



IBM & Consol present: The day after tomorrow Messaging with **Quarkus**

Dr. Marco Bungart
Senior Software Engineer

2021-09-30



```
$> whoami
```

Dr. Marco Bungart

- 2008 – 2013: Studied Bioinformatics/Computer Science in Jena
- 2013 – 2018: Ph.D. student in Kassel
- Since 2018: Software Engineer at ConSol
- Twitter, github, bitbucket, stackoverflow, ... : turing85
- Interests: Keycloak, GraalVM, Quarkus

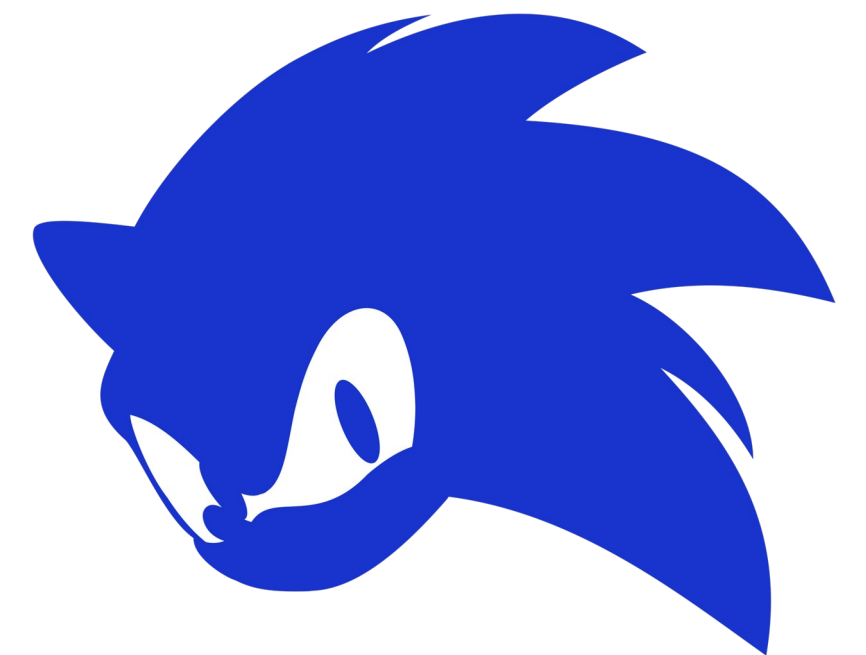




Quarkus!

What is Quarkus?

- “A Kubernetes Native Java stack tailored for OpenJDK HotSpot and GraalVM, crafted from the best of breed Java libraries and standards.” - quarkus.io
- Lightweight in execution (small memory footprint)
- Compilation to native binary possible through GraalVM
 - Even lower memory footprint
 - Single instance may have worse performance, but more deployments possible due to lower resource consumption
→ net gain
- Up & running within minutes



Source:

<https://1000logos.net/sonic-logo/>

Why Quarkus?

- Developer-joy front & center
 - Developed by developers, for developers
 - Excellent Maven- and gradle-integration
 - Powerful dev-mode, tight feedback loop
- Curated list of dependencies, “best of the breed”
 - Red Hat- and Community-developed
- Apps (re)start within milliseconds



Messaging

Messaging

- Loosely coupled systems
 - Do not use when strong coupling is needed/wanted!
- Realized through
 - Microprofile
 - implementation is asynchronous & reactive, easy to swap between different providers
 - Supports JMS/AMQP, Kafka
 - Vendor-specific implementations (e.g. Qpid, Artemis)
 - More fine-grained control, but also more configuration

Messaging – What we will look at today

- Microprofile-base implementation
- Messaging-Provider:
 - ArtemisMQ
 - Kafka
- Communication between two services (customer and order)
- Observability

Messaging – What we will **not** look at **today**

- Vendor-specific implementations
- Kafka-Streams (other usecase)
- Partitioning
- No demo on a cluster today :(but if you want to use a cluster, try the IBM Cloud!
<https://ibm.biz/BdfGea>



Architecture of the system

Architecture of the **system**

- Customer-Service
 - Publishes message when a new customer is created (containing email of customer)
 - Receives message when a new order for a customer is created
- Order-Service
 - Orders need customers, so customers must be received from Customer-service before an order can be created
 - Publishes message when a new order is created (containing email of customer, an order-id, creation-date and total sum)
- Customer-messages via AMQP
- Order-messages via Kafka

Code Examples

- Can be found at <https://github.com/ConSol/quarkusMessaging>
- Slides are included
- Commits + messages are meant as a step-by-step guide for what we are going to do

Code Examples

- Can be found at <https://github.com/ConSol/quarkus-devmode>
- Slides are included
- Commits + messages are meant as a step-by-step guide for what we are going to do

Live Demo



Source: <https://makeameme.org/meme/lets-pray-to-9000c9f697>



Questions?



Thank you!



ConSol
Consulting & Solutions Software
GmbH

Office Düsseldorf
Kanzlerstraße 8
D-40472 Düsseldorf
Germany
Tel.: +49-89-45841-100
Marco.Bungart@consol.de
www.consol.com
Twitter: @turing85