QUARKUS

QUARKUS WORLD TOUR 2021

Tour Stop:

JUG Saxony

bit.ly/quarkus-worldtour-jam

Thanks for hosting us

bit.ly/quarkus-worldtour-jam

Here’s how it’s going to work tonight...

bit.ly/quarkus-worldtour-jam

line up

intros

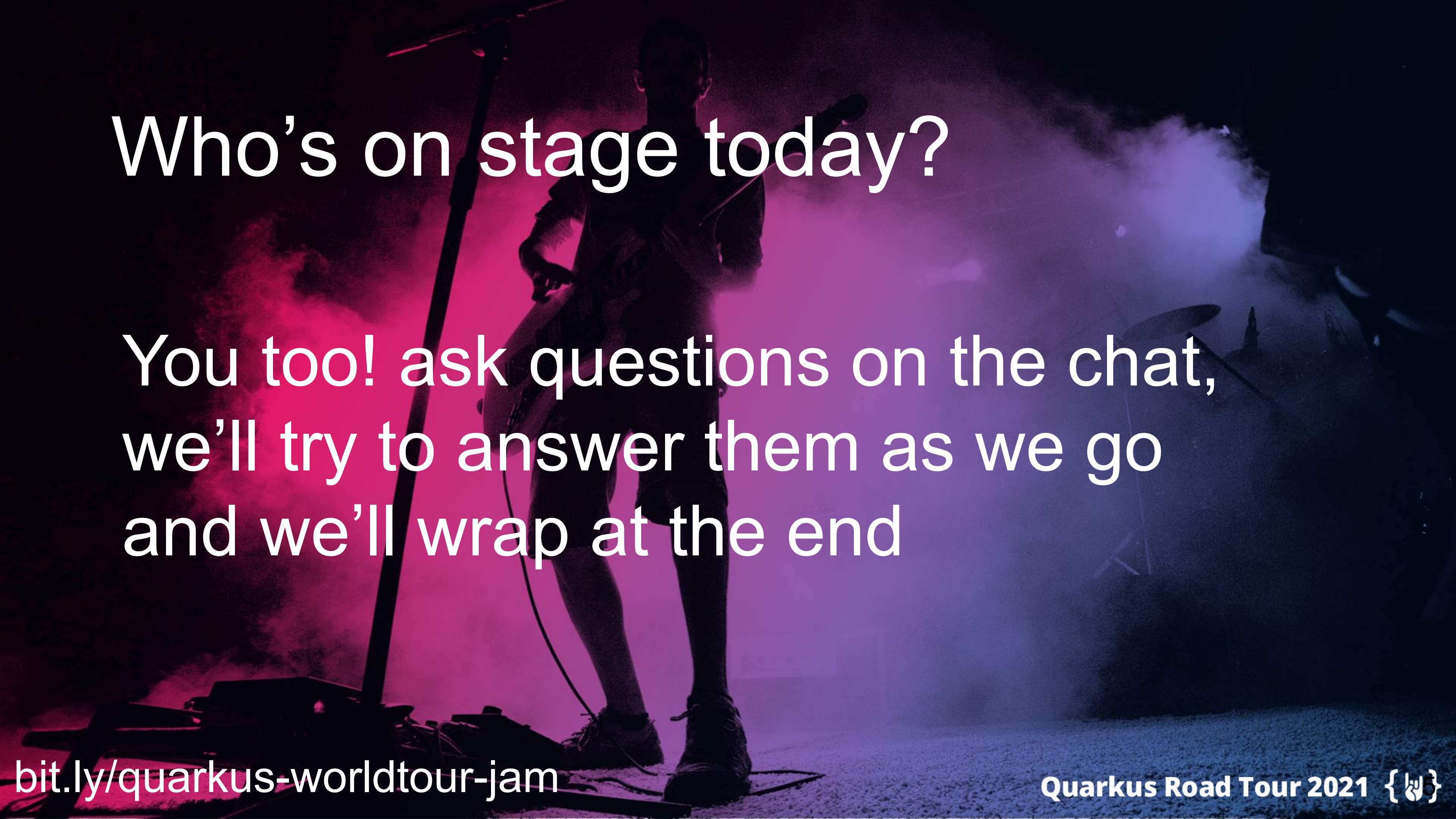
A taster demo

More of an structured introduction to Quarkus

Code Jam: follow along demos and examples with Q&A as we go

wrap up and last questions

bit.ly/quarkus-worldtour-jam



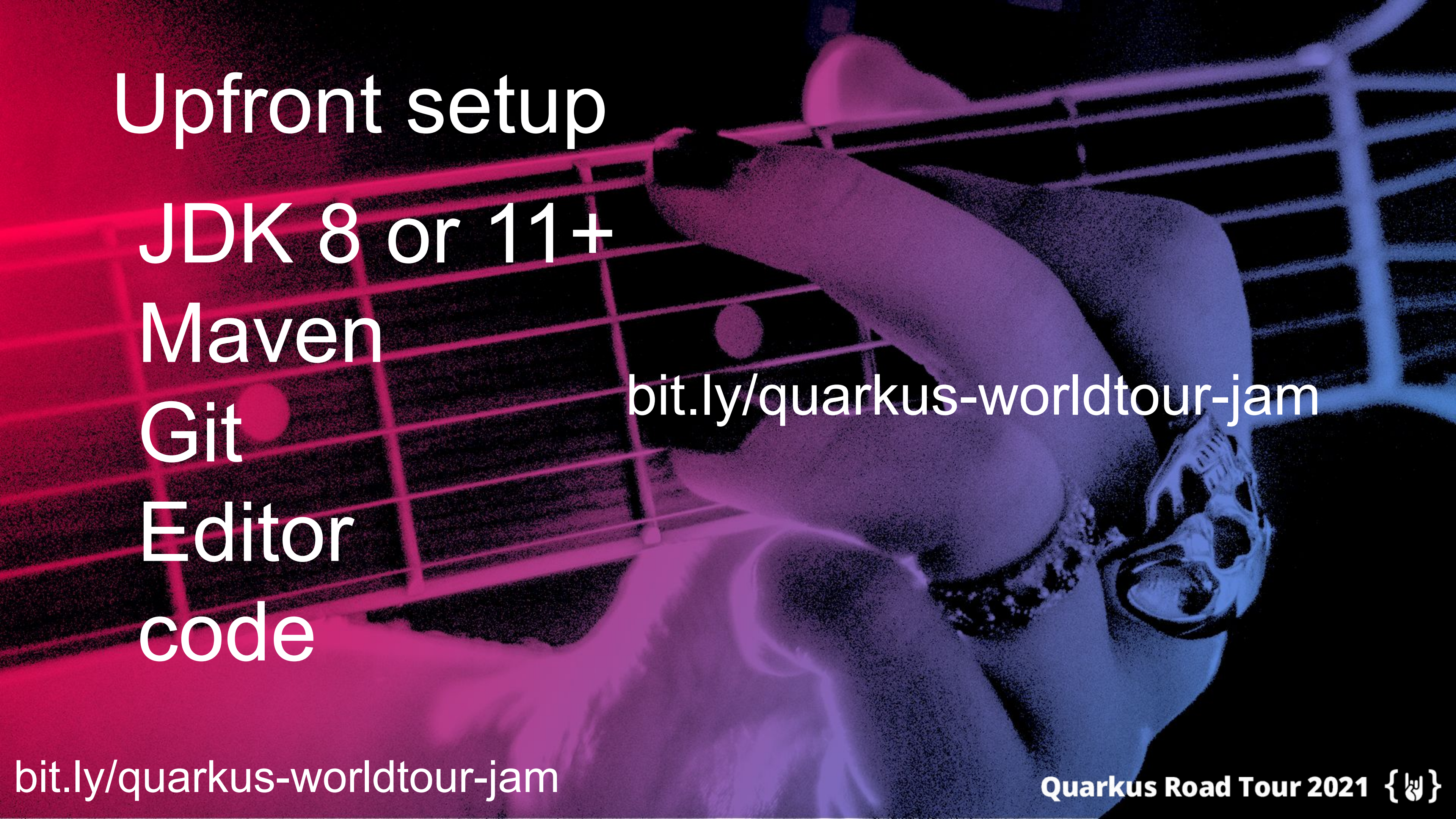
Who’s on stage today?

You too! ask questions on the chat, we’ll try to answer them as we go and we’ll wrap at the end

Upfront setup

- JDK 8 or 11+
- Maven
- Git
- Editor
- code

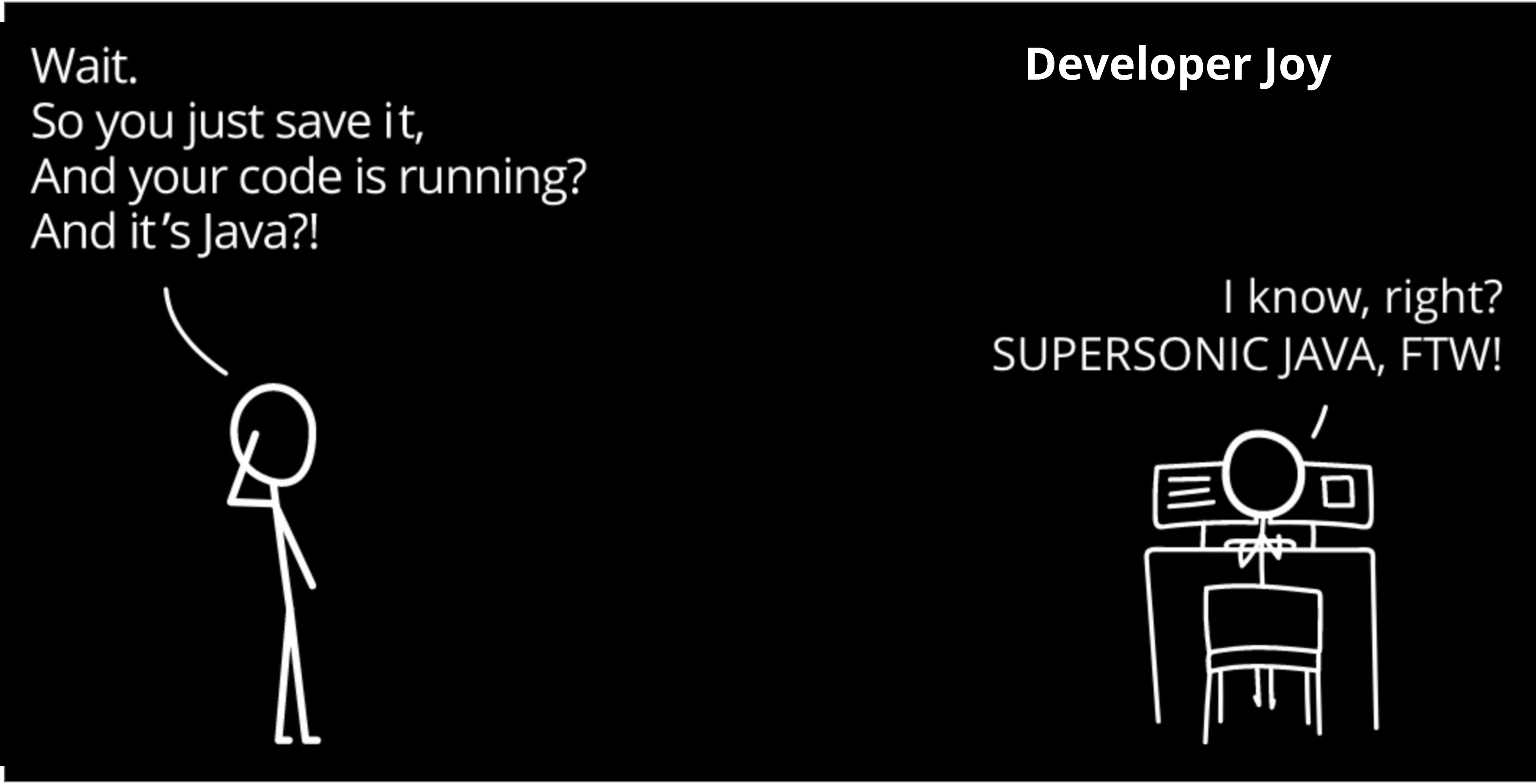
bit.ly/quarkus-worldtour-jam



QUARKUS

bit.ly/quarkus-worldtour-jam

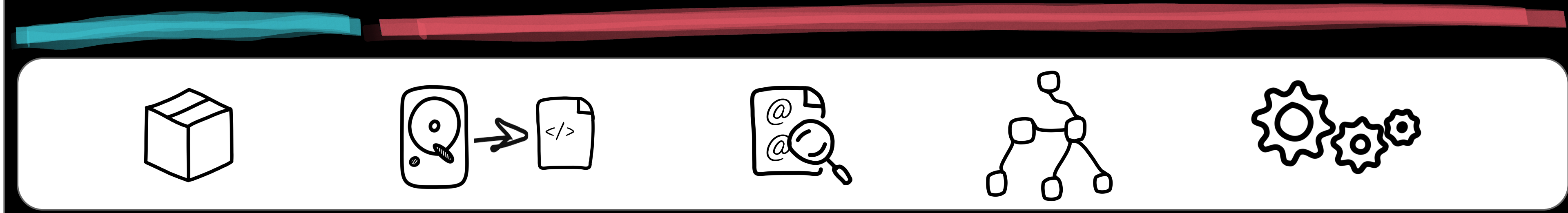
Developer Joy



How Does a Framework Start?

Build Time

Runtime



Packaging (maven,

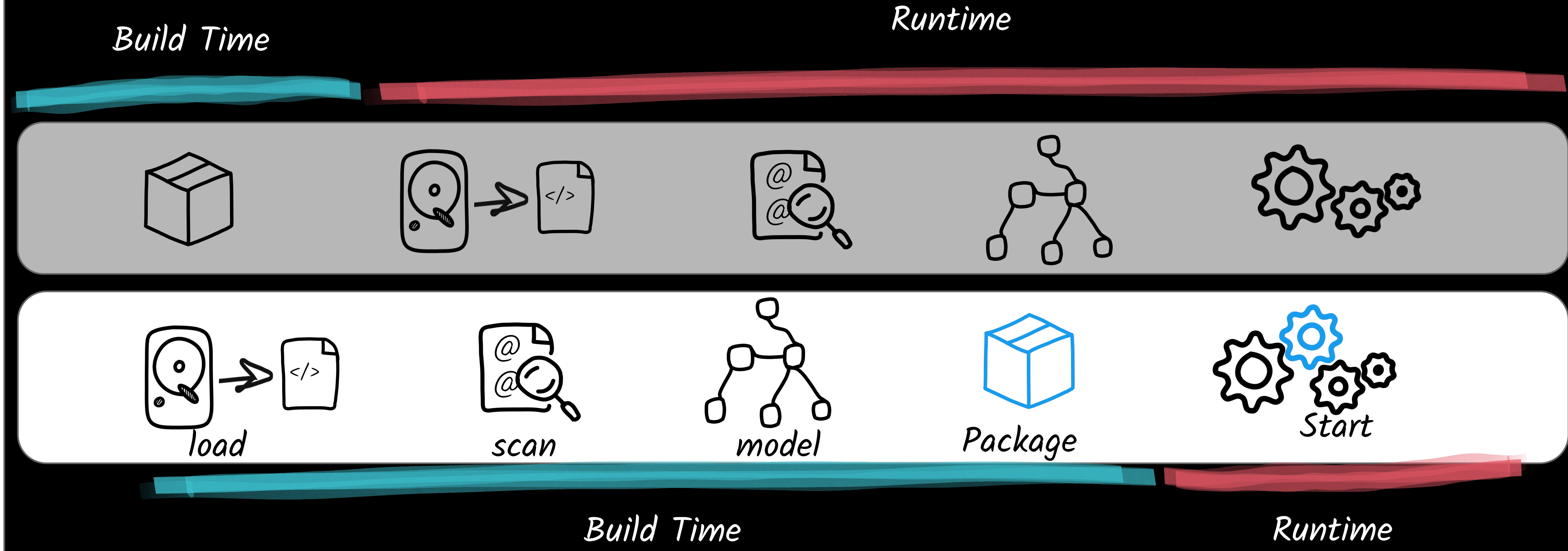
Load config file from file system Parse it

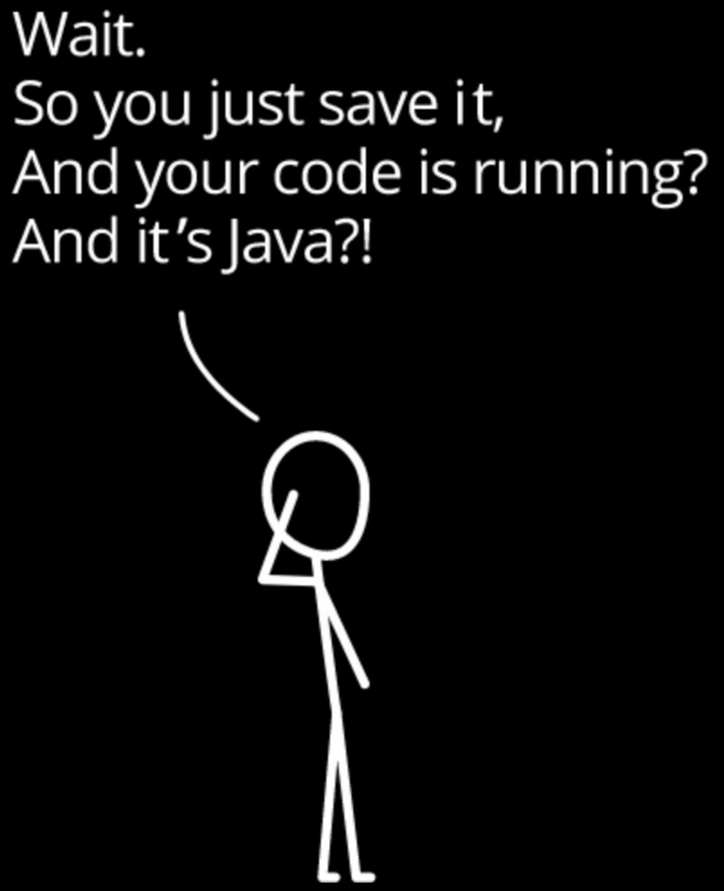
Classpath scanning to find annotated classes Attempt to load class to enable/disable features

Build its model of the world.

Start the management (thread, pool...)

The Quarkus Way





- Quarkus is instrumenting & adapting code for performance
- Let’s use this for better developer tooling!
- less boilerplate,
- better defaults
- container centric development approach

You’ve never seen Hot code replace like this !

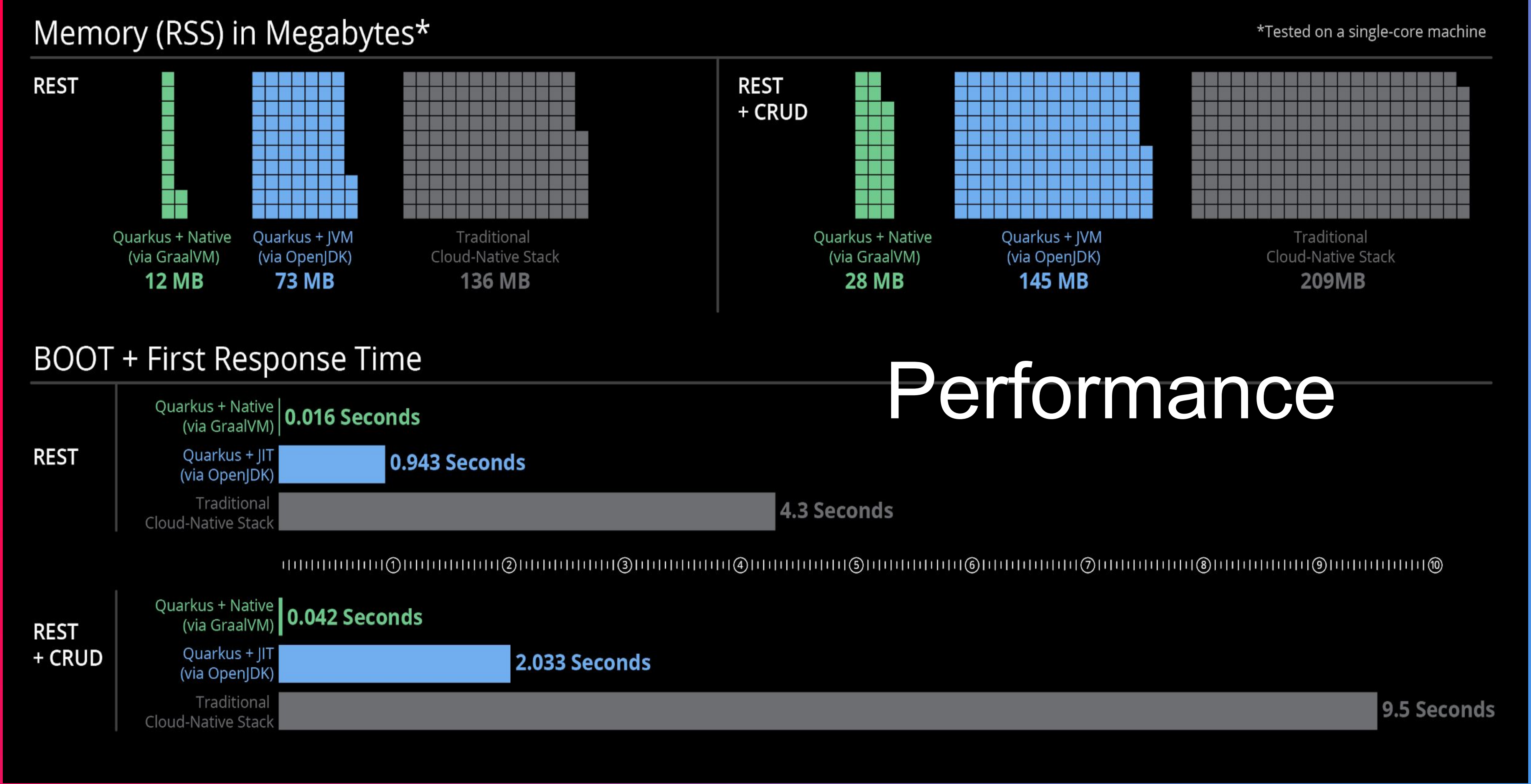
Three things:

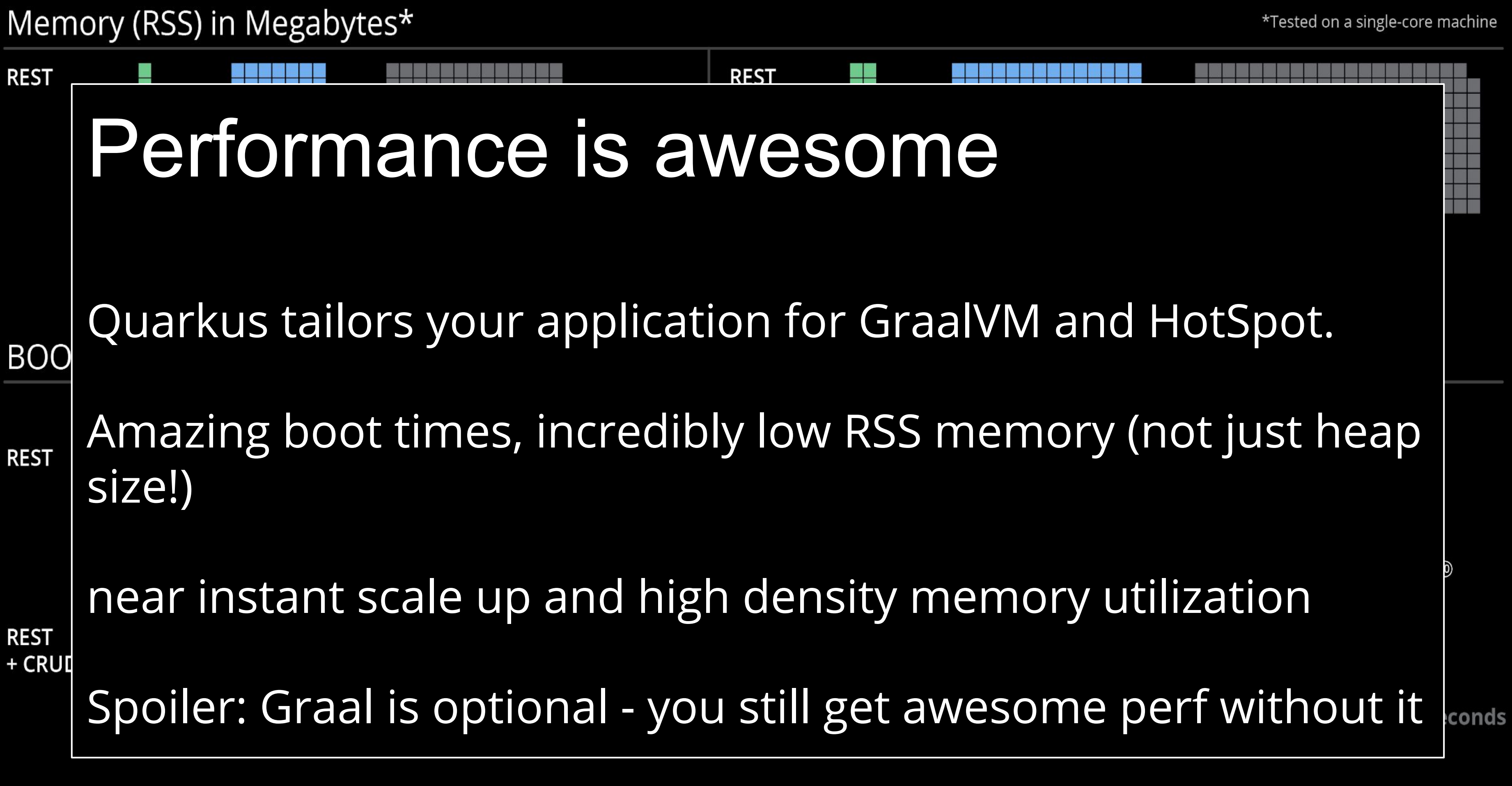
Performance

Capabilities

Community

Performance





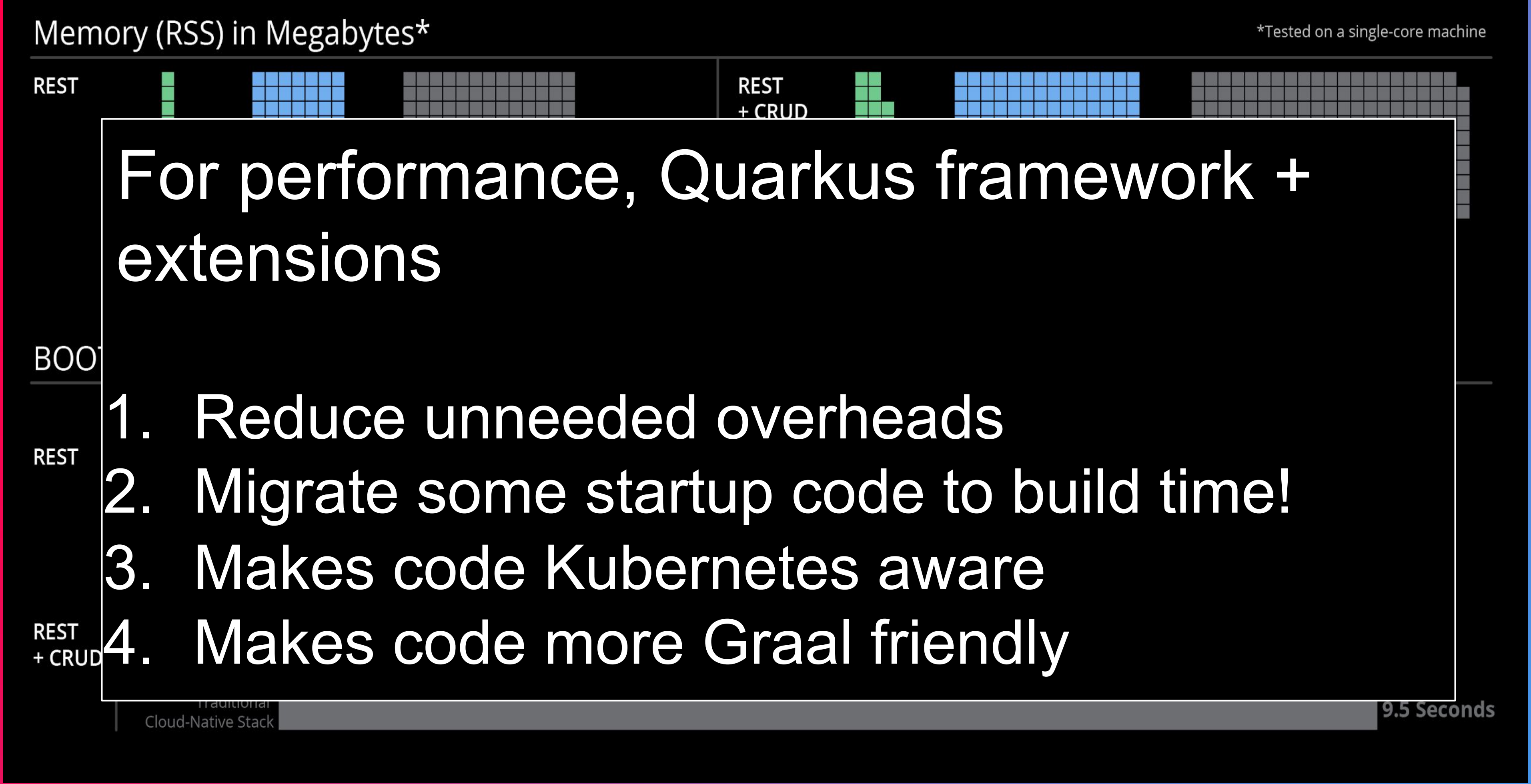
Performance is awesome

Quarkus tailors your application for GraalVM and HotSpot.

Amazing boot times, incredibly low RSS memory (not just heap size!)

near instant scale up and high density memory utilization

Spoiler: Graal is optional - you still get awesome perf without it



For performance, Quarkus framework + extensions

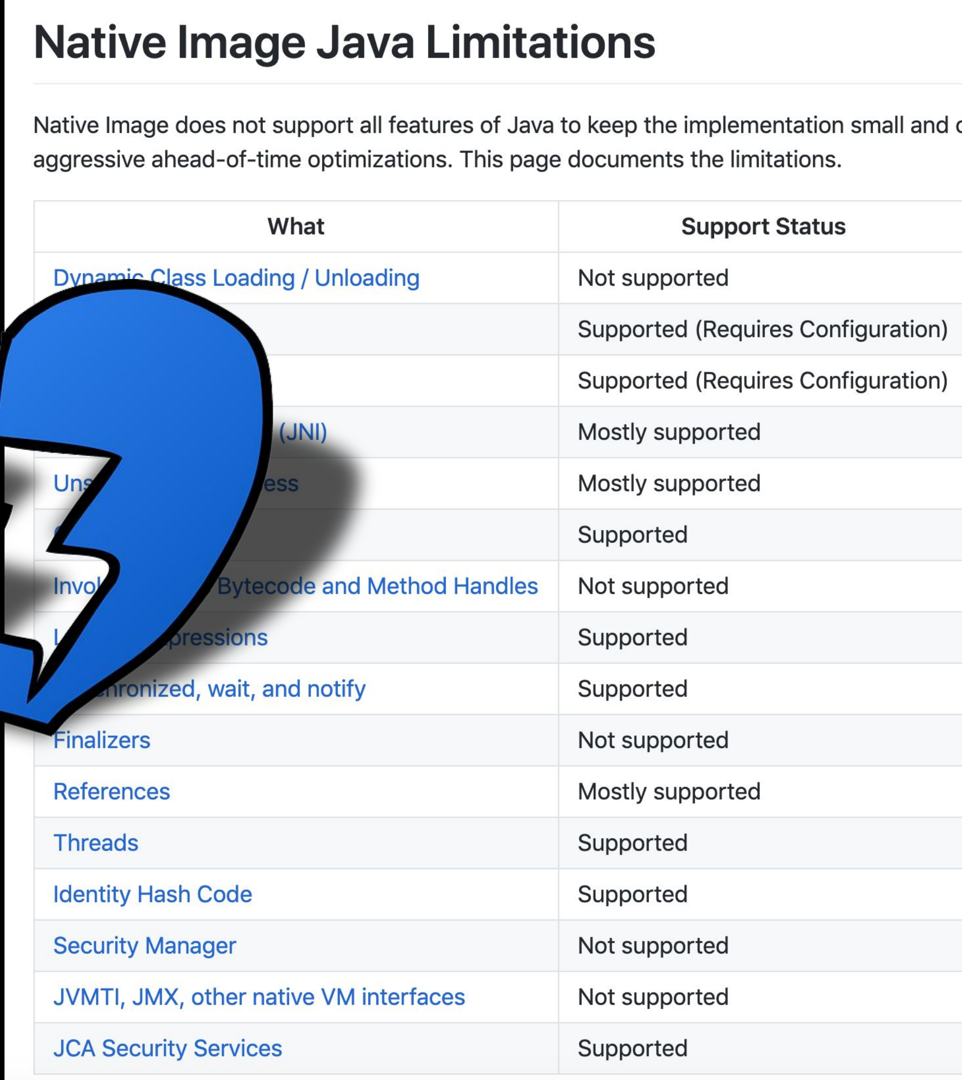
1. Reduce unneeded overheads
2. Migrate some startup code to build time!
3. Makes code Kubernetes aware
4. Makes code more Graal friendly

GraalVM

Standard Java is a challenge.

Java's dynamic, open view is at odds with an AOT approach

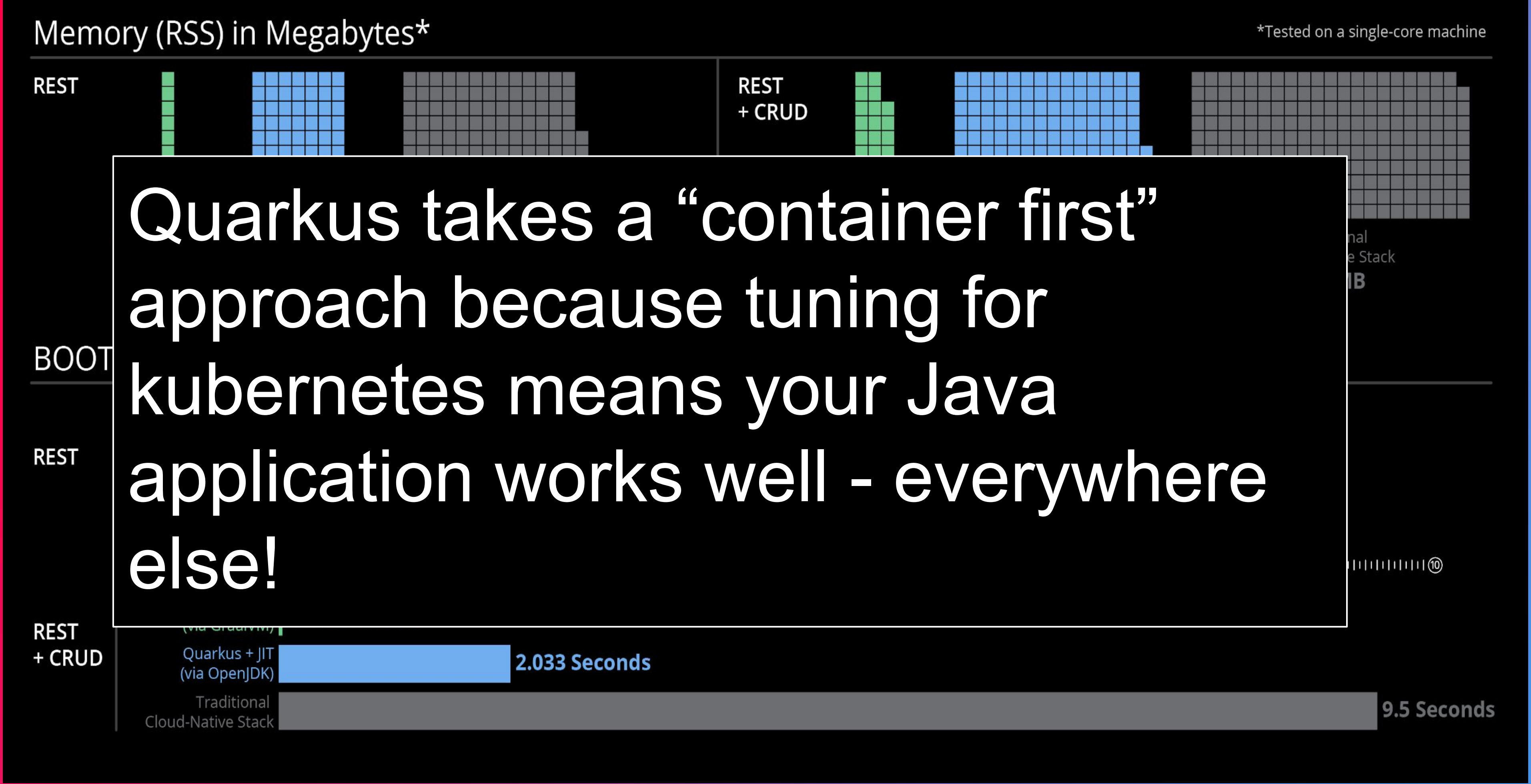
Quarkus makes using Graal so much easier



Native Image Java Limitations

Native Image does not support all features of Java to keep the implementation small and c aggressive ahead-of-time optimizations. This page documents the limitations.

What	Support Status
Dynamic Class Loading / Unloading	Not supported
[illegible]	Supported (Requires Configuration)
[illegible]	Supported (Requires Configuration)
[illegible] (JNI)	Mostly supported
Uns [illegible] ess	Mostly supported
[illegible]	Supported
Invol [illegible] Bytecode and Method Handles	Not supported
L [illegible] pressions	Supported
[illegible] hronized, wait, and notify	Supported
Finalizers	Not supported
References	Mostly supported
Threads	Supported
Identity Hash Code	Supported
Security Manager	Not supported
JVMTI, JMX, other native VM interfaces	Not supported
JCA Security Services	Supported



Quarkus takes a “container first” approach because tuning for kubernetes means your Java application works well - everywhere else!

Three things:

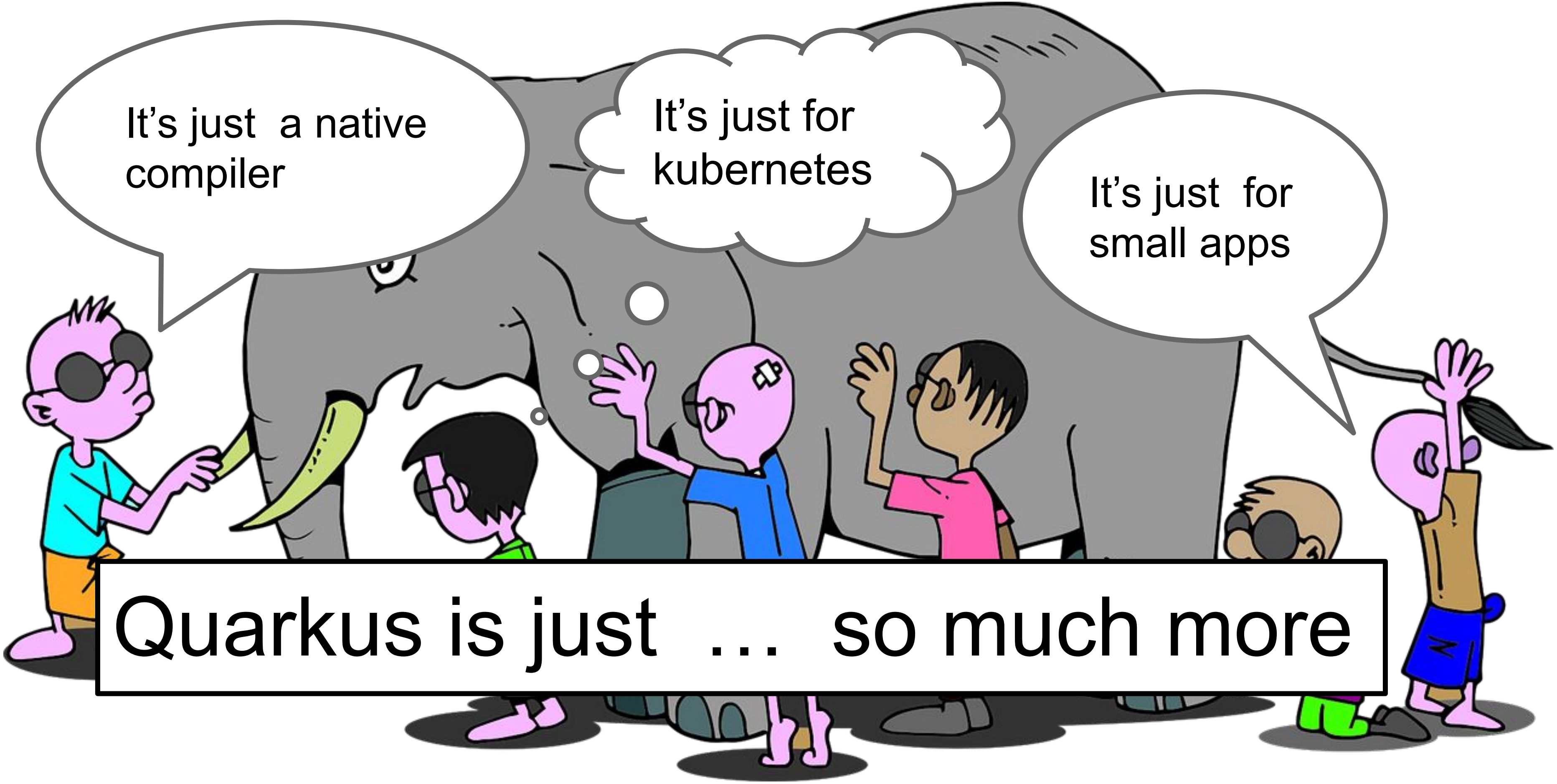
~~Performance~~

Capabilities

Community

You may have heard

Quarkus is just ...



Quarkus is just … so much more

quarkus.io

Supersonic Subatomic Java

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

Comprehensive

A cohesive, fun to use, full-stack framework

leveraging a growing list of over 100 best-of-breed libraries that you love and use.

All wired on a standard backbone

Quarkus is not

Just a native compiler - it’s optional - use it where it makes sense. It’s much easier with Quarkus

Just for new apps - Quarkus is applicable to all forms of Java application - new, old, web based , desktop. Even CLI’s

Just for small apps - Quarkus scales across the range

A Spring-lite copy - Quarkus lets you use almost all your code and dependencies and all of your skills. The only differences are where it’s essential for the cloud

Just another framework!

So what is Quarkus?

Quarkus is a comprehensive, inclusive, community driven, open source, practical reimaging of Java for the Cloud era

Quarkus takes all your frameworks, all your dependencies, all your code and helps make them work together in the best performing way for the Cloud

Quarkus improves the developer experience too - by using the same techniques to give you a much more productive toolkit for cloud application development

Bonus: The performance and productivity improvements work for your existing code - even if its not moving to the cloud

Quarkus Extensions

The jewel of Quarkus is the mechanisms for making your dependencies just work in a container environment

Whether adding container awareness, performance improvements, wiring into kubernetes services (or other clouds) etc

Even the hard work of using Graal is made easier (and often invisible) since the extension framework generates the config, adapter code etc at build time.

There are core extensions that are part of Quarkus
There are many more that are community created and owned.

Agroal

Amazon Alexa

Amazon DynamoDB

Amazon IAM

Amazon KMS

Amazon S3

Amazon SES

Amazon SNS

Amazon SQS

AMQP JMS client

Apache Avro

Apache Camel ActiveMQ

Apache Camel AMQP

Apache Camel ArangoDb

Apache Camel AS2

Apache Camel Async HTTP Client (AHC)

Apache Camel Async HTTP Client (AHC) Websocket

Apache Camel AtlasMap

Compute Cloud (EC2)

Apache Camel AWS 2 Elastic Container Service (ECS)

Apache Camel AWS 2 Elastic Kubernetes Service (EKS)

Apache Camel AWS 2 Eventbridge

Apache Camel AWS 2 Identity and Access Management (IAM)

Apache Camel AWS 2 Key Management Service (KMS)

Apache Camel AWS 2 Kinesis

Apache Camel AWS 2 Lambda

Apache Camel AWS 2 Managed Streaming for Apache Kafka (MSK)

Apache Camel AWS 2 MQ

Apache Camel AWS 2 S3 Storage Service

Apache Camel AWS 2 Security Token Service (STS)

Apache Camel AWS 2 Simple Email Service (SES)

Artemis JMS

AWS Lambda

AWS Lambda HTTP

AWS Lambda HTTP V1 (deprecated)

AWS Lambda X-Ray

Azure Functions HTTP

Blaze-Persistence

Container Image Docker

Container Image Jib

Container Image OpenShift

Container Image S2I

DataStax Apache Cassandra client

Debezium Quarkus Outbox

Eclipse Vertx GraphQL

Elasticsearch REST client

Elasticsearch REST High Level Client

Elytron Security JDBC

Elytron Security LDAP Realm

Three things:

~~Performance~~
~~Capabilities~~
Community

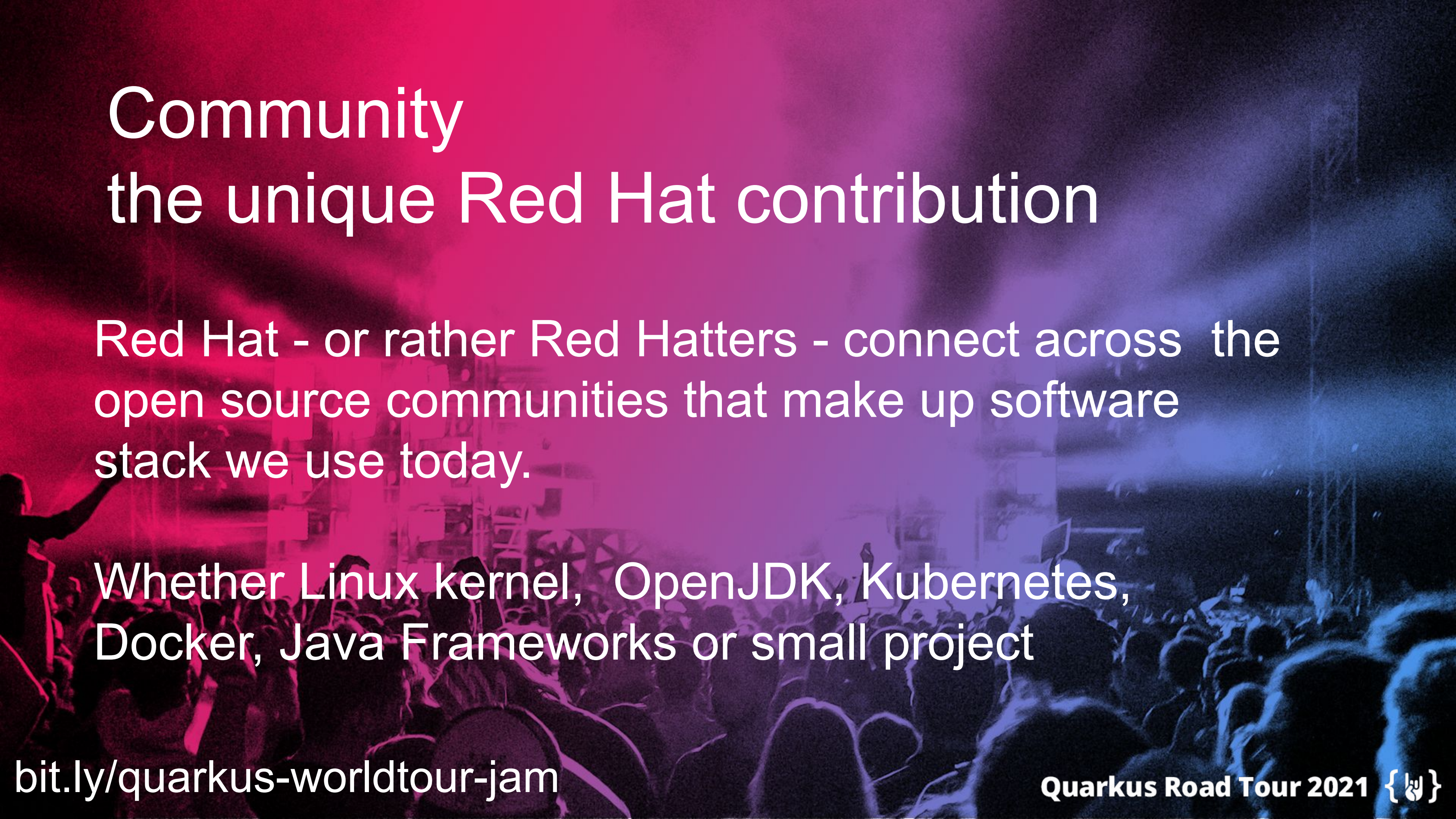
bit.ly/quarkus-worldtour-jam

Community

over 450 committers:
Inclusive, open, practical, pragmatic, inventive. Not all from Red Hat!

But that’s just Quarkus.

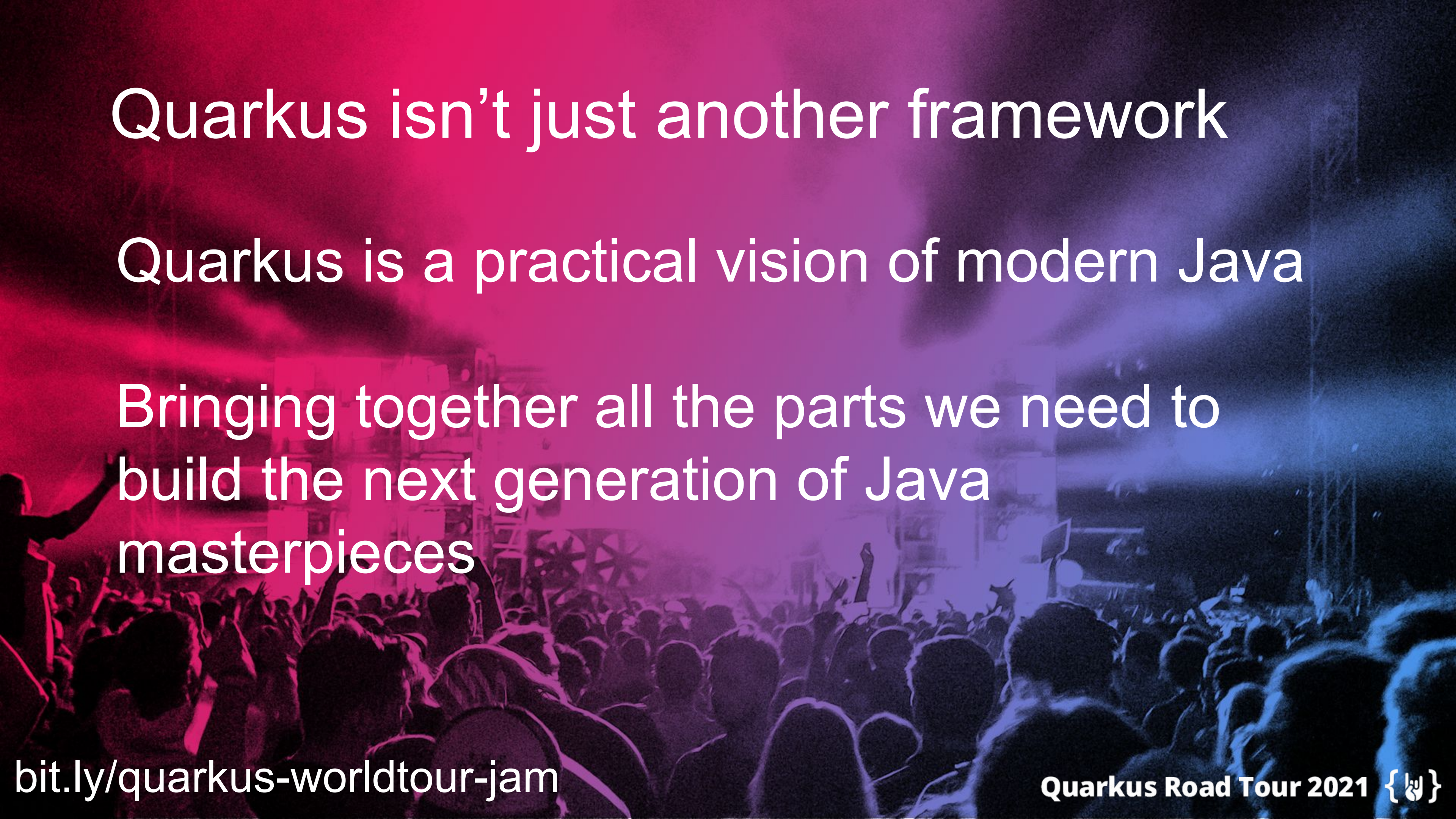
*



Community
the unique Red Hat contribution

Red Hat - or rather Red Hatters - connect across the open source communities that make up software stack we use today.

Whether Linux kernel, OpenJDK, Kubernetes, Docker, Java Frameworks or small project



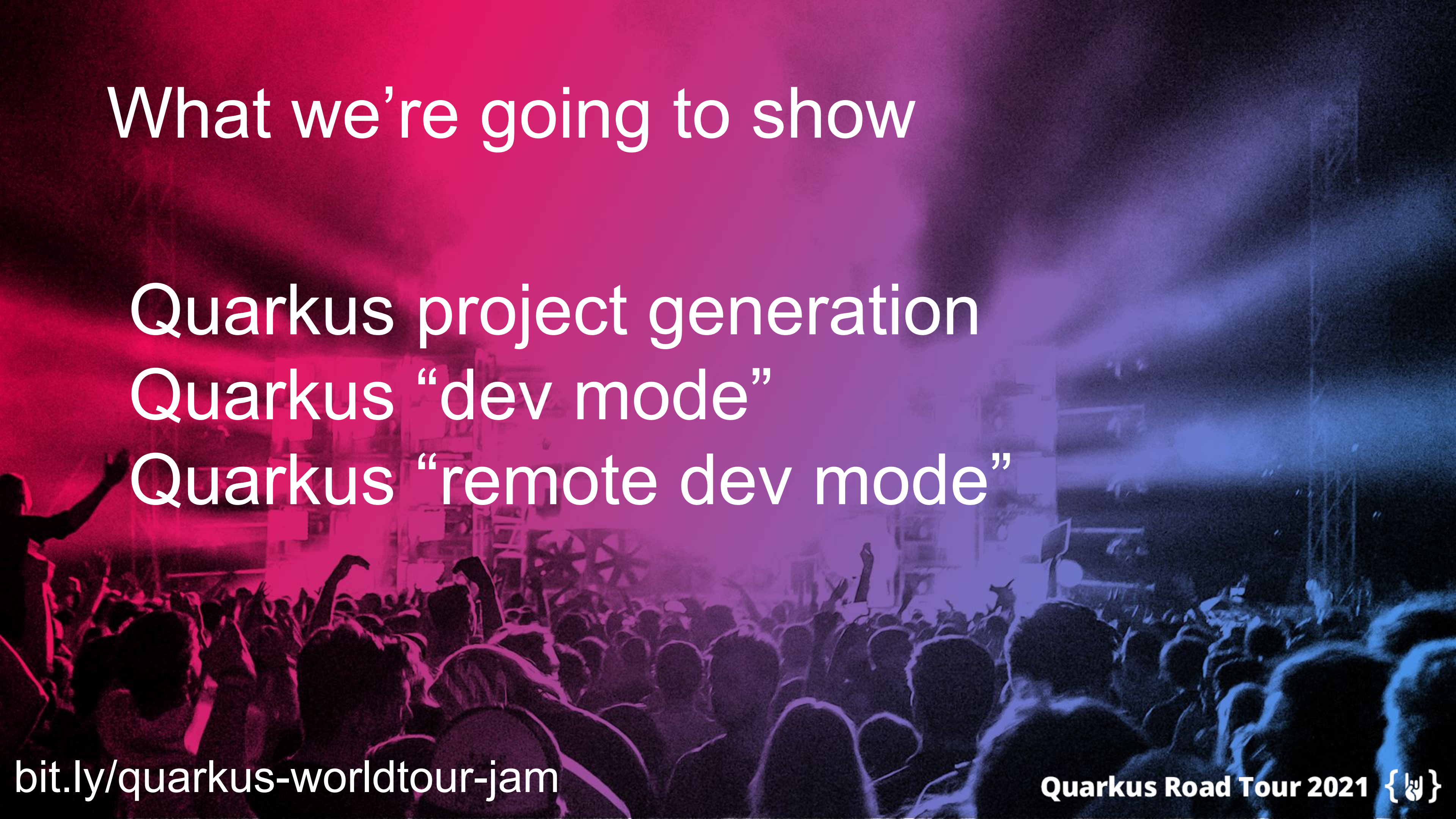
Quarkus isn’t just another framework

Quarkus is a practical vision of modern Java

Bringing together all the parts we need to build the next generation of Java masterpieces

bit.ly/quarkus-worldtour-jam

bit.ly/quarkus-worldtour-jam



What we’re going to show

Quarkus project generation
Quarkus “dev mode”
Quarkus “remote dev mode”

bit.ly/quarkus-worldtour-jam

Q&A

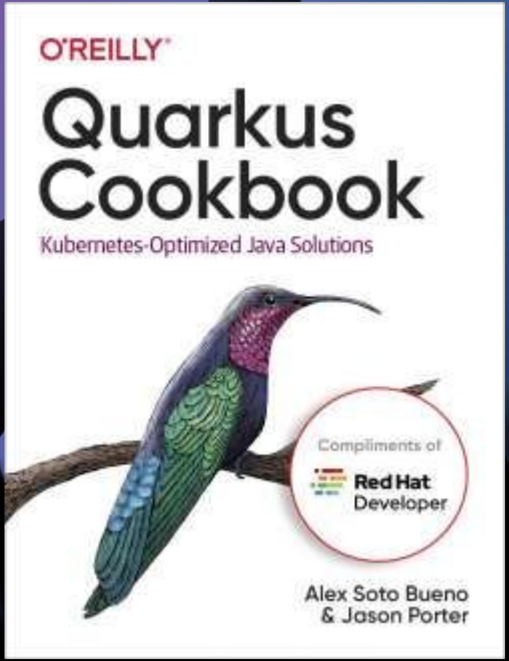
bit.ly/quarkus-worldtour-jam

Had a good time?
let everyone know!
#quarkusrocks

Want to know more?
Links are in the jam doc below

Web	quarkus.io
Talk	quarkusio.zulipchat.com
Twitter	@quarkusio
tour	worldtour@quarkus.io
More	quarkus.io/worldtour/soundcheck
book	bit.ly/q-book

QuarkusIO
@QuarkusIO



O'REILLY

Quarkus Cookbook

Kubernetes-Optimized Java Solutions

Compliments of
Red Hat Developer

Alex Soto Bueno
& Jason Porter

bit.ly/quarkus-worldtour-jam

Swag is sent to your JUG leaders

Swag varies by country

Swag will be as awesome as we can make it !

bit.ly/quarkus-worldtour-jam

Three things:

Quarkus: accelerates your day-2-day Java development
Quarkus: makes all your Java applications supersonic
Quarkus: for Java everywhere

Next tour stop

bit.ly/quarkus-worldtour-jam