

Google Cloud

# Next '24

## Java on Google Cloud:

The enterprise, the serverless,  
and the native

# Rustam Mehmandarov

Chief Engineer,  
Computas



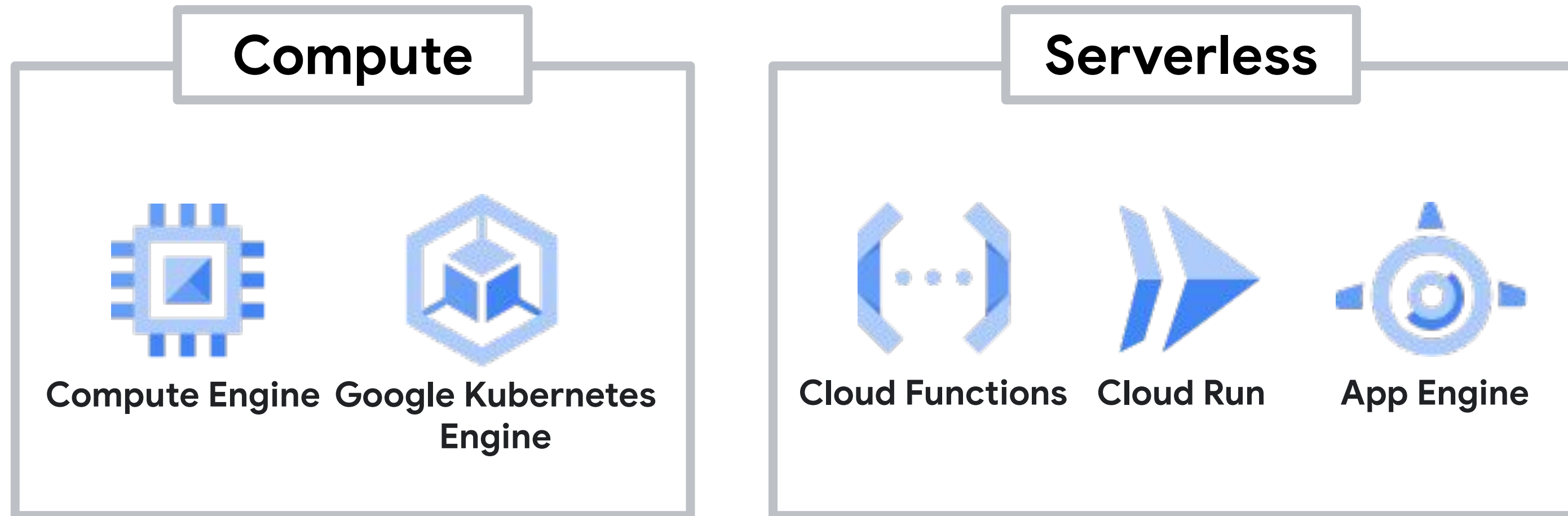
# Agenda

- 01 What are my Options?
- 02 Modern, Enterprise Java
- 03 Go Serverless?
- 04 Optimization  
*Start-up, Native, CRaC, etc.*
- 05 Some Final Thoughts



# What are my Options?

# Running Java Applications



# Focus on: Cloud Run

- ✓ Containerisation
- ✓ Flexibility: Runtimes, optimizations, ++
- ✓ Portability from Knative

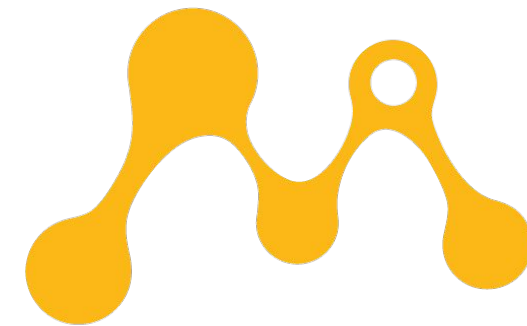


# Modern, Enterprise Java

# Cloud-Native applications



JAKARTA<sup>®</sup> EE



MICROPROFILE<sup>®</sup>



# Jakarta EE 10 Platform

## Jakarta EE 10 Web Profile

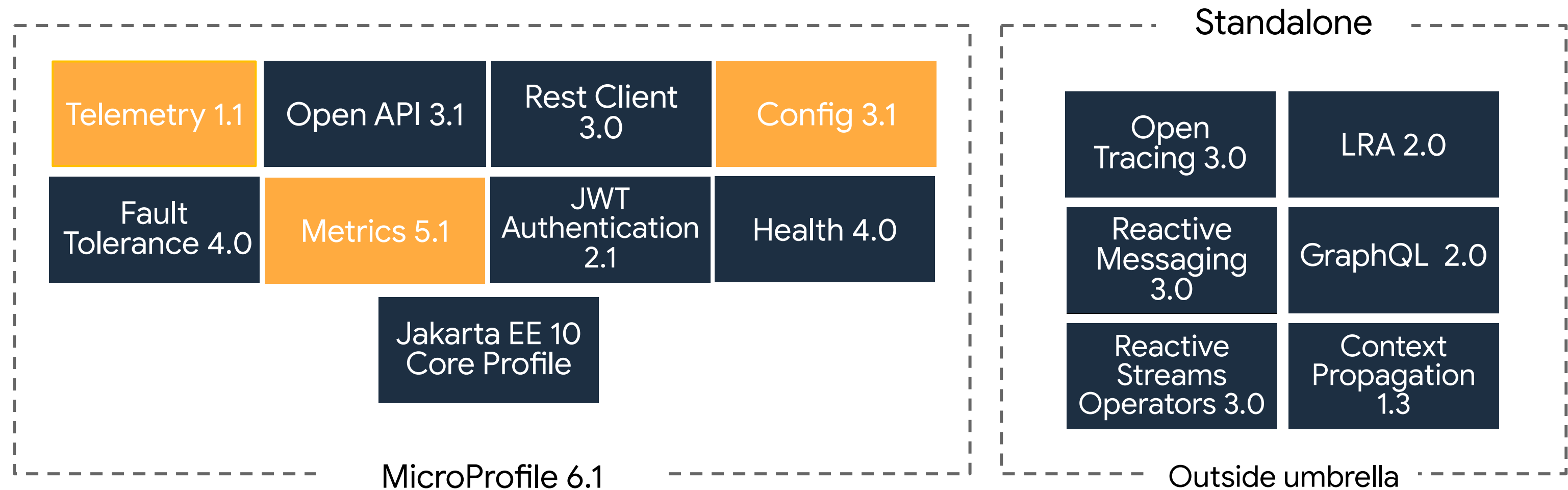
Authorization 2.1	Expression Language 5.0	Server Pages 3.1
Activation 2.1	Authentication 3.0	CDI 4.0
Batch 2.1	Concurrency 3.0	WebSocket 2.1
Connectors 2.1	Persistence 3.1	Bean Validation 3.0
Mail 2.1	Faces 4.0	Debugging Support 2.0
Messaging 3.1	Security 3.0	Enterprise Beans Lite 4.0
Enterprise Beans 4.0	Servlet 6.0	Managed Beans 2.0
	Standard Tag Libraries 3.0	Transactions 2.0

## Jakarta EE 10 Core Profile

CDI Lite 4
JSON Binding 3.0
Annotations 2.1
Interceptors 2.1
Restful Web Services 3.1
Json Processing 2.1
Dependency Injection 2.0

■ Updated ■ Not Updated ■ New

# MicroProfile



□ = New    ■ = Updated    ■ = No change from last release

# Cloud-Native code examples

1

Basic Example:

<https://github.com/mehmandarov/randomstrings>

2

Advanced Example:

<https://github.com/mehmandarov/5-features-talk-demo>

# Let's go Serverless

# Serverless

- ✓ To me: “*Manageless*”
- ✓ Do you *really* need idling infrastructure?
- ✓ Running *costs*: pros and cons
- ✓ Need to think about *startup* times
- ✓ Flexibility && Control over containers

# Optimizations:

Startup times,

Native images,

CRaC, ++

# Runtimes differ

*Some examples*



**Open Liberty**

<https://openliberty.io/>



**Helidon**

<https://helidon.io/>



**Apache TomEE**

<https://tomee.apache.org/>



**Quarkus**

<https://quarkus.io/>

# Container images



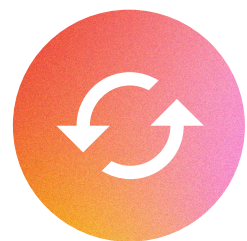
## Image size

Minimal images, multi-stage builds, etc.



## Image build time

Time to build a new image  
vs. number of deploys  
vs. resources it takes to build it.



## Image lifecycle

What happens to your image when scaling

<https://cloud.google.com/blog/topics/developers-practitioners/lifecycle-container-cloud-run>



## Secure Software Lifecycle

Need to scan, update, and redeploy  
container images.



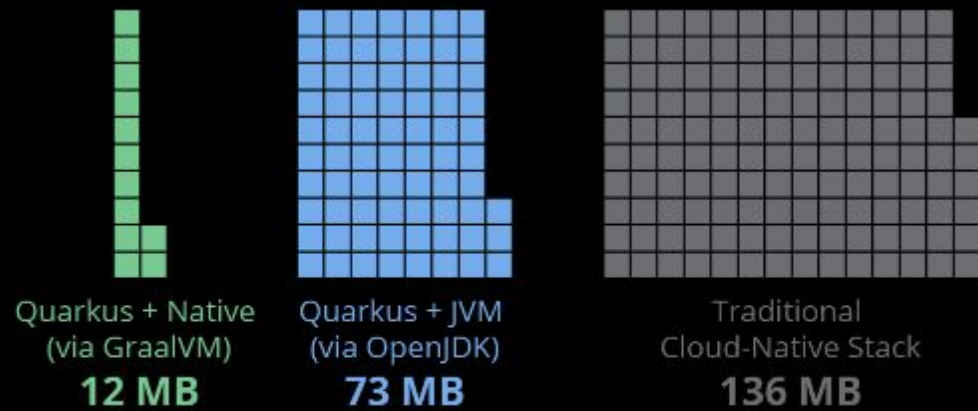
# 1. Quarkus + Cloud Run

Quarkus offers unequaled performance

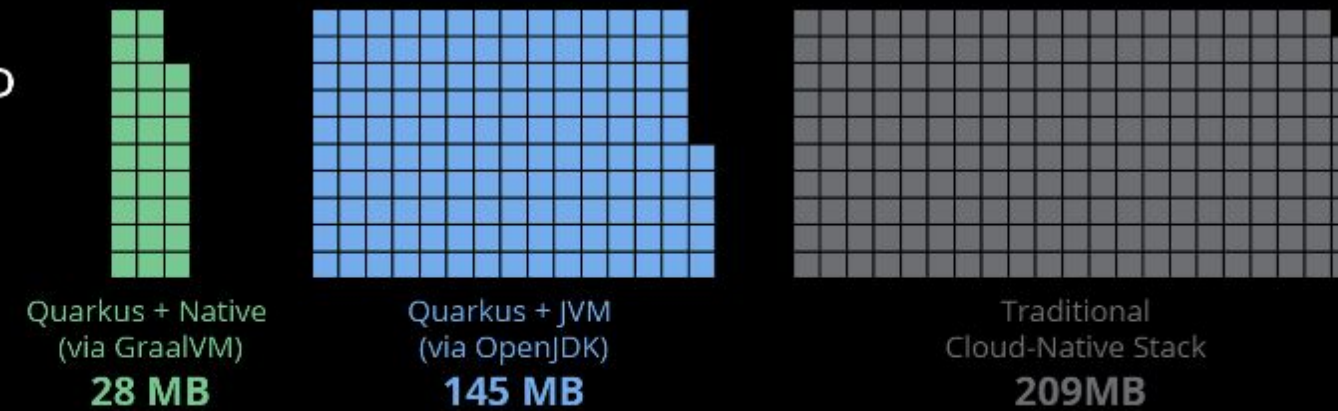
Memory (RSS) in Megabytes\*

\*Tested on a single-core machine

REST



REST + CRUD



BOOT + First Response Time

REST



REST + CRUD



# 2. Native Images with GraalVM



QUARKUS

GraalVM™



Even faster start-up time



Smaller footprint (no JVM)



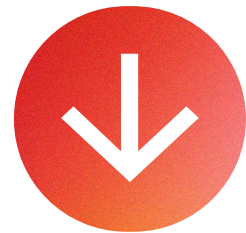
No JVM optimizations



Longer build time && bigger server

# 3. CRaC

## Coordinated Restore at Checkpoint



Source: Gerrit Grunwald, Azul



OpenJDK project



Based on Linux kernel project: **CRIU**



Create checkpoints using code (**API**) or **jcmd**



**Linux** only: X64 / AARCH64



Still needs a **specific** JDK build (for now)

# Optimization Code Examples

- 1 Changing Runtime:  
<https://github.com/mehmandarov/randomstrings> -> *“Local Build and Run”*
- 2 GraalVM native images:  
<https://github.com/mehmandarov/Randomstrings> -> *“Build, Add, Deploy”*
- 3 CRaC:  
<https://github.com/mehmandarov/Randomstrings> -> *“CRaC”*

# Some Final Thoughts

# Java on Google Cloud



## Runtime

Runtimes *differ*.

Many variables to consider:

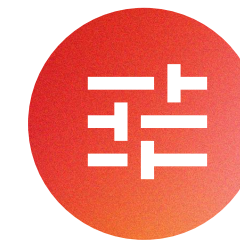
- Startup times.
- Runtime footprint.
- Support for Jakarta EE, Spring, Micronaut, etc.



## Where to deploy

Choose the *right offering* for your app.

- Do you need a VM?
- A k8s cluster?
- Serverless? Or a Function?



## Optimize

Scaling to zero and autoscaling in general *needs optimizations*.

- Start-up times
- Container size
- Build time and resources

# Ready to build what's next?

Tap into **special offers** designed to help you **implement what you learned** at Google Cloud Next.

**Scan the code** to receive personalized guidance from one of our experts.



Or visit [g.co/next/24offers](https://g.co/next/24offers)

# Thank you

Follow me:

<https://rustam.no>

