DESENVOLVIMENTO DE APLICAÇÕES BACKEND COM QUARKUS

AULA 1

@eldermoraes

COMO O QUARKUS SURGIU

COMO O QUARKUS SURGIU

- 1996 a 2013: Java sai de um laboratório da Sun Microsystems e chega ao status de linguagem mais utilizada no mundo
- 2013: Docker é lançado e estabelece um padrão ao uso de containers (que já existiam há muito tempo)
- 2014: Lançamento do Java 8, ainda sem suporte a containers (que só começaria a vir a partir da build 121)
- 2015 a 2018: Inúmeras features voltadas a containers são introduzidas à plataforma Java, culminando na versão 11, que se tornou a versão mínima recomendada para uso de Java em containers
- 2016: Lançamento do Eclipse MicroProfile (eu estava lá!), a primeira especificação voltada ao uso de Java em microservices e sem o uso de application servers
- 2019: Primeira demonstração pública do Quarkus (eu estava lá!), com a promessa do “Supersonic Subatomic Java"

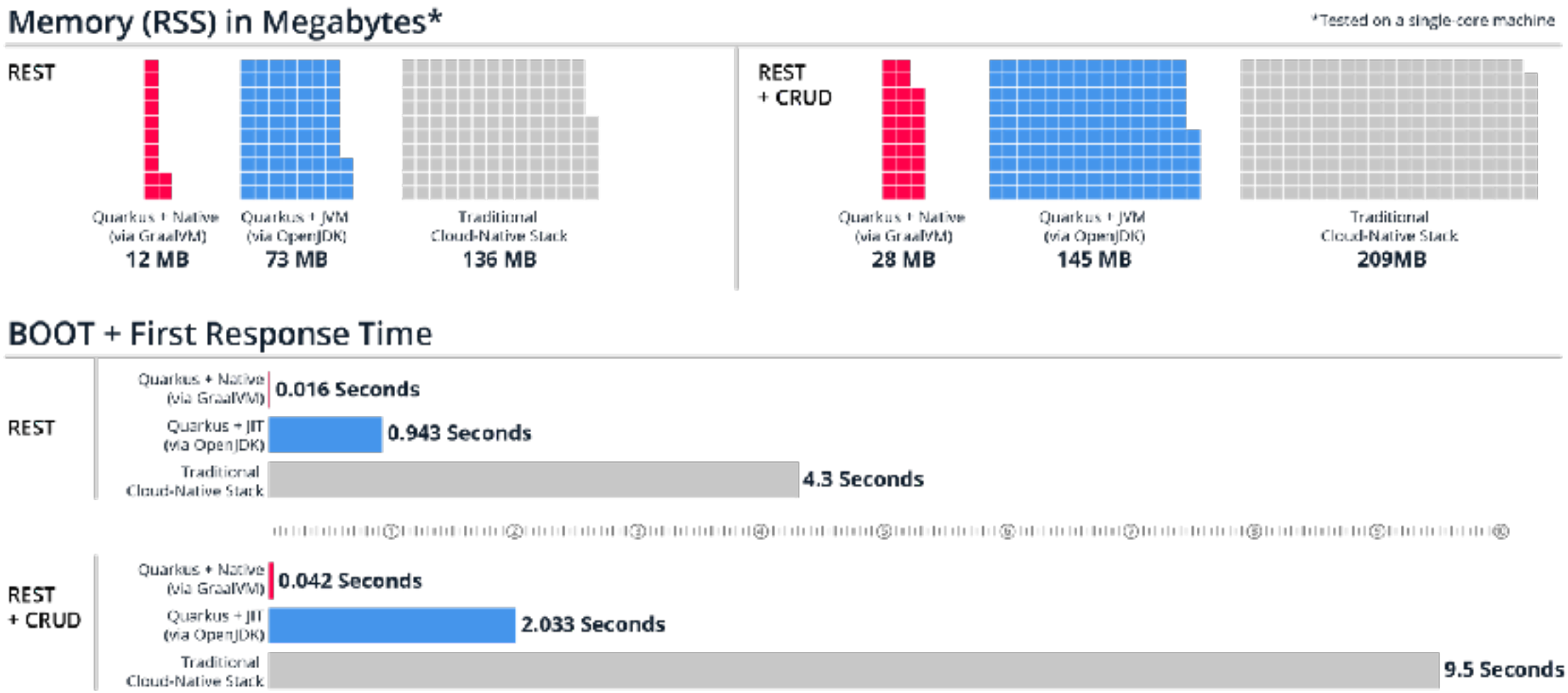
O QUE É QUARKUS?

O QUE É QUARKUS?

- Segundo o próprio site do projeto, "A Kubernetes Native Java stack tailored for OpenJDK HotSpot and GraalVM, crafted from the best of breed Java libraries and standards”
- Ou seja, o Quarkus em si mesmo é uma stack completa que permite o desenvolvimento de aplicações prontas para rodar em Kubernetes (de “containers first” a “Kubernetes Native”), com 100% de compatibilidade tanto com o OpenJDK quanto com compilação nativa (via GraalVM)
- Possibilita o desenvolvimento de todo tipo de aplicação: monólitos, microservices, serverless, command-line tools (CLI), event driven, web app, etc
- Traz em sua plataforma toda integração com os maiores projetos open source do mercado, como: Kubernetes, Camel, Hibernate, Langchain4j, OpenShift, Kafka... além de integração com os principais provedores de serviços, como: AWS, Google e Azure

PORQUE “SUPERSONIC" E “SUBATOMIC"?

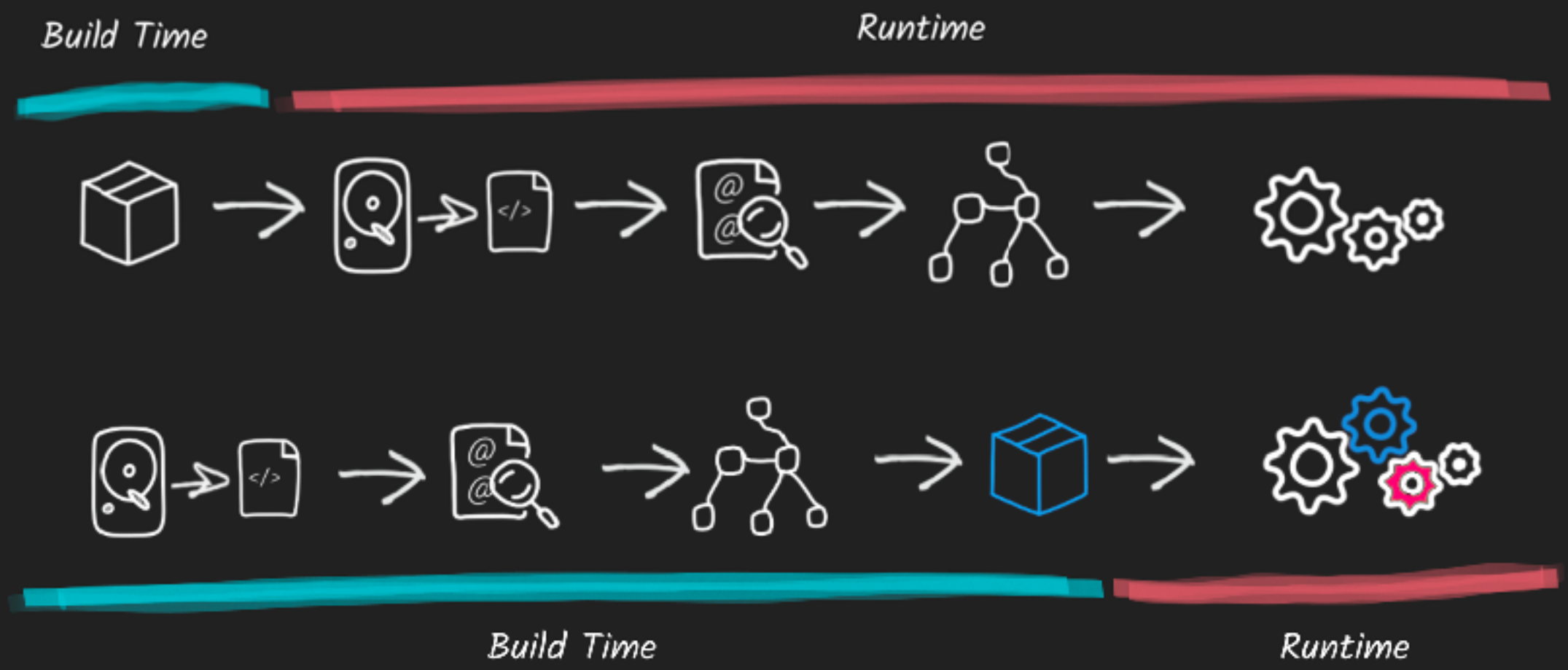
PORQUE “SUPERSONIC" E “SUBATOMIC"?



Fonte: https://quarkus.io

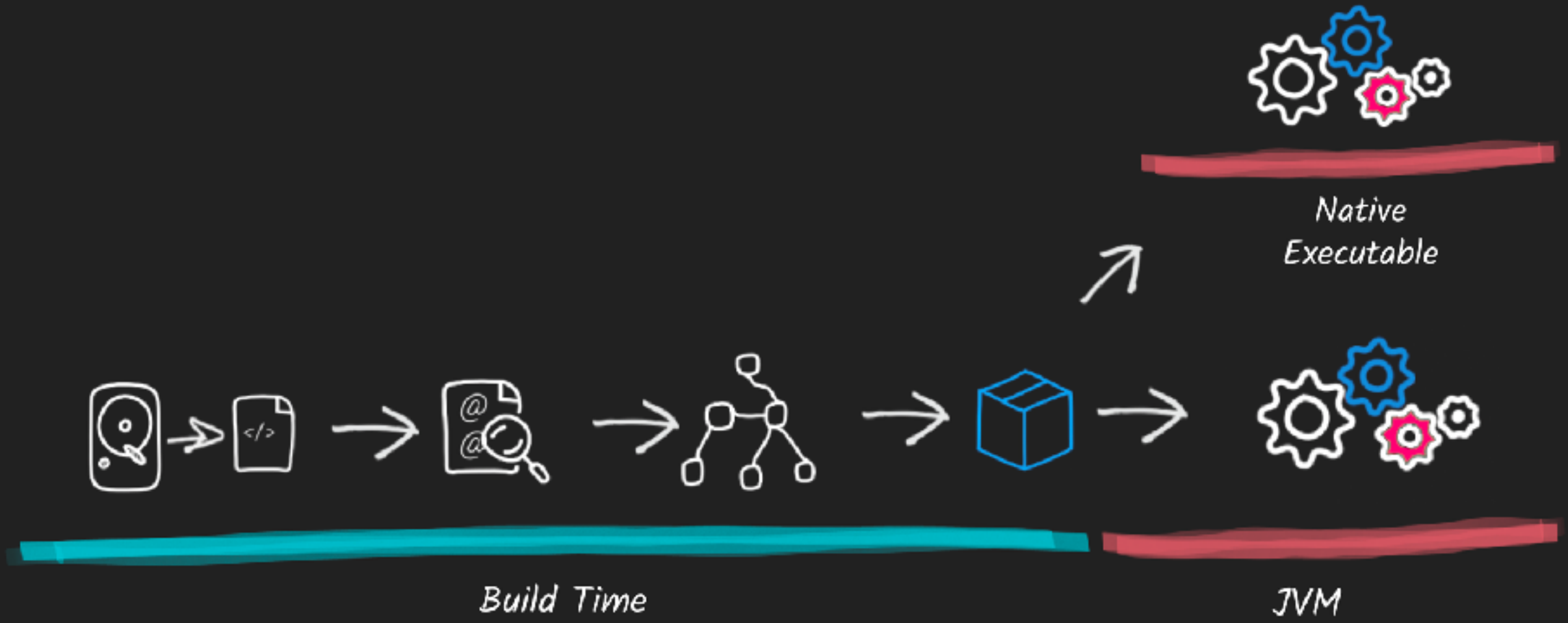
COMO O QUARKUS FAZ SUA “MÁGICA"?

COMO O QUARKUS FAZ SUA “MÁGICA"?



Fonte: https://dn.dev/quarkusmaster

E QUANTO À COMPILAÇÃO NATIVA?



Fonte: https://dn.dev/quarkusmaster

CLOUD NATIVE E ALÉM

CLOUD NATIVE
COMPUTING FOUNDATION

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

The Cloud Native Computing Foundation seeks to drive adoption of this paradigm by fostering and sustaining an ecosystem of open source, vendor-neutral projects. We democratize state-of-the-art patterns to make these innovations accessible for everyone.

Cloud Native Definition

Fonte: https://github.com/bloomberg/cncf-toc/blob/master/DEFINITION.md

CONTAINERS FIRST

- Tudo no Quarkus foi pensado visando o uso em ambientes de containers
  - Otimização de espaço em disco, visando imagens de containers menores
  - Otimização de tempo de startup, visando escalabilidade instantânea
  - Otimização de memória e CPU, visando maior densidade de deployment

KUBERNETES NATIVE

- Já que tudo é feito pra rodar em containers, então faz sentido fazer tudo rodar em Kubernetes
  - Métricas
  - Health check
  - Debugging
  - Tracing
  - ConfigMaps
  - Secrets
  - YAML

EXPERIÊNCIA DO USUÁRIO

- Usuário = Desenvolvedor
- Um dos poucos (talvez o único?) framework pensado para trazer uma experiência de uso realmente diferenciada
  - Live reload em frações de segundo
  - Dev Services (bancos de dados, Kafka, servidores de e-mail... tudo em seu ambiente local com setup automático via TestContainers)
  - Compilação native “out of the box” (não precisa pensar se vai funcionar ou não)

OPEN SOURCE

- Injeção de dependências via CDI
- REST endpoints via JAX-RS
- Persistência via JPA
- Transações via JTA
- IA via Langchain4j
- Etc, etc, etc...

SUA PRIMEIRA APLICAÇÃO USANDO QUARKUS

TE VEJO NA PRÓXIMA AULA!