



Quarkus

Supersonic. Subatomic. Java

Harald Pehl

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



Harald Pehl

Senior Software Engineer at Red Hat

WildFly Management / HAL Console

Quarkus gRPC Extensions

Why Quarkus?

Another framework?

No thanks - I can't take
any more!

“““

FRANK HAYES ■ FRANKLY SPEAKING

Not Dead Yet

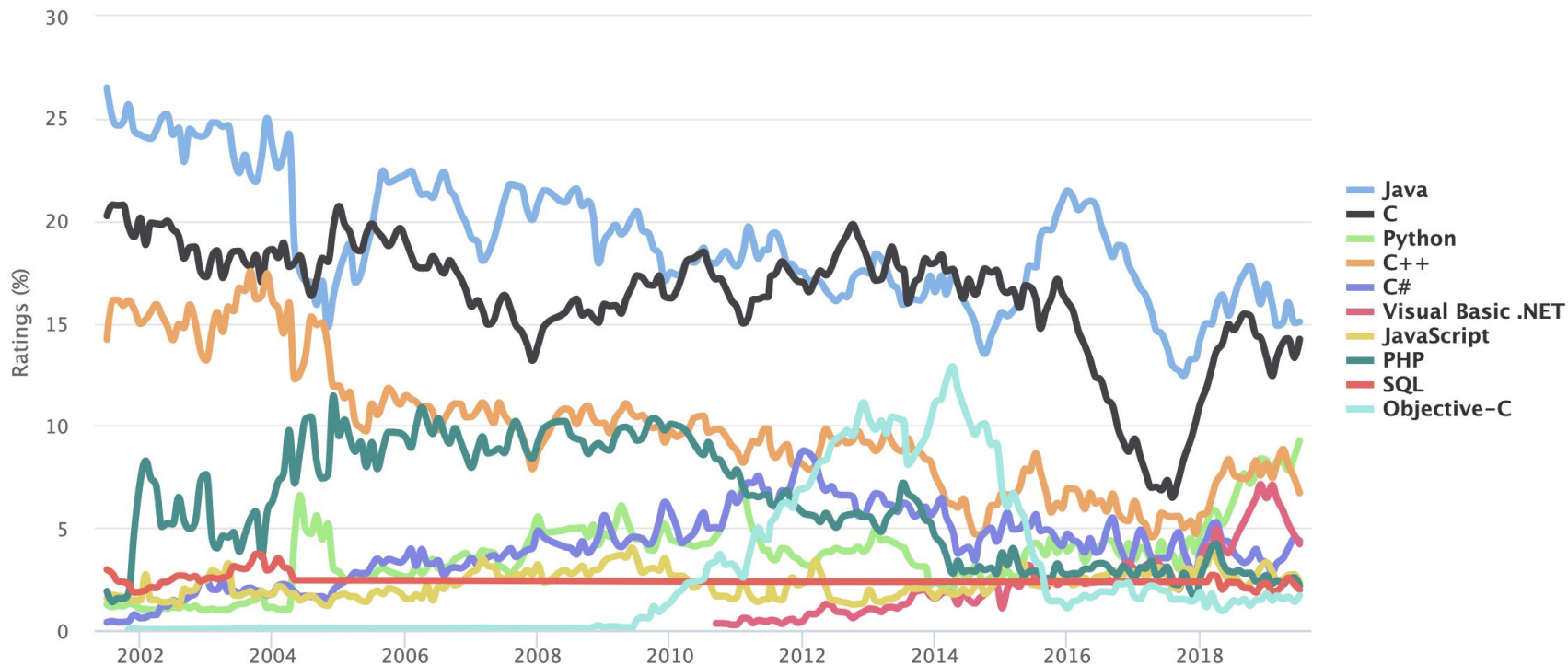
IS JAVA DEAD? Come on, seriously — why else would Sun Microsystems be offering it up to the open-source crowd? (See story, page 1.) A decade ago, Java was the hottest, most exciting thing in IT, a certified Windows-killer that was going to wipe out Microsoft's monopoly and revolutionize the way software was made, distributed and run. Today? Today, Java is old hat. It's been eclipsed by open-source, the *new* hottest thing in IT that's going to wipe out Microsoft's monopoly and revolutionize the way software is made, distributed and run.

Computer World

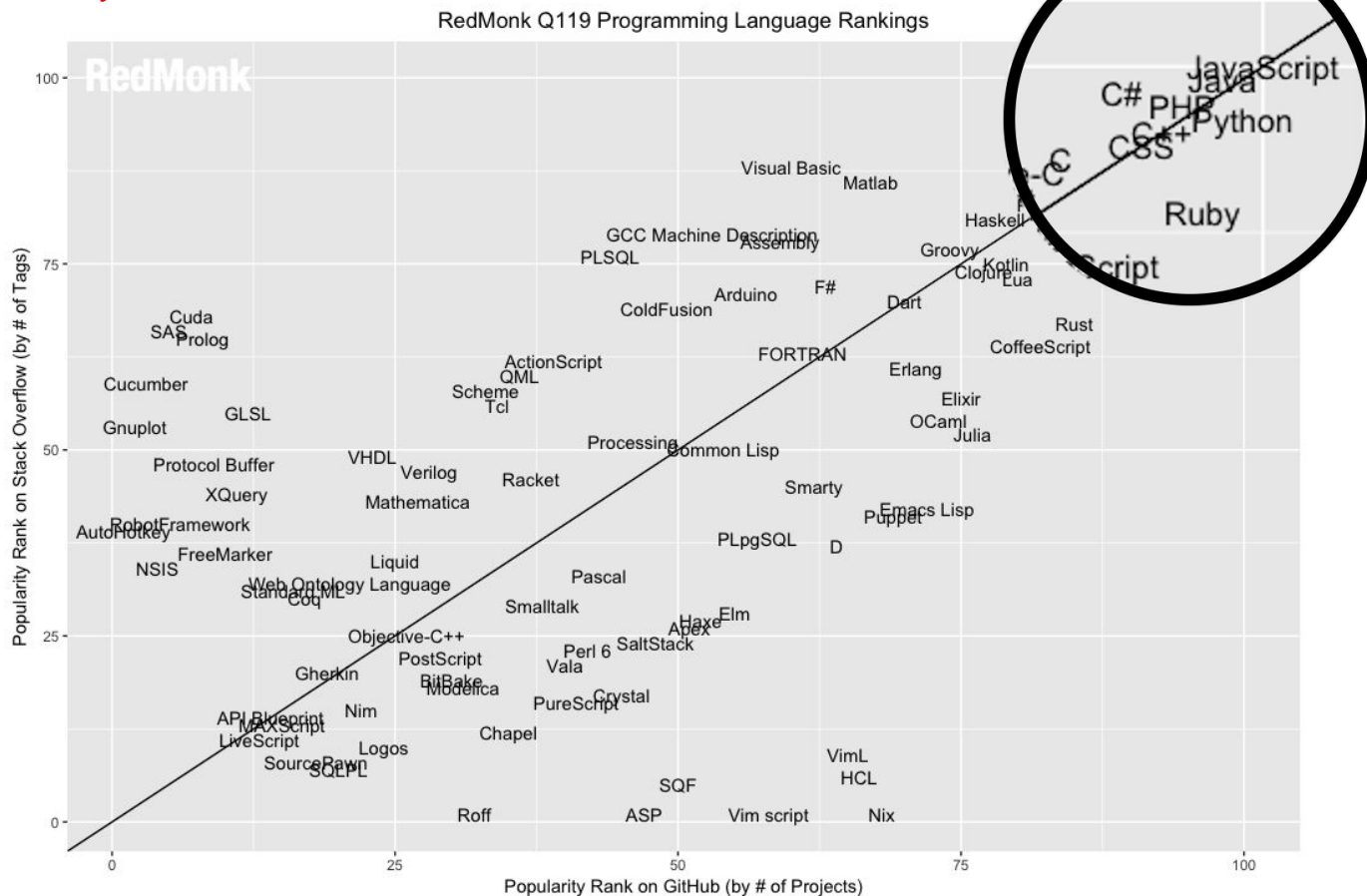
May 22, 2006

TIOBE Programming Community Index

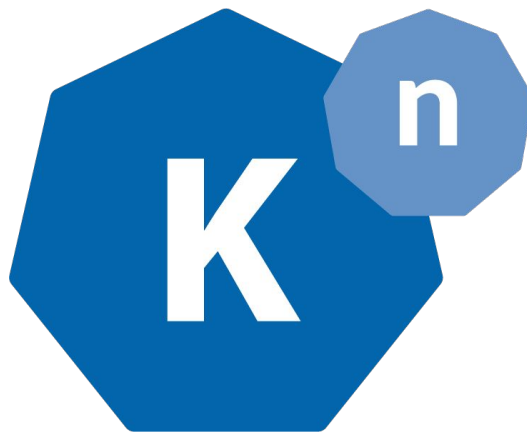
Source: www.tiobe.com

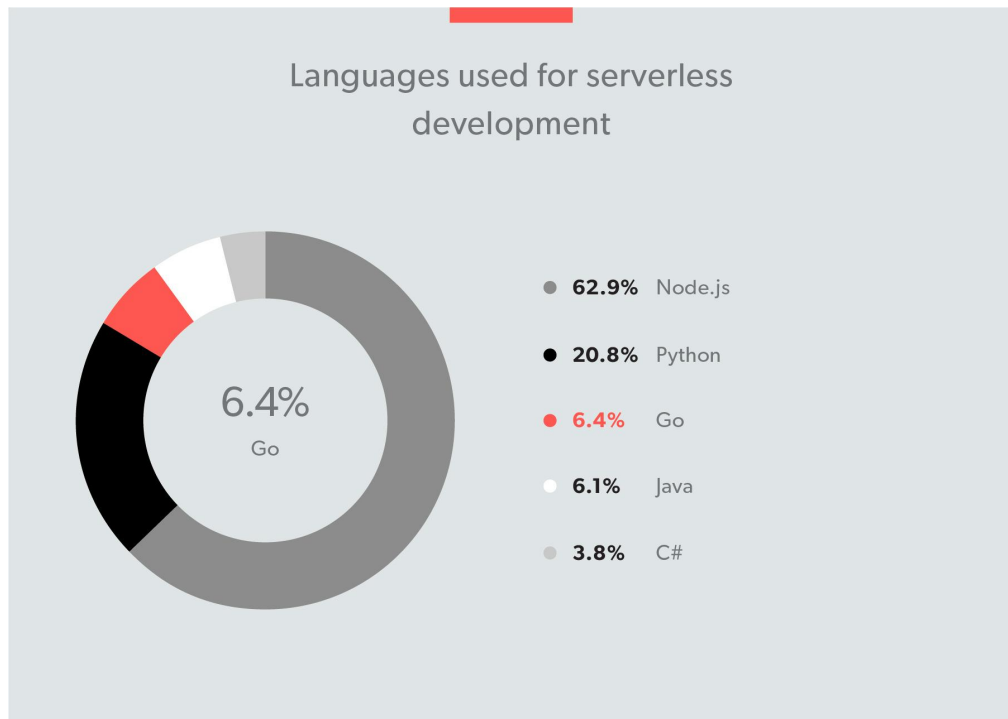


Background



1. JavaScript
2. Java
3. Python
4. PHP
5. C#
6. C++
7. CSS
8. Ruby
9. C
10. Objective-C

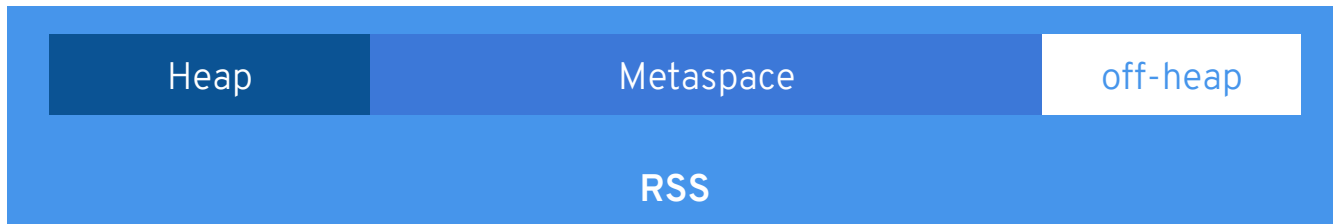




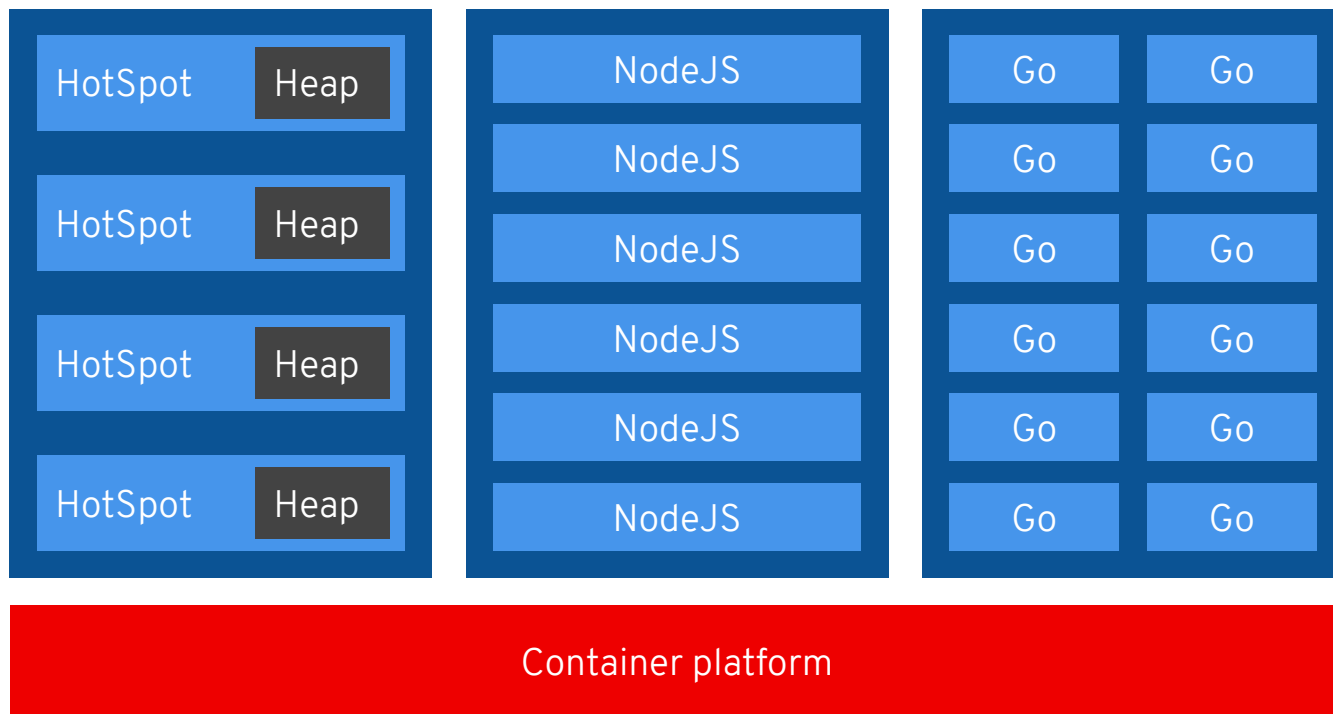
The hidden truth

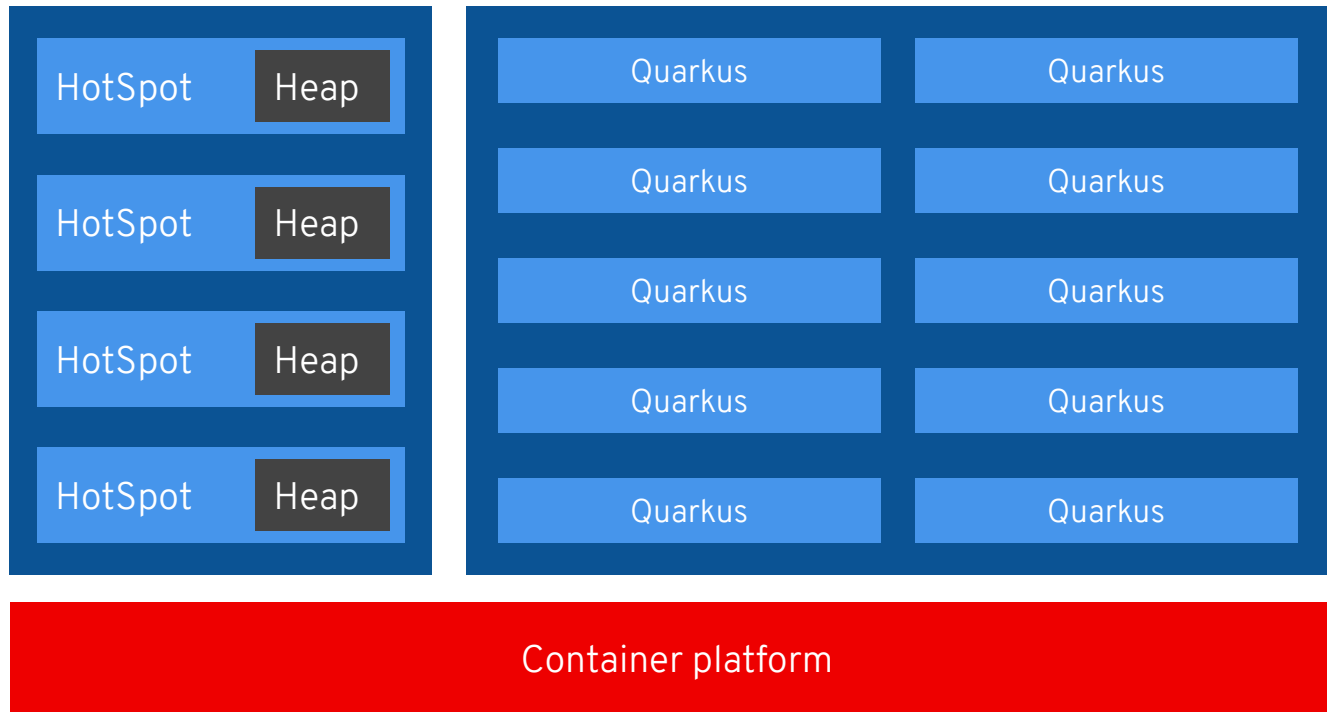
Startup Overhead: # of classes, bytecode, JIT

Memory Overhead: # of classes, metadata, compilation



<https://developers.redhat.com/blog/2017/03/14/java-inside-docker/>





What is Quarkus?

A look into the box

WTF is a Quarkus?

Quark - type of elementary particle and a fundamental constituent of matter

us - the hardest problem in software

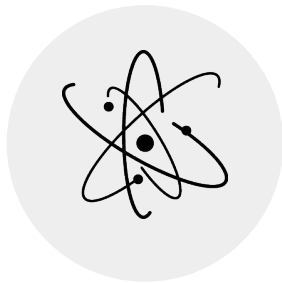
Supersonic. Subatomic. Java.

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



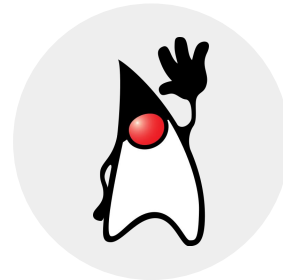
Supersonic

Fast.
Blazing fast to start.
Millisecond fast!



Subatomic

Improve memory consumption
Increase deployment density

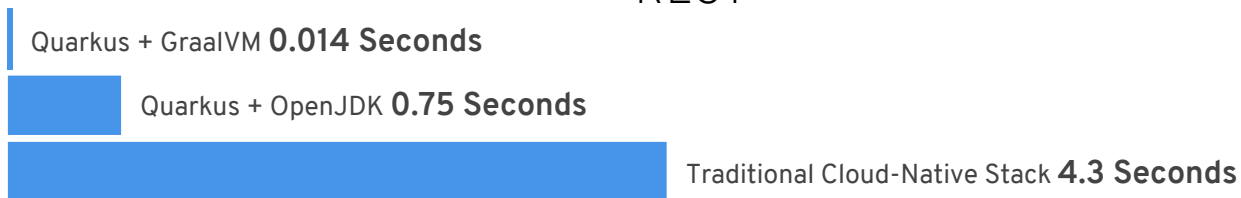


Java

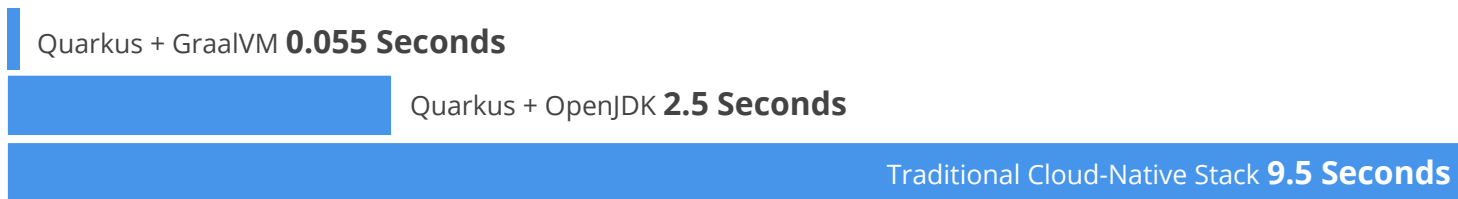
Based on a framework &
specs you love
Blazing fast-hot-reload

Time to first response

REST



REST + CRUD

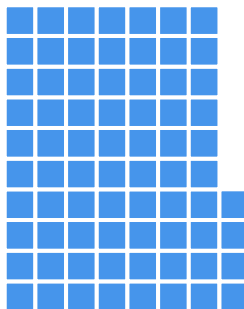


Memory (RSS)

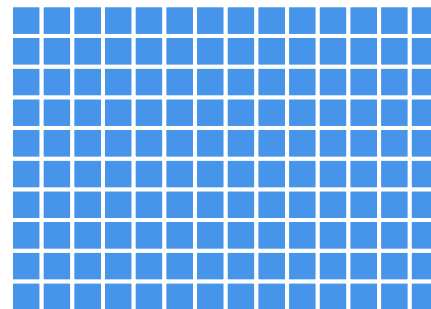
REST



Quarkus + GraalVM
13 MB



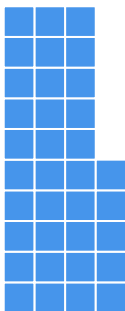
Quarkus + OpenJDK
74 MB



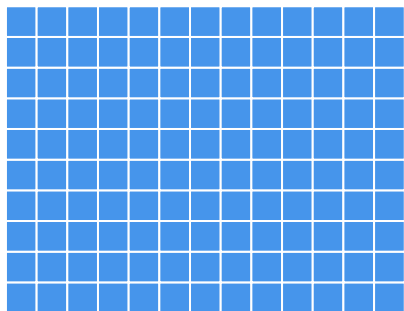
Traditional Cloud-Native Stack
140 MB

Memory (RSS)

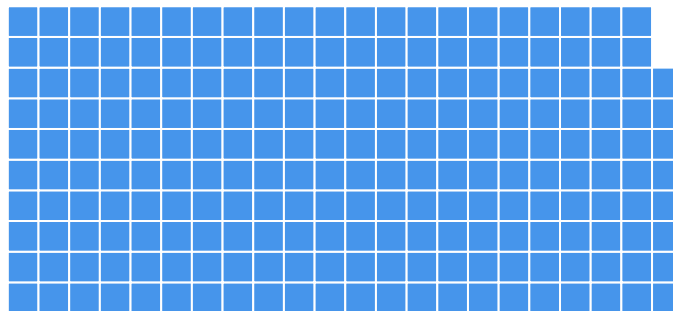
REST + CRUD



Quarkus + GraalVM
35 MB



Quarkus + OpenJDK
130 MB



Traditional Cloud-Native Stack
218 MB

<https://quarkus.io/guides/performance-measure>

Standards

Servlet
JAX-RS
JPA, JDBC
CDI
Bean Validation
Transactions
Logging



Fault Tolerance
Health
JWT
Metrics

OpenAPI
OpenTracing
Reactive
Messaging
Rest Client

Best of Breed



Eclipse Vert.x



Hibernate



RESTEasy



Apache Camel



Eclipse MicroProfile



Netty



Kubernetes



OpenShift



Jaeger



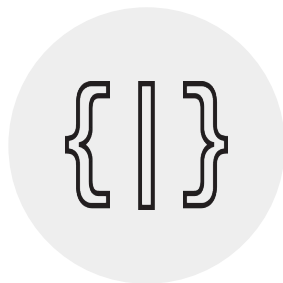
Prometheus



Apache Kafka



Infinispan



Developer Joy

Live reload
Java or Kotlin
Maven or Gradle



No Compromises

Serverless and Microservices
JVM Mode and Native Executables
Imperative and Reactive



Blazing Fast

Lower memory usage
Faster startup
Optimized for short-lived processes

A cohesive platform for developer joy

Based on standards, but not limited

Unified configuration

Zero config, live reload in the blink of an eye

Streamlined code for the 80% common

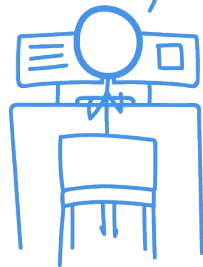
usages, flexible for the 20%

No hassle native executable generation

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



I KNOW, RIGHT?
SUPERSONIC JAVA, FTW!



Unifies

Imperative

```
@Inject
SayService say;

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

Reactive

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

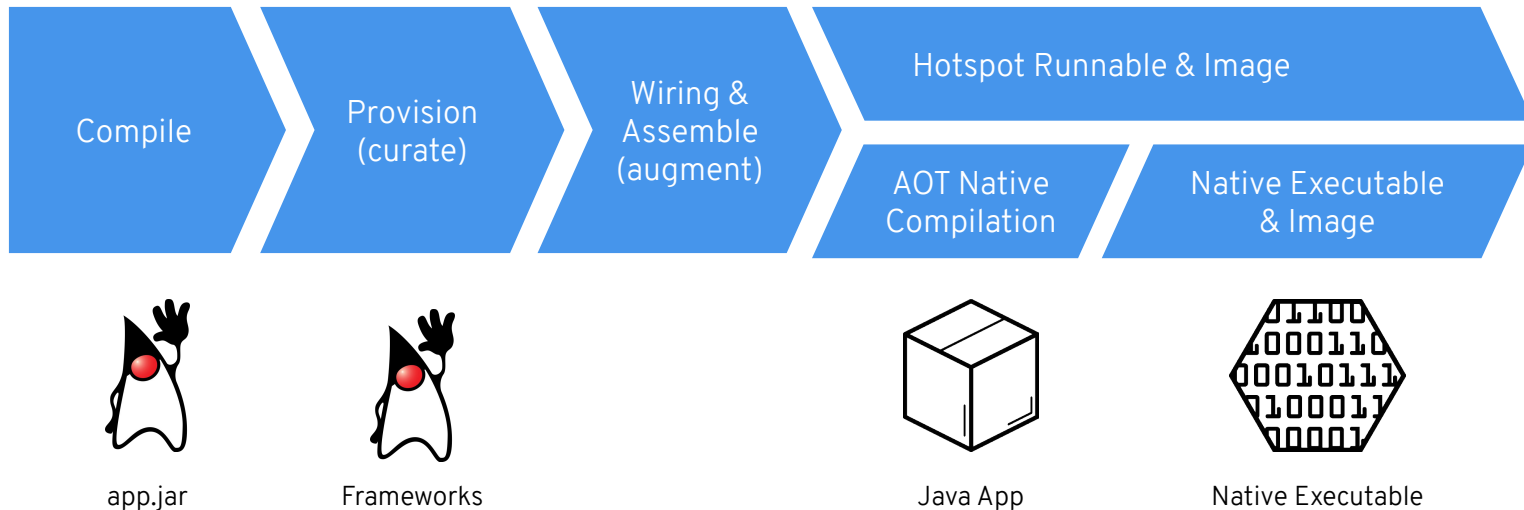
@GET
@Produces(MediaType.SERVER_SENT_EVENTS)
public Publisher<String> stream() {
    return reactiveSay;
}
```

Demo

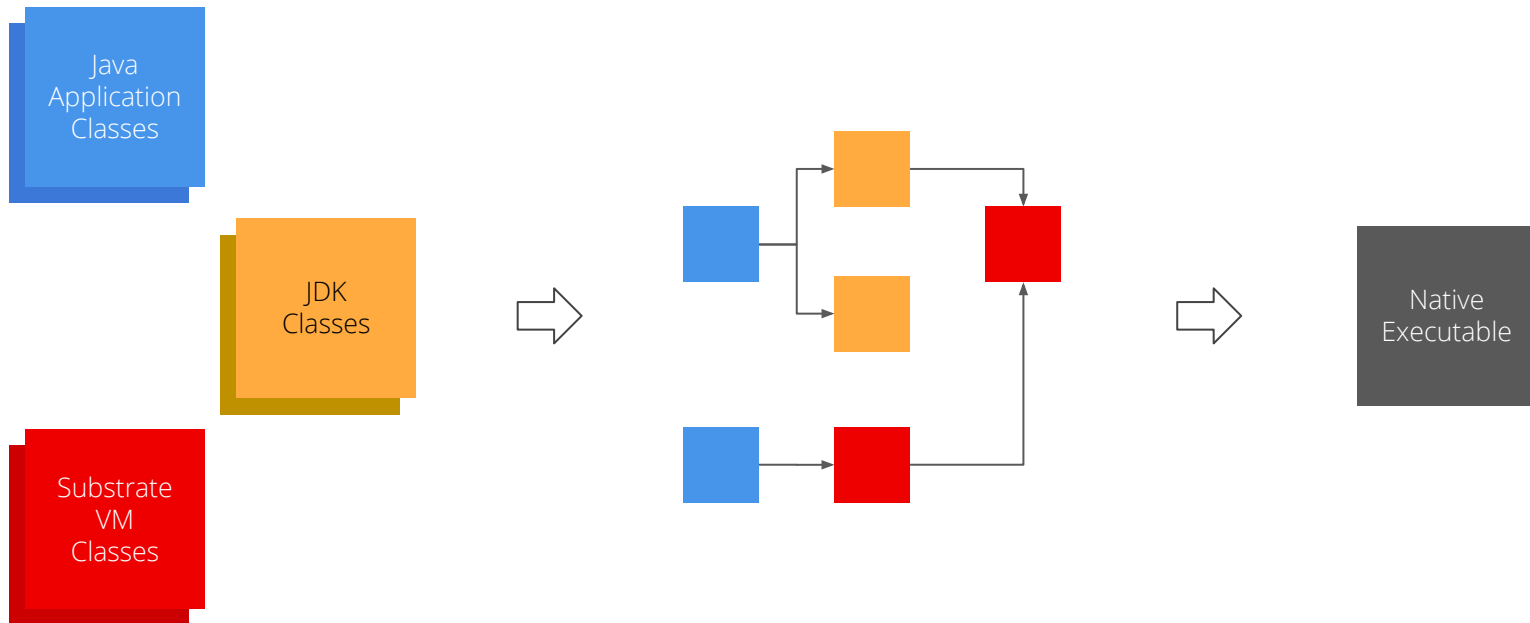
“Truth can only be found in
one place: the code.”

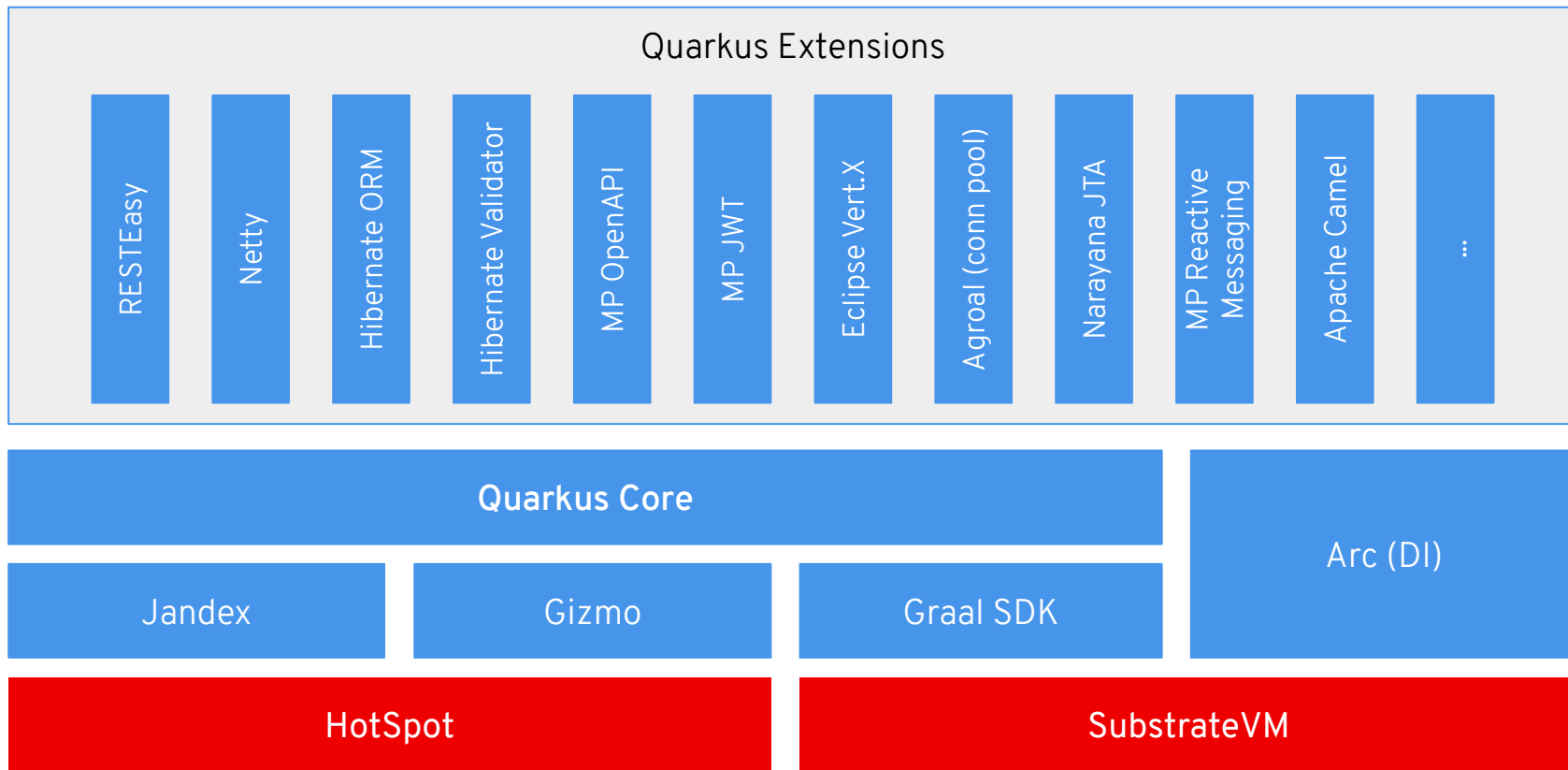
Robert C. Martin, Clean Code

Build Process



Dead code elimination





Links

Memory settings for the
JVM in containers:

<https://serverless.com/blog/2018-serverless-community-survey-huge-growth-usage/>

Performance
measurement in Quarkus

<https://quarkus.io/guides/performance-measure>

Quarkus Todo sample

<https://github.com/burrsutter/quarkus-todo-app>

Quarkus gRPC extensions

<https://github.com/hpehl/quarkus-grpc-extension>
<https://github.com/hpehl/quarkus-grpc-client-extension>
<https://github.com/hpehl/quarkus-grpc-quickstart>

Thank you!
Questions?

<https://quarkus.io>

<https://quarkusio.zulipchat.com>

[@quarkusio](#)

<https://github.com/quarkusio/quarkus>