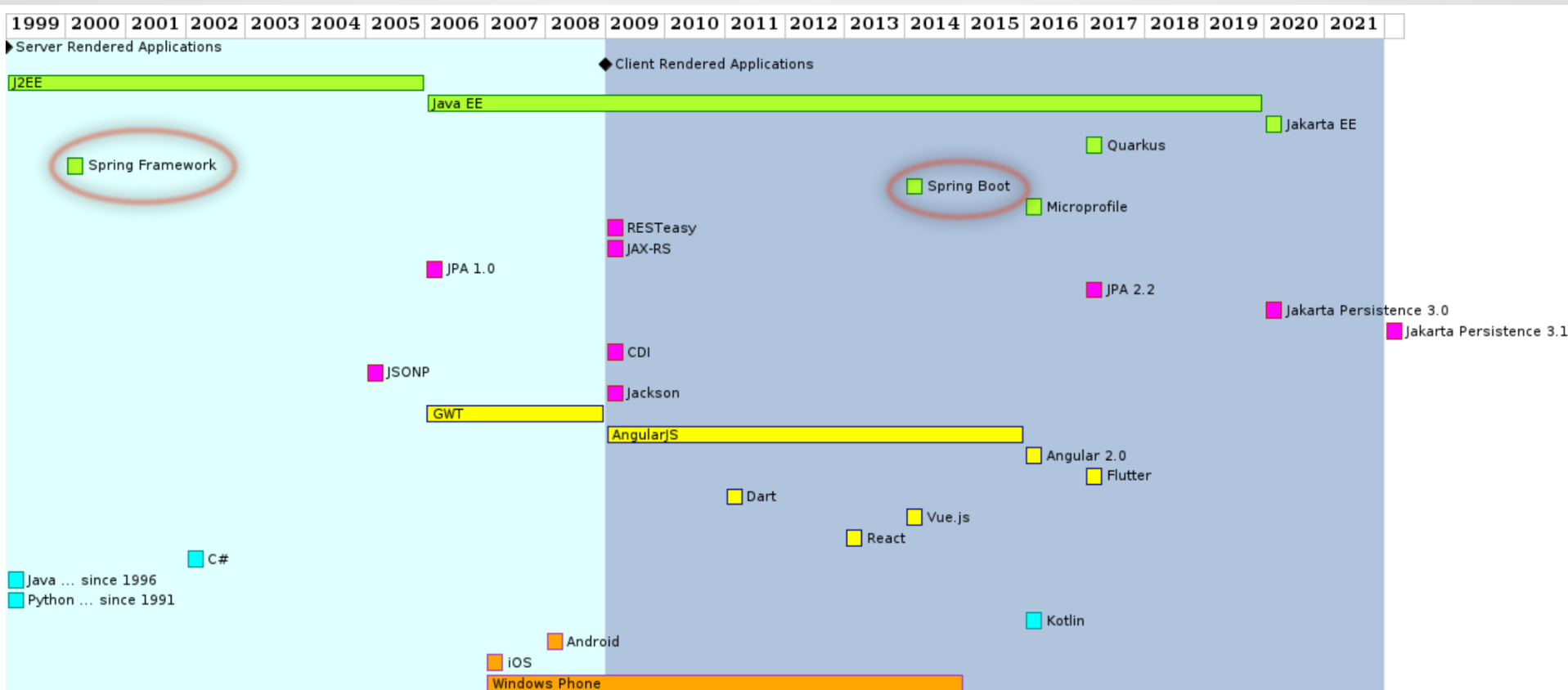


# Spring

Framework

# Spring Framework



- Most widespread Framework
- Still commonly used

# Spring Framework

- 2004
- For Java
- Application Framework
- **I**nversion **O**f **C**ontrol (IOC) container
- Open Source
- No specific programming model
- Has extensions to use if you want to get a specific job done
- Currently de-facto-standard for Java application development

# Spring MVC

- 2005
- Template for using Master-View-Controller pattern
- Has a 'DispatcherServlet' that does the mapping using the URL and routes it to the correct resource (model, view and controller)

# Spring Security

- 2007
- Supports Authentication and Authorization
- Protection against various attacks
- Servlet API integration
- Optional integration with Spring Web MVC

# Spring Data

- 2009
- Repository and custom **O**bject **R**elationship (OR)-mapping abstractions
- Query derivation from method names
- Transparent auditing
- Inegration with Spring MVC controllers

# Spring Boot

- 2014
- Easy creation stand-alone applications
- Embedded Tomcat, Jetty or Undertow
- 'Starter' dependencies
- Autoconfigure where possible
- Provide metrics, health-checks, config
- No code generation or XML config



# Step 1: Start a new Spring Boot project

Use [start.spring.io](https://start.spring.io) to create a “web” project. In the “Dependencies” dialog search for and add the “web” dependency as shown in the screenshot. Hit the “Generate” button, download the zip, and unpack it into a folder on your computer.

The screenshot shows the Spring Initializr web interface. The 'Project' section has 'Maven Project' selected. The 'Language' section has 'Java' selected. The 'Spring Boot' section has '2.2.6' selected. The 'Project Metadata' section has 'Group' as 'com.example', 'Artifact' as 'demo', 'Name' as 'demo', 'Description' as 'Demo project for Spring Boot', and 'Package name' as 'com.example.demo'. The 'Packaging' section has 'Jar' selected. The 'Java' section has '8' selected. The 'Dependencies' section has 'Spring Web' selected. A callout box points to the 'ADD DEPENDENCIES...' button in the Dependencies section, with the text: 'Click on 'Add dependencies', type `Web` in the search box, then click on the dependency 'Spring Web' to select it.'

**Project**

☒ Maven Project ☐ Gradle Project

**Language**

☒ Java ☐ Kotlin ☐ Groovy

**Spring Boot**

☐ 2.3.0 M4 ☐ 2.3.0 (SNAPSHOT) ☐ 2.2.7 (SNAPSHOT) ☒ 2.2.6 ☐ 2.1.14 (SNAPSHOT) ☐ 2.1.13

**Project Metadata**

Group

Artifact

Name

Description

Package name

**Packaging** ☒ Jar ☐ War

Java ☐ 14 ☐ 11 ☒ 8

**Dependencies**

**Spring Web** **WEB**

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

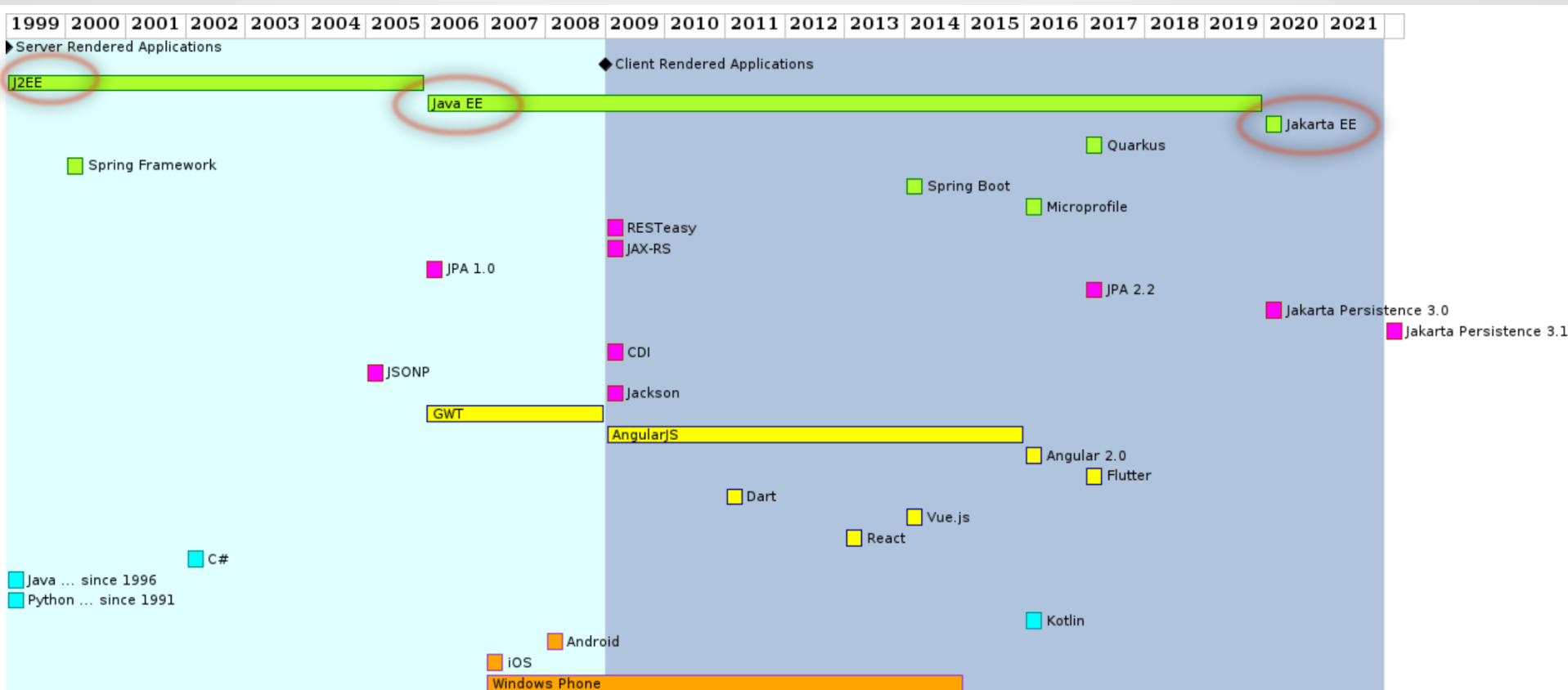
The current version of Spring Boot changes regularly. Just choose the latest release (but not snapshot).

# Jakarta EE

former Java EE

former J2EE

# Jakarta EE



- First Java Enterprise framework
- Parts of it are still widely used...

# J2EE - Java EE - Jakarta EE

- 1999 - 2006 - 2019
- Set of specifications extending Java SE
  - Distributed Computing
  - Web Services
- Run on reference runtimes (servers)
  - Glassfish
  - TomEE
  - IBM WebSphere
  - Oracle WebLogic
  - Resin
  - Wildfly
  - ...

# J2EE - Java EE - Jakarta EE

- **P**lain **O**ld **J**ava **O**bject (POJO) - Java Enterprise **B**ean
- Consists of:
  - Servlet, WebSocket
  - JAX-RS (RESTful Web Services)
  - JSON-P (Processing), JSON-B (Binding)
  - Annotations, Transactions
  - Bean Validation, Managed Beans
  - Interceptors, Dependency Injection
  - Security, Authentication
  - JMS (messaging service)
  - ...

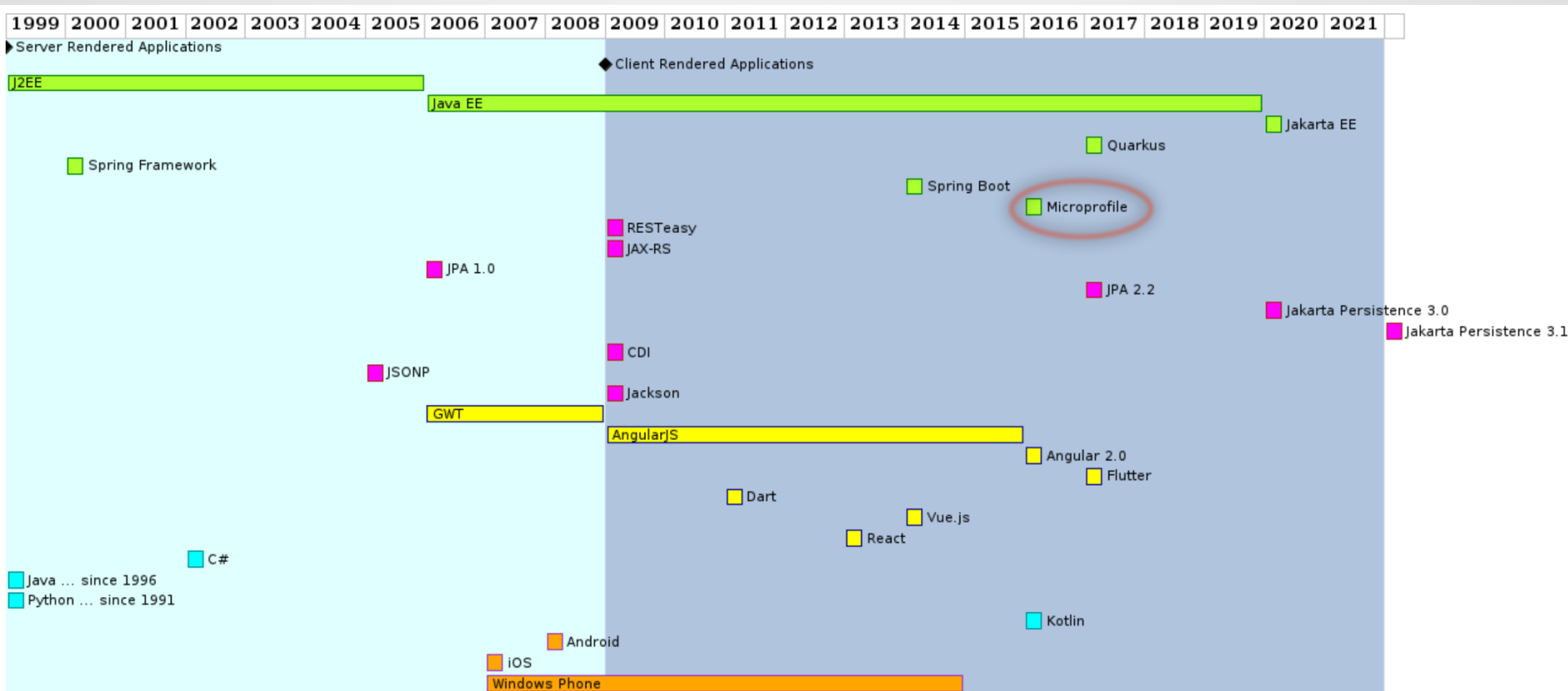
These servers are too heavy for modern development.

Only parts are in the new Jakarta EE Web-Profile.

# Microprofile

by Eclipse

# Microprofile



- To develop Enterprise Java cloud-native microservices

# Microprofile

- 2016
- For developing micro-services
- First version was formed solely on **JAX-RS**, **JSON-P** and **CDI**
  - Cloud-Native approach
  - Small and light
  - Made for micro-services



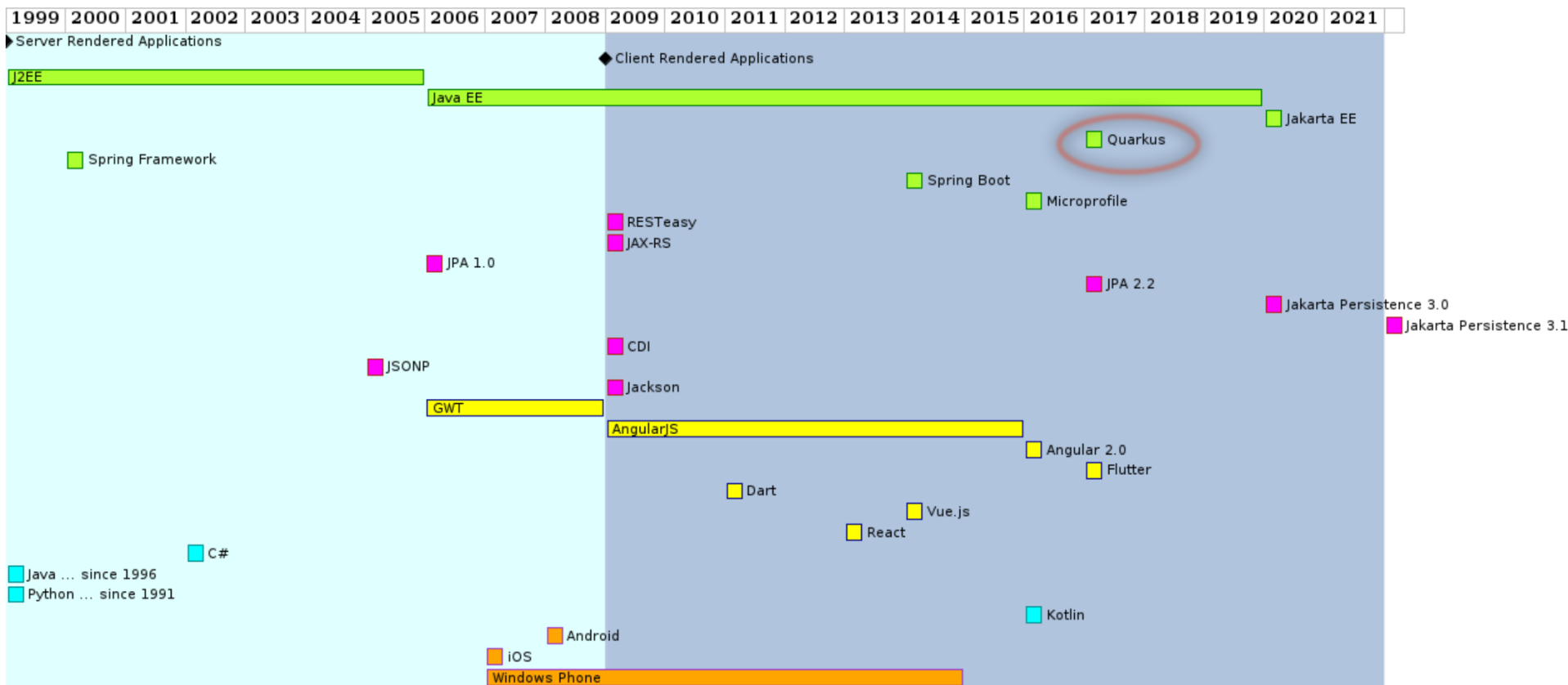
# Microprofile

- Health Check
- Fault Tolerance
- Metrics
- Rest Client
- Open Tracing
- OpenAPI
- JWT
- MicroProfile Config
- ...

# Quarkus

by Red Hat

# Quarkus



- Tailored for deployment on Kubernetes (micro-services) and serverless

# Quarkus

- 2017
- For developing micro-services using Kubernetes
- For serverless development
- Own VM (**GraalVM**) to speed up things
- But may build on OpenJDK HotSpot as well

# Quarkus

- When used with GraalVM natively...
  - You sacrifice the write-once / run-everywhere paradigm
  - You gain stupendous startup- and execution-speed benefits along with an extremely small memory-footprint
  - Quarkus tries to compile everything - even runtime-logic like annotation-processing and so on...

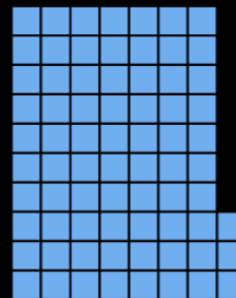
# Quarkus

REST



Quarkus + Native  
(via GraalVM)

**12 MB**



Quarkus + JVM  
(via OpenJDK)

**73 MB**



Traditional  
Cloud-Native Stack

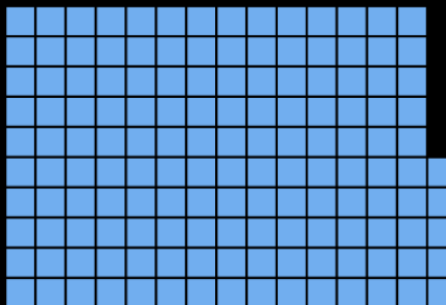
**136 MB**

REST  
+ CRUD



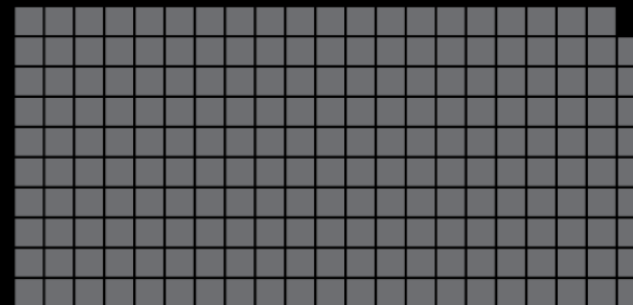
Quarkus + Native  
(via GraalVM)

**28 MB**



Quarkus + JVM  
(via OpenJDK)

**145 MB**

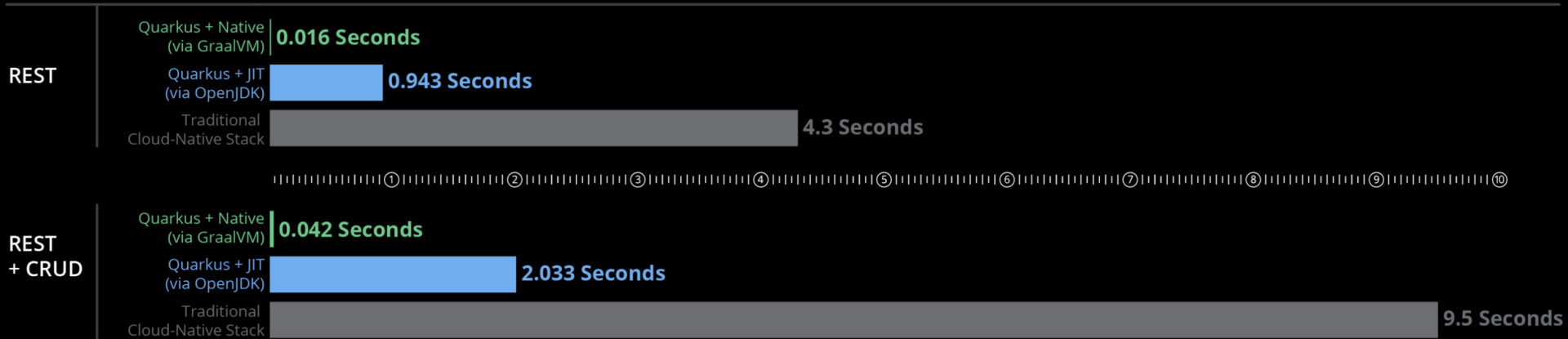


Traditional  
Cloud-Native Stack

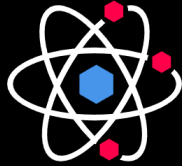
**209MB**

# Quarkus

## BOOT + First Response Time



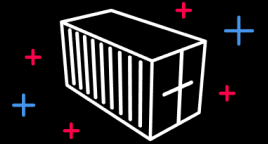
# Quarkus



Unifies imperative and reactive



Community and Standards



Container First



Developer Joy.

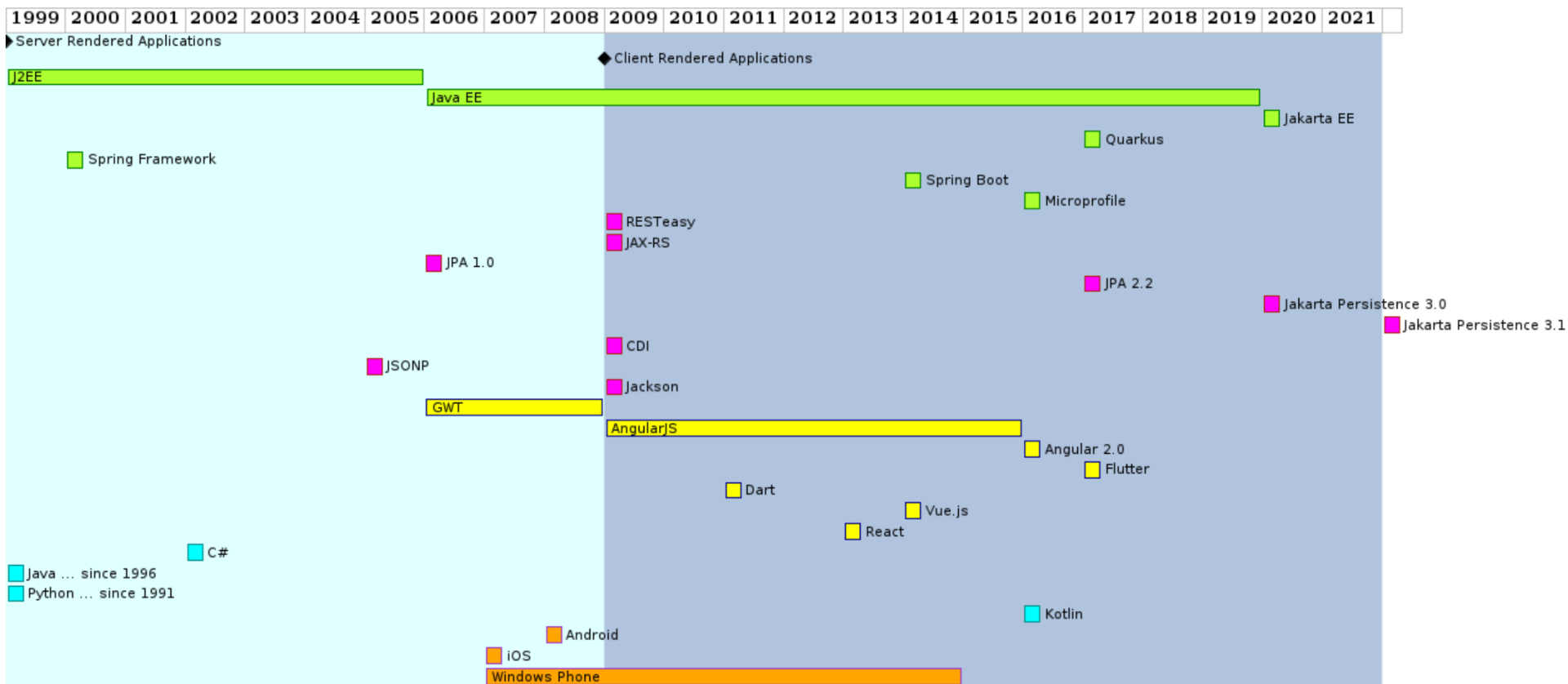


Kube-Native



# Quarkus

- Adheres to standards:
  - Microprofile
  - CDI
  - JAX-RS
  - JPA
  - JTA (Java Transaction API)
  - ...





# References

- [https://en.wikipedia.org/wiki/Spring\\_Framework#:~:text=The%20first%20version%20was%20written,was%20released%20in%20March%202004.](https://en.wikipedia.org/wiki/Spring_Framework#:~:text=The%20first%20version%20was%20written,was%20released%20in%20March%202004.)
- <https://projects.eclipse.org/projects/technology.microprofile>
- <https://entwickler.de/java/nebenbuhler-001>
- <https://quarkus.io/blog/quarkus-eclipse-microprofile-3-3/>
- <https://www.jrebel.com/blog/eclipse-microprofile#:~:text=Initially%20released%20in%20September%202016,JSON%2DP%2C%20and%20CDI.>
- [https://de.wikipedia.org/wiki/Jakarta\\_EE](https://de.wikipedia.org/wiki/Jakarta_EE)
- <https://chat.openai.com/auth/login>
-