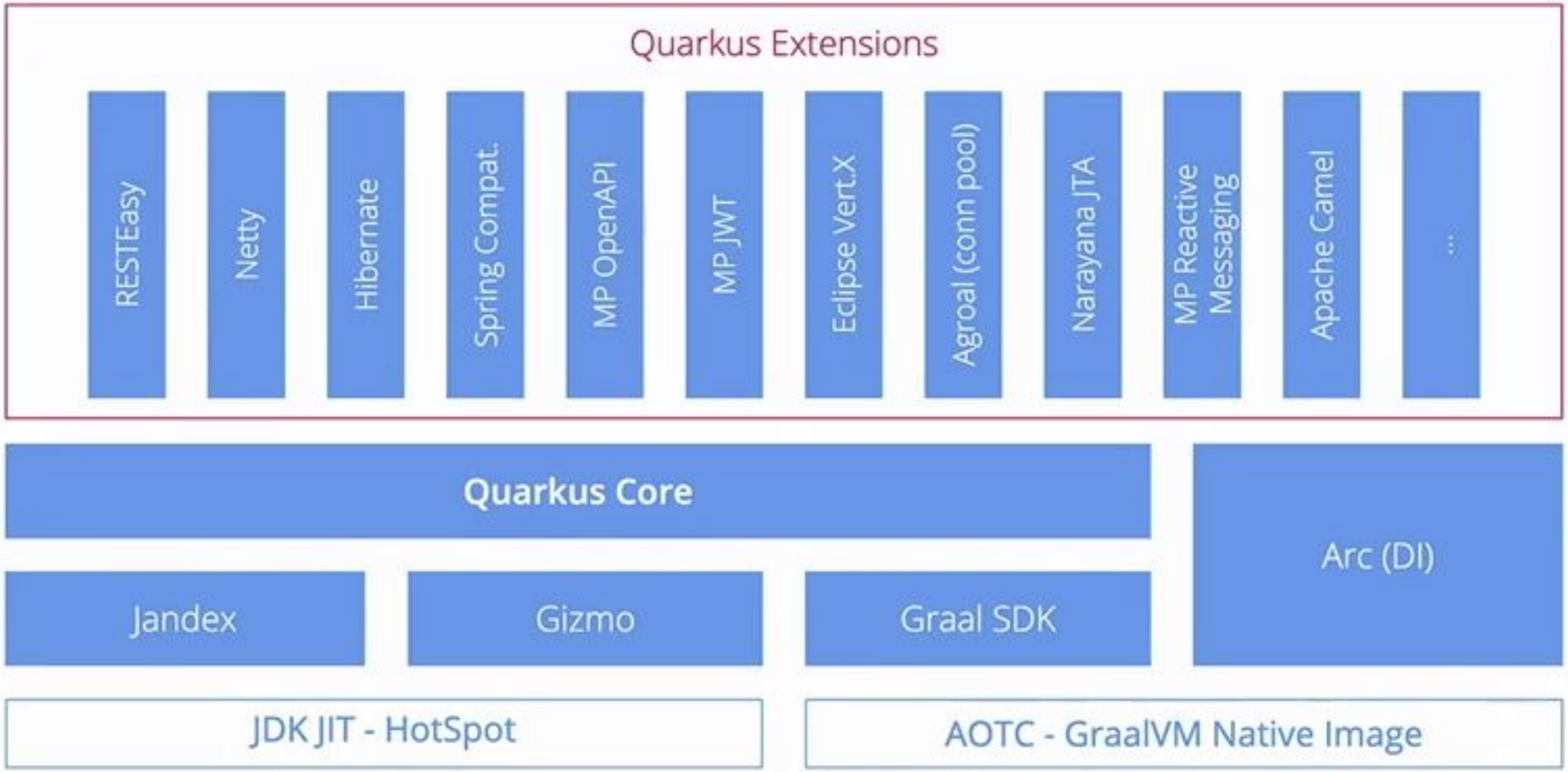
JIT - OpenJDK HotSpot

- High memory density
- Fast startup time
- Best raw performance (CPU)
- Best garbage collectors
- Higher heap size usage
- Known monitoring tools
- Compile Once, Run anywhere
- Libraries that only work in standard JDK

GraalVM native image

- Highest memory density
- Highest Request/s/MB
  - for low heap size usages
- Faster startup time
  - 10s of ms for Serverless

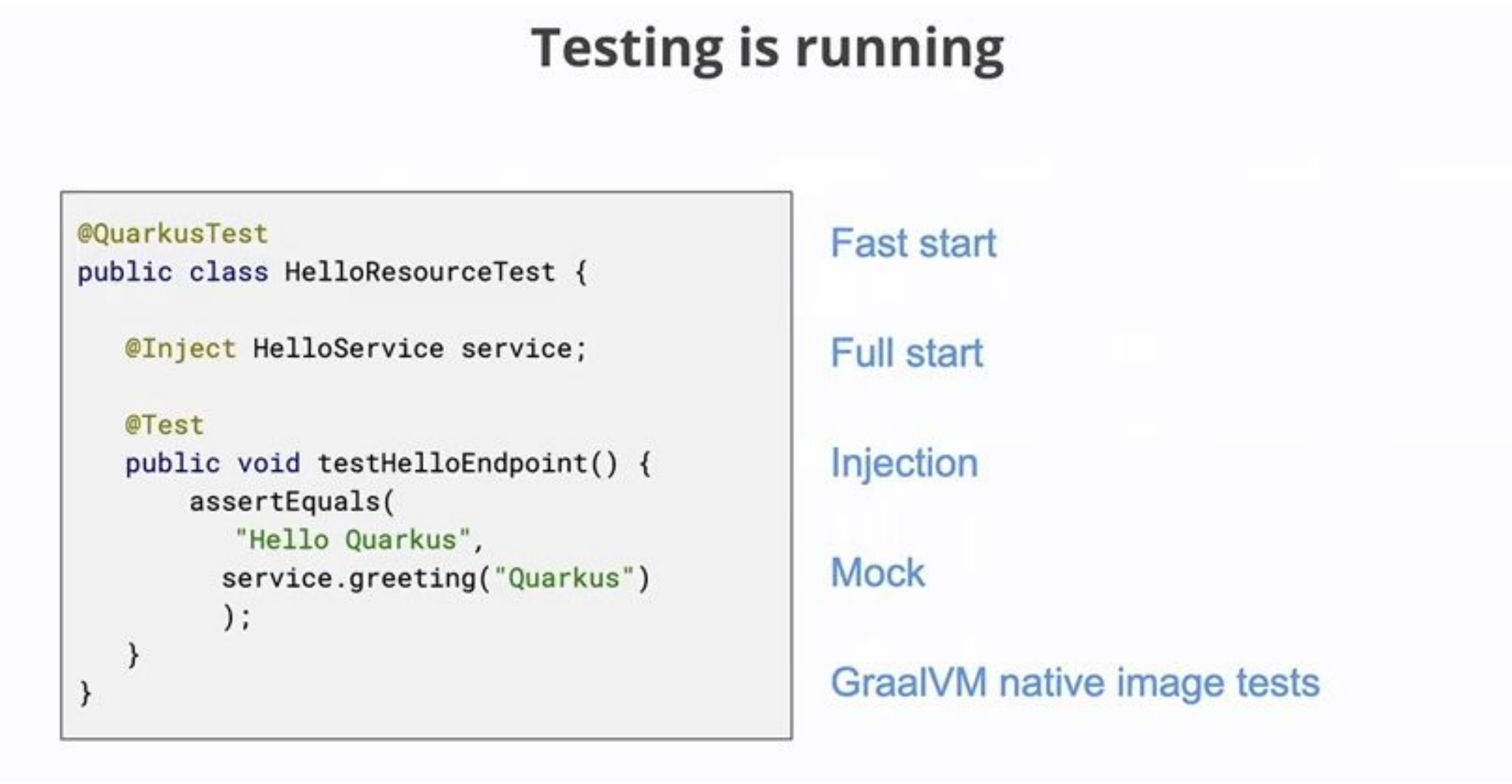
Quarkus as Extensible Framework



Adding your own Dependencies

- Custom dependencies work out-of the Box with Quarkus on JDK
- May work with GraalVM
- Possible to write custom Extension
    - Like adding a dependency + ...
    - Build-time startup and memory improvements
    - Better dead code elimination
    - Developer Joy

Quarkus embraces proper Testing

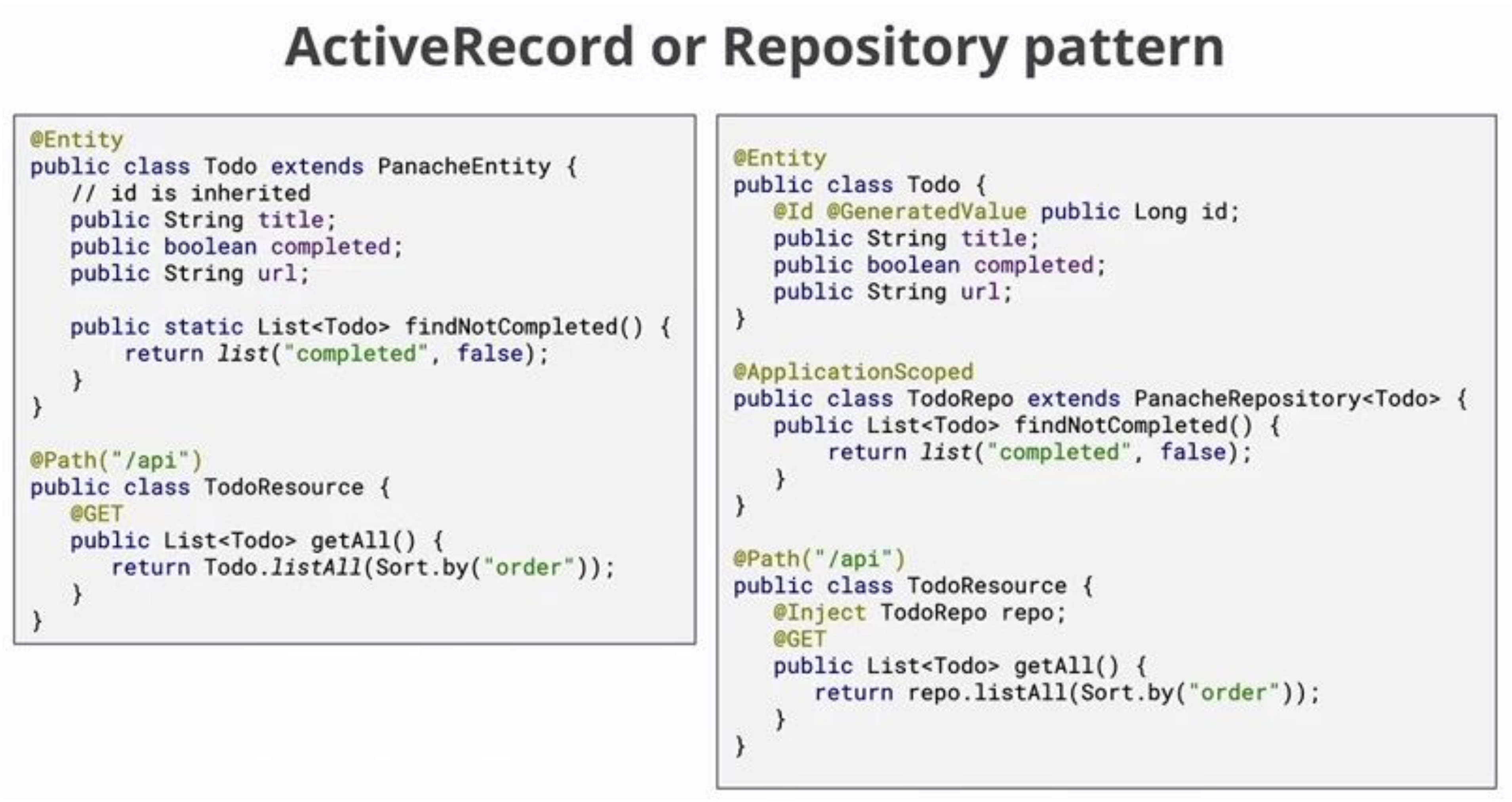


Testing is running

```
@QuarkusTest
public class HelloResourceTest {

   @Inject HelloService service;

   @Test
   public void testHelloEndpoint() {
       assertEquals(
          "Hello Quarkus",
         service.greeting("Quarkus")
         );
   }
}
```

- Fast start
- Full start
- Injection
- Mock
- GraalVM native image tests

Traditional or Opinionated Persistence



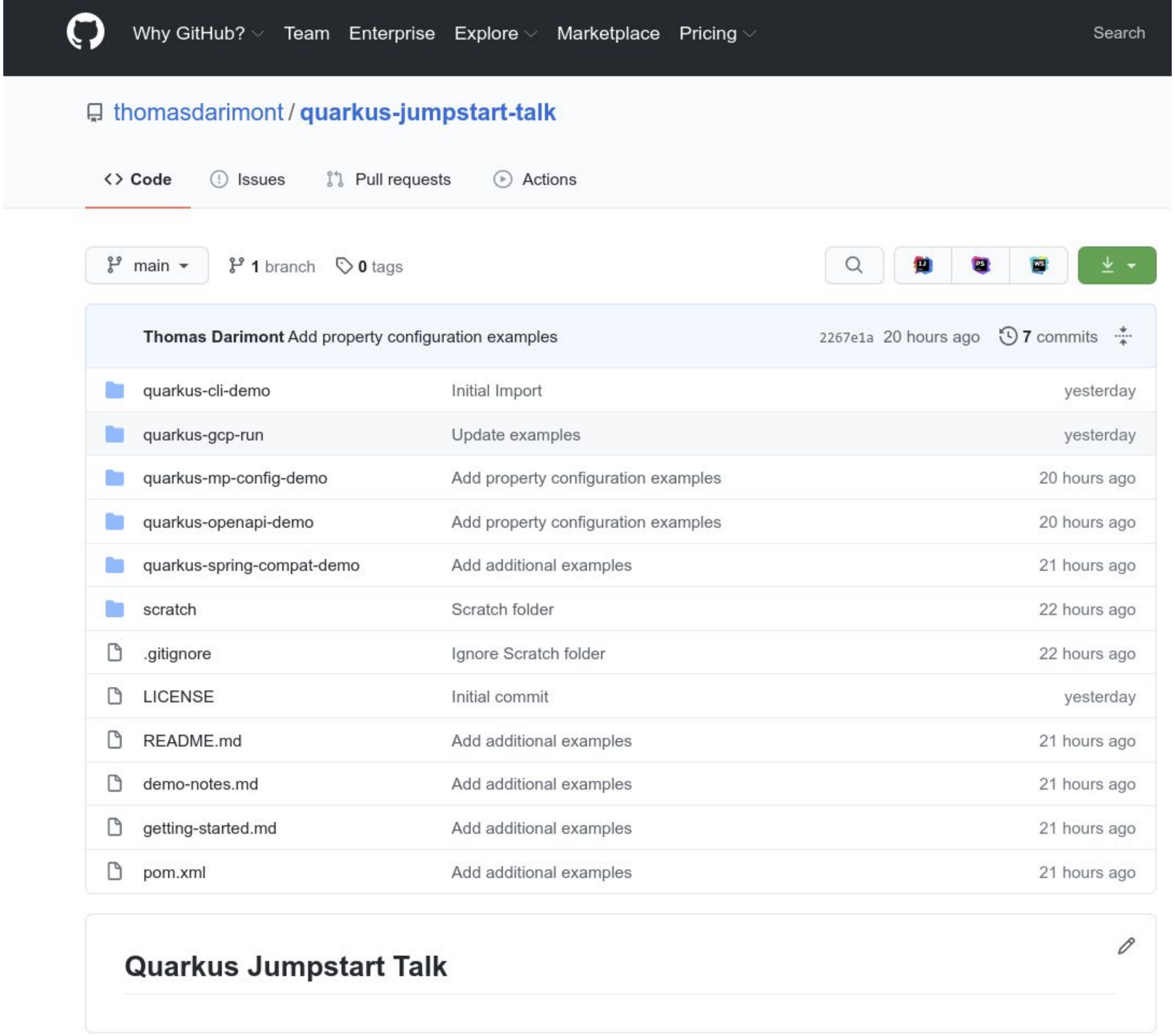
ActiveRecord or Repository pattern

```java
@Entity
public class Todo extends PanacheEntity {
   // id is inherited
   public String title;
   public boolean completed;
   public String url;

   public static List<Todo> findNotCompleted() {
       return list("completed", false);
   }
}

@Path("/api")
public class TodoResource {
   @GET
   public List<Todo> getAll() {
      return Todo.listAll(Sort.by("order"));
   }
}
```

```java
@Entity
public class Todo {
   @Id @GeneratedValue public Long id;
   public String title;
   public boolean completed;
   public String url;
}

@ApplicationScoped
public class TodoRepo extends PanacheRepository<Todo> {
   public List<Todo> findNotCompleted() {
       return list("completed", false);
   }
}

@Path("/api")
public class TodoResource {
   @Inject TodoRepo repo;
   @GET
   public List<Todo> getAll() {
      return repo.listAll(Sort.by("order"));
   }
}
```

Quarkus Use-cases

- Command-line Apps
    - Think of Java+GraalVM as the new Go :)
- Server-less Functions
    - Sub ms start-up times reduce request latency
- Scalable Microservices
    - Lower resource consumption
    - More Services on same hardware!


Why GitHub? Team Enterprise Explore Marketplace Pricing Search

thomasdarimont / quarkus-jumpstart-talk

Code Issues Pull requests Actions

main 1 branch 0 tags

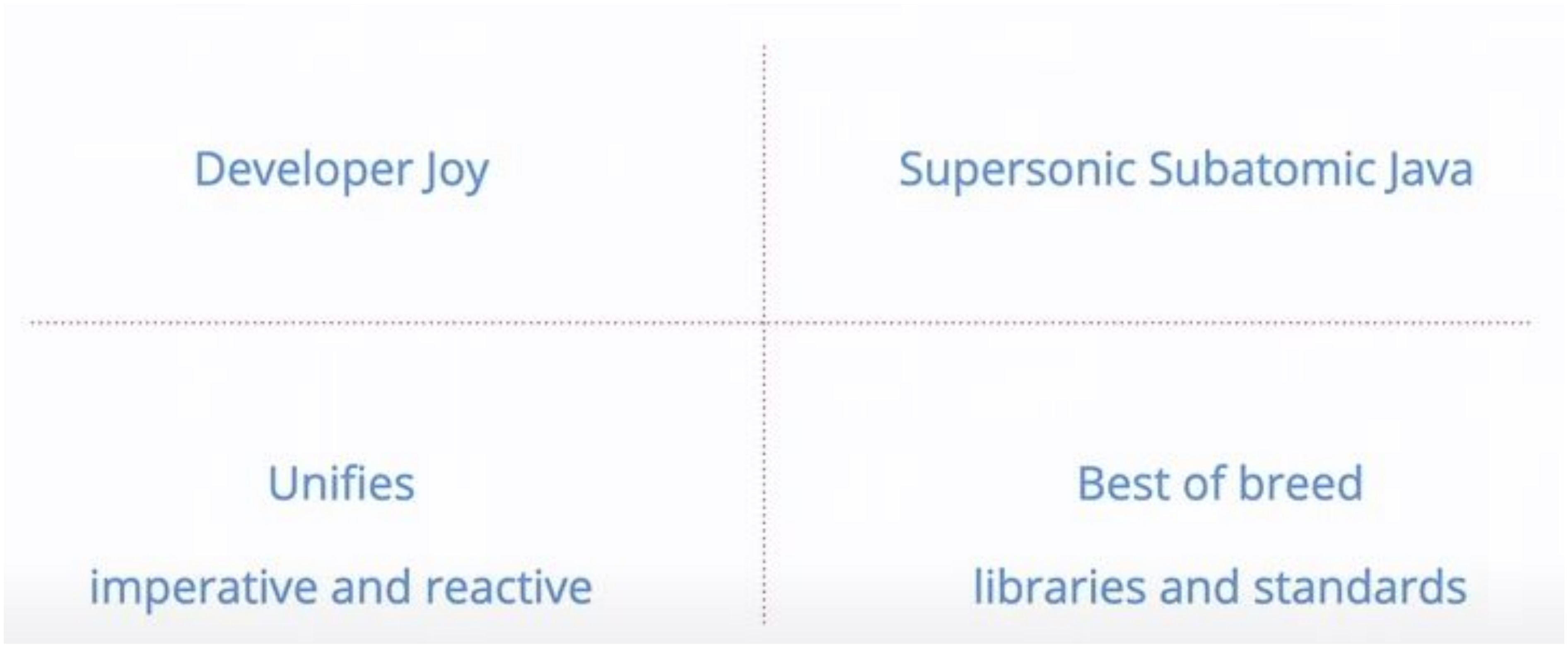
Thomas Darimont Add property configuration examples 2267e1a 20 hours ago 7 commits

quarkus-cli-demo	Initial Import	yesterday
quarkus-gcp-run	Update examples	yesterday
quarkus-mp-config-demo	Add property configuration examples	20 hours ago
quarkus-openapi-demo	Add property configuration examples	20 hours ago
quarkus-spring-compat-demo	Add additional examples	21 hours ago
scratch	Scratch folder	22 hours ago
.gitignore	Ignore Scratch folder	22 hours ago
LICENSE	Initial commit	yesterday
README.md	Add additional examples	21 hours ago
demo-notes.md	Add additional examples	21 hours ago
getting-started.md	Add additional examples	21 hours ago
pom.xml	Add additional examples	21 hours ago

Quarkus Jumpstart Talk

Quarkus Demo 2

Quarkus Benefits



Developer Joy

Supersonic Subatomic Java

Unifies imperative and reactive

Best of breed libraries and standards

Links

- Quarkus Jumpstart on Github
- Quarkus Workshop
- Quarkus Cheat Sheet
- code.quarkus.io
- Quarkus Why, How and What

Major Inspiration for this Talk was Quarkus why, how and what by Emmanuel Bernard