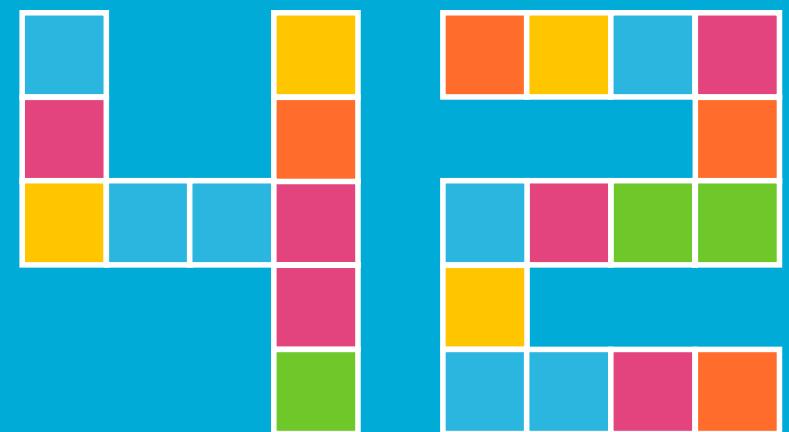


Soul Mates and Partners in Crime

What Quarkus and Spring Boot Can Learn from Each Other

Patrick Baumgartner
42talents GmbH, Zürich, Switzerland

@patbaumgartner
patrick.baumgartner@42talents.com



TALENTS

Abstract

Soul Mates and Partners in Crime

What Quarkus and Spring Boot Can Learn from Each Other

Discover the untapped potential of collaboration between Quarkus and Spring Boot in the realm of Java-based microservices and cloud-native development. While Quarkus excels in startup times and resource efficiency, Spring Boot offers a rich ecosystem and a vast developer community. This talk explores how these frameworks can complement each other, demonstrating how Quarkus' GraalVM-native capabilities can enhance Spring Boot applications, and vice versa. Join us to learn how Quarkus and Spring Boot can become soul mates and partners in crime, working together to push the boundaries of Java development.

Soul Mates and Partners in Crime

What Quarkus and Spring Boot Can Learn from Each Other

Patrick Baumgartner
42talents GmbH, Zürich, Switzerland

@patbaumgartner
patrick.baumgartner@42talents.com





Disclaimer

This presentation serves to clarify that any discussions or opinions expressed regarding Spring Boot or Quarkus are solely based on personal experience and expertise. No affiliation or endorsement with either team should be inferred. Any references made are purely for illustrative purposes within the context of technical discussions.

Introduction



Patrick Baumgartner

Technical Agile Coach @ 42talents

My focus is on the development of software solutions *with* humans.

Coaching, Architecture,
Development, Reviews, and
Training.

Lecturer @ Zurich University of Applied Sciences ZHAW

[@patbaumgartner](https://twitter.com/patbaumgartner)

Agenda

Agenda

- Quarkus and Spring Boot
- Performance
- Tooling
- Ecosystem
- Community
- Support
- Migration
- Final Thoughts
- Conclusion



Indicator	SpringBoot	Quarkus
Application startup time (sec)	~ 9	~ 7
Number of loaded classes	14 691	~ 37
Average CPU usage - moderate load (%)	2	1
Average CPU usage - high load (%)	4	7
Heap memory size (MB)	541.0	83.8
Heap memory used (MB)	398.5	31.0
Memory metaspace size (MB)	77.7	49.7
Memory metaspace used (MB)	76	48.8
Native threads	78	56
Average request latency (ms)	1	3.8

No Benchmarks!

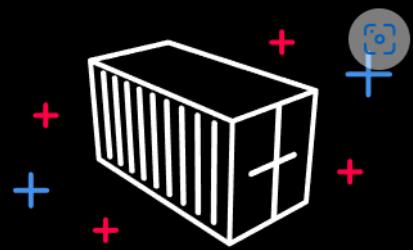
**Reviews, guides, and
comparisons are mostly based
on “Hello world” examples.**



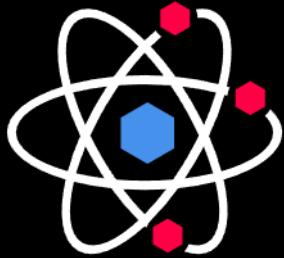




QUARKUS



Container First



Unifies imperative and
reactive



Community and Standards



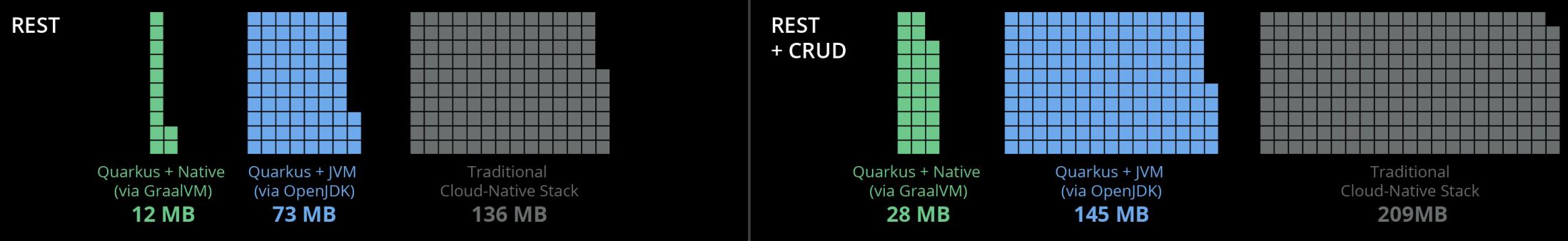
Kube-Native



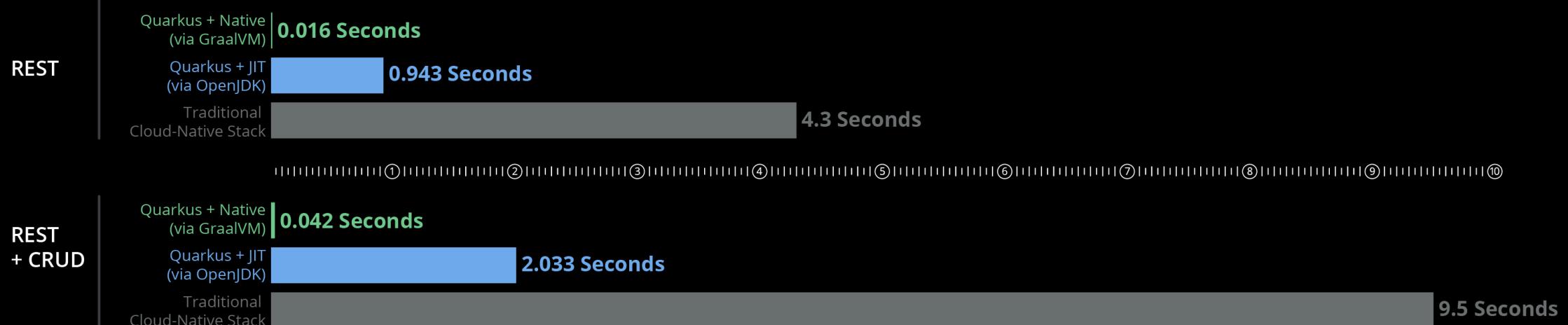
Developer Joy

Memory (RSS) in Megabytes*

*Tested on a single-core machine

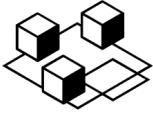


BOOT + First Response Time





What Spring can do



Microservices

Quickly deliver production-grade features with independently evolvable microservices.



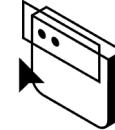
Reactive

Spring's asynchronous, nonblocking architecture means you can get more from your computing resources.



Cloud

Your code, any cloud—we've got you covered. Connect and scale your services, whatever your platform.



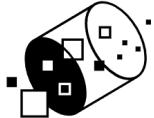
Web apps

Frameworks for fast, secure, and responsive web applications connected to any data store.



Serverless

The ultimate flexibility. Scale up on demand and scale to zero when there's no demand.



Event Driven

Integrate with your enterprise. React to business events. Act on your streaming data in realtime.



Batch

Automated tasks. Offline processing of data at a time to suit you.

Spring Boot

- 2014
- Based on Spring Framework (2004) and ecosystem
- Runtime reflection and dynamic bytecode generation
- Leverages ecosystem projects and third-party libraries
- Opinionated runtime & production-ready features

Quarkus

- 2019
- Based on Java/Jakarta EE technologies
- Compile-time approach & reactive core
- Leverages Quarkus Extensions and ecosystem projects
- Promises smaller artifacts and faster startup times
- Optimized for cloud, serverless and containerized environments

Compile Time

Startup Time

Memory Consumption

CPU Utilization

Spring Boot

- JVM Runtime

Quarkus

- AOT
- Graal VM Native Image
- JVM Runtime



JVM

Runtime

AOT

Build time

Native
Image

Build time
+ GraalVM
Native Images

Spring Boot 2.7

Spring Native

Spring Framework 6, Spring Boot 3, ecosystem projects

There is more ...

Coordinated Restore at Checkpoint (CRaC),

Application Class Data Sharing (AppCDS),

Project Leyden,

... and more Graal VM Native Image stuff.

Spring Boot

- Spring Boot Devtools
- Testcontainers integration
- Docker Compose support
- Buildpacks
- Version Management

Quarkus

- Developer Mode
- Testcontainers integration
- Dev Services
- Dev UI
- Containerisation
- Version Management

**The Quarkus Team
knows how to create a good
Developer Experience.**



Ecosystem

Spring Boot

- Spring Security
- Spring Data
- Spring Batch
- Spring Cloud
- Spring AI
- ... and many more

Quarkus

- Vert.x
- SmallRye
- Panache
- Camel
- Langchain4j
- ... and many more extensions

Both offer huge ecosystems.

Spring allows you to use any library and Quarkus prefers an extension. 

Community

Spring Boot

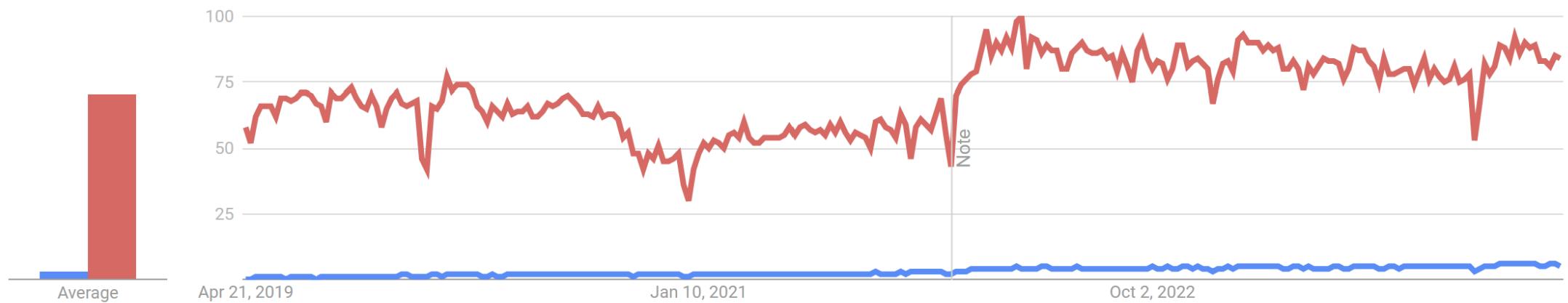
- Large Community
- High Popularity
- **150k** Questions on Stack Overflow
- Mature documentation
- Integrates Java/Jakarta EE Standards
- Spring integrates third-party libraries or vice versa

Quarkus

- Big Community
- Growing popularity
- **4.5k** Questions on Stack Overflow
- "Lighter" and more straightforward documentation
- Leverages Java/Jakarta EE and Microprofile
- Big Quarkiverse Extension catalog

Trend: Spring Boot vs Quarkus

Interest over time ?

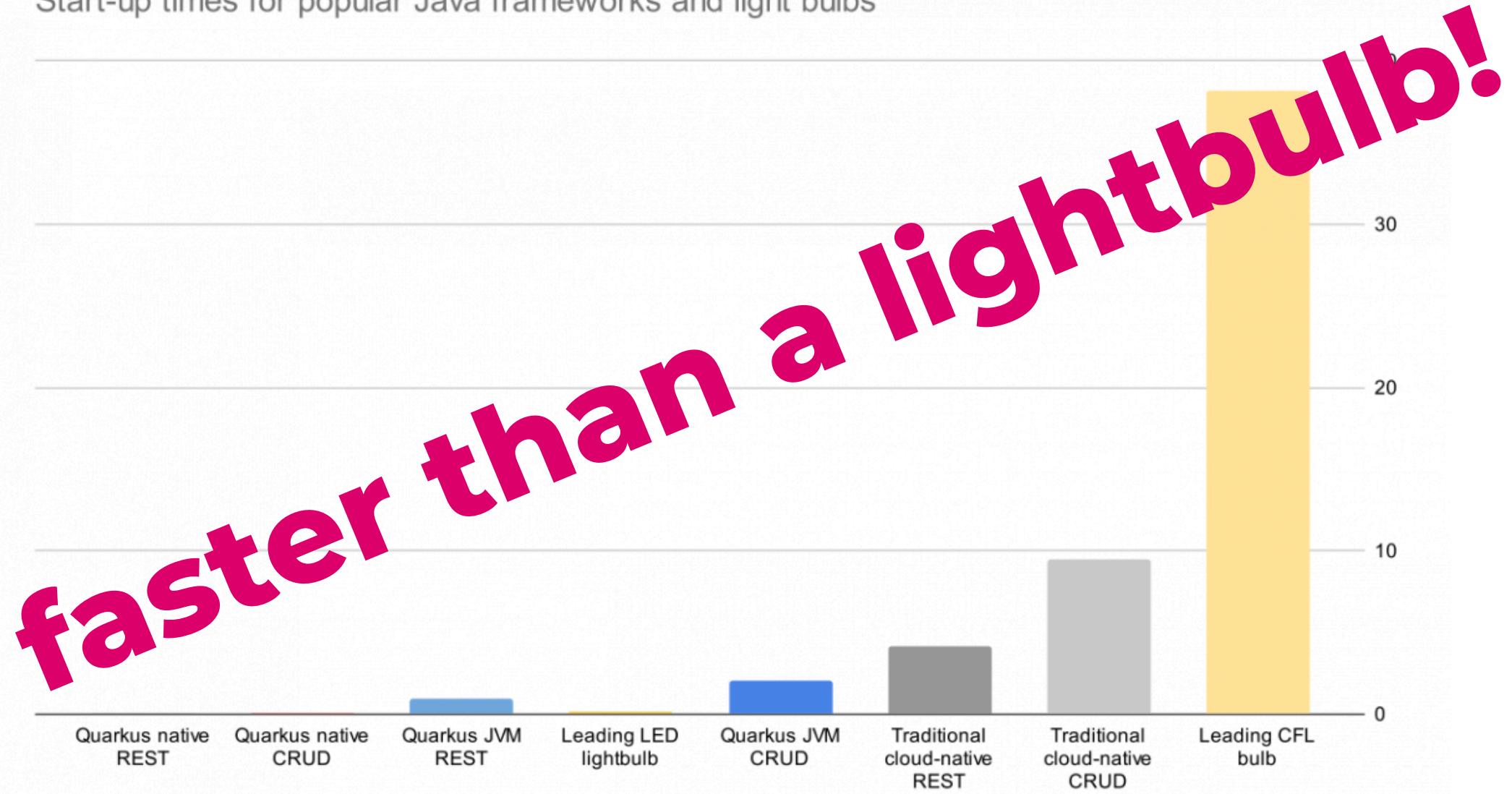


**The Spring Boot team
knows how to write good
documentation.**



**The Quarkus Team
does great marketing.** 

Start-up times for popular Java frameworks and light bulbs



Spring Boot

- Biannual minor releases
- Monthly bugfix releases
- Release branch for 1 year supported
- OSS and Enterprise Support

Quarkus

- "Monthly" minor releases
- Biannual Long-Term Support (LTS)
- Red Hat build of Quarkus (Subscription)

Not sure who does a better job.

Maby both? Who needs

support anyway?



Spring Boot

- Spring Boot Migrator
- Open Rewrite

Quarkus

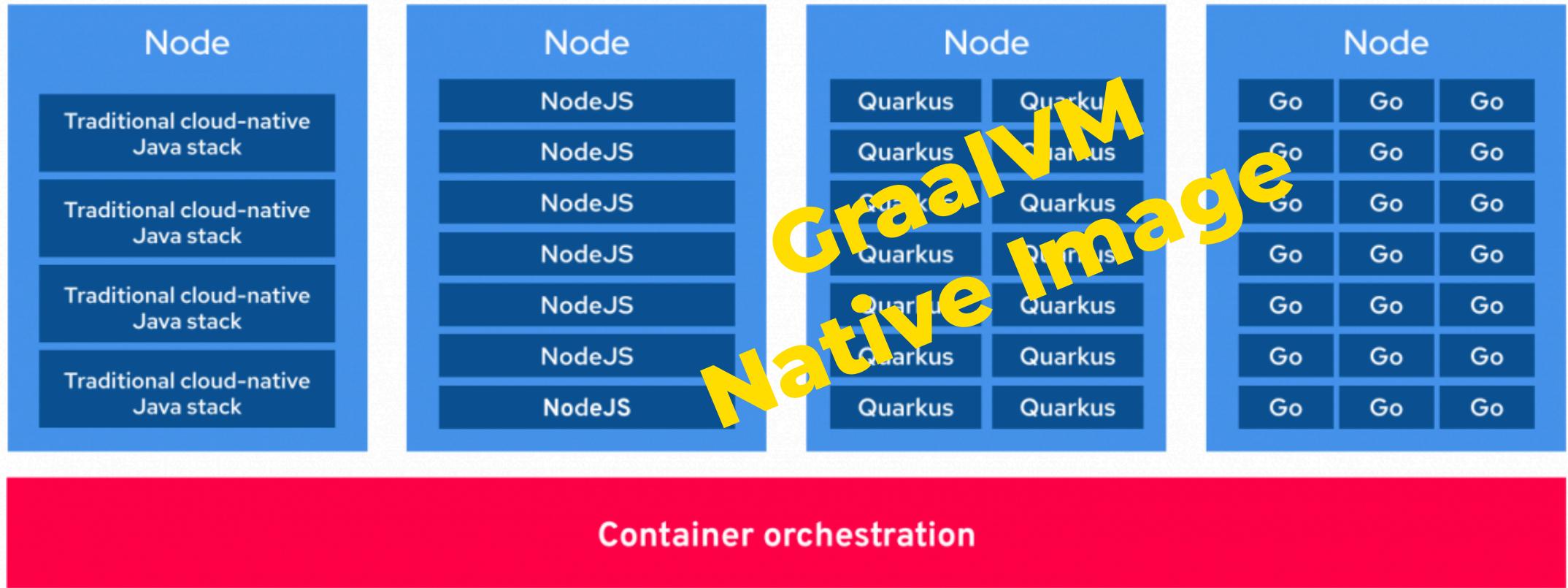
- Migration Toolkit for Applications (MTA)
- Windup
- Open Rewrite
- Compatibility Libraries

**Good News - There are tools to
support you!**



**Quarkus is based on Java/Java EE
standards. But your Spring Boot
application may use Spring Cloud,
Spring Security and Spring Data ...**

Developing Cloud Native Applications





The choice of programming language or framework is not critical to success. Don't panic! The development team will get it done.



**Response time is more
important than startup time.**



**Time to deployment is more
important than start-up time.**



**Native images are not always
the best choice.**



Both frameworks are catching up in areas where they lag, and competition will only benefit the developer community.





**Spring learns what works well.
It then adapts to it, but at a
much slower pace.**





Did I miss something? 

Let me/us know! 

... or not! 

Soul Mates and Partners in Crime

What Quarkus and Spring Boot Can Learn from Each Other

Patrick Baumgartner
42talents GmbH, Zürich, Switzerland

@patbaumgartner
patrick.baumgartner@42talents.com

