



# QUARKUS

QUARKUS WORLD TOUR 2022

INTRODUCTION TO Quarkus

# WHO'S ON STAGE TODAY?

YOANN RODIERE



QUARKUS



HIBERNATE

- Hibernate team at Red Hat
- Hibernate Search lead developer
- Contributor to
  - Hibernate (Search, ORM, Validator)
  - Quarkus (Search, ORM, Validator, Core)
  - Whatever gets in the way...



# QUARKUS

SUPERSONIC, SUBATOMIC JAVA

# DEMO



<https://code.quarkus.io/?e=resteasy&e=hibernate-orm-panache&e=jdbc-postgresql&e=smallrye-openapi>

(Use a desktop browser to download the project...)

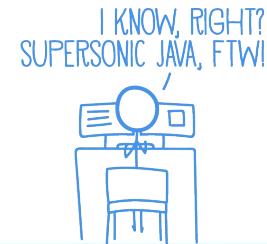
# DEVELOPER JOY

A cohesive platform for optimized developer joy:

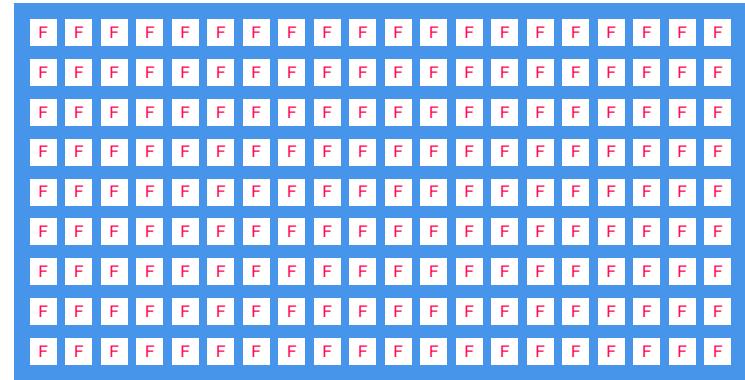
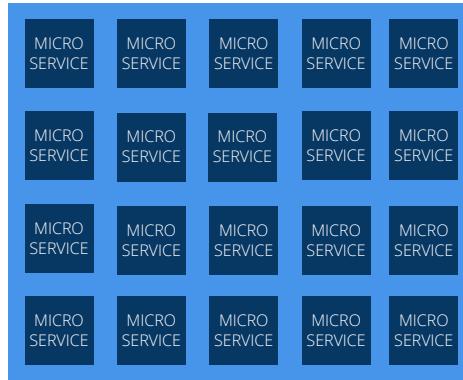
- Based on standards, but not limited
- Unified configuration
- Streamlined code for the 80% common usages, flexible for the 20%
- No hassle native executable generation
- Zero config, live reload in the blink of an eye
- Seamless Development and testing setup with dev services starting databases, SSO, Kafka etc.



WAIT.  
SO YOU JUST SAVE IT,  
AND YOUR CODE IS RUNNING?  
AND IT'S JAVA?!

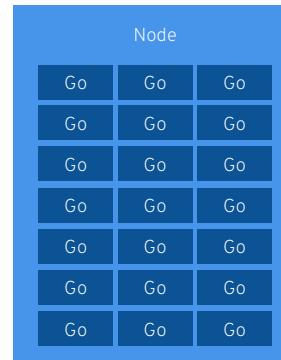
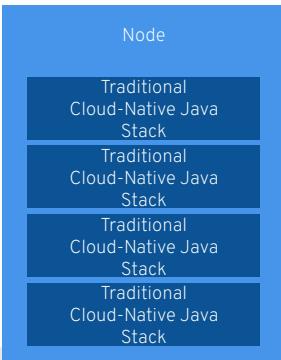


# FROM MONOLITH TO...



- 1 monolith  $\approx$  20 microservices  $\approx$  200 functions
- Scale to 1 vs scale to 0
- Start up time

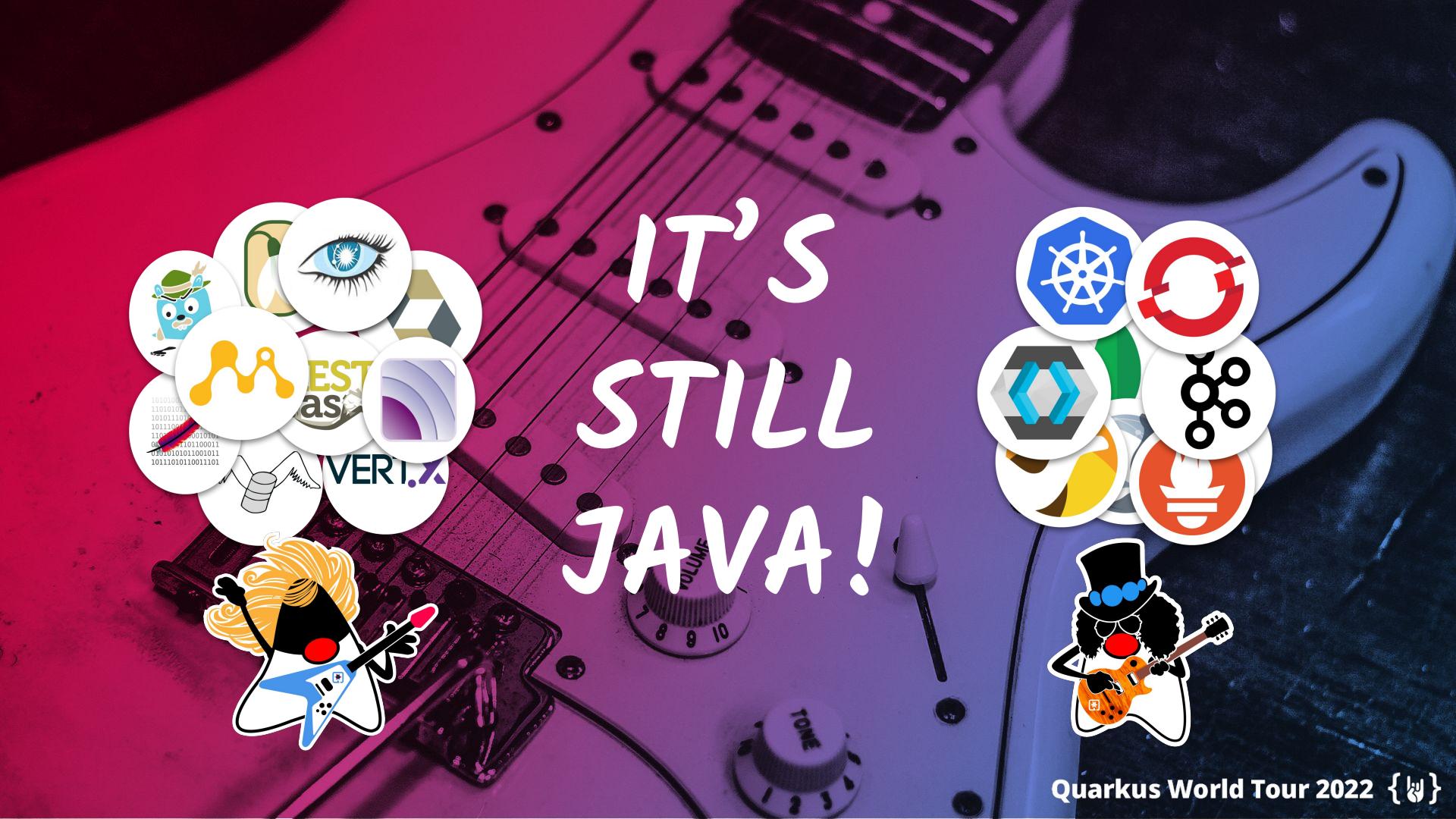
# SUPersonic SUBatomic JAVA



CONTAINER ORCHESTRATION

# DEMO





# IT'S STILL JAVA!

# BEST OF BREED LIBRARIES & STANDARDS



Eclipse  
Vert.x



Eclipse MicroProfile



Apache Kafka



Prometheus



Netty



Hibernate



OpenShift



Jaeger



RESTEasy



Apache  
Camel



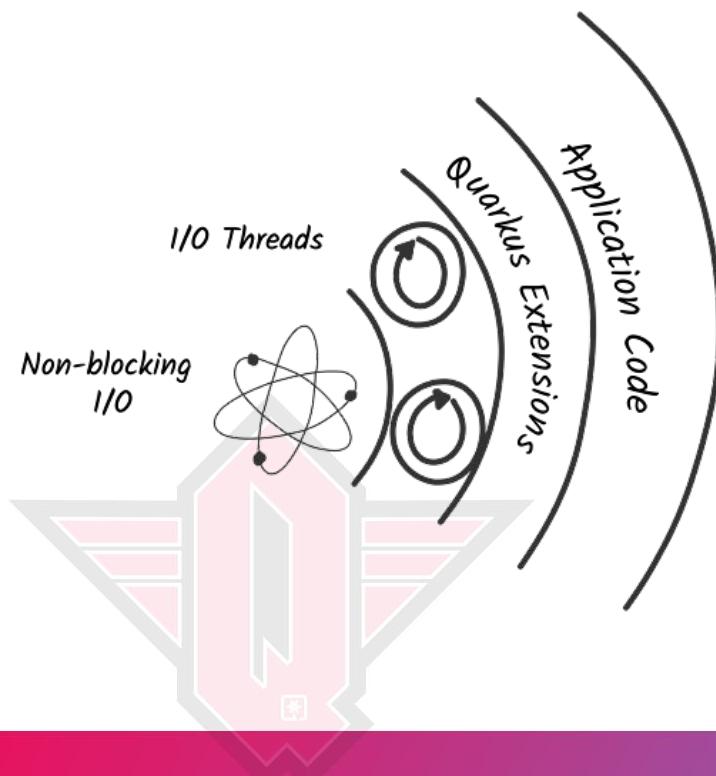
Kubernetes



Spring  
Compat



# UNIFICATION OF IMPERATIVE AND REACTIVE



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

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

- Combine both Reactive and imperative development in the same application
- Use the technology that fits your use-case
- Key for reactive systems based on event driven apps

# QUARKUS BENEFITS

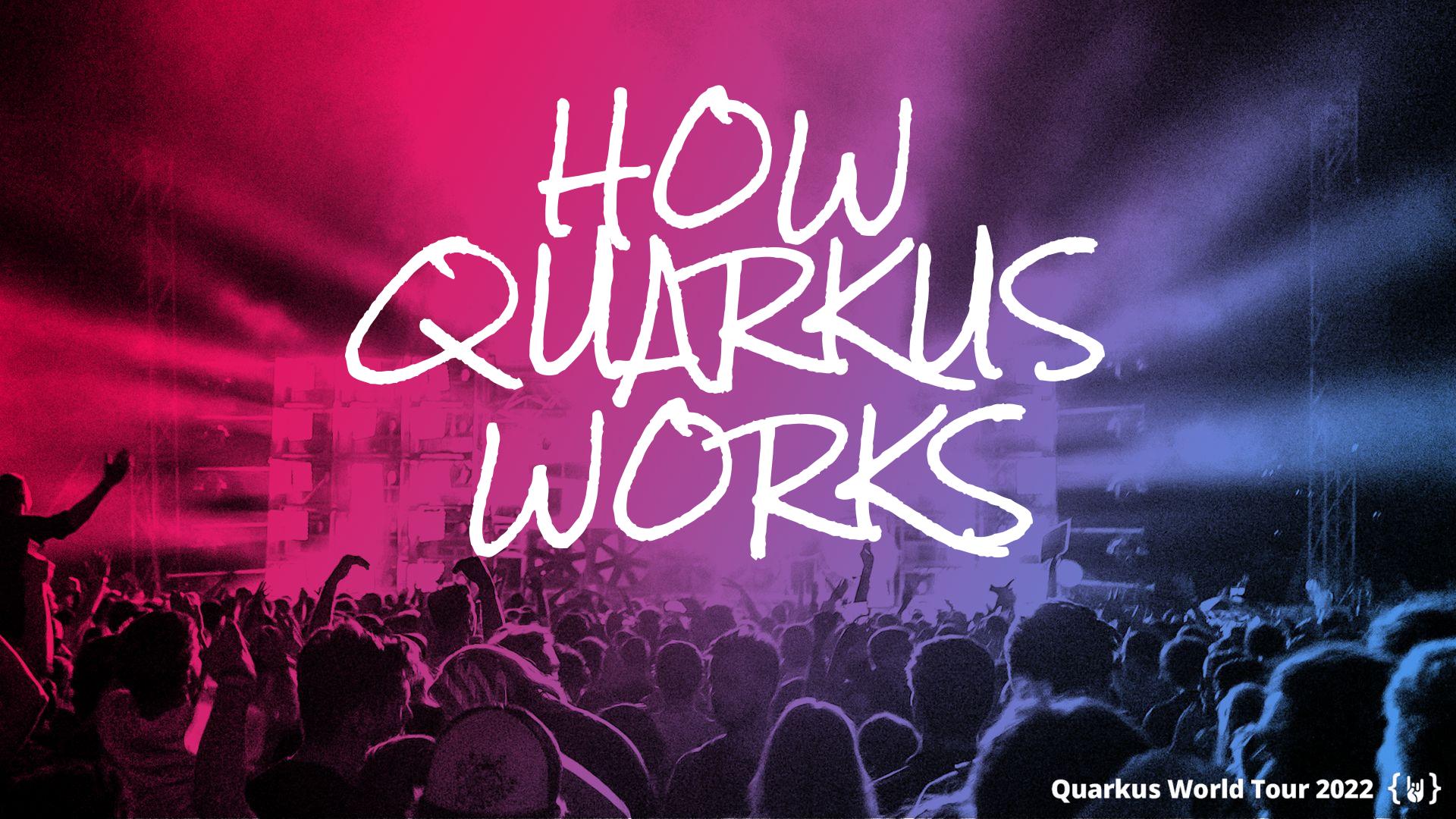
DEVELOPER JOY

SUPersonic  
SUBatomic Java

BEST OF BREED  
LIBRARIES AND  
STANDARDS

UNIFIES  
IMPERATIVE AND  
REACTIVE



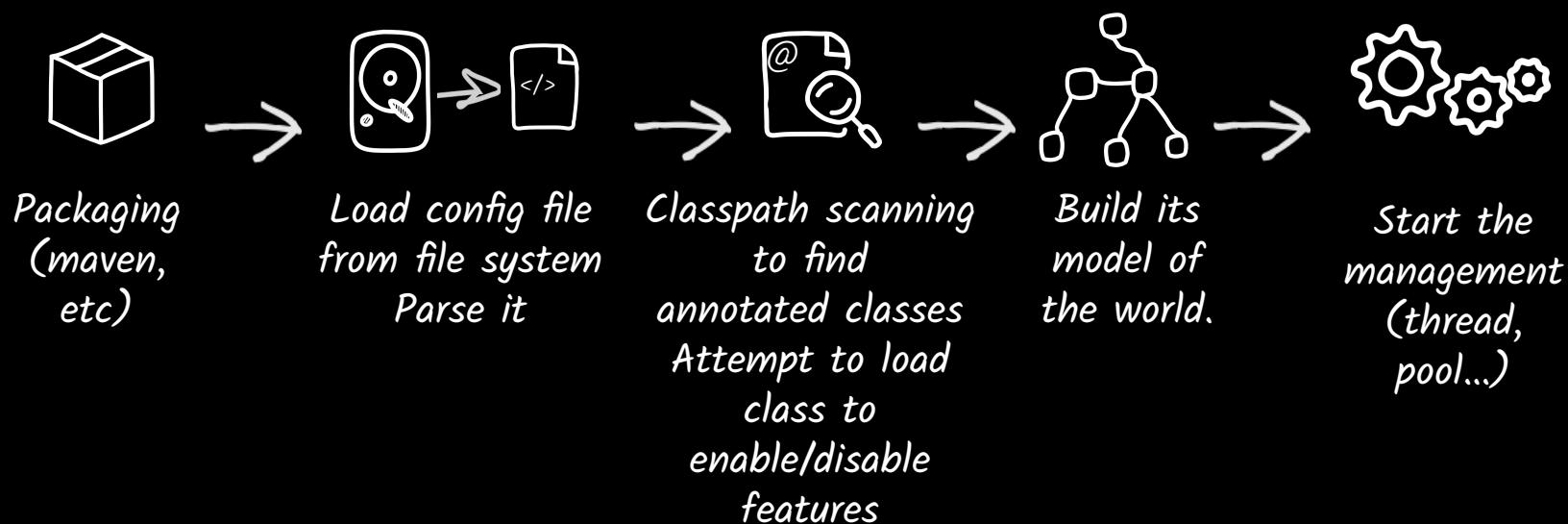
The background of the image is a dark, grainy photograph of a large crowd at a concert or festival. Red and blue stage lights create a vibrant atmosphere, with beams of light cutting through the darkness. The text "HOW QUARKUS WORKS" is overlaid in the center in a large, white, hand-drawn style font.

# HOW QUARKUS WORKS

# HOW DOES A FRAMEWORK START?

Build Time

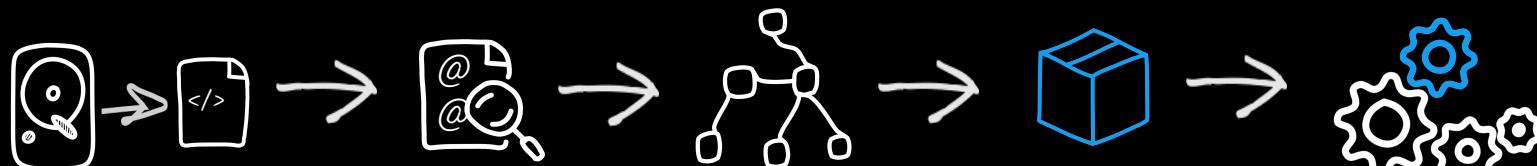
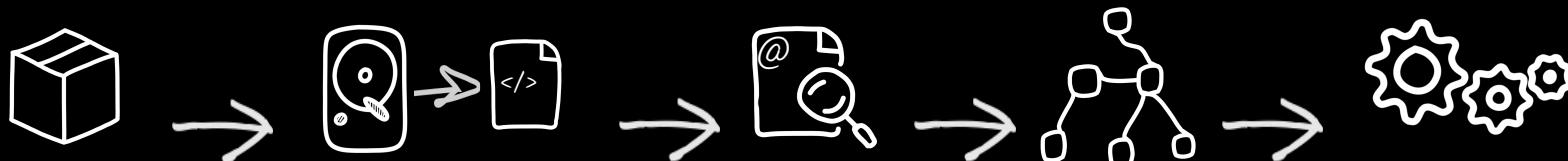
Runtime



# THE QUARKUS WAY

Build Time

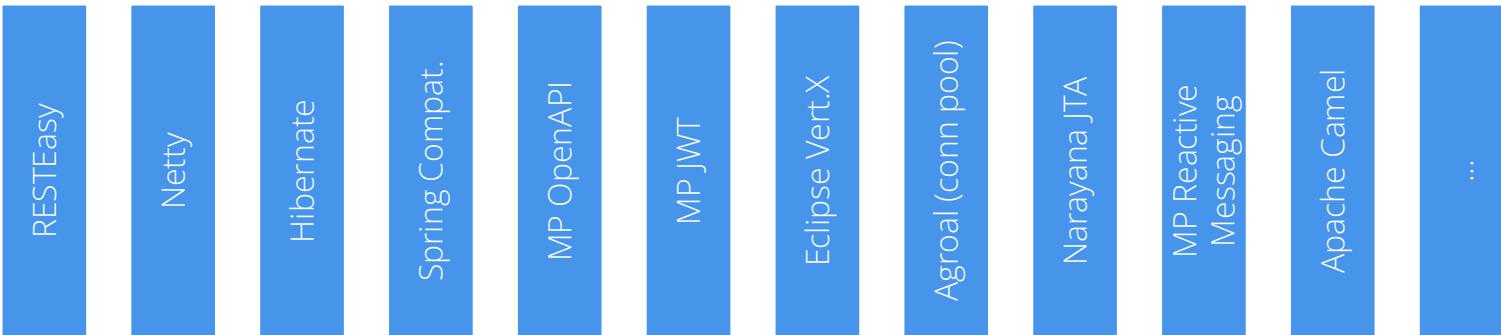
Runtime



Build Time

Runtime

## Quarkus Extensions



## Quarkus Core

Jandex

Gizmo

Graal SDK

Arc (DI)

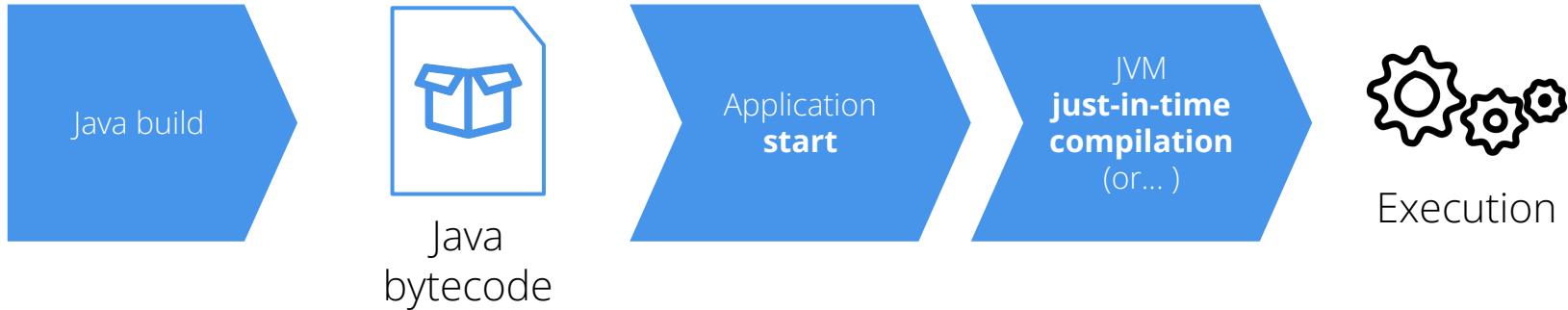
JDK JIT - HotSpot

AOTC - GraalVM Native Image

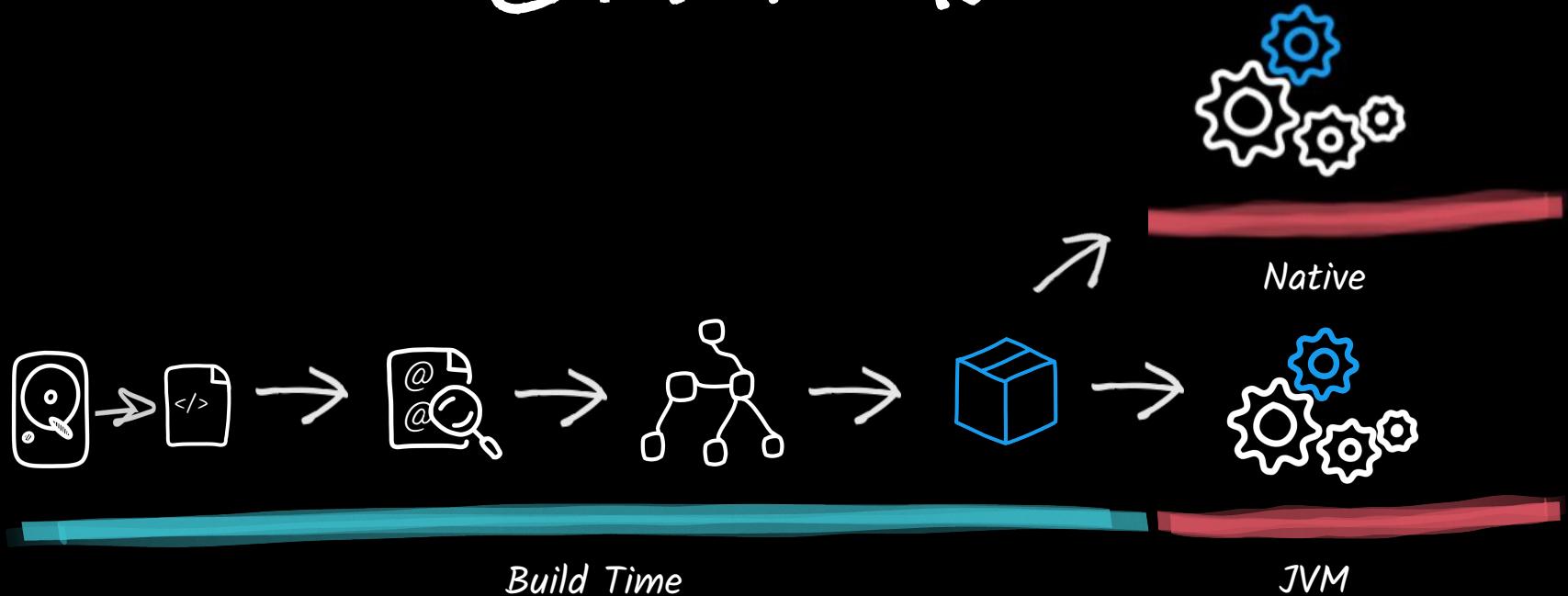


# NATIVE COMPILATION

# NATIVE = AHEAD-OF-TIME



# THE QUARKUS WAY ENABLES NATIVE COMPILATION



# DEMO



# THE DARK SIDE OF GRAALVM

## NOT SUPPORTED

- Dynamic classloading
- InvokeDynamic
- Finalizer
- Security manager
- JVMTI, JMX, native VM Interfaces



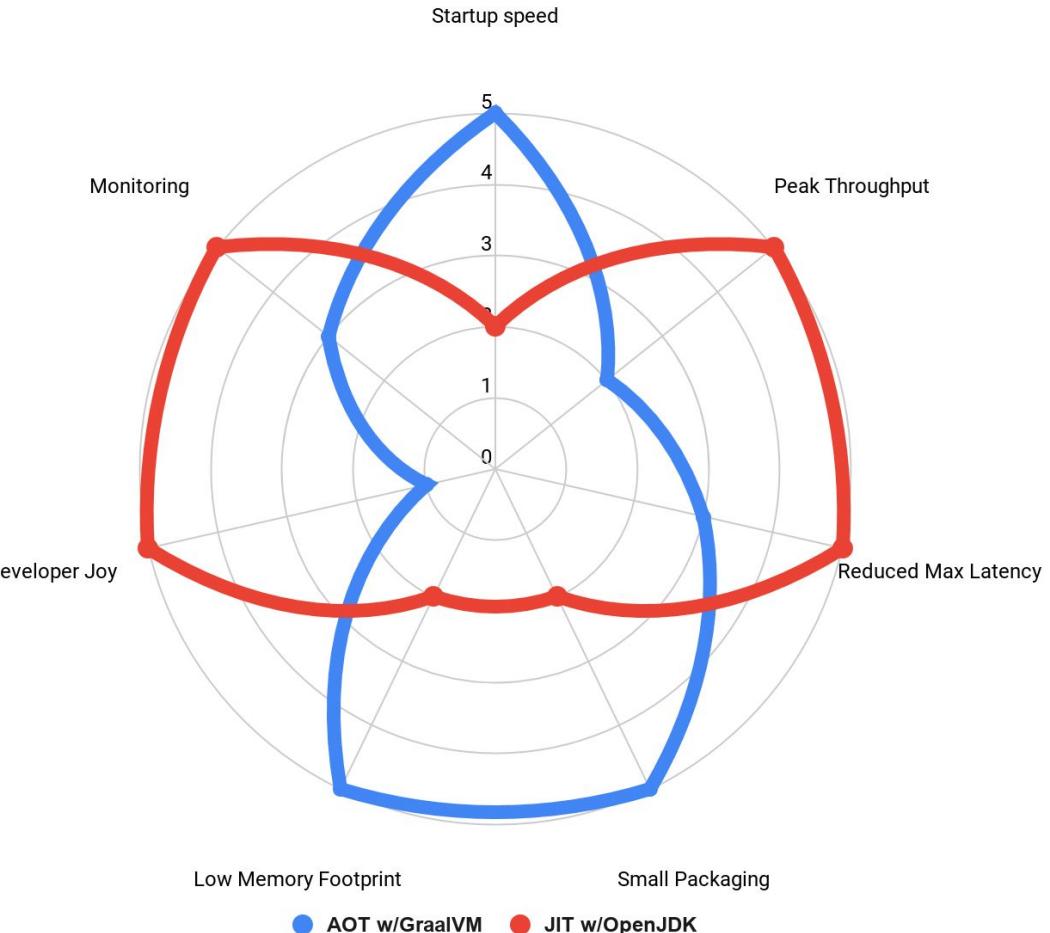
## OK WITH CAVEATS IN USAGE

- Reflection (manual list)
- Method handles
- Dynamic proxy (manual list)
- JNI (manual list)
- Static initializers (eager)
- References (similar)

RED: THE "ACTUALLY GOOD" PARTS

GREY: THE "NOT SO BAD" PARTS

# JVM vs. NATIVE



# NATIVE COMPILATION WITH MANDREL

- Direct downstream distribution of GraalVM
- Near drop-in replacement for GraalVM
- Uses standard OpenJDK instead of GraalVM's custom version
- Focused on native compilation for Java only
- Red Hat contributes upstream!
  - Debugging support
  - ARM support
  - Java Flight Recorder
  - Quarkus part of the upstream GraalVM test suite

# USERS OWN STORIES

<https://quarkus.io/blog/tag/user-story/>



# DON'T TAKE OUR WORD FOR IT!

 <b>asiakastieto</b>	"We went from 1 min startup times to 400 ms."	<a href="#">RH Press Release Community</a>
 <b>vodafone</b>	"We became increasingly worried about resource consumption that Spring Boot was having while being deployed on the Kubernetes cluster... It became increasingly cumbersome to find ways to circumvent the methodology we were using just to squeeze every little bit of performance out of Spring Boot" - Christos Sotiriou DXL Backend Chapter Lead, Vodafone Greece	<a href="#">Information Week Community</a>
 <b>Lufthansa</b>	"We could run 3x denser deployments without sacrificing availability and response times of service."  "Quarkus is close to what our developers are already doing with Spring and it's familiar to them. This is a big benefit"	<a href="#">Community</a>
 <b>wipro</b>	"Before we introduced Quarkus, many of our customers had started to look at alternative stacks like Go and node.js to improve performance and efficiency. These customers were weary of selecting a new language and having to hire new developers or retrain their existing Java developers." - Arijit Mazumdar  "There was a low learning curve with Quarkus. It took one of our developers one week to get up to speed on Quarkus and another week to migrate a Spring application to Quarkus." - Arijit Mazumdar  "Quarkus and the Spring API compatibility reduced the migration time and complexity which is critically important for our customers." - Arijit Mazumdar	<a href="#">Community</a>

# DON'T TAKE OUR WORD FOR IT!



"using Spring with AWS Lambda would have been prohibitive because the startup time of Spring in AWS Lambda is too big from my research" - Dennis Baerten

[Blog](#)

"As costs increase, this is when the benefit of using Quarkus will be experienced due to its more efficient use of cloud resources and fast startup time compared to plain Java and Spring Boot" - Dennis Baerten

"It took me about 3 days to get familiar with the Quarkus stack" - Dennis Berten, Spring Developer



"Some of Payair's developers had mainly Spring experience, we were concerned that it would be difficult for them to "switch sides". It turned out that our fear of the unknown was completely unfounded. Quarkus leverages some good old Jakarta EE standards that all Java developers are familiar with. We did not have to learn a bunch of new APIs." - Hubert Lewandowski

[Blog](#)

"As a long term Spring developer I realized that Spring is slowly becoming the very thing it swore to destroy. The initial premises of Spring (which basically can be summed up as a lightweight alternative to Jakarta EE) are way past the expiry date now. Spring is the undisputed heavyweight champion that can handle everything you imagine but is not your best option for fast and light services. And that applies to Spring Boot as well. - Hubert Lewandowski



"When you adopt Quarkus, you will be productive from day one since you don't really need to learn new technologies." -TalkDesk

[Blog](#)



"After deploying, we found that Quarkus used about 15% of the CPU, 12% of the memory compared to Spring Boot. So far, we are sticking with Quarkus!" - Sam Dacanay, Lead Software Engineer

[Blog](#)

But I ALREADY KNOW SPRING...



# WHY MIGRATE?

DEVELOPER JOY

BEST OF BREED  
LIBRARIES AND  
STANDARDS

SUPersonic  
SUBatomic JAVA

UNIFIES  
IMPERATIVE AND  
REACTIVE

- What if there isn't an extension for my library?
- GraalVM compilation time / is memory intensive...
- How stable is it? There are lots of issues on GitHub...

# STARTING FROM SCRATCH



<https://quarkus.io/guides/>

The screenshot shows the Quarkus website's 'Guides - Latest' section. At the top, there's a navigation bar with links for 'ABOUT', 'LEARN', 'COMMUNITY', 'START CODING', and a dropdown menu. Below the navigation is a search bar labeled 'Find a Guide' and a dropdown for 'Select Guides Version' set to '2.13 - Latest'. The main content area has a heading 'Guides - Latest' and a 'View Category' section with links like 'Getting Started', 'Core', 'Web', etc. To the right, there's a 'Quarkus Cheat Sheet' box with a download link and a note about the Red Hat Developers website. Below that are four cards: 'Getting Started', 'Be Guided Through First Application', 'Getting Started with Reactive', 'Building Native Executables', and 'Using our Tooling'.

# STARTING THE MIGRATION WITH SPRING COMPATIBILITY EXTENSIONS

```
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import com.acme.todo.domain.TodoEntity;

@Repository
public interface TodoRepository
    extends JpaRepository<TodoEntity, Long> {
}
```

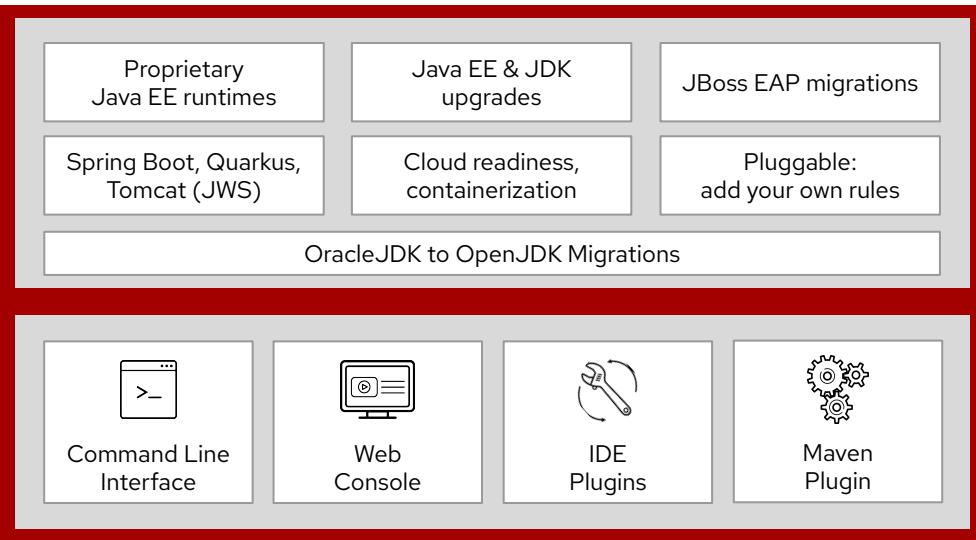
- Dependency injection
- Spring Web/MVC
- Spring Data JPA
- Spring Data REST
- Spring Security
- Spring Cache
- Spring Boot Configuration Properties
- Spring Scheduled
- Spring Cloud Config Client



This is intended to help with migration, not for long-term production use.

# Migration Toolkit for Applications

"Simplifies the Migration of Spring Apps to Quarkus"



<https://developers.redhat.com/products/mta>

- Automate application analysis
- Estimate level of effort
- Accelerate code transformation & migration
- Includes rules for DI, metrics, security, web, shell, & more

# DEMO

<https://red.ht/spring-to-quarkus-todo>



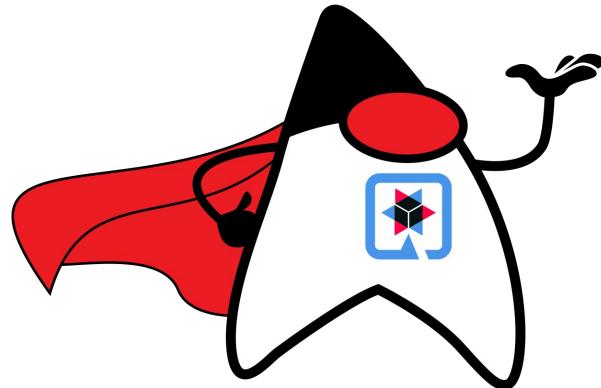


Quarkus World Tour 2022 { }

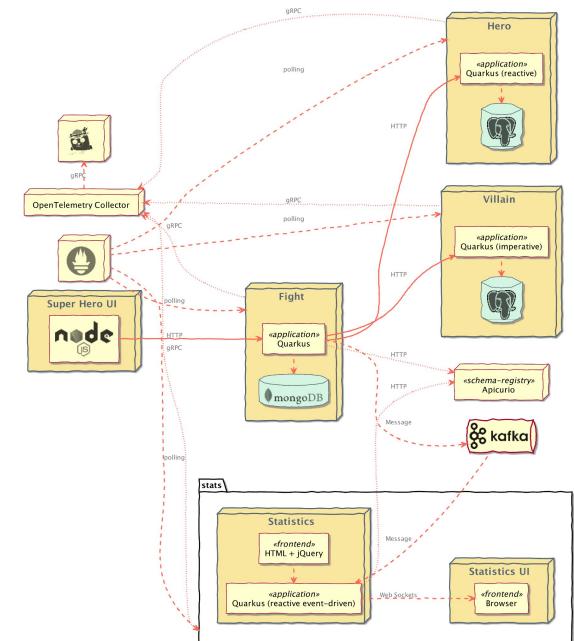


<https://red.ht/quarkus-spring-devs>

# Quarkus Superheroes



<https://github.com/quarkusio/quarkus-super-heroes>



# DEVELOPER SANDBOX

Add

Select a way to create an Application, component or service from one of the options.

## Getting started resources ?

■ Create applications using samples

Choose a code sample to get started creating an application with.

Basic Quarkus →

Basic Spring Boot →

[View all samples](#)

■ Build with guided documentation

Follow guided documentation to build applications and familiarize yourself with key features.

Get started with Quarkus using s2i →

Get started with Spring →

[View all quick starts](#)

Details on



■ Explore new developer features

Explore new features and resources within the developer perspective.

Discover certified Helm Charts →

Start building your application quickly in topology →

[What's new in OpenShift ↗](#)

- 30 days
- Limited access
- 4 vCPU, 8 GB RAM
- 3 projects
- Cloud IDE, Pipelines



[red.ht/dev-sandbox](http://red.ht/dev-sandbox)



## Quick Starts

Learn how to create, import, and run applications on OpenShift with step-by-step instructions and tasks.

←

Status ▼

### Developer Catalog

■ All services

Browse the catalog to discover, deploy and connect to services

### Database

Browse the catalog to discover database services to add to your Application

### Operator Backed

Browse the catalog to discover and deploy operator managed services

### Helm Chart

Browse the catalog to discover and install Helm Charts

### Managed Services

Discover managed services to simplify deployments and reduce operational overhead & complexities

### Git Repository

❖ From Git

Import code from your Git repository to be built and deployed

❖ From Devfile

Import your Devfile from your Git repository to be built and deployed

❖ From Dockerfile

Import your Dockerfile from your Git repository to be built and deployed

### Container images

Deploy an existing Image from an Image registry or Image stream tag

❖ Samples

Create an Application from a code sample

### From Local Machine

❖ Import YAML

Create resources from their YAML or JSON definitions

❖ Upload JAR file

Upload a JAR file from your local desktop to OpenShift



Binding your Quarkus application to Streams for Apache Kafka  
10 minutes

Binding your Quarkus application to the OpenShift Streams for Apache Kafka cloud service

Prerequisites (3) 1



Get started with Quarkus using a Helm Chart  
10 minutes

Deploy a Quarkus application using a Helm Chart.



Get started with Quarkus using s2i  
10 minutes

Import a Quarkus Application from git, build, and deploy it onto OpenShift.

Thank You!



<https://yrodierie.github.io/presentation/2022-10-06-jug-nancy-intro-to-quarkus/>

-  <https://quarkus.io>
-  <https://quarkusio.zulipchat.com>
-  [@quarkusio](https://twitter.com/quarkusio)
-  <https://youtube.com/c/quarkusio>
-  [@yoannrodiere](https://twitter.com/yoannrodiere)



Quarkus World Tour 2022 { } 