A Cloud-Native Dev Is Eating My Java Cheese!

(How to Get It Back?)

Eder Ignatowicz

Sr. Software Engineer @ Red Hat

Alex Porcelli

Principal Software Engineer @ Red Hat

Red Hat

The World has changed

Red Hat

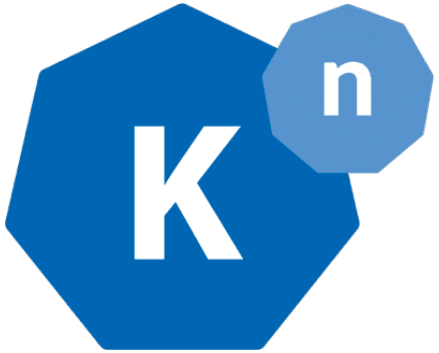
kubernetes

Istio

Cloud Native Applications

Functions

Microservices



OPENSHIFT

Red Hat

Cloud Native Requirements

Microservices scenarios

Existing Java Tools and Frameworks Not Optimized

- Cold Starts

- Low Memory consumption

How Java Platform/Ecosystem perform on those scenarios?

Not so good

How is your dev experience developing Cloud Native Apps in Java?

Bad! :(

But how can we get back to the game?

Let’s dive a little in the history before answer that...

Red Hat

199x - Java Early Adopters

Red Hat

199x - Java Early Adopters

Big reliable “fault-tolerant” servers

Application servers

Application always up and running

JVM optimized for the long run

State of the art

200x - Java Majority

200x - Java Majority Cloud Computing

Commodity Hardware

(expect failure)

Cloud Generations

Cloud Generations

G1: Virtualization

Red Hat

Hypervisor

Application Servers as the best-practice (JavaEE)

tomcat/jetty as outliers

Java did great!

201x - Java Late Majority

Cloud Generations

G2: Containers

Microservices rise

Docker

SpringBoot

go lang, nodejs as outliers

Java OK-ish

Cloud Generations

G3: Orchestration

Microservice maturity

Kubernetes

Go and node skyrocketing

Operator, Service Mesh as Outliers

Lagger: JAVA, but why?

- Image sizing
- Memory footprint
- Startup/Restart time on ephemeral instances

So? Should I learn go or node?

Yes, you should! World is polyglot.

What about Java?

Why should I consider Java for G3?

Java for G3

Microprofile Standard

Rich library ecosystem

Tons of developers

IDE (Tooling)

Static type language that everybody in this room loves it

Keeps evolving

(Don't forget how many times Java saved our lives)

But Java still has issues for G3

How to have back my happiness programming in Java?

QUARKUS

Red Hat

Java EE standards + "NodeJS Dev Experience" + the size of a GO binary

Build on the shoulders of giants

- GraalVM
- Java standards API

DEMO

Kogito

Kogito is the next generation business automation toolkit based on:

- **Drools** - the open source rule engine to provide decision and rule management capabilities
- **jBPM** - the open source process engine to provide automation and orchestration capabilities

It takes advantage of years of battle-tested features, while at the same time modernizing usage to fit the cloud-native ecosystem

Kogito

Next-gen Cloud-Native Business Automation

Cloud-Native Business Automation for building intelligent applications, backed by battle-tested capabilities

Powerful Developer Experience

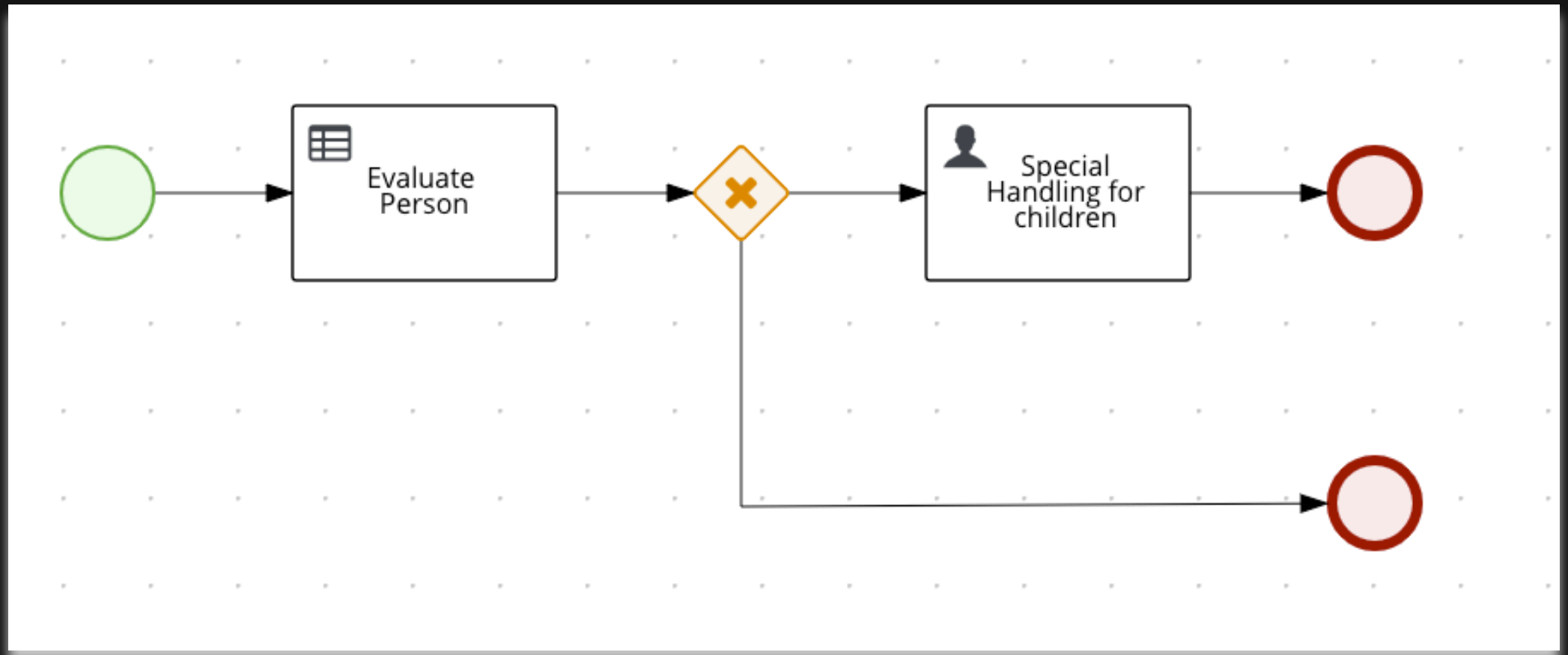
- Instant developer efficiency
- Embeddable tooling
- Codegen for 80% of work
- Flexible, customizable, use only what you need
- Simplified local development and debugging
- Leveraging / integrating many other (DevExp) technologies

Built for Cloud from ground up

- Kube-native, built for OpenShift
- Fit into Knative serverless
- Run on Quarkus / SpringBoot
- Superfast boot time, low footprint (native image)
- Operator-driven service lifecycle management
- Leveraging / integrating many other (cloud) technologies

Ready for Business

- Domain-specific services and APIs
- Context-based data searches, dashboards
- Out-of-the-box customizable, embeddable UI building blocks (e.g. inbox)
- Domain-specific event-based operational visualization



Red Hat

DEMO

Technology progress is inevitable

With GraalVM + Quarkus

202x - Java back to the Game!

Let’s talk about future?

Cloud Generations

G4: Serverless

G4: Serverless

Functions (i.e. AWS Lambda)

Front-end as function orchestrator

non-functional requirement provided at Platform Level Service (Operator)

-> ie. Security, Caching, etc.

JavaScript -> browser native language (function orchestrator) / front-end

GO -> kubernetes native language (operators)

Java -> function development

World is polyglot

With GraalVM + Quarkus

Great dev experience

Low resource footprint

really quick start

Makes Java the **best option** for App Dev for G3 (orchestration) and G4 (serverless)

Java is back to the game!

Thank you!

Eder Ignatowicz

Sr. Software Engineer @ Red Hat

@ederign

Alex Porcelli

Principal Software Engineer @ Red Hat

@porcelli