

KARSTEN SILZ, 22. MÄRZ 2023

# WANN LOHNT SICH NATIVE JAVA MIT GRAALVM FÜR MICH?

**JAVALAND 2023**



**WANN LOHNT SICH  
NATIVE JAVA MIT  
GRAALVM FÜR MICH?**

FÜR DIE MEISTEN:

JETZT NICHT

# ZUSAMMENFASSUNG

IN DER CLOUD IST  
JAVA **TEURER** ALS  
KONKURRENTEN WIE GO

**NATIVE JAVA MACHT**

**JAVA BILLIGER,**

**BRAUCHT ABER...**

...MEHR ENTWICKLER-  
ZEIT, UM OPERATIVE  
KOSTEN ZU SPAREN...

**...UND IST MANCHMAL**

**UNMÖGLICH ODER**

**UNWIRTSCHAFTLICH**



# SPRING BOOT 2 $\Rightarrow$ QUARKUS: STARTUP

**600** MS  $\Rightarrow$  **10** MS

**107** MB  $\Rightarrow$  **6.7** MB

**MICRO-SERVICES IN**

**KUBERNETES:**

**MEHR PODS/NODE (SO  
NICHT CPU-LIMITIERT)**

**FRAMEWORK: SPRING  
BOOT 3, QUARKUS &  
MICRONAUT**

**WAS MACHT MICH  
ZUM EXPERTEN?**

**EDITOR EINER 6-TEILIGEN INFOQ-  
ARTIKEL-SERIE ÜBER NATIVE JAVA**

**DATEN ALS BEWEIS**

**JAVA-EXPERTEN HALFEN**

**NEUTRAL – NICHT MIT PROJEKTEN  
VERBUNDEN, VERKAUFE NIX**



**BETTER  
PROJECTS  
FASTER**

# FOLIEN & MEHR



**BPELI/LAM**

# AGENDA

PROBLEM? UND WARUM JETZT?

LÖSUNG: NATIVE JAVA

LÖSUNG: JIT JAVA

WANN LOHNT ES SICH?



**PROBLEM?**

**"THE LONG-TERM PAIN POINTS OF JAVA'S SLOW  
STARTUP TIME, SLOW TIME TO PEAK  
PERFORMANCE, AND LARGE FOOTPRINT"**

**JAVA LANGUAGE ARCHITECT MARK REINHOLD,  
ORACLE, APRIL 2020**

**SLOW** STARTUP,  
**SLOW** TIME TO PEAK  
PERFORMANCE,  
**LARGE** FOOTPRINT

**WARUM?**

# SO FUNKTIONIERT JAVA

# JAVA-COMPILER

QUELL-  
CODE

BYTE-  
CODE

BUILD TIME

JVM

JIT COMPILER

BYTE-  
CODE

KLASSEN-  
LISTE

INIT JDK &  
FRAMEWORK

INIT  
APP

...

MASCHI-  
NEN-CODE

RUNTIME

**SLOW** STARTUP,  
**SLOW** TIME TO PEAK  
PERFORMANCE,  
**LARGE** FOOTPRINT

**SLOW** STARTUP,  
**SLOW** TIME TO PEAK  
PERFORMANCE,  
**LARGE** FOOTPRINT



# JAVA-COMPILER

QUELL-CODE

BYTE-CODE

IMMER WIEDER MIT IMMER  
GLEICHEM ERGEBNIS:  
VIELE JAVA-OBJEKTE

BUILD TIME

# JVM

# JIT COMPILER

BYTE-CODE

KLASSEN-LISTE

INIT JDK &  
FRAMEWORK

INIT  
APP

...

MASCHI-NEN-CODE

RUNTIME

**SLOW STARTUP,  
SLOW TIME TO PEAK  
PERFORMANCE,  
LARGE FOOTPRINT**

# JAVA-COMPILER

QUELL-  
CODE

BYTE-  
CODE

BUILD TIME

PROFILING,  
SCHNELLE COMPILIERUNG (C1),  
PROFILING,  
OPTIMIERTE COMPILIERUNG (C2)

# JVM

# JIT COMPILER

BYTE-  
CODE

KLASSEN-  
LISTE

INIT JDK &  
FRAMEWORK

INIT  
APP

...

MASCHI-  
NEN-CODE

RUNTIME

**SLOW** STARTUP,  
**SLOW** TIME TO PEAK  
PERFORMANCE,  
**LARGE** FOOTPRINT

# JAVA-COMPILER

QUELL-  
CODE

BYTE-  
CODE

PROFILER & COMPILER  
LÄUFT MIT

BUILD TIME

JVM

JIT COMPILER

BYTE-  
CODE

KLASSEN-  
LISTE

INIT JDK &  
FRAMEWORK

INIT  
APP

...

MASCHI-  
NEN-CODE

RUNTIME

**WARUM JETZT?**

**KUBERNETES**  
**MICROSERVICES**  
**CLOUD**

## **KUBERNETES**

**BACK-ENDS IN ALLEN  
PROGRAMMIER-  
SPRACHEN, ERSETZT  
APP SERVER**

## **MICROSERVICES**

**NEUE SPRACHEN &  
FRAMEWORK MIT  
KLEINEM RISIKO  
TESTEN**

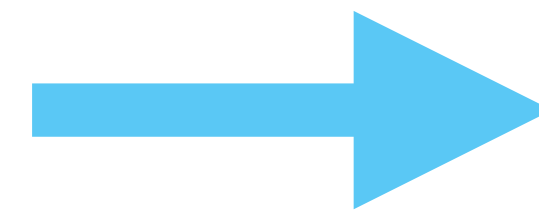
## **CLOUD**

**MEHR SPEICHER &  
STARTUP-ZEIT IST OFT  
TEURER**



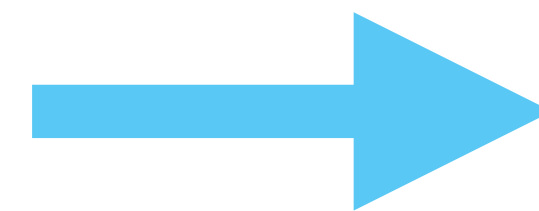
# JAVA-UMFELD

**UNIX-VARIANTEN, WINDOWS,  
MAINFRAME**



**LINUX-CONTAINER AUF LINUX**

**LANG-LAUFENDE  
APPLIKATIONEN MIT VIEL  
SPEICHER (JAVA EE  
APPLICATION SERVERS)**



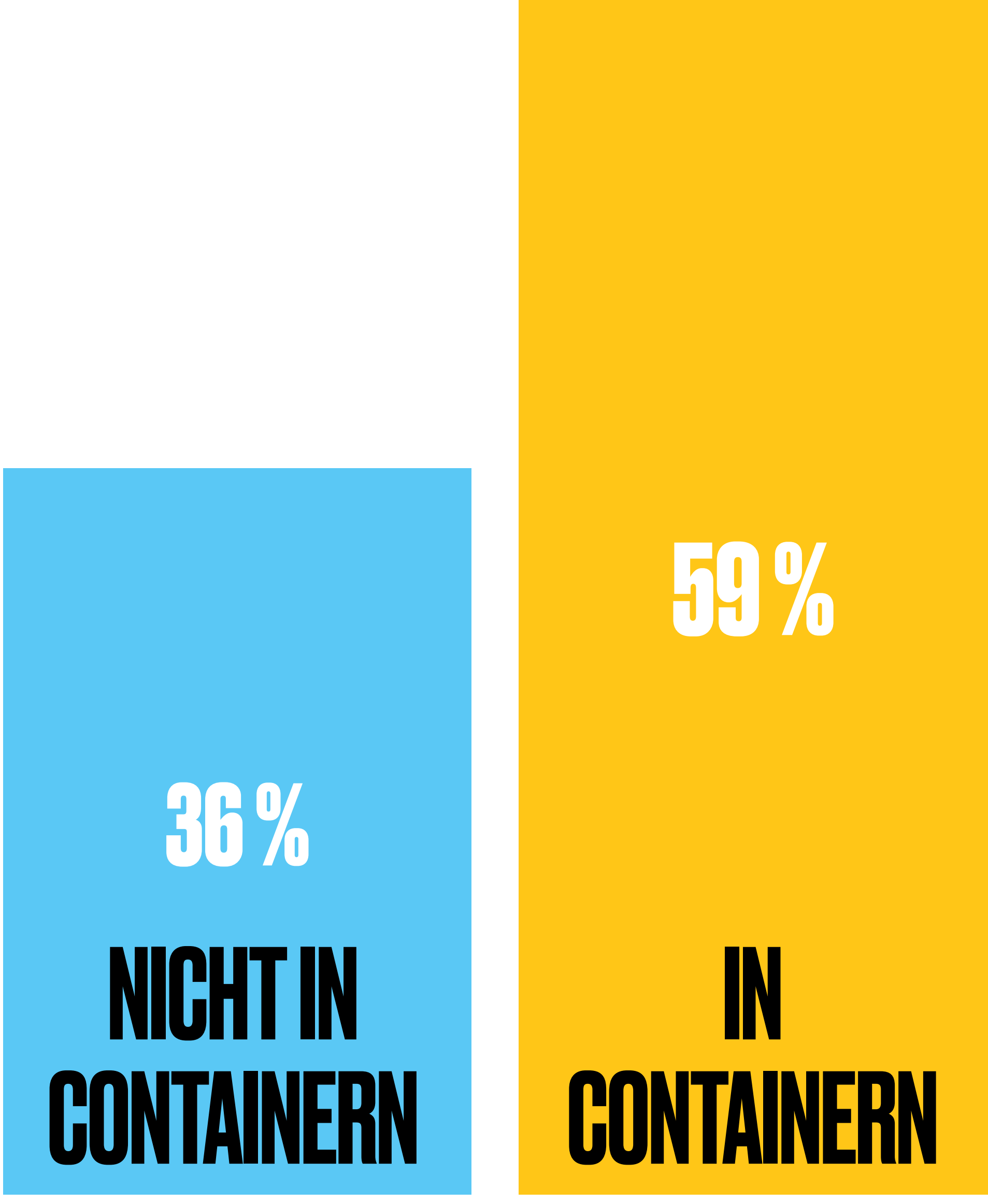
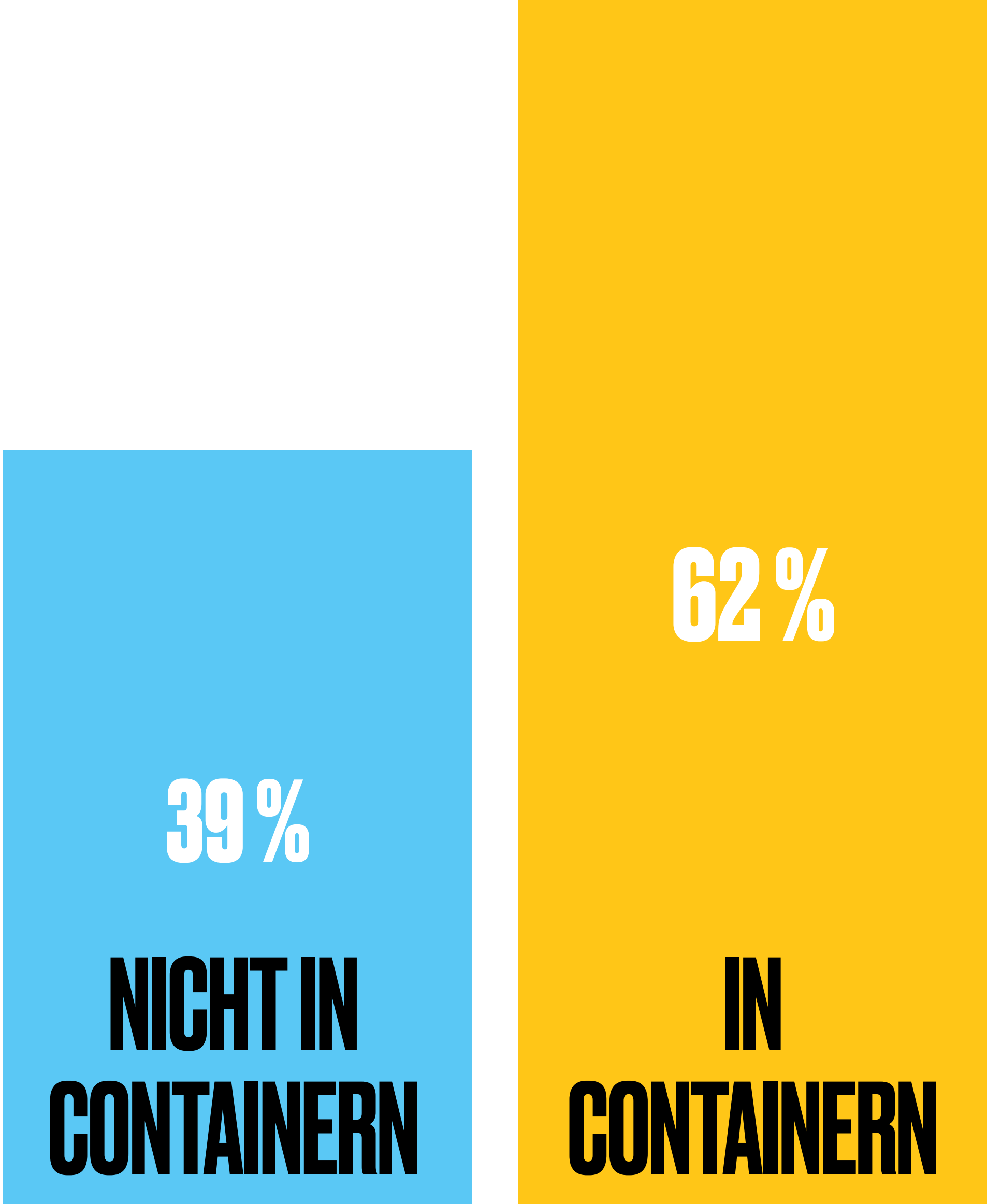
**MICROSERVICES MIT WENIG  
SPEICHER, ZUNEHMEND IN  
KUBERNETES**

**NEW RELIG:**

**"DATA FROM MILLIONS OF APPLICATIONS" (APR 2022)**

# 1-2 CPU-KERNE

≤ 1 GB RAM



**WENIGER** SPEICHER &  
CPU-KERNE

**BEDROHUNG FÜR**  
**JAVAS ENTERPRISE-**  
**DOMINANZ!**

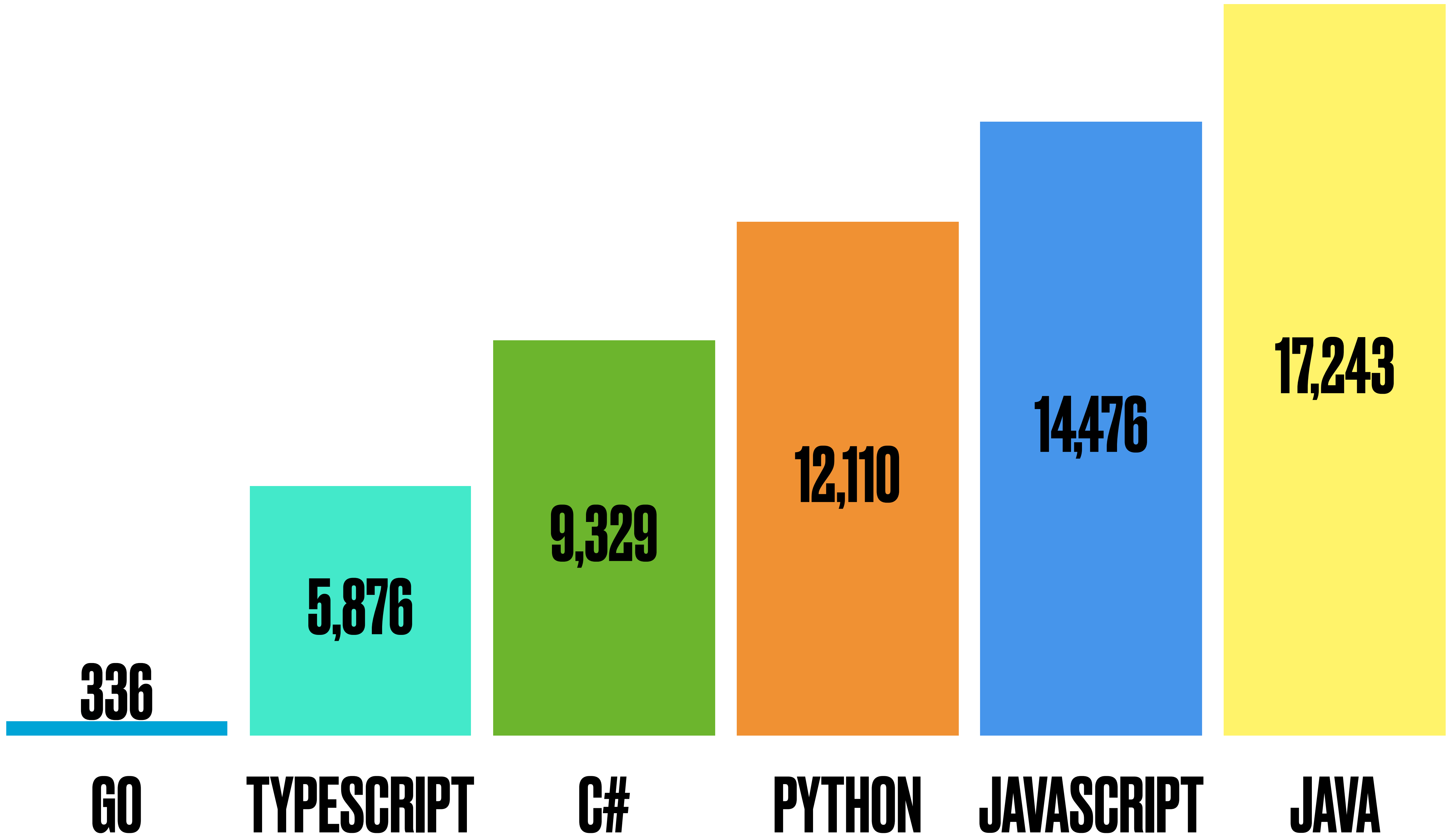
**MÜSSEN WIR JAVA-  
ENTWICKLER JETZT**

**GO LERNEN?**



NENNUNGEN IN **JOB-**  
**ANZEIGEN** BEI INDEED.DE  
ALS NACHFRAGE-PROXY  
(FEB 2022)





**NEIN**



**ZWISCHENFAZIT**

**SLOW** STARTUP,  
**SLOW** TIME TO PEAK  
PERFORMANCE,  
**LARGE** FOOTPRINT

...WEGEN **LANG**-

LAUFENDER PROZESSE

MIT **VIEL** SPEICHER

TREND: WENIGER  
SPEICHER & CPU, MEHR  
KONKURRENZ FÜR JAVA

**PROBLEM? UND  
WARUM JETZT?**



# AGENDA

~~PROBLEM? UND WARUM JETZT?~~

LÖSUNG: NATIVE JAVA

LÖSUNG: JIT JAVA

WANN LOHNT ES SICH?

**LÖSUNG: NATIVE JAVA**

**NATIVE JAVA?**

**"NATIVE JAVA" =**

**"STATIC JAVA" =**

**"AOT JAVA"**

**"GRAALVM"**

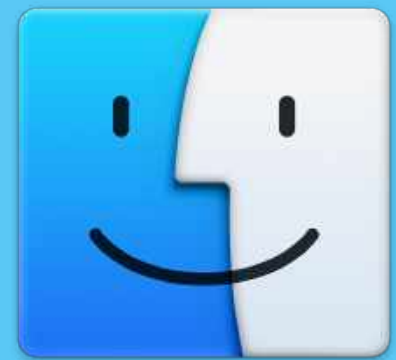
# STATT 1X BYTECODE...

```
java -jar my-app-1.2.jar
```

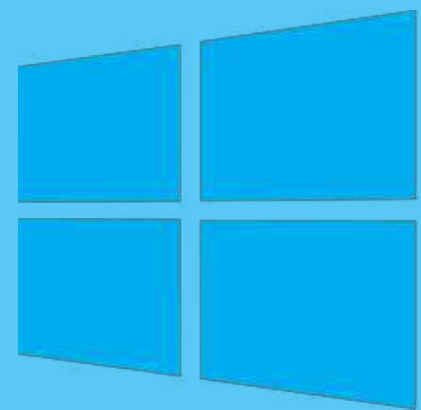
# ...3 AUSFÜHRBARE PROGRAMME!



`./my-app-1.2-runner`



`./my-app-1.2-runner`



`my-app-1.2-runner.exe`

WIE STARTUP-ZEIT UND  
SPEICHER-VERBRAUCH  
REDUZIEREN?

# SPRING BOOT 2 $\Rightarrow$ QUARKUS: STARTUP

**600** MS  $\Rightarrow$  **10** MS

**107** MB  $\Rightarrow$  **6.7** MB



**"DEMO MICROSERVICE":**

**80 BILDER ZU PDF, SPRING  
BOOT 2 & 3 UND QUARKUS**

# JAVA-COMPILER

QUELL-  
CODE

BYTE-  
CODE

## BUILD TIME

# JVM

# JIT COMPILER

BYTE-  
CODE

KLASSEN-  
LISTE

INIT JDK &  
FRAMEWORK

INIT  
APP

...

MASCHI-  
NEN-CODE

## RUNTIME



**RUNTIME  $\Rightarrow$  BUILD TIME**

**WENIGER SPEICHER**

# JAVA-COMPILER

QUELL-  
CODE

BYTE-  
CODE

## BUILD TIME

# JVM

# JIT COMPILER

BYTE-  
CODE

KLASSEN-  
LISTE

INIT JDK &  
FRAMEWORK

INIT  
APP

...

MASCHI-  
NEN-CODE

## RUNTIME

# JAVA-COMPILER

QUELL-  
CODE

BYTE-  
CODE

## BUILD TIME

# JVM

# JIT COMPILER

BYTE-  
CODE

KLASSEN-  
LISTE

INIT JDK &  
FRAMEWORK

INIT  
APP

...

MASCHI-  
NEN-CODE

## RUNTIME

**JAVA-COMPILER**

**QUELL-  
CODE**

**BYTE-  
CODE**

**JIT COMPILER**

**MASCHI-  
NEN-CODE**

**BUILD TIME**

**JVM**

**BYTE-  
CODE**

**KLASSEN-  
LISTE**

**INIT JDK &  
FRAMEWORK**

**INIT  
APP**

...

**RUNTIME**



**JAVA-COMPILER**

**QUELL-  
CODE**

**BYTE-  
CODE**

**JIT COMPILER**

**MASCHI-  
NEN-CODE**

**BUILD TIME**

**JVM**

**BYTE-  
CODE**

**KLASSEN-  
LISTE**

**INIT JDK &  
FRAMEWORK**

**INIT  
APP**

...

**RUNTIME**

**JAVA-COMPILER**

**QUELL-  
CODE**

**BYTE-  
CODE**

**AOT COMPILER**

**MASCHI-  
NEN-CODE**

**BUILD TIME**

**JVM**

**BYTE-  
CODE**

**KLASSEN-  
LISTE**

**INIT JDK &  
FRAMEWORK**

**INIT  
APP**

...

**RUNTIME**

**JAVA-COMPILER**

**QUELL-  
CODE**

**BYTE-  
CODE**

**BUILD TIME**

**AOT: AHEAD-OF-TIME  
(ÄHNELT C++-COMPILER)**

**AOT COMPILER**

**MASCHI-  
NEN-CODE**

**JVM**

**BYTE-  
CODE**

**KLASSEN-  
LISTE**

**INIT JDK &  
FRAMEWORK**

**INIT  
APP**

...

**RUNTIME**

**JAVA-COMPILER**

**QUELL-  
CODE**

**BYTE-  
CODE**

**AOT COMPILER**

**MASCHI-  
NEN-CODE**

**BUILD TIME**

**JVM**

**BYTE-  
CODE**

**KLASSEN-  
LISTE**

**INIT JDK &  
FRAMEWORK**

**INIT  
APP**

...

**RUNTIME**

**JAVA-COMPILER**

**QUELL-  
CODE**

**BYTE-  
CODE**

**AOT COMPILER**

**MASCHI-  
NEN-CODE**

**BUILD TIME**

**JVM**

**MASCHI-  
NEN-CODE**

**KLASSEN-  
LISTE**

**INIT JDK &  
FRAMEWORK**

**INIT  
APP**

**RUNTIME**

**JAVA-COMPILER**

**AOT COMPILER**

**QUELL-  
CODE**

**BYTE-  
CODE**

**KLASSEN-  
LISTE**

**MASCHI-  
NEN-CODE**

**BUILD TIME**

**JVM**

**MASCHI-  
NEN-CODE**

**INIT JDK &  
FRAMEWORK**

**INIT  
APP**

**RUNTIME**

WIE SPART "WENIGER  
SPEICHER" **KOSTEN?**

**KUBERNETES:**

**MEHR PODS/NODE (SO  
NICHT CPU-LIMITIERT)**



**SCHNELLERER START**

**JAVA-COMPILER**

**AOT COMPILER**

**QUELL-  
CODE**

**BYTE-  
CODE**

**KLASSEN-  
LISTE**

**MASCHI-  
NEN-CODE**

**BUILD TIME**

**JVM**

**MASCHI-  
NEN-CODE**

**INIT JDK &  
FRAMEWORK**

**INIT  
APP**

**RUNTIME**

# JAVA-COMPILER

QUELL-CODE

BYTE-CODE

KLASSEN-LISTE

# AOT COMPILER

INIT JDK & FRAMEWORK

INIT APP

MASCHINEN-CODE

## BUILD TIME

JVM

MASCHINEN-CODE

INIT JDK & FRAMEWORK

INIT APP

## RUNTIME

# JAVA-COMPILER

QUELL-CODE

BYTE-CODE

KLASSEN-LISTE

# AOT COMPILER

VIEL INIT JDK & FRAMEWORK

VIEL INIT APP

MASCHINEN-CODE

## BUILD TIME

JVM

MASCHINEN-CODE

INIT JDK & FRAMEWORK

INIT APP

## RUNTIME

# JAVA-COMPILER

QUELL-CODE

BYTE-CODE

KLASSEN-LISTE

# AOT COMPILER

VIEL INIT JDK & FRAMEWORK

VIEL INIT APP

MASCHINEN-CODE

## BUILD TIME

JVM

MASCHINEN-CODE

WENIG INIT JDK & FRAMEWORK

WENIG INIT APP

## RUNTIME

WIE SPART  
"SCHNELLERER  
START" **KOSTEN?**

WENIGER **STANDBY-**  
SERVER/-NODES, WEIL  
**SCHNELLERER** START

ICH GLAUBE: "WENIGER  
**SPEICHER**" SPART MEHR  
ALS "SCHNELLERER START"



**GRAALVM?**

3 VERSCHIEDENE

PROJEKTE – HIER: NATIVE

IMAGE ALS AOT COMPILER

AB 2023 TEIL VON  
**OPENJDK** – GEMEINSAME  
RELEASES

**SCHNELLERER** START

**WENIGER** SPEICHER


**KLEINERE** DATEIEN

**BESSERE** SICHERHEIT

WO IST DER **HAKEN**?

**CWA!**

**CWA?**

A photograph of two male wrestlers in a blue and red singlet embracing on a wrestling mat. The background is blurred, showing spectators and arena lighting. The text 'CZECH WRESTLING ASSOCIATION?' is overlaid in large white letters.

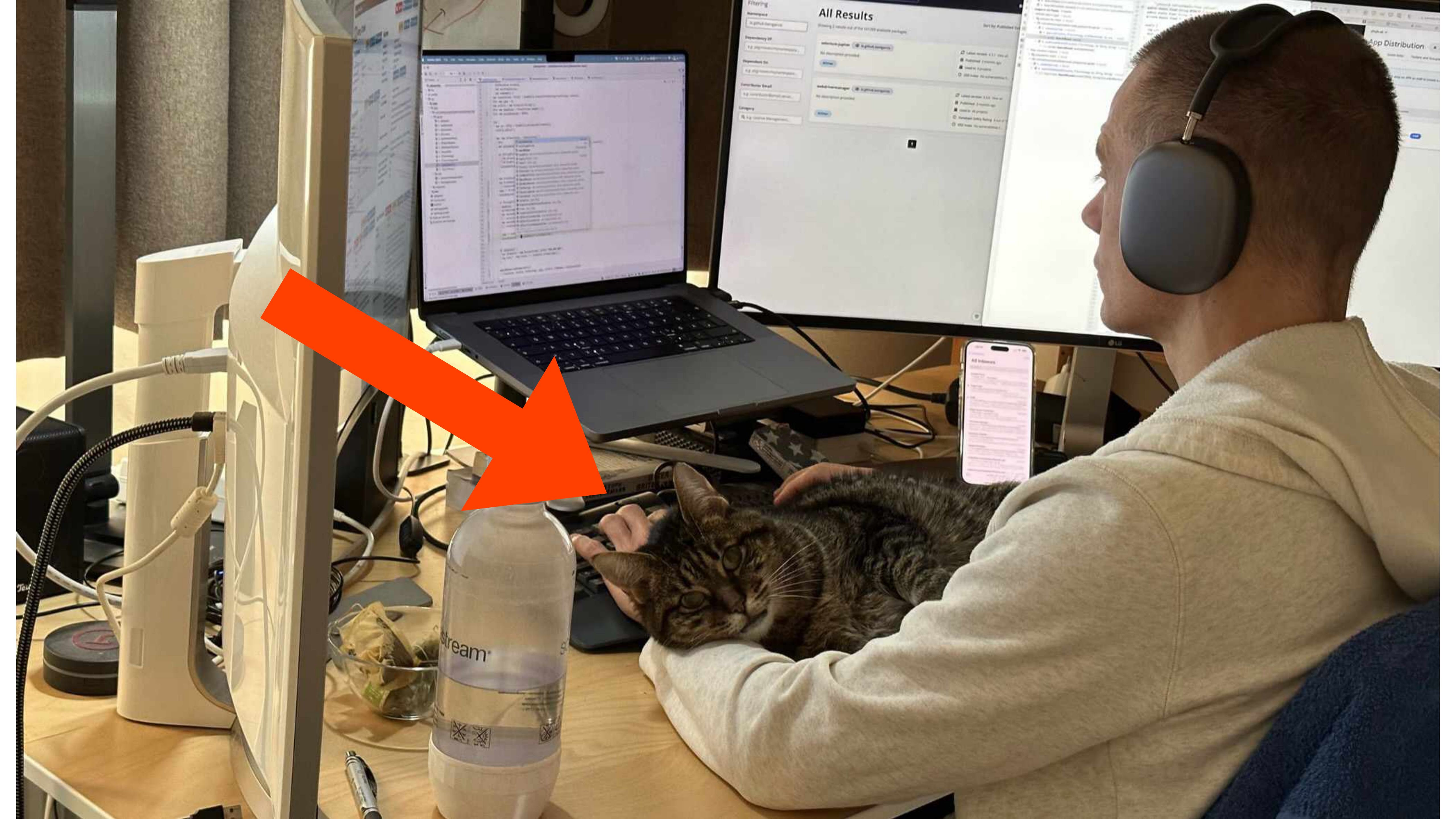
# CZECH WRESTLING ASSOCIATION?



How and Why to Add Cats  
to Your Fiction Writing



CAT WRITING  
ASSOCIATION?





**JAVA-COMPILER**

**QUELL-  
CODE**

**BYTE-  
CODE**

**KLASSEN-  
LISTE**

**AOT COMPILER**

**VIEL INIT JDK &  
FRAMEWORK**

**VIEL INIT  
APP**

**MASCHI-  
NEN-CODE**

**BUILD TIME**

**JVM**

**MASCHI-  
NEN-CODE**

**WENIG INIT JDK &  
FRAMEWORK**

**WENIG INIT  
APP**

**RUNTIME**

# CWA: ZUR BUILD TIME

## ALLE KLASSEN BEKANNT

AUS UNSERER APPLIKATION,  
UNSEREN BIBLIOTHEKEN, DEM  
FRAMEWORK & DEM JDK

## ALLE RESSOURCEN DA

ALLE DATEIEN, DIE WIR LADEN  
WOLLEN - PROPERTY FILES,  
RESOURCE BUNDLE, ETC.

**NATIVE IMAGE ENTFERNT**

**ALLES, WAS NICHT**

**GEBRAUCHT WIRD**

...UND IST MANCHMAL

UNMÖGLICH

ODER

UNWIRTSCHAFTLICH

WAS IST MIT GRAALVM

MÖGLICH - UND

WAS NICHT?



# MÖGLICH ODER NICHT?

## IMMER UNMÖGLICH

NEUE KLASSEN &  
METHODEN ZUR  
LAUFZEIT ERZEUGEN

BELIEBIGEN  
BYTECODE  
AUSFÜHREN

## TEILWEISE & IN ARBEIT

AWT (NICHT AUF MACOS)

JMX

JAVA FLIGHT RECORDER

TEST-FRAMEWORKS

## IMMER MÖGLICH

REFLECTION & CLASS  
LOADING - ABER MIT  
KONFIGURATION

SUBSTRATE VM: GARBAGE  
COLLECTION, THREAD  
MANAGEMENT

CLASSLOADING &

REFLECTION: LAUFZEIT-

**CRASH**, WENN KLASSE **FEHLT**

AUCH BIBLIOTHEKEN  
KÖNNEN UNMÖGLICHE  
SACHEN MACHEN...

**...UND BRAUCHEN**  
**KONFIGURATION**

...MEHR ENTWICKLER-

ZEIT, UM OPERATIVE

KOSTEN ZU SPAREN...

**WARUM MEHR  
ENTWICKLER-ZEIT?**

# MEHR ENTWICKLER-ZEIT

## KONFIGURATION

REFLECTION & CLASS  
LOADING: MANUELL ODER  
PER TRACING AGENT  
(BELAUSCHT JIT-JAVA)

EVTL. BUILD-TIME-INIT  
VERBIETEN

## BUILD & DEPLOYMENT

BUILDS DAUERN VIEL LÄNGER  
UND ERZEUGEN 1 PROGRAMM  
PRO PLATTFORM (KEINE  
CROSS-COMPILIERUNG)

DEBUGGING AUF MACOS &  
WINDOWS DAUERT VIEL  
LÄNGER (LINUX-UMWEG)

## MONITORING & PRODUCTION

EINIGE OBSERVABILITY  
TOOLS LAUFEN GAR NICHT,  
ANDERE LIEFERN WENIGER  
DATEN

TROUBLESHOOTING  
DAUERT LÄNGER

**BEST PRACTICE:**  
**ENTWICKLUNG MIT**  
**JIT JAVA**



# ENTWICKLUNG FÜR NATIVE JAVA

## MEISTENS

ENTWICKLER NEHMEN JIT JAVA -  
WIE SONST AUCH

CI/CD PIPELINE BAUT NATIVE  
JAVA-PROGRAMME

## SELTEN

ENTWICKLER BAUEN NATIVE JAVA-  
PROGRAMME SELBST "VOR GROßEN  
MERGE"

DEBUGGING AUF MAC & WINDOWS -  
NERVT!

**GRAALVM EMPFIEHLT:**

**FRAMEWORK NEHMEN**

# FRAMEWORKS MIT GRAALVM PRODUCTION SUPPORT

## WARUM FRAMEWORKS?

KONFIGURIEREN NATIVE IMAGE &  
BIBLIOTHEKEN

HABEN KOMPATIBLE BIBLIOTHEKEN

ZUSATZ-FEATURES (Z.B. HOT CODE  
RELOAD MIT QUARKUS)

## WELCHE FRAMEWORKS?

NEU: QUARKUS & MICRONAUT  
– NICHT HELIDON, KAUM JOB-NACHFRAGE

ALT: SPRING BOOT 3.0 (NOV. 2022)

JAKARTA EE: NEIN

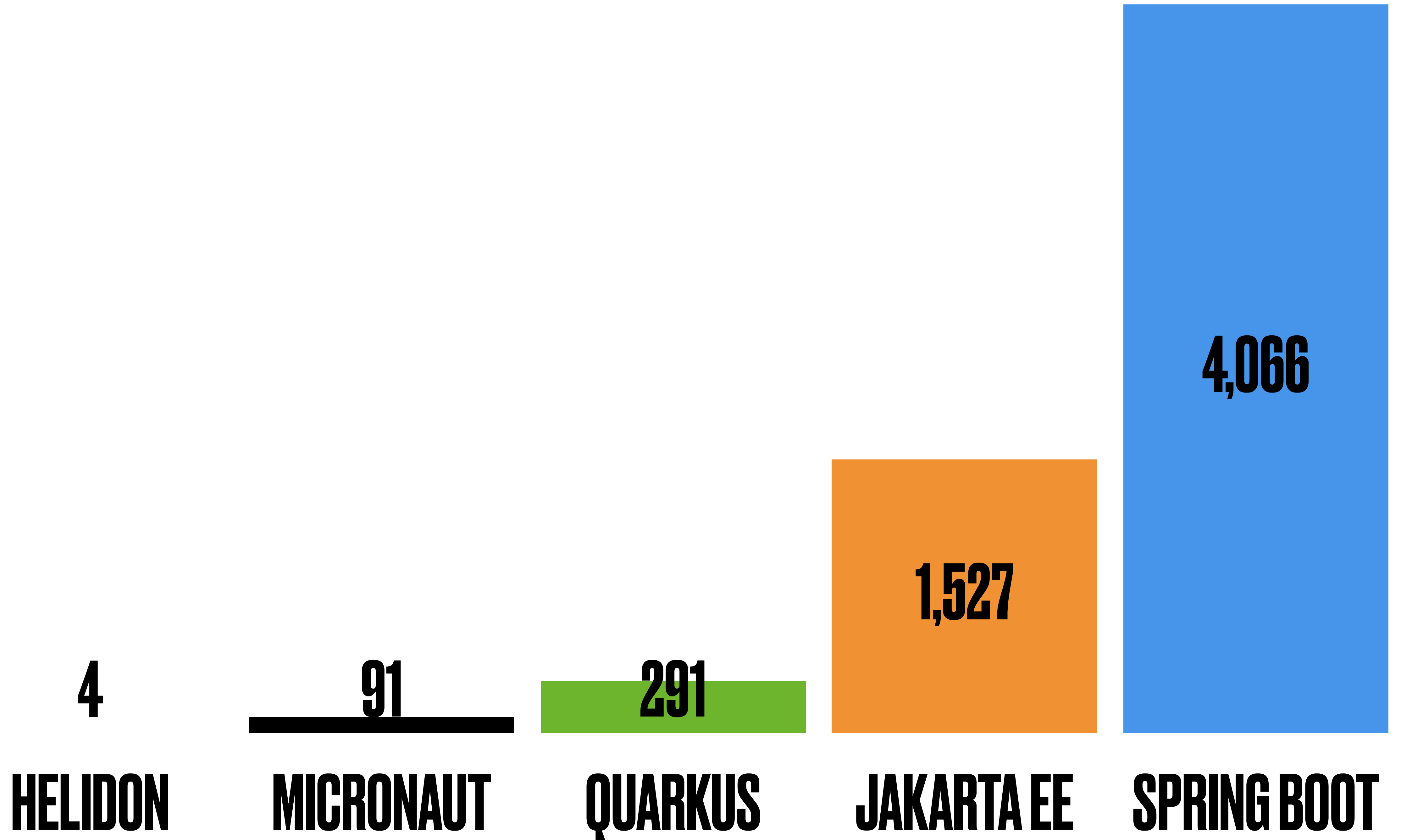
**ANDERE NACHTEILE**

**VON GRAALVM**

GERINGE **JOB**-NACHFRAGE

(**JOB-ANZEIGEN** BEI

INDEED.DE, FEB 2022)



**SCHWER, GRAALVM-**

**EXPERTEN**

**EINZUSTELLEN**

PEAK PERFORMANCE:  
SCHLECHTER ALS JIT



# PEAK PERFORMANCE

## GRAALVM COMMUNITY EDITION

**SERIAL GARBAGE COLLECTOR: STOP THE WORLD, SINGLE-THREADED**

**NUR GLOBALE OPTIMIERUNGEN**

## GRAALVM ENTERPRISE EDITION: €€€

**G1 GARBAGE COLLECTOR: NICHT "STOP THE WORLD", MULTI-THREADED**

**PROFILE-GUIDED OPTIMIZATION (PGO):  
NATIVE IMAGE NUTZT AGENT-DATEN  
FÜR SPEZIFISCHE OPTIMIERUNGEN**

**NATIVE JAVA IST NEU:  
WENIGER TRAINING,  
WENIGER ANTWORTEN**

**SCHNELLERER** START

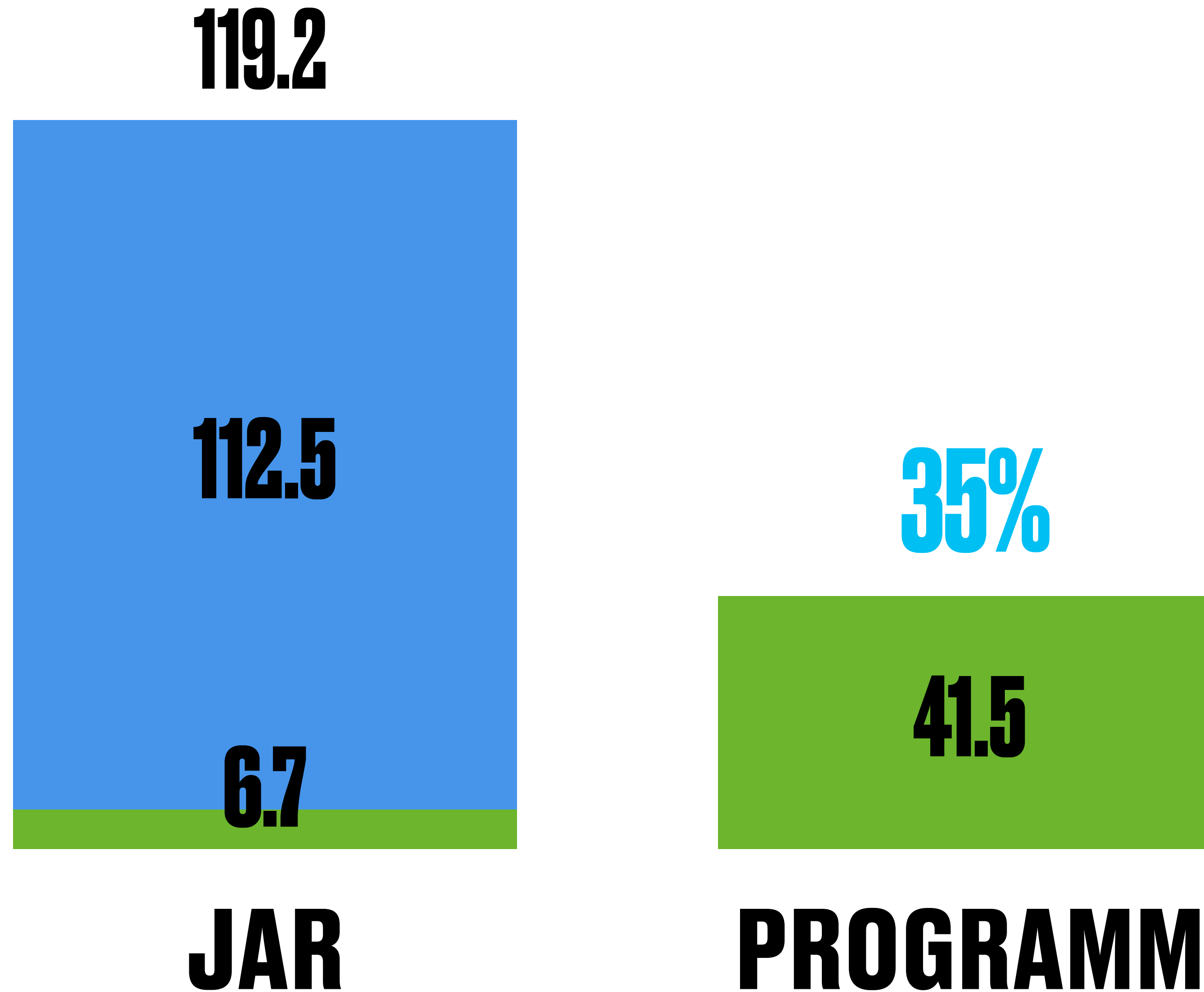
**WENIGER** SPEICHER

**KLEINERE** DATEIEN

**BESSERE** SICHERHEIT

**KLEINERE DATEIEN**

# QUARKUS: DATEI-GRÖÖE (MB)



**BESSERE SICHERHEIT**

# BESSERE SICHERHEIT

## KLEINERE ANGRIFFS-FLÄCHE

UNBENUTZTEN CODE ENTFERNEN  
LÖSCHT AUCH (POTENTIELLE)  
SICHERHEITSLÜCKEN IN  
FRAMEWORKS & BIBLIOTHEKEN

## KEIN UNBEKANNTER CODE AUSGEFÜHRT

ANGRIFFE, WIE LOG4SHELL, BRINGEN  
JAVA DAZU, "ANGRIFFS-CODE" ZU LADEN  
& AUSZUFÜHREN

CWA VERHINDERT DAS: CODE MUSS  
SCHON ZUR BUILD TIME BEKANNT SEIN

**ZWISCHENFAZIT**



GRAALVM **NATIVE IMAGE**  
ERSETZT JIT COMPILER &  
ARBEITET ZUR **BUILD TIME**

**SCHNELLERER** START

**WENIGER** SPEICHER

**KLEINERE** DATEIEN

**BESSERE** SICHERHEIT

WENIGER **SPEICHER** &  
SCHNELLERER **START**  
SPAREN **KOSTEN**

**CWA** SCHLIEßT EINIGE  
JAVA-FEATURES AUS

AUCH BIBLIOTHEKEN

KÖNNEN GRAALVM

SCHIEDERN LASSEN

**MEHR** ENTWICKLUNGS-  
AUFWAND, VOR ALLEM  
BEIM **TROUBLESHOOTING**

**FRAMEWORK** NEHMEN:

**SPRING BOOT 3,  
QUARKUS, MICRONAUT**

**LÖSUNG: NATIVE JAVA**



# AGENDA

~~PROBLEM? UND WARUM JETZT?~~

~~LÖSUNG: NATIVE JAVA~~

LÖSUNG: JIT JAVA

WANN LOHNT ES SICH?

**LÖSUNG: JIT JAVA**

**SCHNELLERER START &  
WENIGER SPEICHER  
OHNE GRAALVM?**

# JAVA-COMPILER

QUELL-  
CODE

BYTE-  
CODE

BUILD TIME

JVM

JIT COMPILER

BYTE-  
CODE

KLASSEN-  
LISTE

INIT JDK &  
FRAMEWORK

INIT  
APP

...

MASCHI-  
NEN-CODE

RUNTIME



**JAVA-COMPILER**

**QUELL-  
CODE**

**BYTE-  
CODE**

**IMMER WIEDER MIT IMMER  
GLEICHEM ERGEBNIS:  
VIELE JAVA-OBJEKTE**

**BUILD TIME**

**JVM**

**JIT COMPILER**

**BYTE-  
CODE**

**KLASSEN-  
LISTE**

**INIT JDK &  
FRAMEWORK**

**INIT  
APP**

...

**MASCHI-  
NEN-CODE**

**RUNTIME**

# JAVA-OBJEKTE SPEICHERN & LADEN

**NUR SCHNELLERER  
START, NICHT  
WENIGER SPEICHER**



# JAVA-COMPILER

QUELL-CODE

BYTE-CODE

IMMER WIEDER MIT IMMER  
GLEICHEM ERGEBNIS:  
VIELE JAVA-OBJEKTE

BUILD TIME

# JVM

# JIT COMPILER

BYTE-CODE

KLASSEN-LISTE

INIT JDK &  
FRAMEWORK

INIT  
APP

...

MASCHI-NEN-CODE

RUNTIME

# APPLICATION CLASS DATA SHARING FÜR **KLASSEN-** **LISTE**

# APPLICATION CLASS DATA SHARING

## WAS?

**SCHNELLERER START: SPEICHERT  
KLASSEN-LISTE BEIM ERSTEN LAUF &  
LÄDT AB 2. LAUF  
  
REDUZIERT START-ZEIT UM CA. 10-20%**

## WIE?

**JAVA-PARAMETER IN OPENJDK 17  
  
FRAMEWORKS (WIE QUARKUS)  
BIETEN EXTRA-SUPPORT FÜR  
BESSERE ERGEBNISSE**

# JAVA-COMPILER

QUELL-  
CODE

BYTE-  
CODE

IMMER WIEDER MIT IMMER  
GLEICHEM ERGEBNIS:  
VIELE JAVA-OBJEKTE

BUILD TIME

# JVM

# JIT COMPILER

BYTE-  
CODE

KLASSEN-  
LISTE

INIT JDK &  
FRAMEWORK

INIT  
APP

...

MASCHI-  
NEN-CODE

RUNTIME

**INIT** JDK, FRAMEWORK  
& APP MIT **CRAC**



**NICHT DIESES CRACK!**

OPENJDK-PROJEKT:  
COORDINATED **R**ESTORE  
**A**T **C**CHECKPOINT

# CRAC

## WAS?

**SCHNELLERER START: SPEICHERT  
KOMPLETTEN JAVA-HEAP BEIM ERSTEN LAUF &  
LÄDT AB 2. LAUF**

**BRAUCHT FRAMEWORK-UNTERSTÜTZUNG:  
DATEIEN & NETZWERK-VERBINDUNGEN VOR  
SPEICHERN SCHLIEßEN & DANN WIEDER  
ÖFFNEN**

## WIE?

**NUR AUF LINUX: NUTZT CRIU-FEATURE**

**APP BESTIMMT LADE- UND SPEICHER-  
ZEITPUNKT**

**NUR MICRONAUT HAT DAS IN PRODUKTION,  
AMAZON LAMBDA "SNAPSTART FOR JAVA"  
BENUTZT ES**



# CRAC VS. NATIVE IMAGE

## VORTEILE CRAC

**SCHNELLERER START: ERGEBNISSE ALLER  
INIT-OPERATIONEN LADEN – NATIVE IMAGE  
IST DA UNVOLLSTÄNDIG**

**APP ENTSCHIEDET SPEICHER-ZEITPUNKT  
– KANN AUCH "WARM-UP-ERGEBNISSE"  
ENTHALTEN**

## NACHTEILE CRAC

**NUR MIT MICRONAUT**

**SEPARATE DATEI – NATIVE IMAGE:  
IM PROGRAMM**

**OPENJDK-PROJEKT**

**LEYDEN**

# PROJECT LEYDEN

## ERSTER VERSUCH

**ZIEL: NATIVE JAVA IN JAVA-SPEZIFIKATION**

**START IM JUNI 2020...**

**...UND DANN 2 JAHRE NIX PASSIERT**

## ZWEITER VERSUCH

**ZIELE: JIT JAVA VERBESSERN & NATIVE JAVA IN JAVA-SPEZIFIKATION**

**START IM MAI 2022**

**FRÜHESTES LTS-RELEASE: JAVA 25 IM SEPTEMBER 2025**

**RELEVANTE  
JAVALAND-TALKS:  
VORTRAGS-SEITE**

**LÖSUNG: JIT JAVA**

# AGENDA

~~PROBLEM? UND WARUM JETZT?~~

~~LÖSUNG. NATIVE JAVA~~

~~LÖSUNG. JIT JAVA~~

WANN LOHNT ES SICH?

**WANN LOHNT ES SICH?**

# MEIN "FÜNF-PUNKTE- PLAN FÜR NATIVE JAVA"



JEDER PUNKT: "GO/NO  
GO"-ENTSCHEIDUNG

**1. INTERESSIERT'S DEN CHEF?**

**2. RECHNET SICH DAS?**

**3. KÖNNTE ES KLAPPEN?**

**4. KLEIN ANFANGEN**

**5. GROSS RAUSKOMMEN**

**INTERESSIERT'S  
DEN CHEF?**

UNSER JOB IST,  
"BUSINESS VALUE"  
ZU ERZEUGEN...

...**NICHT**, AUF **TWITTER** MIT  
MILLISEKUNDEN-START-  
ZEITEN ZU **PROTZEN!**

# INTERESSIERT'S DEN CHEF?

## UNSER JOB

"BUSINESS VALUE" ERZEUGEN -  
NEUE FUNKTIONEN & FEHLER  
BEHEBEN

"BUSINESS VALUE" NICHT  
ZWINGEND "KOSTEN SPAREN"

## CHEF NICHT INTERESSIERT, WEIL...

NEUE FUNKTIONEN GERADE WICHTIGER SIND

MÖCHTE NICHT DIE ARBEIT MACHEN – UND  
OPS HEIMST DANN DEN RUHM EIN

KOSTEN-EINSPARUNGEN ÄNDERN SEIN IT-  
BUDGET KAUM

**~~1. INTERESSIERT'S DEN CHEF?~~**

**2. RECHNET SICH DAS?**

**3. KÖNNTE ES KLAPPEN?**

**4. KLEIN ANFANGEN**

**5. GROSS RAUSKOMMEN**

**RECHNET SICH  
DAS?**



...MEHR **ENTWICKLER-**

**ZEIT,** UM OPERATIVE

**KOSTEN** ZU SPAREN...

**SCHLECHT FÜR NATIVE JAVA:**

**ENTWICKLER-ZEIT WIRD**

**TEURER, OPERATIVE**

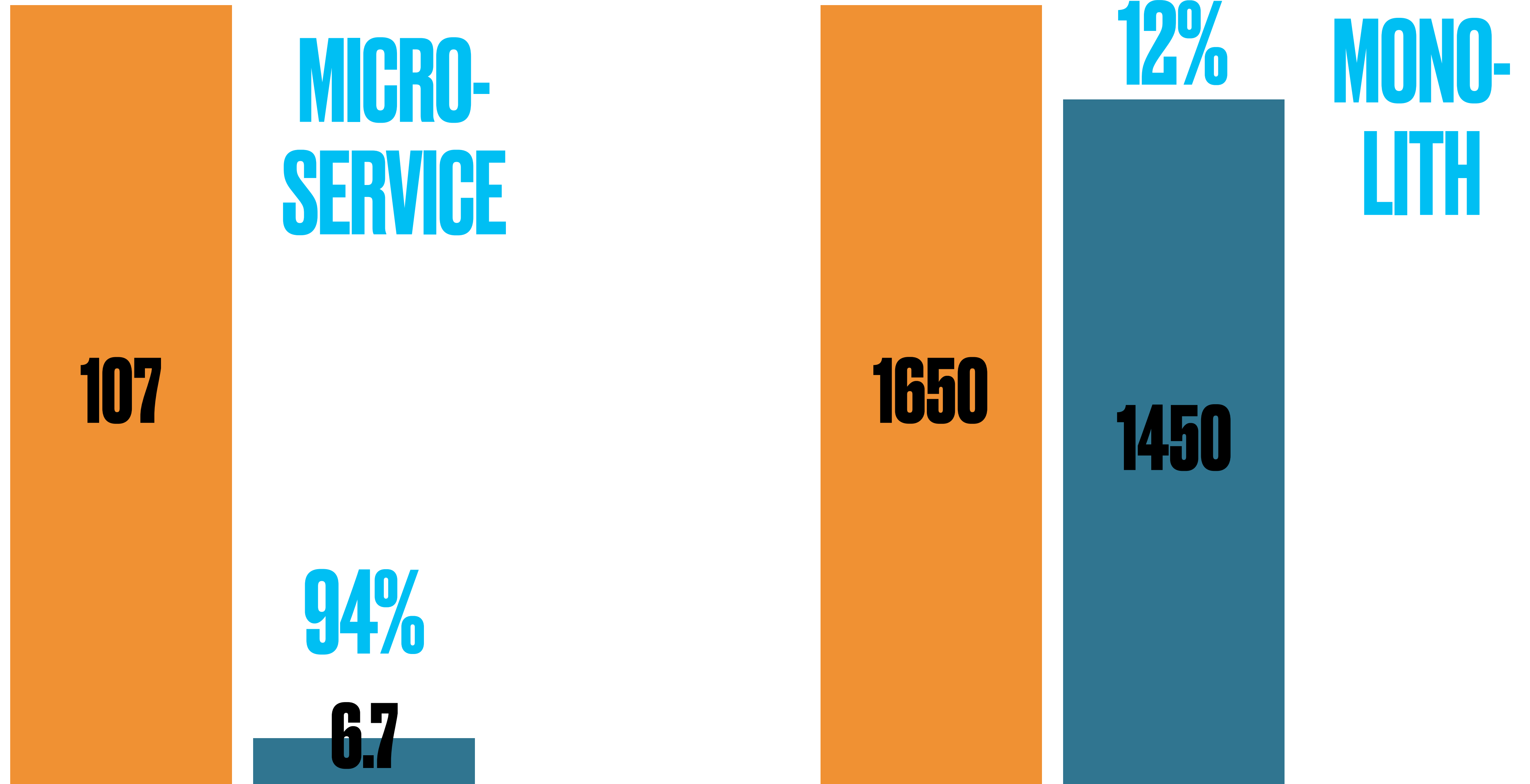
**KOSTEN BILLIGER**

**...UND IST MANCHMAL**

**UNMÖGLICH ODER**

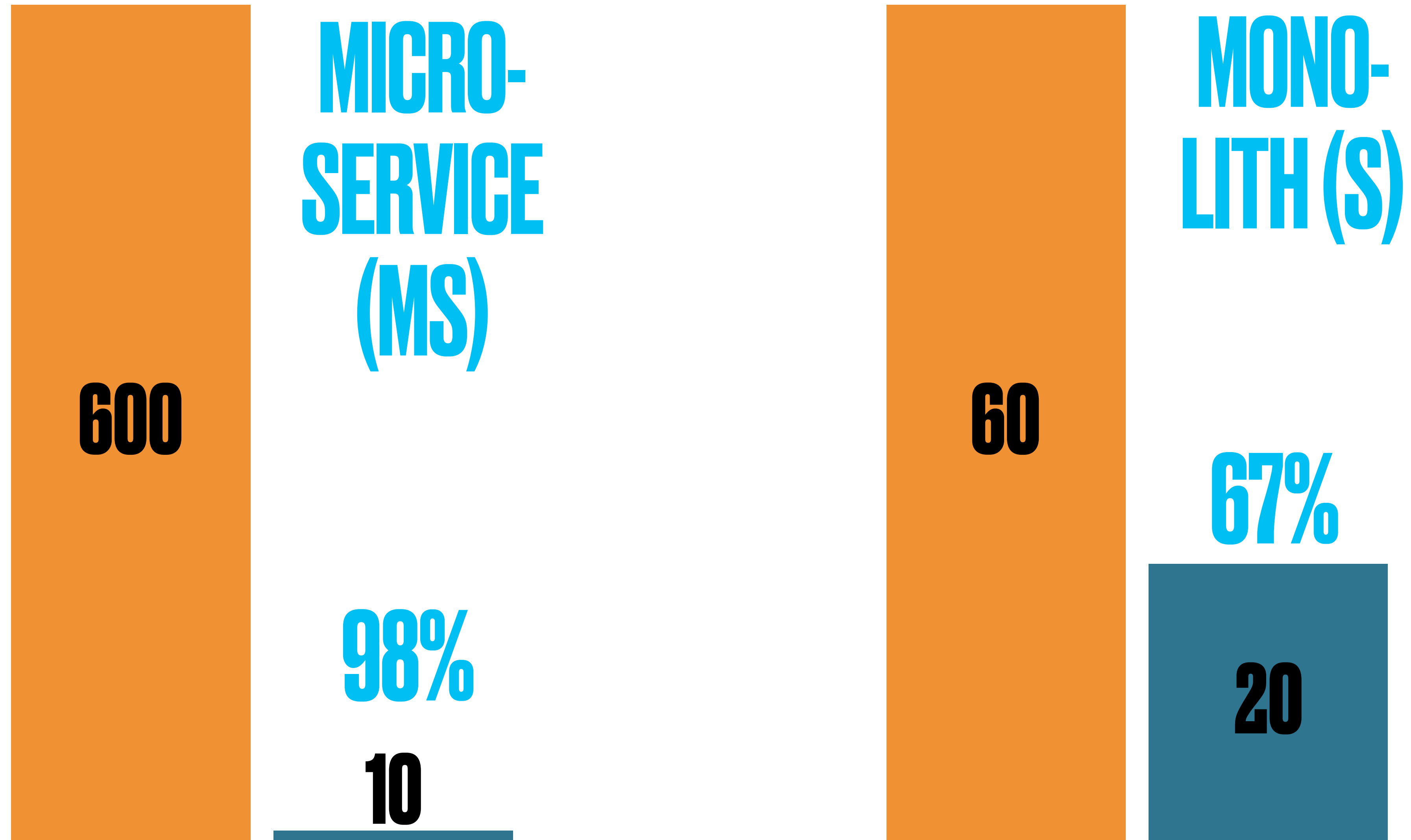
**UNWIRTSCHAFTLICH**

# NATIVE JAVA: SPEICHER-EINSPARUNGEN



WAS NÜTZEN IHNEN  
BEI **1.6 GB** HEAP SPACE  
**200 MB** MEHR?

# NATIVE JAVA: START-ZEIT



WAS NÜTZEN IHNEN **40**  
**SEKUNDEN** WENIGER START-  
ZEIT **ALLE ZWEI WOCHEN?**

**50%** DER KUBERNETES-  
KOSTEN PRO JAHR **SPAREN:**  
**10K FÜR SIE – ODER 10M?**



NATIVE JAVA **IDEAL** FÜR  
**SERVERLESS** – ABER  
SEHR GERINGER  
MARKTANTEIL FÜR JAVA

**SERVERLESS JAVA**

**TEURER/SCHLECHTER**

**ALS KUBERNETES? ODER**

**LEGT JAVA JETZT ZU?**

**WAS SIND DIE KOSTEN  
VON NATIVE JAVA?**

# KOSTEN VON NATIVE JAVA

## EINMAL

**NEUES FRAMEWORK (QUARKUS, MICRONAUT) ODER UPGRADE (SPRING BOOT 3: JAVA 17, JAKARTA EE 9) – BIBLIOTHEKEN!**

**WIE VIELE ENTWICKLER SCHULEN?**

**ENTWICKLER-PCS, BUILD PIPELINE, CI/CD, OBSERVABILITY, ...**

## LAUFEND

**TROUBLESHOOTING DAUERT LÄNGER, WENIGER ANTWORTEN, OBSERVABILITY SCHLECHTER**

**WENIG ENTWICKLER MIT GRAALVM-ERFAHRUNG**

JE **KLEINER** APP-  
GRÖÖE & -LAUFZEIT,  
DESTO **BESSER**  
RECHNET ES SICH

**SCHÄTZUNG – WIR  
WISSEN ES NICHT!**

**1. INTERESSIERT'S DEN CHEF?**

**2. RECHNET SICH DAS?**

**3. KÖNNTE ES KLAPPEN?**

**4. KLEIN ANFANGEN**

**5. GROSS RAUSKOMMEN**

**KÖNNTE ES  
KLAPPEN?**



# KÖNNTE ES KLAPPEN?

## WAS WIR WISSEN

MACHEN UNSERE APPLIKATIONEN  
DINGE, DIE IN NATIVE JAVA NICHT  
KLAPPEN?

FUNKTIONIERT DIE  
OBSERVABILITY IN NATIVE JAVA?

## WAS WIR NICHT WISSEN

FUNKTIONIEREN DIE  
BIBLIOTHEKEN IN NATIVE JAVA?

~~1. INTERESSIERT'S DEN CHEF?~~  
1. INTERESSIERT'S DEN CHEF?

~~2. RECHNET SICH DAS?~~  
2. RECHNET SICH DAS?

~~3. KÖNNTE ES KLAPPEN?~~  
3. KÖNNTE ES KLAPPEN?

4. KLEIN ANFANGEN

5. GROSS RAUSKOMMEN

**KLEIN ANFANGEN**

# KLEIN ANFANGEN

## FÜR MICROSERVICES

EINEN NEU BEGINNEN/NEU  
IMPLEMENTIEREN IN NATIVE JAVA

KLEINES TEAM, NIEDRIGE KOSTEN

ZIEL: STIMMT UNSERE ROI-  
RECHNUNG?

## FÜR MONOLITHEN

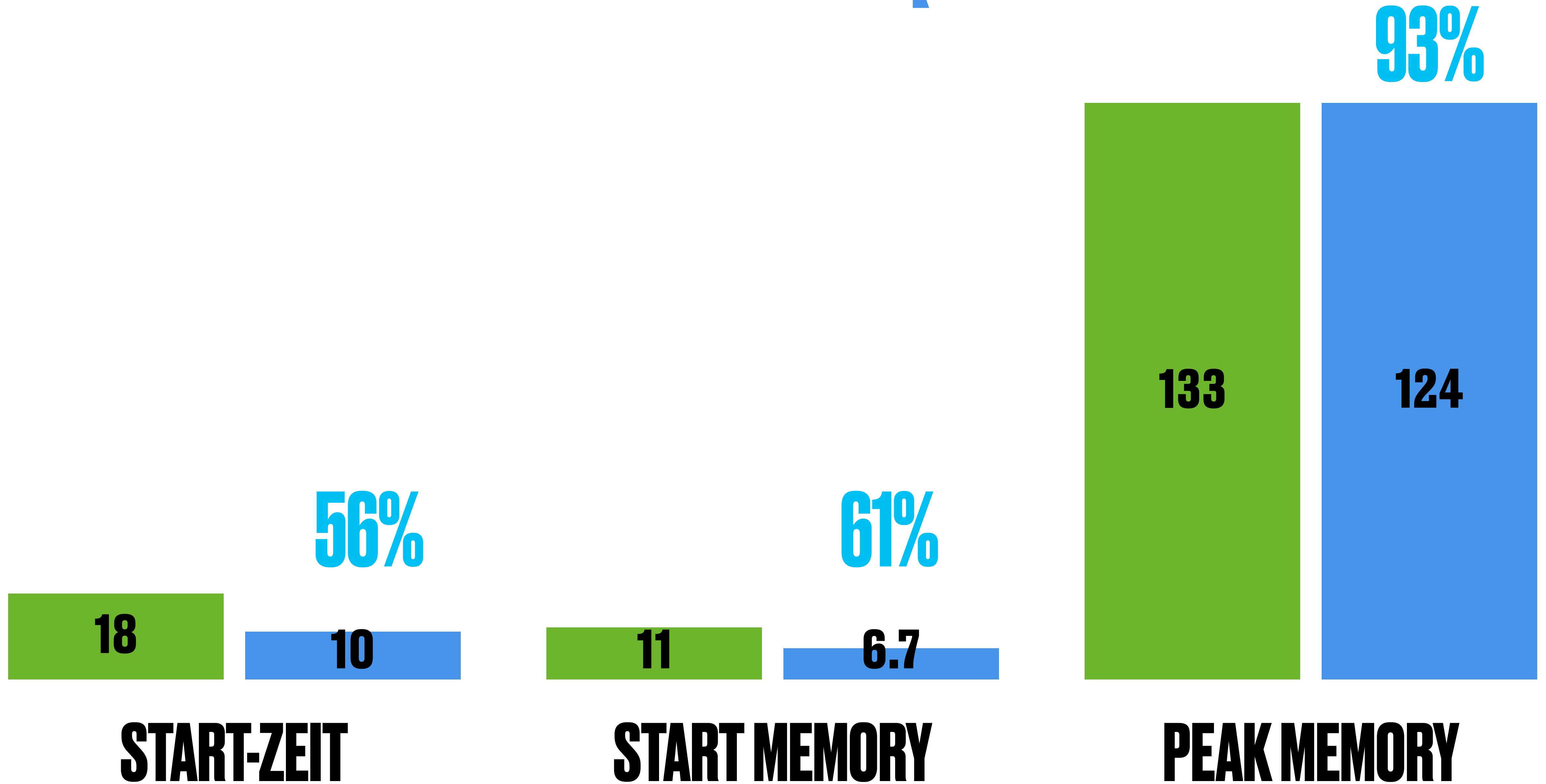
EINEN KLEINEN TEIL ALS  
MICROSERVICE HERAUSBRECHEN

DANN: SIEHE LINKS

**WELCHES**  
**FRAMEWORK?**

**SPRING BOOT 3**  
**VS. QUARKUS**  
**(MEIN BEISPIEL!)**

# SPRING BOOT VS. QUARKUS



**QUARKUS: BESTE START-**

**ZEIT & SPEICHER-**

**VERBRAUCH...**



...VS. WECHSEL-KOSTEN  
VON SPRING BOOT

## **KUBERNETES**

**BACK-ENDS IN ALLEN  
PROGRAMMIER-  
SPRACHEN**

## **MICROSERVICES**

**NEUE SPRACHEN &  
FRAMEWORK MIT  
KLEINEM RISIKO  
TESTEN**

## **CLOUD**

**MEHR SPEICHER &  
STARTUP-ZEIT IST OFT  
TEURER**

## **ALLE FRAMEWORKS**

**PROBIEREN SIE MAL EIN  
ANDERES FRAMEWORK  
AUS - SPRING BOOT 3,  
QUARKUS ODER  
MICRONAUT**

## **SPRING BOOT**

**AUF 3.0 MIGRIEREN  
  
DANN NATIVE JAVA  
TESTEN**

## **QUARKUS & MICRONAUT**

**DABEI BLEIBEN, WENN  
ES PASST**

~~1. INTERESSIERT'S DEN CHEF?~~  
1. INTERESSIERT'S DEN CHEF?

~~2. RECHNET SICH DAS?~~  
2. RECHNET SICH DAS?

~~3. KÖNNTE ES KLAPPEN?~~  
3. KÖNNTE ES KLAPPEN?

~~4. KLEIN ANFANGEN~~  
4. KLEIN ANFANGEN

5. GROSS RAUSKOMMEN

**GROSS  
RAUSKOMMEN**

**GROSS RAUSKOMMEN**

**SORGFÄLTIG NATIVE JAVA WEITER AUSROLLEN**

**STÄNDIG NACHJUSTIEREN**

**APPLIKATIONEN NUR DANN NACH NATIVE JAVA  
MIGRIEREN, WENN ES SICH RECHNET!**

**1. INTERESSIERT'S DEN CHEF?**  
**1. INTERESSIERT'S DEN CHEF?**

**2. RECHNET SICH DAS?**  
**2. RECHNET SICH DAS?**

**3. KÖNNTE ES KLAPPEN?**  
**3. KÖNNTE ES KLAPPEN?**

**4. KLEIN ANFANGEN**  
**4. KLEIN ANFANGEN**

**5. GROSSE HAUSKOMMEN**  
**5. GROSSE HAUSKOMMEN**

FÜR DIE MEISTEN:

JETZT NICHT



**WARUM JETZT NICHT?**

GERINGE **JOB**-NACHFRAGE –  
UND ES HAKT  
BEI PUNKT **1&2**

**WANN LOHNT ES SICH?**

# AGENDA

~~PROBLEM? UND WARUM JETZT?~~  
~~PROBLEM? UND WARUM JETZT?~~

~~LÖSUNG. NATIVE JAVA~~  
~~LÖSUNG. NATIVE JAVA~~

~~LÖSUNG. JIT JAVA~~  
~~LÖSUNG. JIT JAVA~~

~~WANN LOHNT ES SICH?~~  
~~WANN LOHNT ES SICH?~~

**ZUSAMMENFASSUNG**

IN DER CLOUD IST  
JAVA **TEURER** ALS  
KONKURRENTEN WIE GO

**NATIVE JAVA MACHT**

**JAVA BILLIGER,**

**BRAUCHT ABER...**

...MEHR ENTWICKLER-  
ZEIT, UM OPERATIVE  
KOSTEN ZU SPAREN...



**...UND IST MANCHMAL**

**UNMÖGLICH ODER**

**UNWIRTSCHAFTLICH**

# SPRING BOOT 2 $\Rightarrow$ QUARKUS: STARTUP

**600** MS  $\Rightarrow$  **10** MS

**107** MB  $\Rightarrow$  **6.7** MB

**MICRO-SERVICES IN**

**KUBERNETES:**

**MEHR PODS/NODE (SO  
NICHT CPU-LIMITIERT)**

**FRAMEWORK: SPRING  
BOOT 3, QUARKUS &  
MICRONAUT**

- 1. INTERESSIERT'S DEN CHEF?**
- 2. RECHNET SICH DAS?**
- 3. KÖNNTE ES KLAPPEN?**
- 4. KLEIN ANFANGEN**
- 5. GROSS RAUSKOMMEN**

**WANN LOHNT SICH  
NATIVE JAVA MIT  
GRAALVM FÜR MICH?**

FÜR DIE MEISTEN:

JETZT NICHT



THANK  
YOU



**QUARKUS TEAM, RED HAT**

**BEN EVANS**

**DIMITRIS ANDREADIS**

**FOLVOS ZAKKAK**

**GALDER ZAMARRENO**

**HOLLY CUMMINS**

**MAX RYDAHL ANDERSEN**

**MICHAEL KAM BARBACEC**

**PATRICK BAUMGARTNER**

**SANNE GRINOVERO**

**OPENJ9 TEAM, RED HAT**

**DAN HEIDINGA**

**GRAALVM TEAM, ORACLE**

**ALINA YURENKO**

**AZUL**

**SIMON RITTER**

The image features a classic 'The End' title card. The text 'The End' is written in a white, cursive script font with a black drop shadow, centered within a black circle. This circle is surrounded by several concentric red circles that fade out towards the edges of the frame, creating a tunnel-like effect. The overall background is a dark red color.

*The End*

MEIN **VORTRAG** HÖRT  
NICHT AUF, WENN ICH  
AUFHÖRE **VORZUTRAGEN!**

**FOLIEN**



**ZUSATZ-INFORMATIONEN**

**REPOS FÜR BEISPIEL-CODE**

**MIT NATIVE JAVA LOSLEGEN**

**MEIN "POPULARITÄTS-NEWSLETTER"**



**BPELI/LAM**