

# Microframeworks

von Fabian Prieschl

# Warum Microframeworks?

- wachsendes Interesse an “Serverless-Architekturen”
- kleine leichtgewichtige Applikationen statt großen Monolithen
- Docker, Kubernetes
- kompakt und verständlich
- große Framework-Vielfalt

# Eclipse MicroProfile

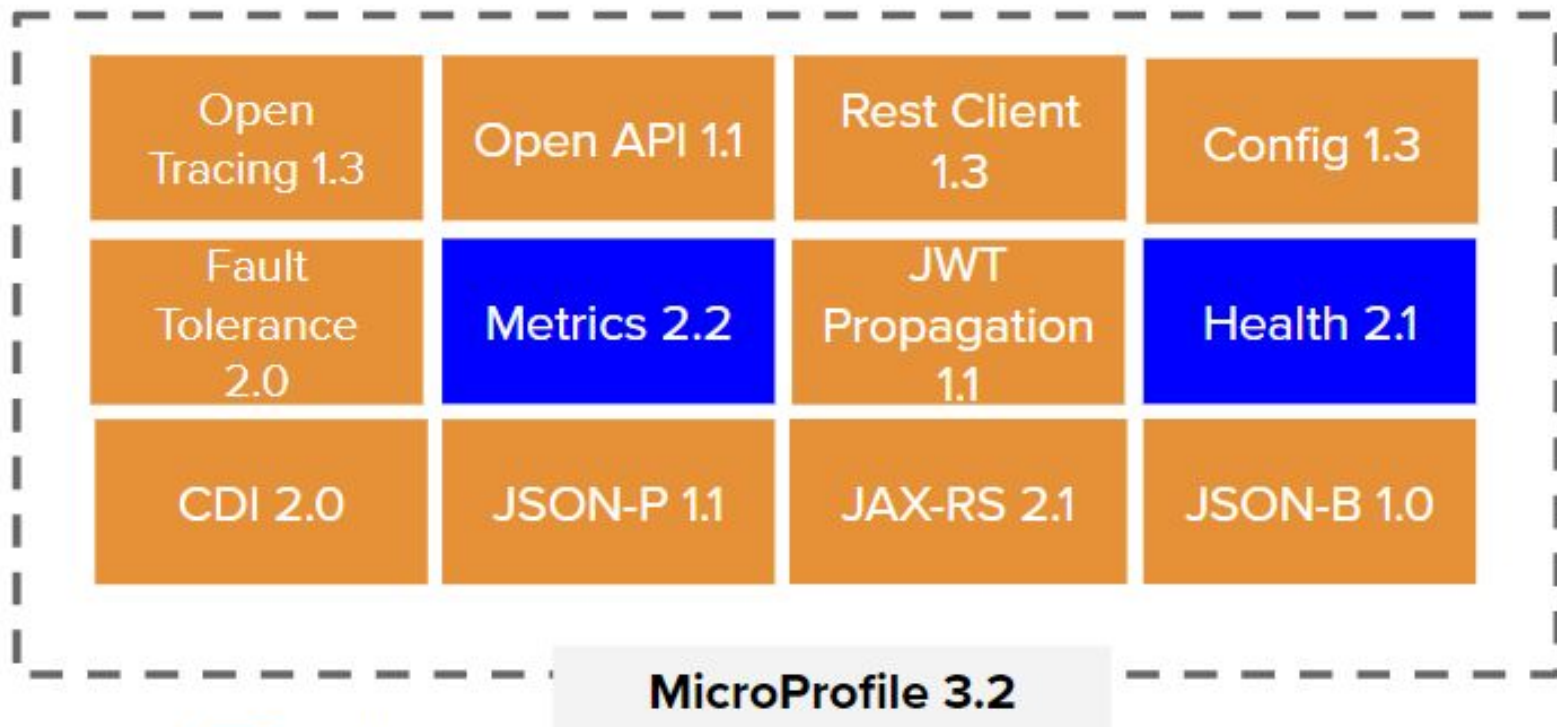



- Open-Source-Community-Spezifikation für Enterprise Java Microservices
- MicroProfile Implementierungen
  - Quarkus
  - Thorntail
  - Payara Micro
  - Helidon
- Einzelspezifikationen von Jakarta EE




QUARKUS





 = New

 = Updated

 = No change from last release (MicroProfile 3.1)

# Spring Boot

- bekanntestes Microframework
- große Community
- umfangreiche Dokumentation
- sehr einfache und schnelle Konfiguration
- langsamer als andere Frameworks
- DataRepository bzw. CrudRepository
- Server wird in main Methode gestartet

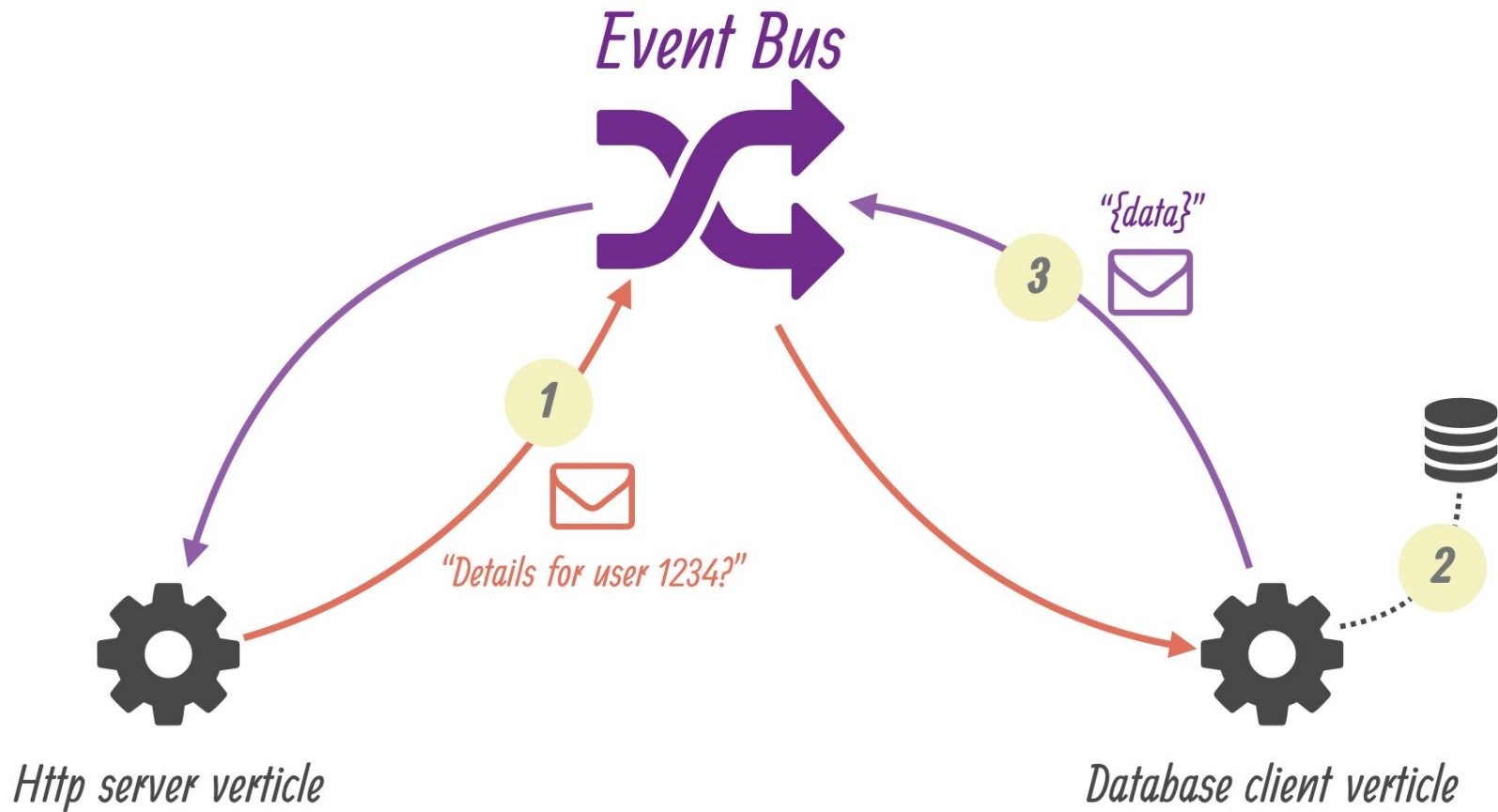


**Spring  
Boot**

# Vert.x

- sehr leichtgewichtig
- top Performance
- nicht sehr intuitiver Einstieg
- JPA? (Alternative - jOOQ)
- Event Loop - (Don't block!)
- Worker vs Non-Worker Verticles
- Blocking code
- Großes Ökosystem, sehr modular gehalten
- Hollywood Prinzip - node.js Ähnlichkeiten (Reactor Pattern)
- Multi Reactor Pattern - mehrere Event Loops
- Vert.x Event Bus





# Micronaut

- Dependency Injection bereits zur Compilezeit
  - schnellere Startup Time
  - weniger Speicherverbrauch
- allgemein sehr schnell und schlank
- JPA
- in main Methode gestartet





# Helidon

- Helidon SE vs Helidon MP
- schnell und leichtgewichtig
- leicht verständlich
- schnell einsetzbar
- JPA
- ähnlich wie Quarkus
- in main Methode gestartet



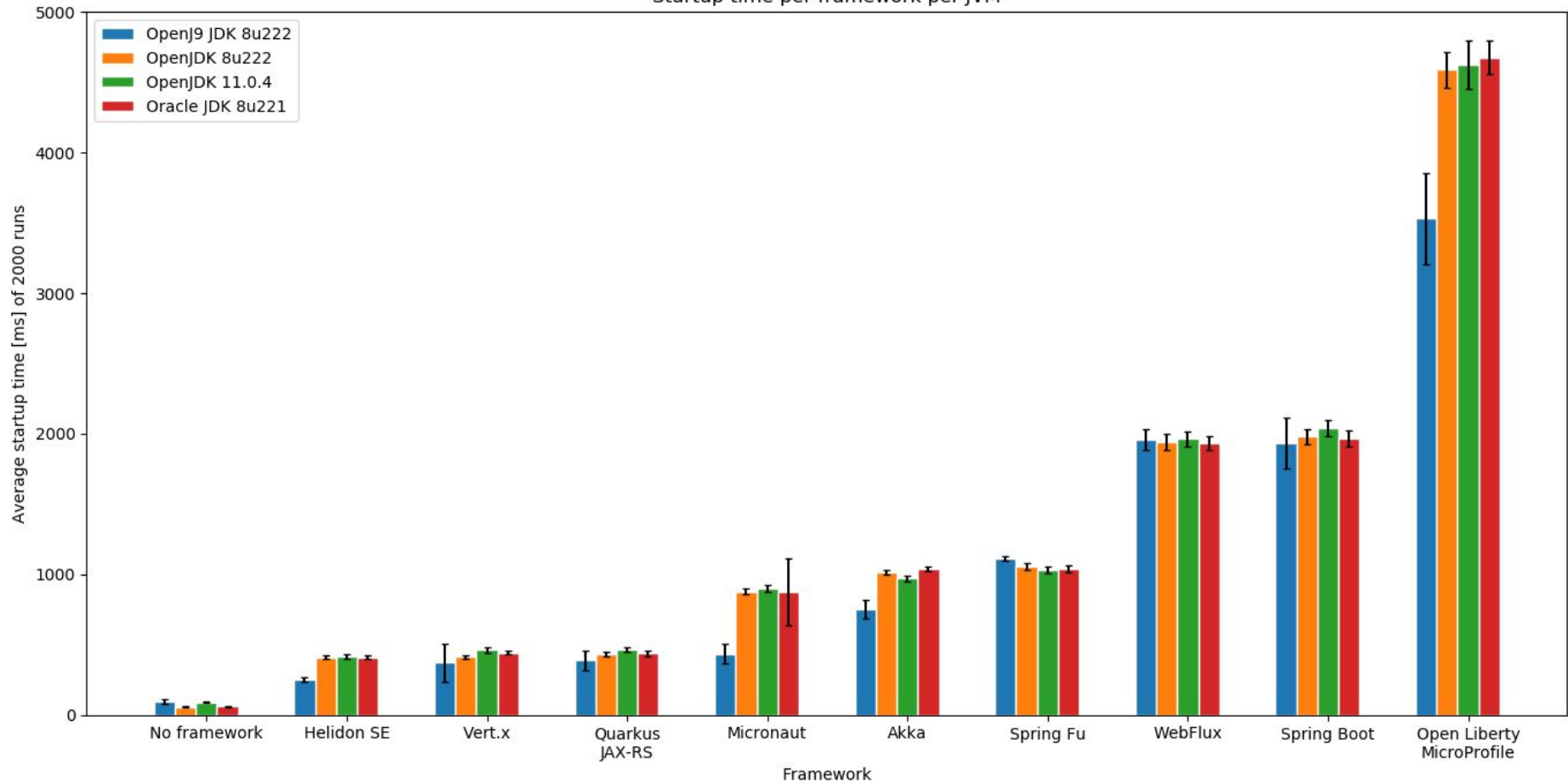
# Quarkus

- leichtgewichtig und schnell
- Panache
- keine main Methode
- einfache Datasource Konfiguration
- Spring Boot DataRepositories
- vert.x

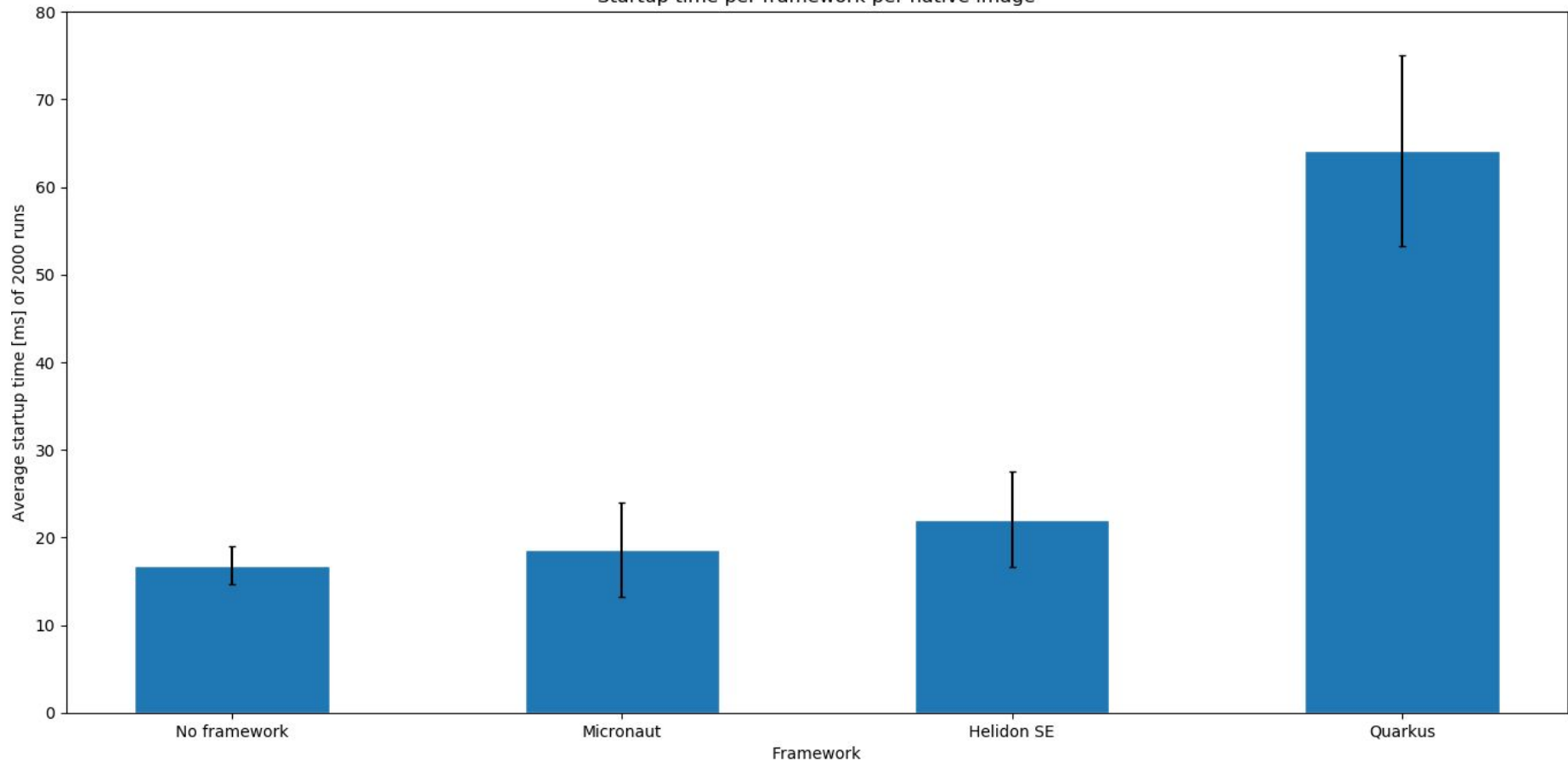


QUARKUS

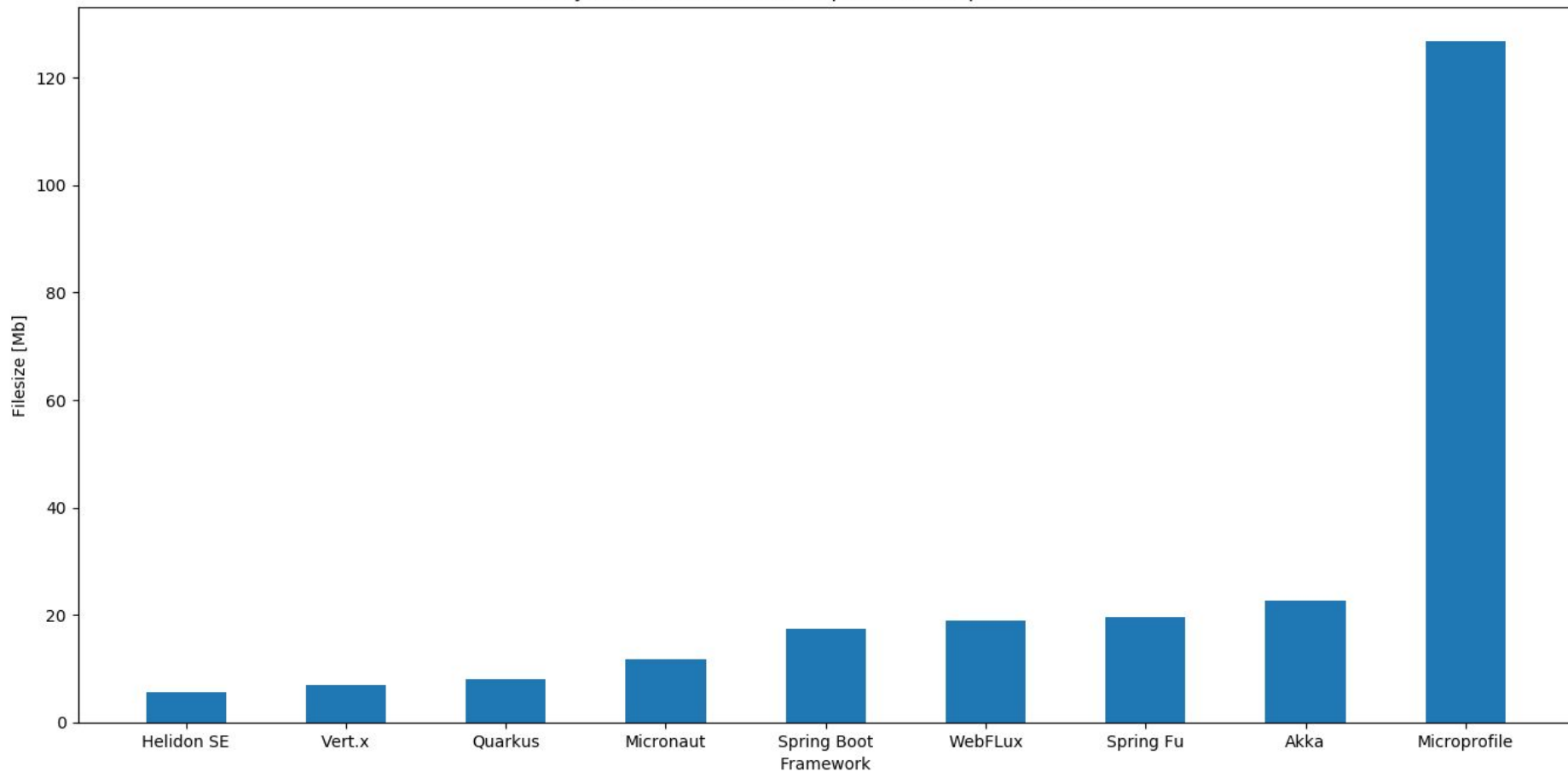
Startup time per framework per JVM



Startup time per framework per native image



Fat JAR file size of minimal implementation per framework



# Quellen

- <https://blogs.itemis.com/de/4-microservice-frameworks-im-vergleich>
- <https://quarkus.io/>
- <https://spring.io/guides/gs/rest-service/>
- <https://technology.amis.nl/2019/09/03/microservice-framework-startup-time-on-different-jvms-aot-and-jit/>
- <https://de.wikipedia.org/wiki/Ahead-of-time-Compiler>
- <https://spring.io/guides/gs/rest-service/>
- <https://spring.io/guides/gs/accessing-data-rest/>
- [https://medium.com/@levon\\_t/java-vert-x-starter-guide-part-2-worker-verticles-c49866df44ab](https://medium.com/@levon_t/java-vert-x-starter-guide-part-2-worker-verticles-c49866df44ab)
- <https://www.opencodez.com/java/introduction-to-micronaut-framework.htm>
- <https://quarkus.io/guides/vertx>
- <https://quarkus.io/guides/spring-data-jpa>

**Vielen Dank für Ihre  
Aufmerksamkeit!**