Il framework Quarkus

Una panoramica generale

Alessio Giovanni Baroni

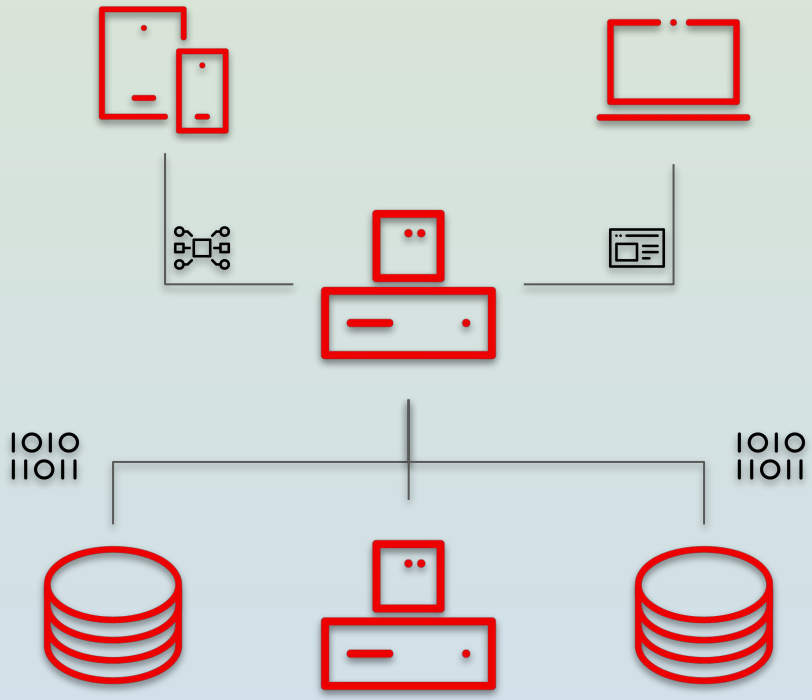
Senior Consultant

- Architettura generale ed informazioni di base
- Red Hat Developer Program
- Demo

Red Hat

Perché Quarkus?

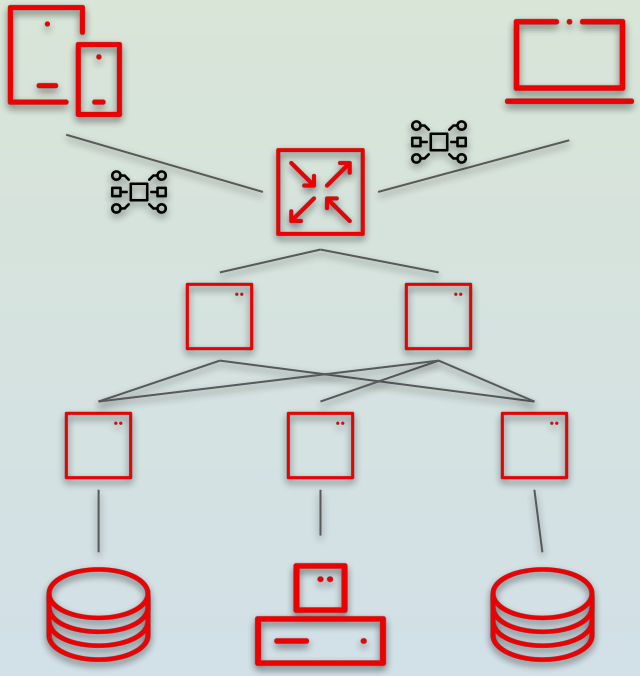
Architetture tradizionali



- Logiche sempre più complesse, quindi applicazioni sempre più grandi
- Tempi di avvio più lunghi
- Aggiornamenti più rari ma più consistenti
- Cose grandi e complesse sono sicure?

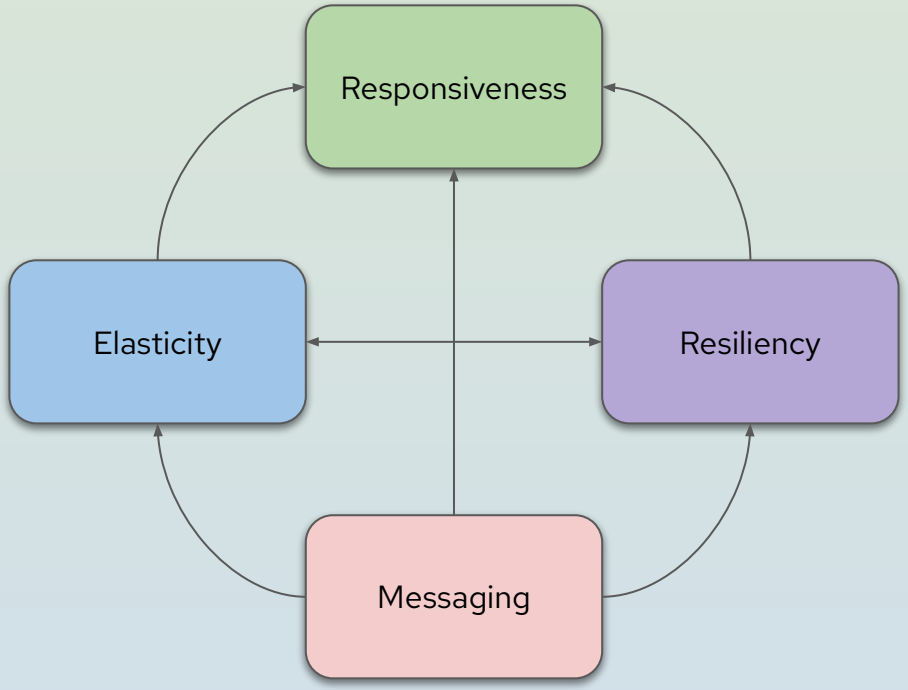
Perché Quarkus?

Architetture moderne



- Implementiamo la logica in tanti componenti piccoli e facilmente gestibili
- Tempi di avvio (molto) più corti
  - Buona *responsiveness*
- Aggiornamenti (molto) più frequenti
- Analisi di sicurezza (molto) semplificata

Quarkus è un framework *Reactive*



Responsive

I massimi tempi di risposta sono prevedibili ed affidabili

Resilient

Il sistema è capace di sostenere il verificarsi di errori

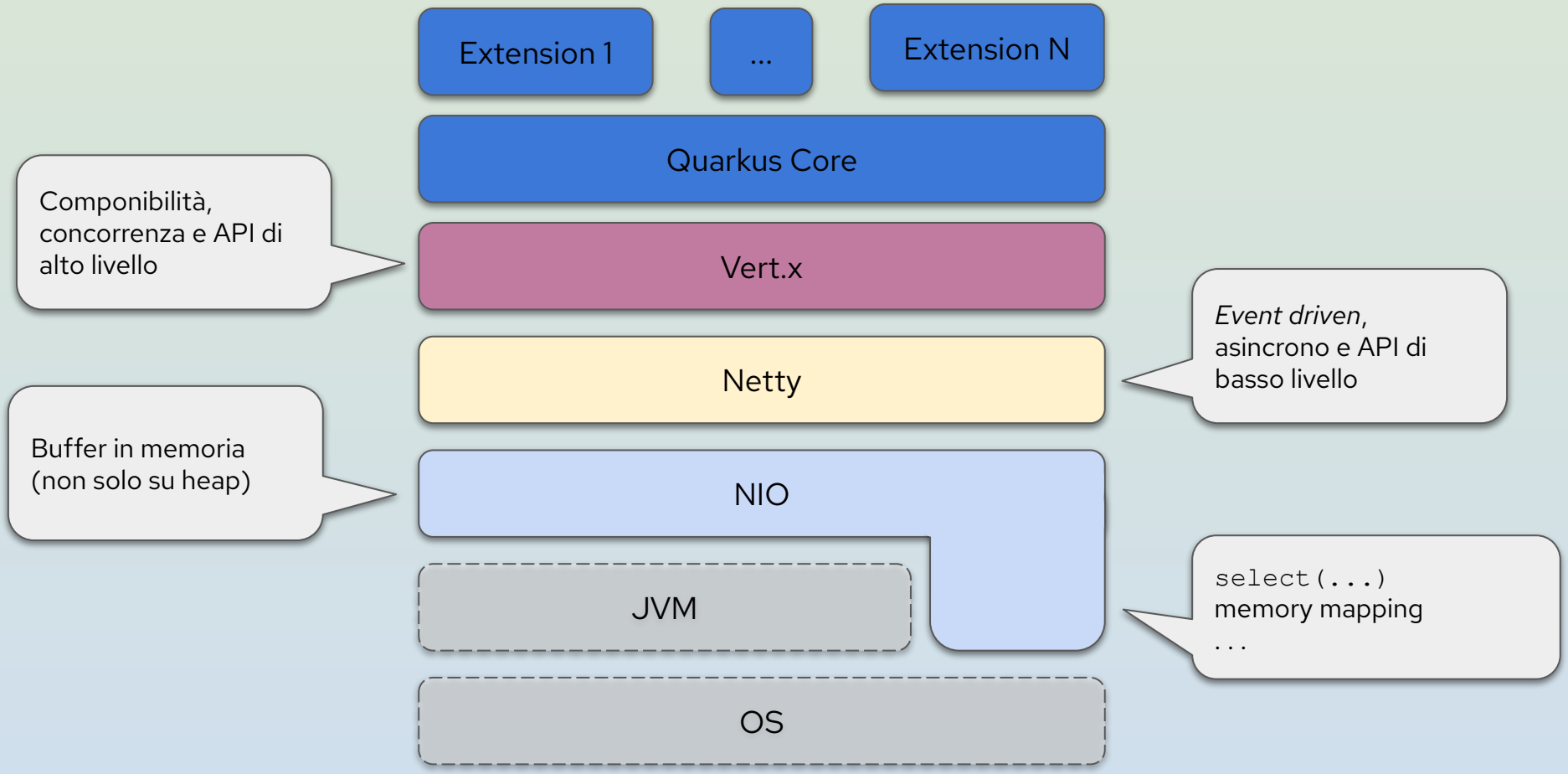
Elastic

Il sistema è in grado di sostenere variazioni di carico

Message driven

Comunicazione asincrona per aumentare la coesione

Architettura generale



Vert.x

Esempio

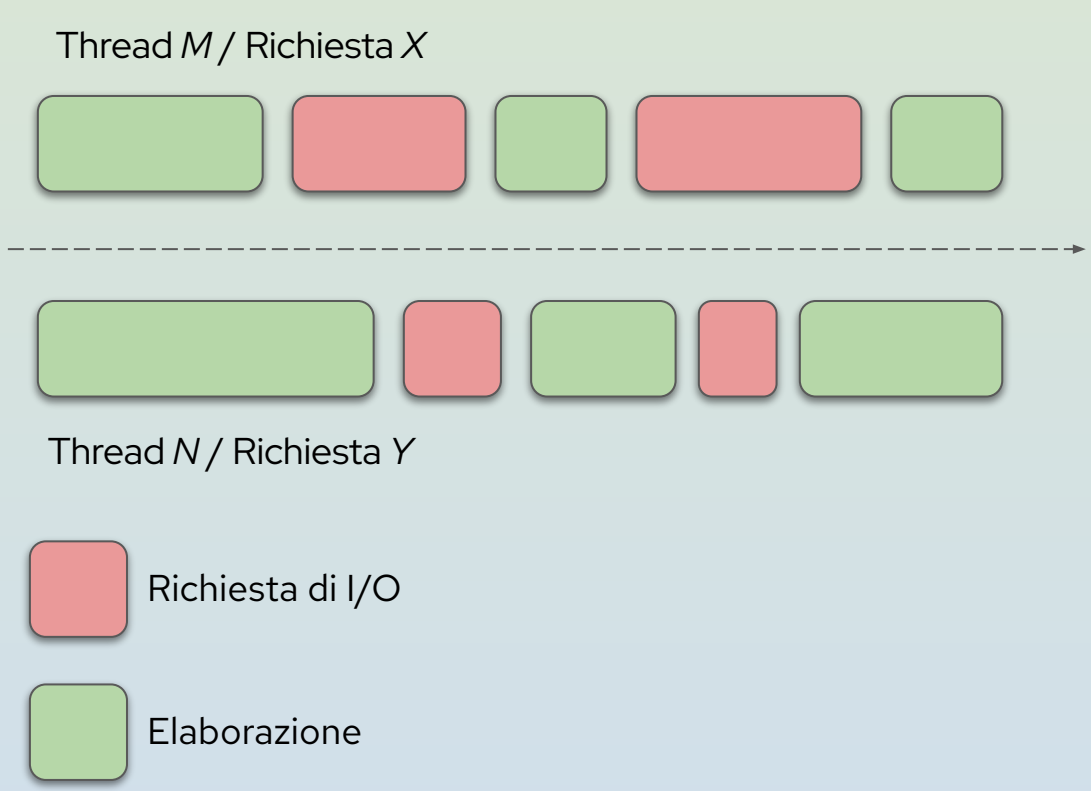
```
import io.vertx.core.buffer.Buffer;
import io.vertx.core.Vertx;
import io.vertx.core.http.HttpServerOptions;
import io.vertx.core.net.SelfSignedCertificate;

public class Main {
    public static void main(String... args) {
        var vertx = Vertx.vertx();
        var options = new HttpServerOptions()
                          .setUseAlpn(true)
                          .setSsl(true)
                          .setKeyCertOptions(SelfSignedCertificate
                                             .create()
                                             .keyCertOptions())
                          .setLogActivity(true);
        var server = vertx.createHttpServer(options);

        server.requestHandler(request -> {
                request.bodyHandler(body -> {
                        System.console().writer().println(body);
                    });

                request.response().end();
            });

        server.listen(8080);
    }
}
```

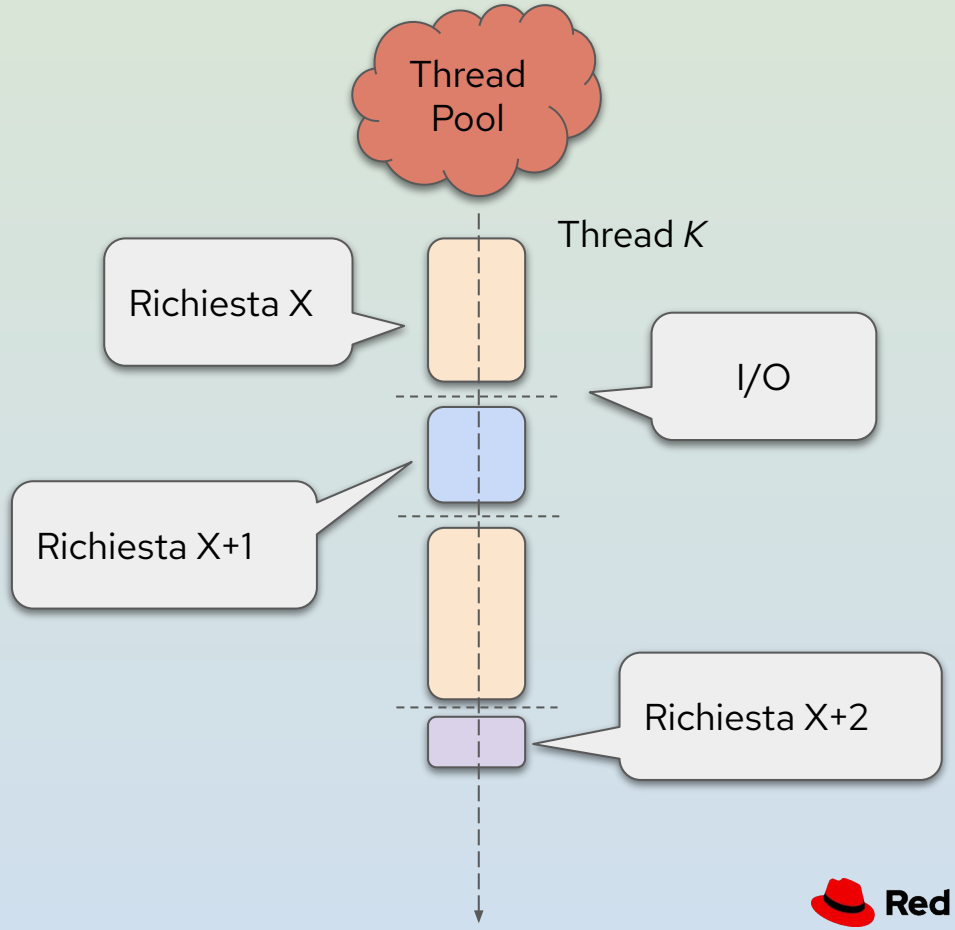
```
[alessio@pc-arch-rh example-vertx]$ java -jar target/example-vertx-0.1.0-SNAPSHOT.jar
ciao mondo
```

```
[alessio@pc-arch-rh example-vertx]$ curl -k -X POST -d 'ciao mondo' https://localhost:8080/
[alessio@pc-arch-rh example-vertx]$
[alessio@pc-arch-rh example-vertx]$
```

https://www.martinfowler.com/bliki/FluentInterface.html

I/O non bloccante





Red Hat

Mutiny (1)

```
private static int k = 0;

private static String mark(String s) {
    return String.format("%d %s", ++k, s);
}

private static <T, R> Uni<R> genericComputation(T t, long n, Function<T, R> f) {
    return Uni.createFrom().completionStage(CompletableFuture.supplyAsync(() -> {
                var s = System.currentTimeMillis();
                var k = n * 1000L;

                while (System.currentTimeMillis() - s <= k);

                return f.apply(t);
            }));
}

private static Uni<String> getMessage(String initial, long n) {
    return genericComputation(initial, n, s -> s);
}

private static Uni<String> handleMessage(String text, long n) {
    return genericComputation(mark(text), n, String::toUpperCase);
}
```

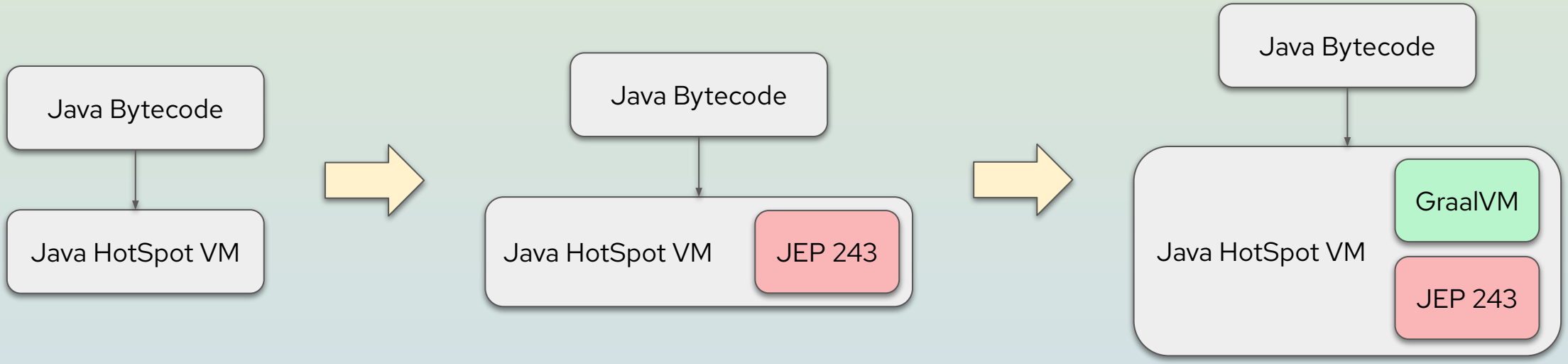
Mutiny (2)

```
public static void main(String... args) {
    System.console().writer().println(
        Uni.combine().all().unis(getMessage("ciao", 3)
                                    .onItem()
                                    .transform(s1 -> handleMessage(s1, 1)
                                                        .await()
                                                        .indefinitely()),
                                getMessage("mondo", 1)
                                    .onItem()
                                    .transform(s1 -> handleMessage(s1, 2)
                                                        .await()
                                                        .indefinitely()))
                .asTuple().await().indefinitely()
    );
}
```

```
[alessio@pc-arch-rh example-mutiny]$ java -jar target/example-mutiny-0.1.0-SNAPSHOT.jar
Tuple{item1=2 CIAO, item2=1 MONDO}
[alessio@pc-arch-rh example-mutiny]$
```

GraalVM

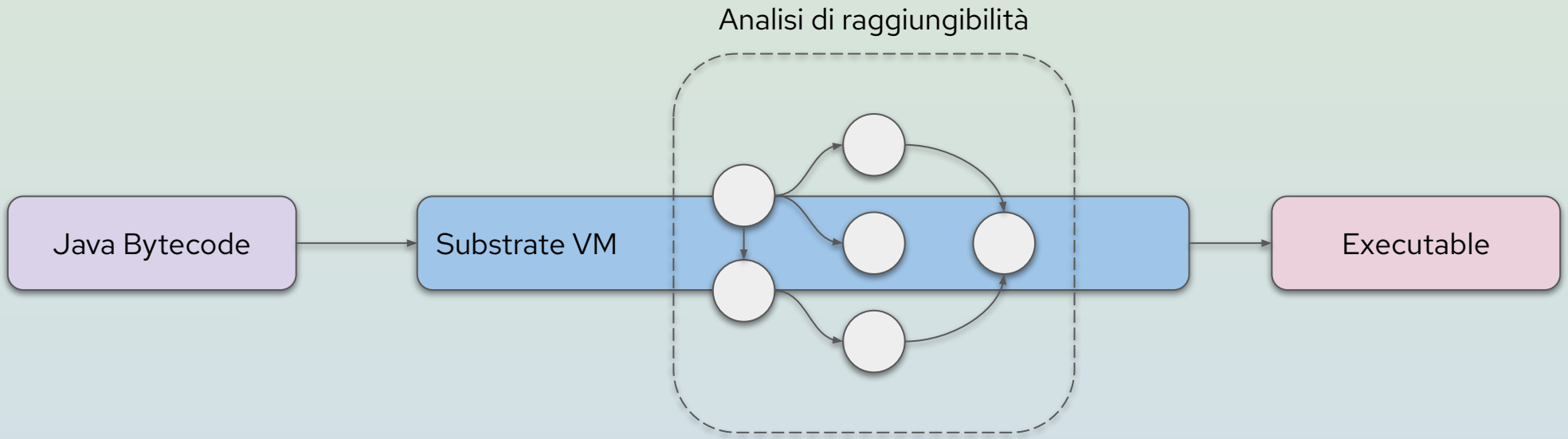
Runtime mode



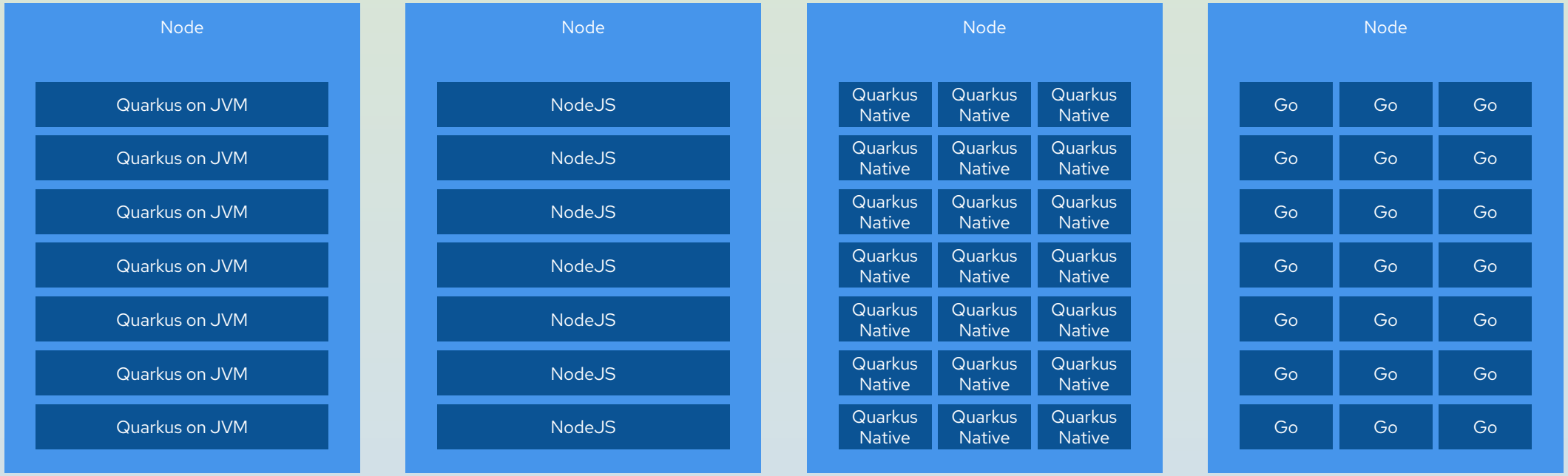
JEP 243 Summary:
"Develop a Java based JVM compiler interface (JVMCI) enabling a compiler written in Java to be used by the JVM as a dynamic compiler."

GraalVM

Native images

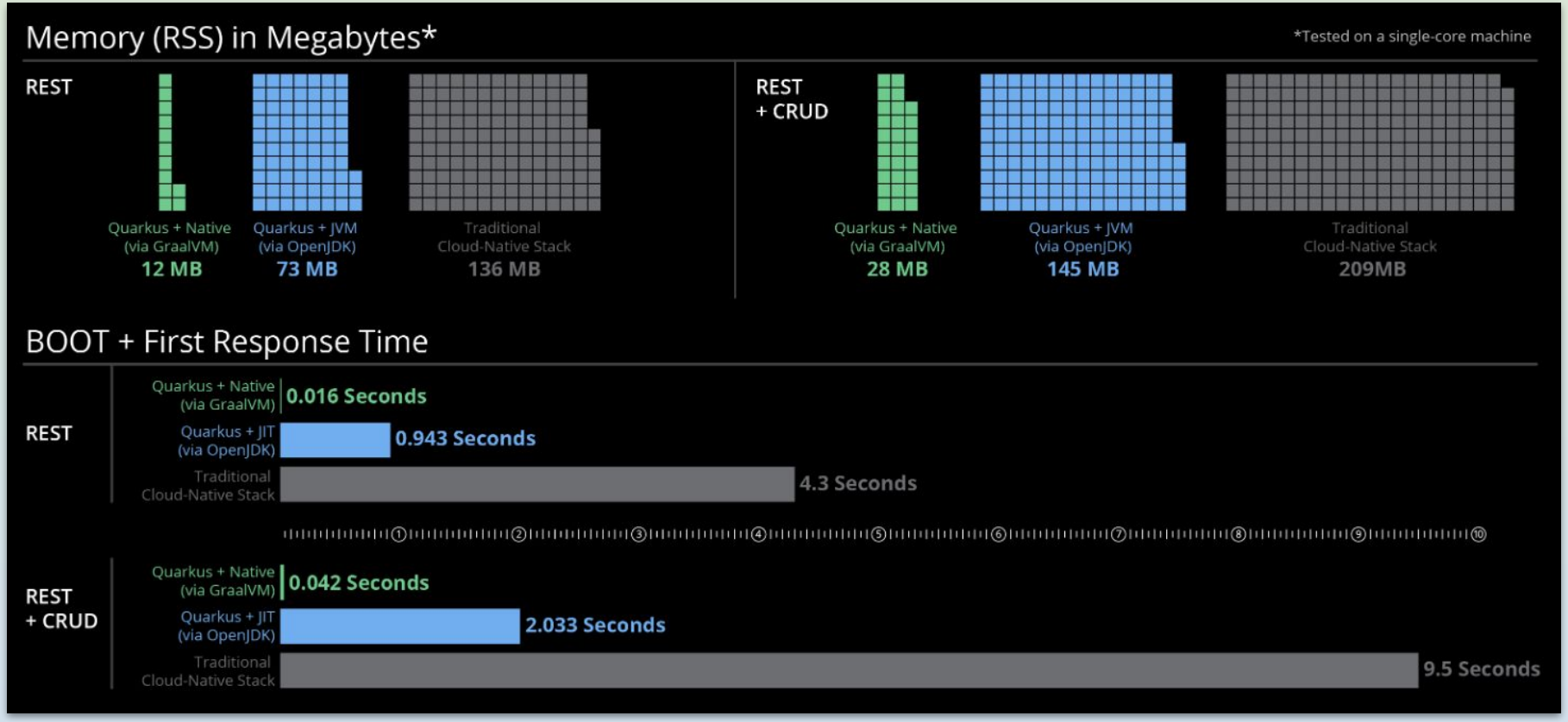


Uso efficiente delle risorse



Red Hat

Quarkus native



Red Hat

Creare un nuovo progetto

```
...]$ mvn \
        io.quarkus.platform:quarkus-maven-plugin:2.7.6.Final:create \
        -DprojectGroupId=io.github.agbaroni \
        -DprojectArtifactId=test \
        -Dextensions="config-yaml,resteasy,resteasy-jsonb"
```

```
...]$ ./mvnw quarkus:add-extension -Dextensions="hibernate-validator,hibernate-orm,jdbc-postgresql"
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------------< io.github.agbaroni:test >-----------------------
[INFO] Building test 1.0.0-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- quarkus-maven-plugin:2.7.6.Final:add-extension (default-cli) @ test ---
[INFO] [SUCCESS] ✅  Extension io.quarkus:quarkus-hibernate-validator has been installed
[INFO] [SUCCESS] ✅  Extension io.quarkus:quarkus-hibernate-orm has been installed
[INFO] [SUCCESS] ✅  Extension io.quarkus:quarkus-jdbc-postgresql has been installed
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.525 s
[INFO] Finished at: 2022-10-21T12:02:37+02:00
[INFO] ------------------------------------------------------------------------
```

Configurazione Maven

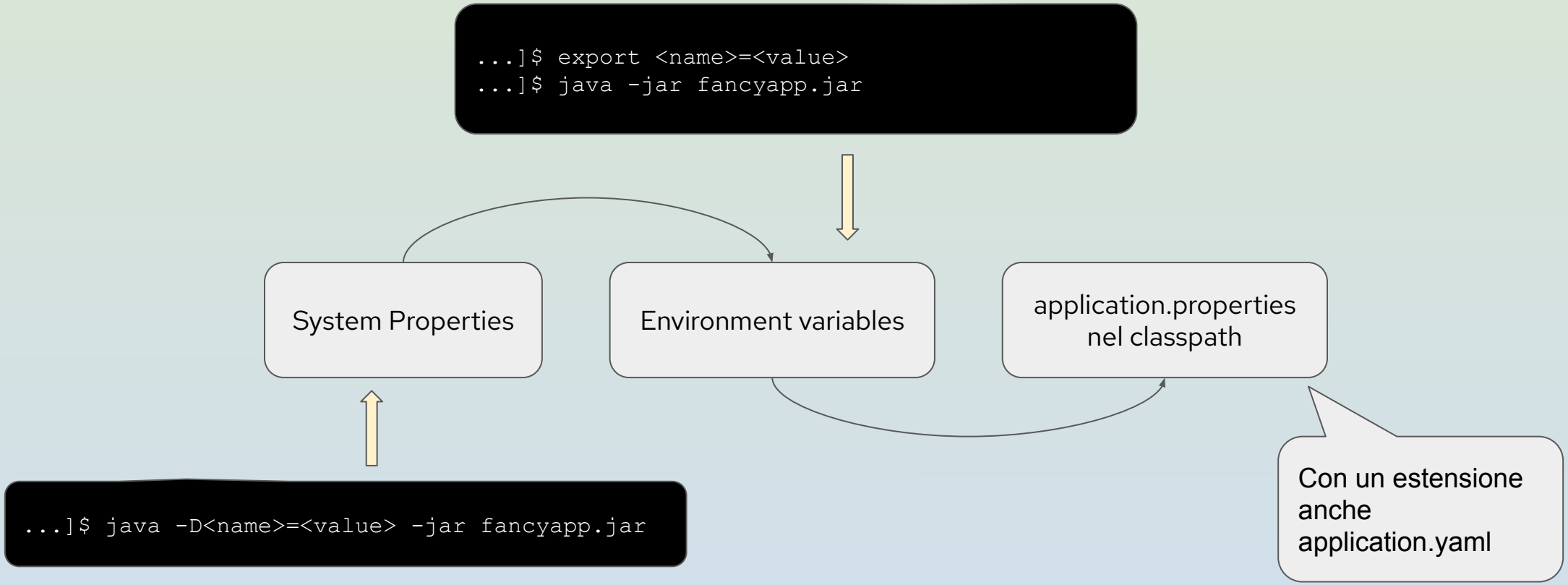
```
<plugin>
  <artifactId>quarkus-maven-plugin</artifactId>
  <groupId>${quarkus.platform.groupId}</groupId>
  <executions>
    <execution>
      <goals>
        <goal>build</goal>
        <goal>generate-code</goal>
        <goal>generate-code-tests</goal>
      </goals>
    </execution>
  </executions>
  <extensions>true</extensions>
  <version>${quarkus.version}</version>
</plugin>
```

```
<dependencyManagement>
  <dependencies>
    <dependency>
      <artifactId>quarkus-bom</artifactId>
      <groupId>${quarkus.platform.groupId}</groupId>
      <scope>import</scope>
      <type>pom</type>
      <version>${quarkus.version}</version>
    </dependency>
  </dependencies>
</dependencyManagement>
```

```
<dependency>
  <artifactId>quarkus-resteasy-reactive</artifactId>
  <groupId>${quarkus.groupId}</groupId>
</dependency>
```

Sorgenti di configurazione di Quarkus

Microprofile Config



Red Hat

Red Hat Developer Program

The universal, supported, OS

- Red Hat è un'azienda *open source*
- Tutti i suoi prodotto sono *open source*
- Red Hat fornisce la professionalità dei suoi dipendenti
- Chiunque può creare un account Red Hat gratuito
- Chiunque può scaricare la versione aggiornata di Red Hat Enterprise Linux

Scaricate e condividete!

Red Hat

Demo

https://github.com/agbaroni/LinuxDay22