Descubriendo **Quarkus**:

Java sub-atómico en acción

Aurea Munoz, Red Hat

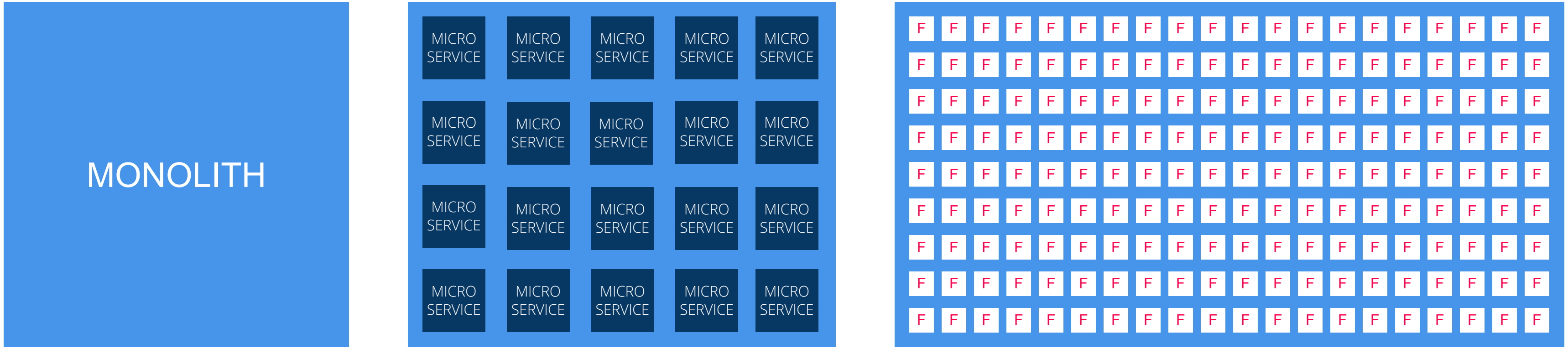
¿Quién soy?

Aurea Muñoz
Senior Software engineer
Red Hat - Snowdrop team

La nueva era de las aplicaciones

Cloud native & Container platforms

Monolitos->Microservicios->serverless

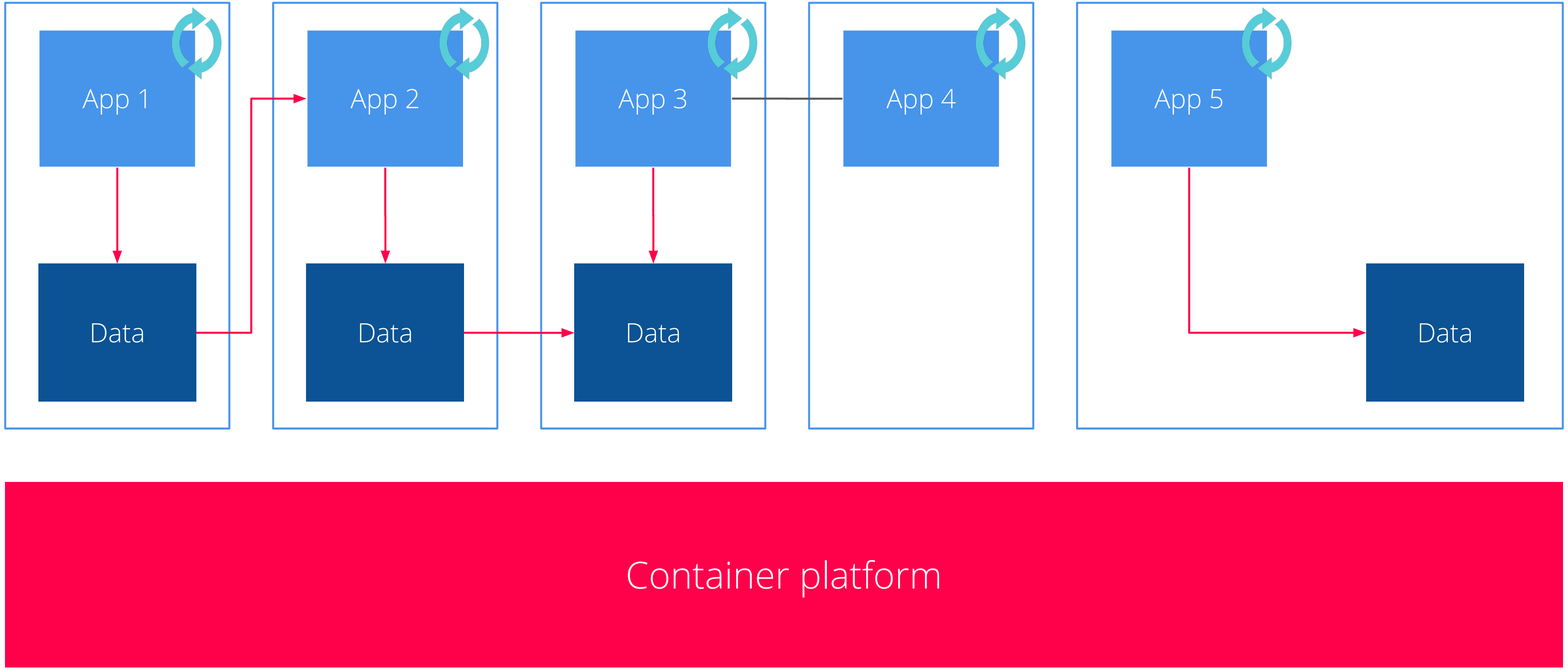

MONOLITH
MICRO SERVICE MICRO SERVICE MICRO SERVICE MICRO SERVICE MICRO SERVICE
MICRO SERVICE MICRO SERVICE MICRO SERVICE MICRO SERVICE MICRO SERVICE
MICRO SERVICE MICRO SERVICE MICRO SERVICE MICRO SERVICE MICRO SERVICE
MICRO SERVICE MICRO SERVICE MICRO SERVICE MICRO SERVICE MICRO SERVICE
F F F F F F F F F F F F F F F F F F F F
F F F F F F F F F F F F F F F F F F F F
F F F F F F F F F F F F F F F F F F F F
F F F F F F F F F F F F F F F F F F F F
F F F F F F F F F F F F F F F F F F F F
F F F F F F F F F F F F F F F F F F F F
F F F F F F F F F F F F F F F F F F F F
F F F F F F F F F F F F F F F F F F F F
F F F F F F F F F F F F F F F F F F F F

- 1 monolith ≈ 20 microservices ≈ 200 functions
- Scale to 1 vs scale to 0
- Start up time

¿Por qué complicarnos la vida?

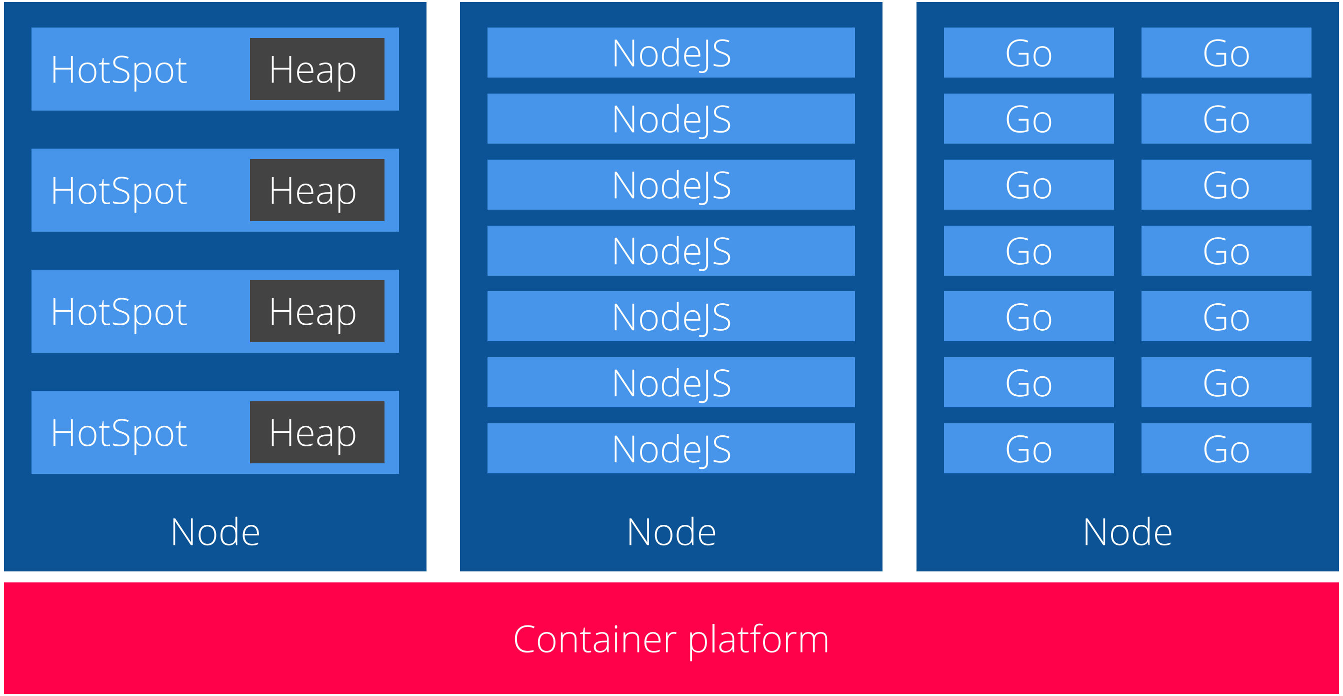
Microservices
Serverless
JAKE-CLARK.TUMBLR
imgflip.com

Agilidad, escalabilidad, mayor reactividad



When containers meet java

La verdad tras la JVM



HotSpot Heap
HotSpot Heap
HotSpot Heap
HotSpot Heap
Node

NodeJS
NodeJS
NodeJS
NodeJS
NodeJS
NodeJS
NodeJS
Node

Go Go
Go Go
Go Go
Go Go
Go Go
Go Go
Go Go
Node

Container platform

Java on cloud native: a Red Hat journey

Standards

- Java EE streamlining
- Eclipse MicroProfile

Runtimes

- WildFly on OpenShift

Hardware architectures

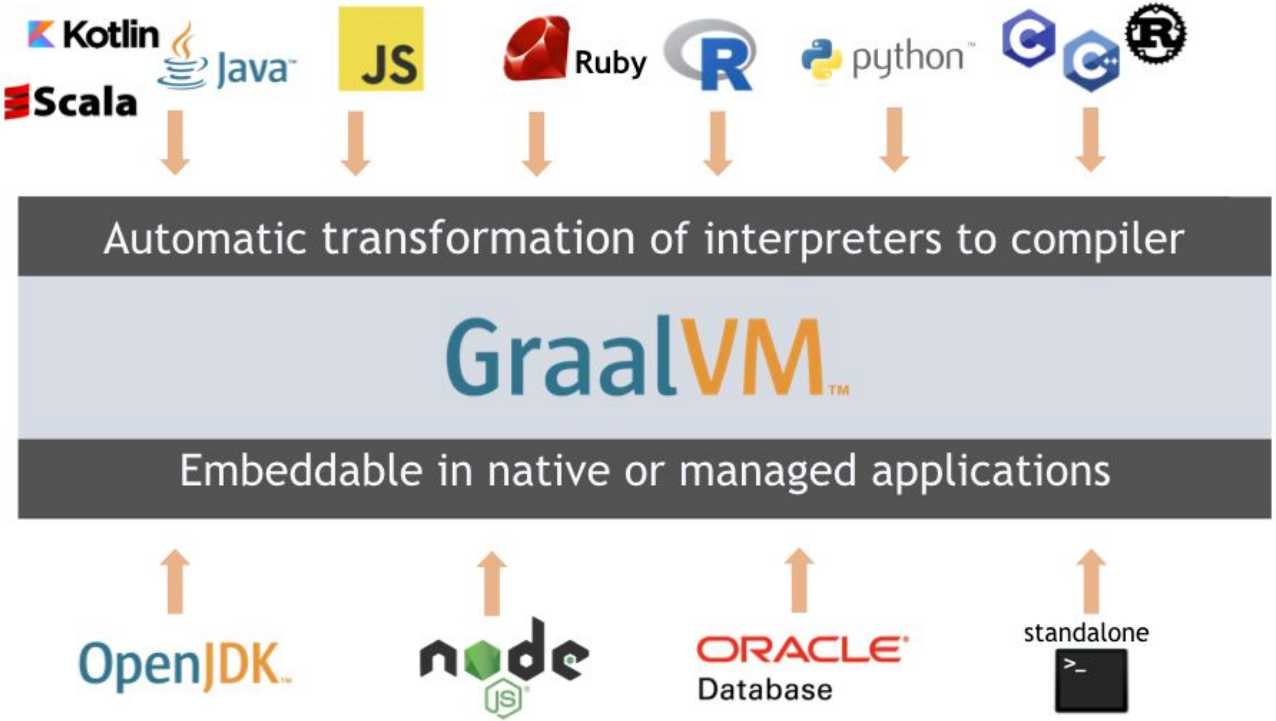
- Raspberry Pi and Plug Computer
- Android

Ahead of time compilation

- Lead gcj
- Looked at Dalvik, Avian, Excelsior JET

OpenJDK

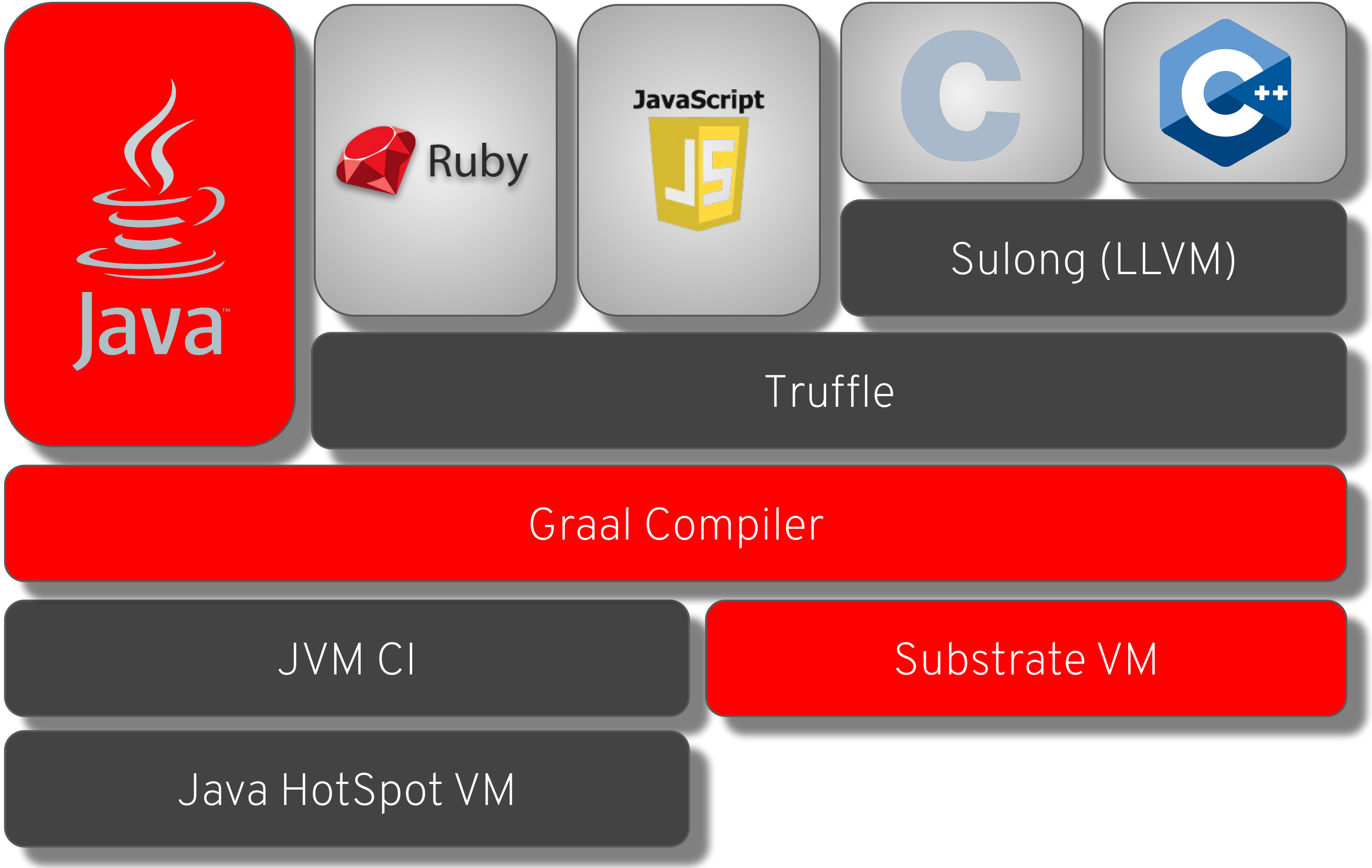
- Container ergonomics
- JVM metadata reduction



Automatic transformation of interpreters to compiler

GraalVM

Embeddable in native or managed applications



Sulong (LLVM)

Truffle

Graal Compiler

JVM CI

Substrate VM

Java HotSpot VM

El problema de Graal

No soporta

- **Dynamic classloading**
- InvokeDynamic & Method handles
- Finalizer
- Security manager
- JVMTI, JMX, native VM Interfaces

Con limitaciones

- **Reflection (manual list)**
- **Dynamic proxy (manual list)**
- JNI (manual list)
- **Static initializers (eager)**
- **Lambda, Threads (OK, pfff!)**
- References (similar)

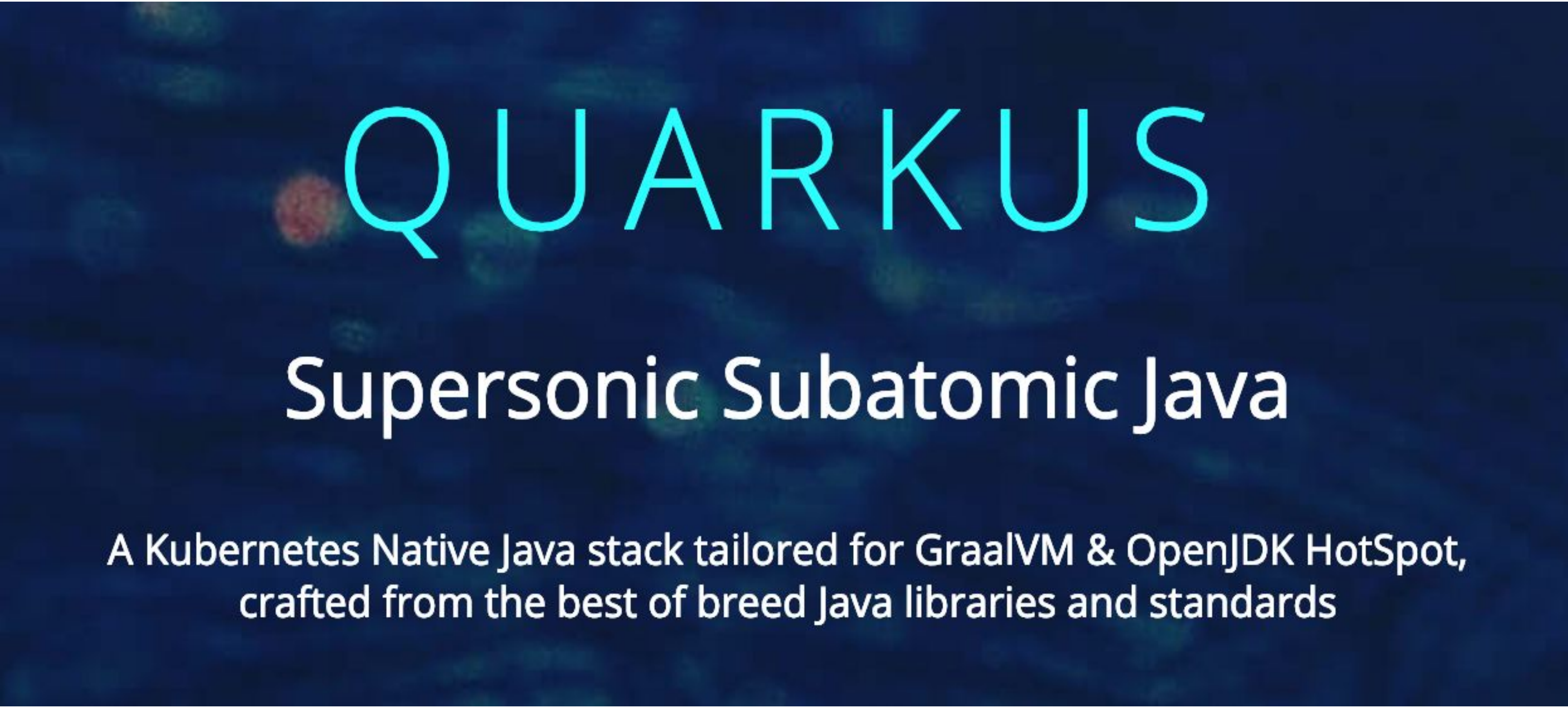
¿Qué es Quarkus?

QUARK: elementary particle

US: hardest thing in computer science

GraalVM™

Java



QUARKUS

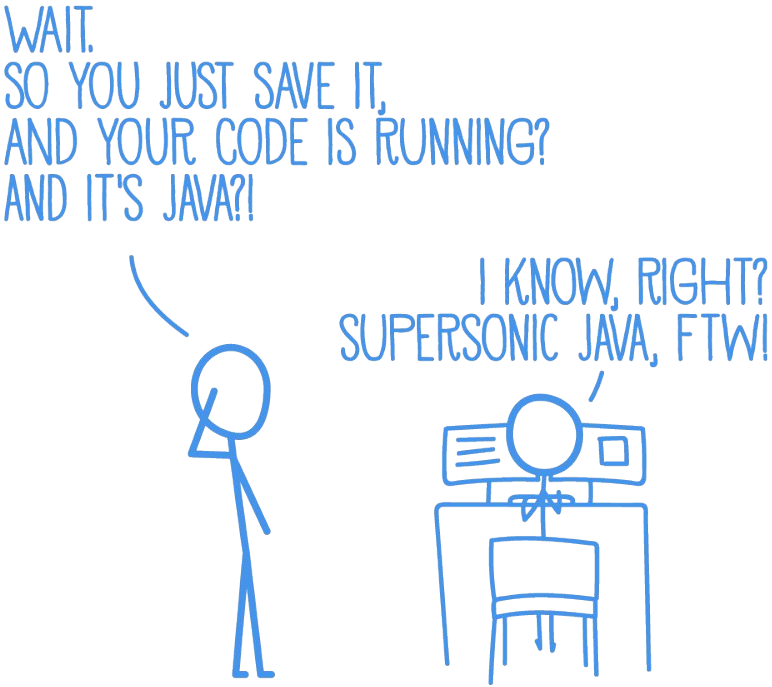
Supersonic Subatomic Java

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

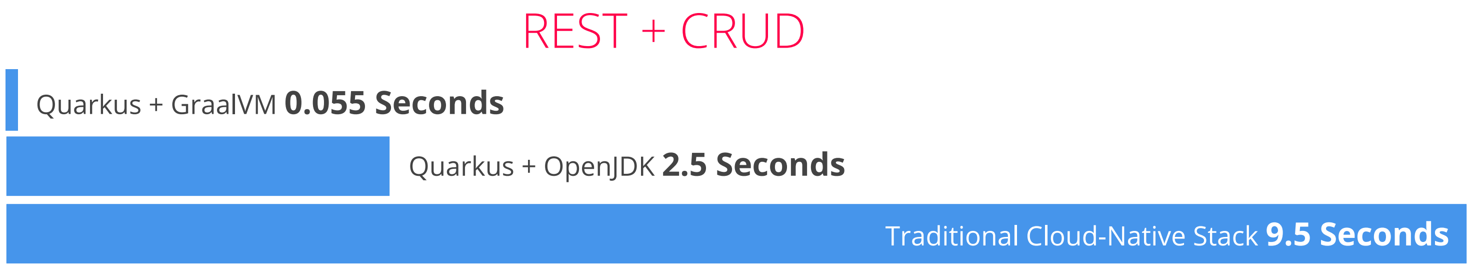
An Open Source stack to write Java apps

Cloud Native, Microservices, Serverless

Beneficio 1: Felicidad del programador



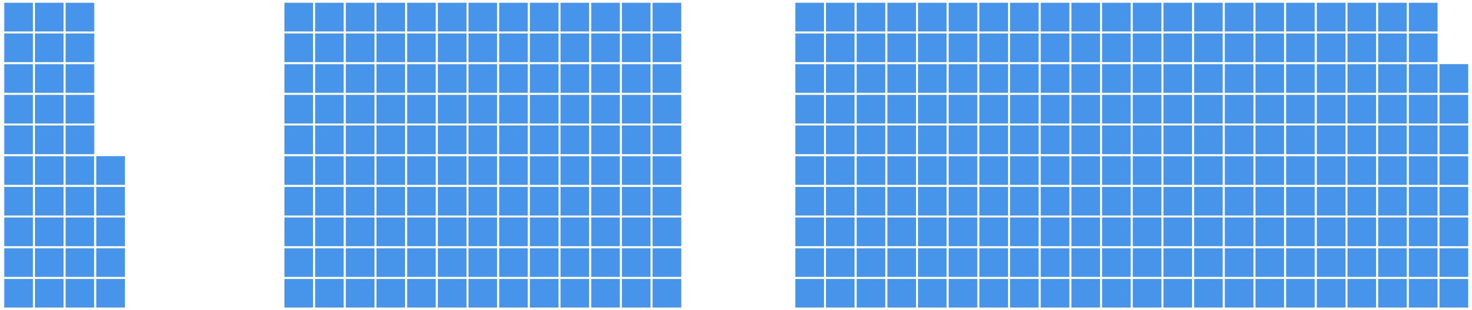
Beneficio 2: Supersonic Subatomic Java



Time to first response

Beneficio 2: Supersonic Subatomic Java

REST + CRUD



Quarkus + GraalVM
35 MB

Quarkus + OpenJDK
130 MB

Traditional Cloud-Native Stack
218 MB

Beneficio 3: Unifica Imperativo + Reactivo

```
@Inject
Servicio 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;
}
```

Beneficio 4:
Los mejores Frameworks y Standards

Eclipse Vert.x

Eclipse MicroProfile

Spring Compat

Hibernate

RESTEasy

Apache Camel

Kubernetes

OpenShift

Jaeger

Prometheus

Apache Kafka

Netty

Beneficios de Quarkus

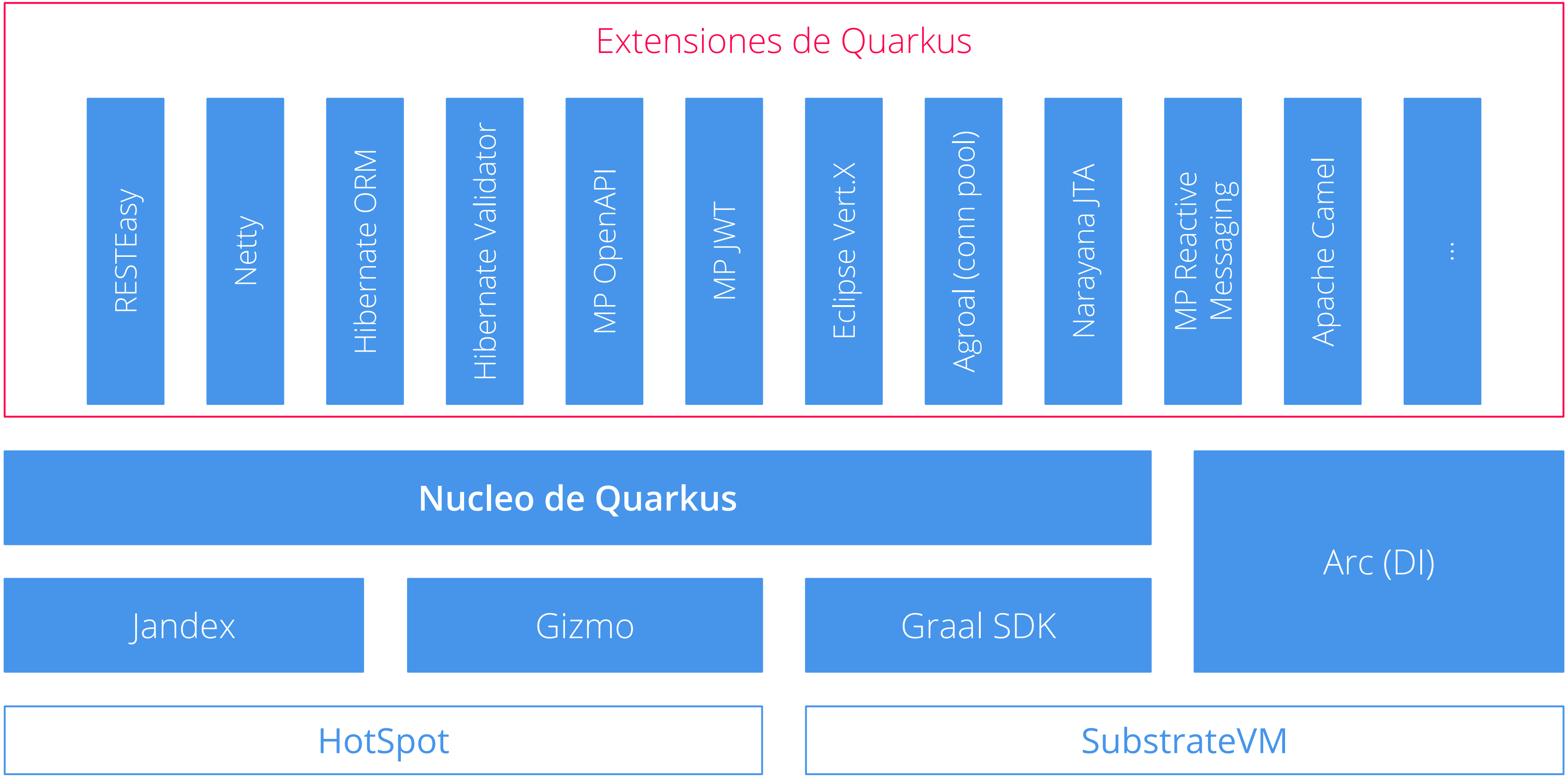
Felicidad del programador

Supersonic Subatomic Java

Unifica programación reactiva e imperativa

Las mejores librerías y estándares

Quarkus: nucleo + extensiones



Extensiones de Quarkus

RESTEasy

Netty

Hibernate ORM

Hibernate Validator

MP OpenAPI

MP JWT

Eclipse Vert.X

Agroal (conn pool)

Narayana JTA

MP Reactive Messaging

Apache Camel

...

Nucleo de Quarkus

Arc (DI)

Jandex

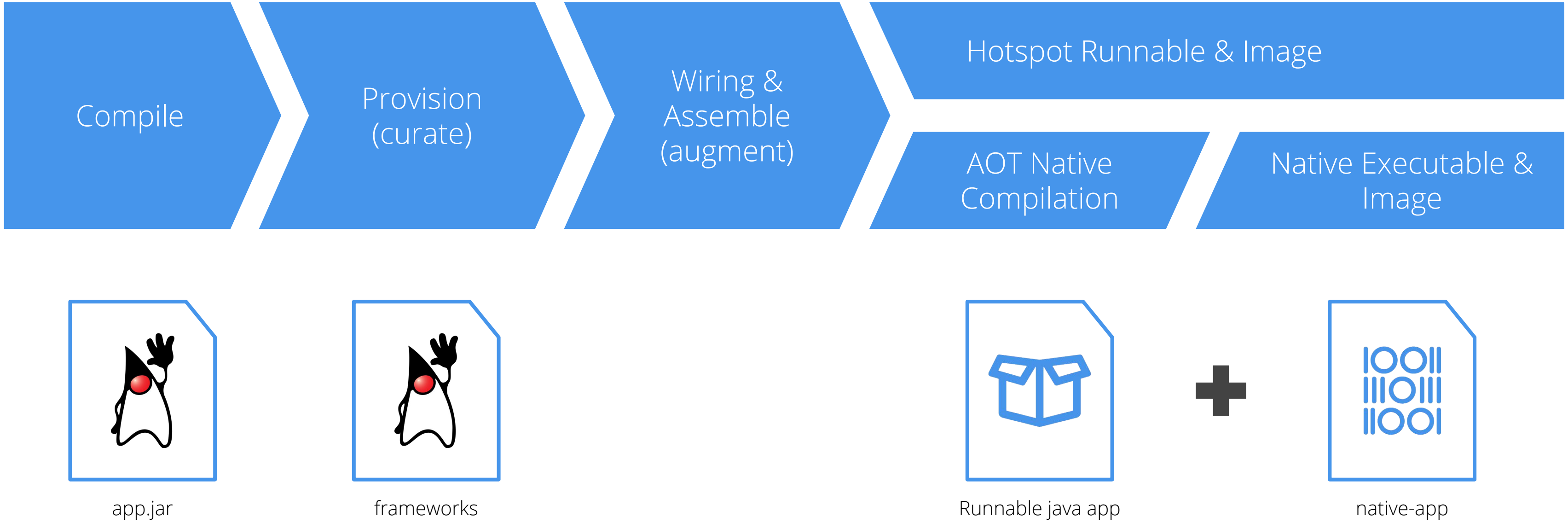
Gizmo

Graal SDK

HotSpot

SubstrateVM

Proceso de build



Compile

Provision (curate)

Wiring & Assemble (augment)

Hotspot Runnable & Image

AOT Native Compilation

Native Executable & Image

app.jar

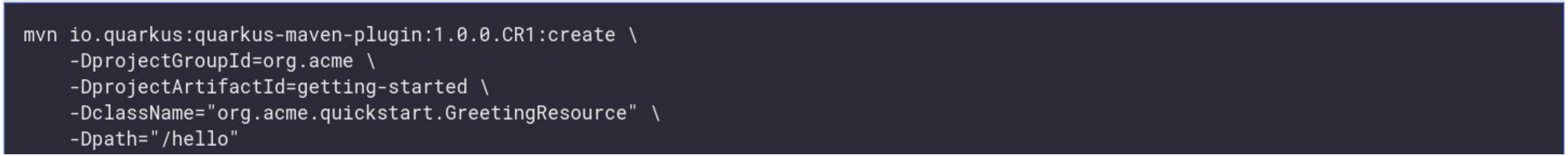
frameworks

Runnable java app

native-app

TALK IS CHEAP
SHOW ME THE CODE
imgflip.com
the Superficial

Simple REST endpoint, returning "hello" to requests on "/hello"



```
mvn io.quarkus:quarkus-maven-plugin:1.0.0.CR1:create \
    -DprojectGroupId=org.acme \
    -DprojectArtifactId=getting-started \
    -DclassName="org.acme.quickstart.GreetingResource" \
    -Dpath="/hello"
```

https://quarkus.io/guides/getting-started#bootstrapping-the-project

Beneficio 4:
Los mejores Frameworks y Standards

Eclipse Vert.x

Eclipse MicroProfile

Spring Compat

Hibernate

RESTEasy

Apache Camel

Kubernetes

OpenShift

Jaeger

Prometheus

Apache Kafka

Netty

Spring Apps con Quarkus

Por qué Spring en Quarkus?

- Capitalizar el conocimiento Spring
- Familiaridad
- Acelerar el aprendizaje y onboarding.

Spring DI - Spring Web

Dependency injection model

- @Component, @Autowired, etc

Spring Web - Developing REST APIs

- @RestController, @RequestMapping etc

Spring REST APIs

pom.xml

```xml
<dependency>
   <groupId>io.quarkus</groupId>
   <artifactId>quarkus-spring-web</artifactId>
</dependency>
```

GreetingController.java

```java
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class GreetingController {

   @GetMapping("/hello")
   public String hello(@RequestParam(defaultValue = "world") String name) {
       return "hello " + name;
   }
}
```

TALK IS CHEAP
SHOW ME THE CODE
imgflip.com
the Superficial

Spring Data JPA

Spring Data JPA

- Derived methods
- Custom query methods
- Fragments
- Transactional support

Spring Data JPA

pom.xml

```
<dependency>
 <groupId>io.quarkus</groupId>
 <artifactId>quarkus-spring-data-jpa</artifactId>
</dependency>
```

BookRepository.java

```
import java.util.List;
import org.springframework.data.jpa.repository.Modifying;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.CrudRepository;
import org.springframework.data.repository.query.Param;

public interface BookRepository extends CrudRepository<Book, Integer> {

  List<Book> findByPublicationYearBetween(Integer lower, Integer higher);

 }
```

Book.java

```
import javax.persistence.Entity;
import javax.persistence.Id;

@Entity
public class Book {
  @Id
  private Integer bid;
  private Integer publicationYear;
  private String name;
}
```

TALK IS CHEAP
SHOW ME THE CODE
imgflip.com
theSuperficial

QUARKUS CONSIGUE UN MEJOR JAVA EN ENTORNOS DE CONTENEDORES Y SERVERLESS

QUARKUS

https://quarkus.io

https://quarkusio.zulipchat.com

@quarkusio

Docs:

https://github.com/quarkusio/quarkus-quickstarts.git

https://quarkus.io/guides/spring-di-guide

https://quarkus.io/guides/spring-web

https://quarkus.io/guides/spring-data-jpa

https://quarkus.io/guides/hibernate-orm

https://learn.openshift.com/middleware/courses/middleware-quarkus/getting-started

Demos:

https://github.com/aureamunoz/spring-quarkus-jug
https://github.com/aureamunoz/quarkus-getting-started

Slides:

https://github.com/aureamunoz/spring-quarkus-jug/blob/master/Madrid%20JUG%20Descubriendo%20Quarkus%2C%20java%20sub-at%C3%B3mico%20en%20acci%C3%B3n.pdf

@auritamh

Gracias!

MADRIDJUG
JAVA USER GROUP