

Chapter 10. Cloud-native Application Development

Bilkent University | CS443 | 2020, Spring | Dr. Orçun Dayıbaş

Cloud-native Applications

- What is “Cloud-nativeness”?

- CNCF Definition ([v1.0](#))
 - Cloud native technologies empower organizations to build and **run scalable applications in modern, dynamic environments** such as public, private, and hybrid clouds.
 - Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.
 - These techniques enable loosely coupled systems that are **resilient, manageable, and observable**. Combined with robust **automation**, they allow engineers to make high-impact changes frequently and predictably with minimal toil.
- Older version: “container packaged, dynamically scheduled, microservices-based application development & operations”
- Landscape: <https://landscape.cncf.io/>

Cloud-native Applications

● Maturity of Cloud-native Applications



Cloud-native Applications

- **Design principles**

- **Distribution**
 - Containers, microservices, API-driven dev.
- **Performance**
 - Resource efficient, concurrent, responsive
- **Automation**
 - Automated DevOps tasks
- **Resiliency**
 - Fault-tolerant, self-healing
- **Elasticity**
 - Scales dynamically and react to stimuli
- **Observability**
 - Logs, metrics and traces

Cloud-native Applications

- **Amazon API Mandate (2002)**



Jeff Bezos' Big Mandate

(circa 2002 — paraphrased)

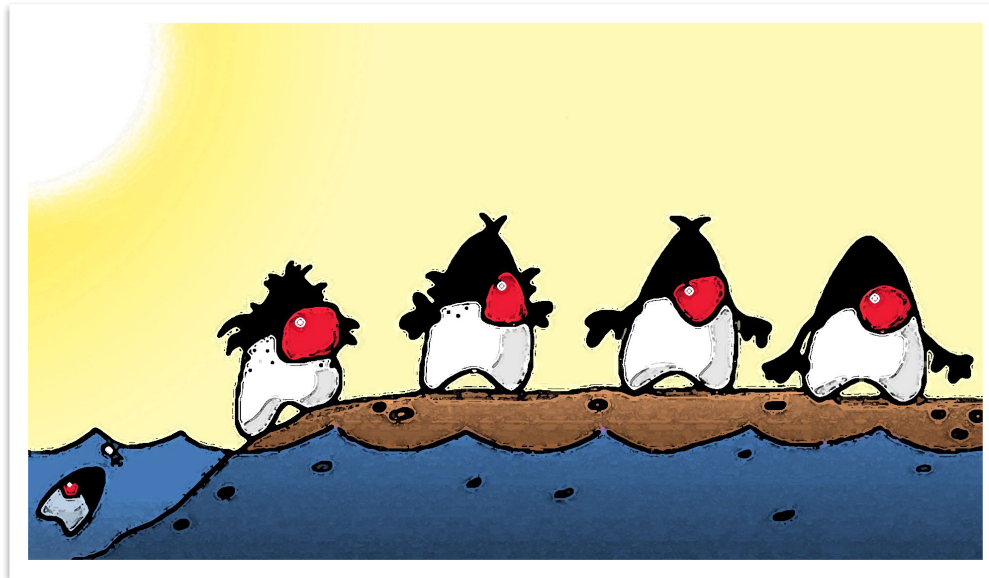
1. All teams will henceforth expose their data and functionality through service interfaces.
2. Teams must communicate with each other through these service interfaces.
3. No other communication is allowed other than service interfaces over the network.
4. It doesn't matter what technology they use.
5. All service interfaces must be designed to be **externalizable**.
6. Anyone who doesn't do this will be fired.

<https://plus.google.com/+RipRowan/posts/eVeauesvaVX>

Java Ecosystem

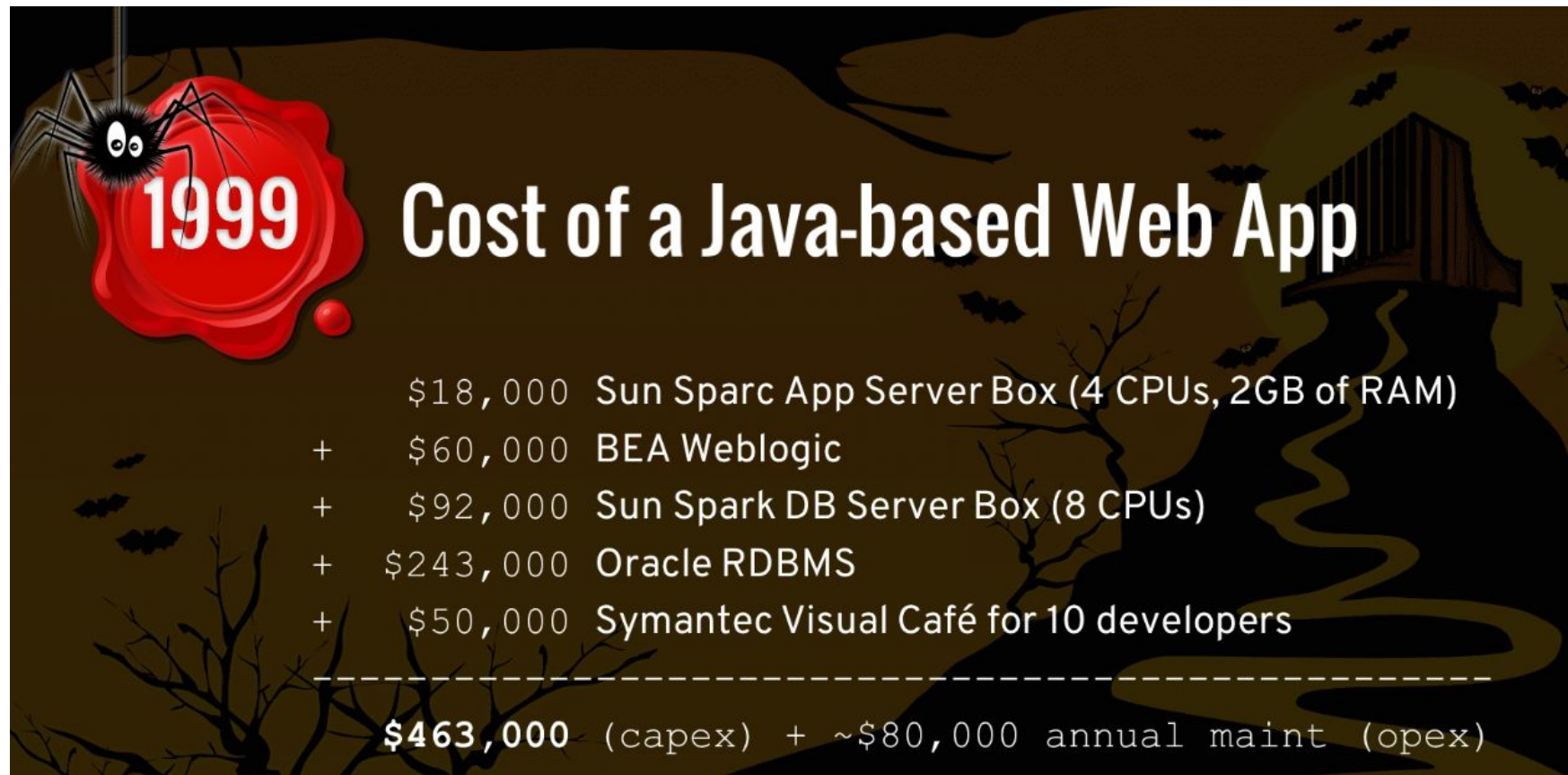
- **Why Java?**

- [Most Popular Programming Languages 1965 - 2019](https://www.tiobe.com/tiobe-index/)
- <https://www.tiobe.com/tiobe-index/>
- Very well established ecosystem (tooling, community, etc.)
- Still evolving to cope with emerging languages/platforms/ecosystems...



Java Ecosystem

- ~20 years ago

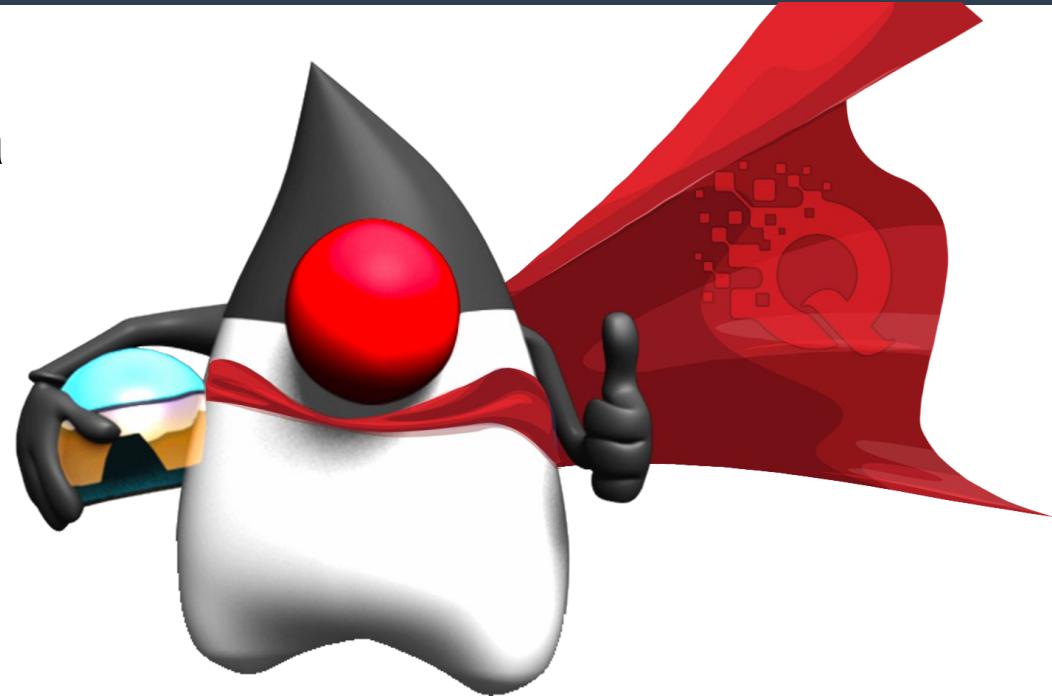
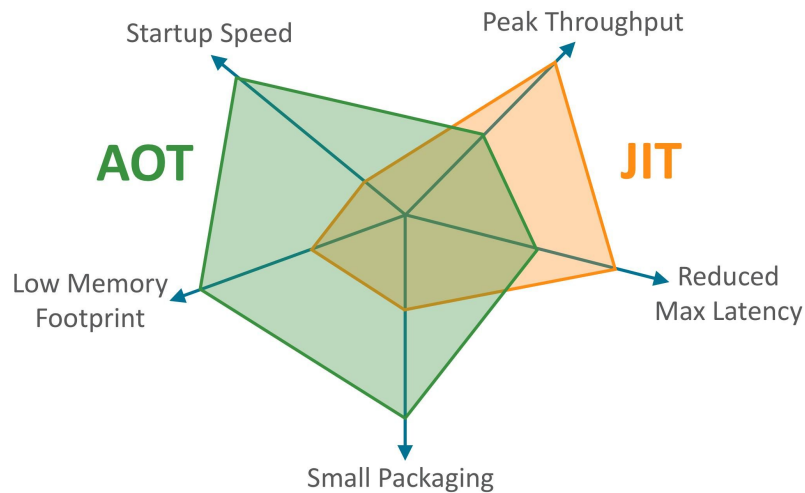


source: <http://bit.ly/clujnapoca2019>

Java Ecosystem

- **Quarkus**

- Supersonic, subatomic Java
- <https://quarkus.io/>
- GraalVM images
 - AOT vs JIT ([src](#))

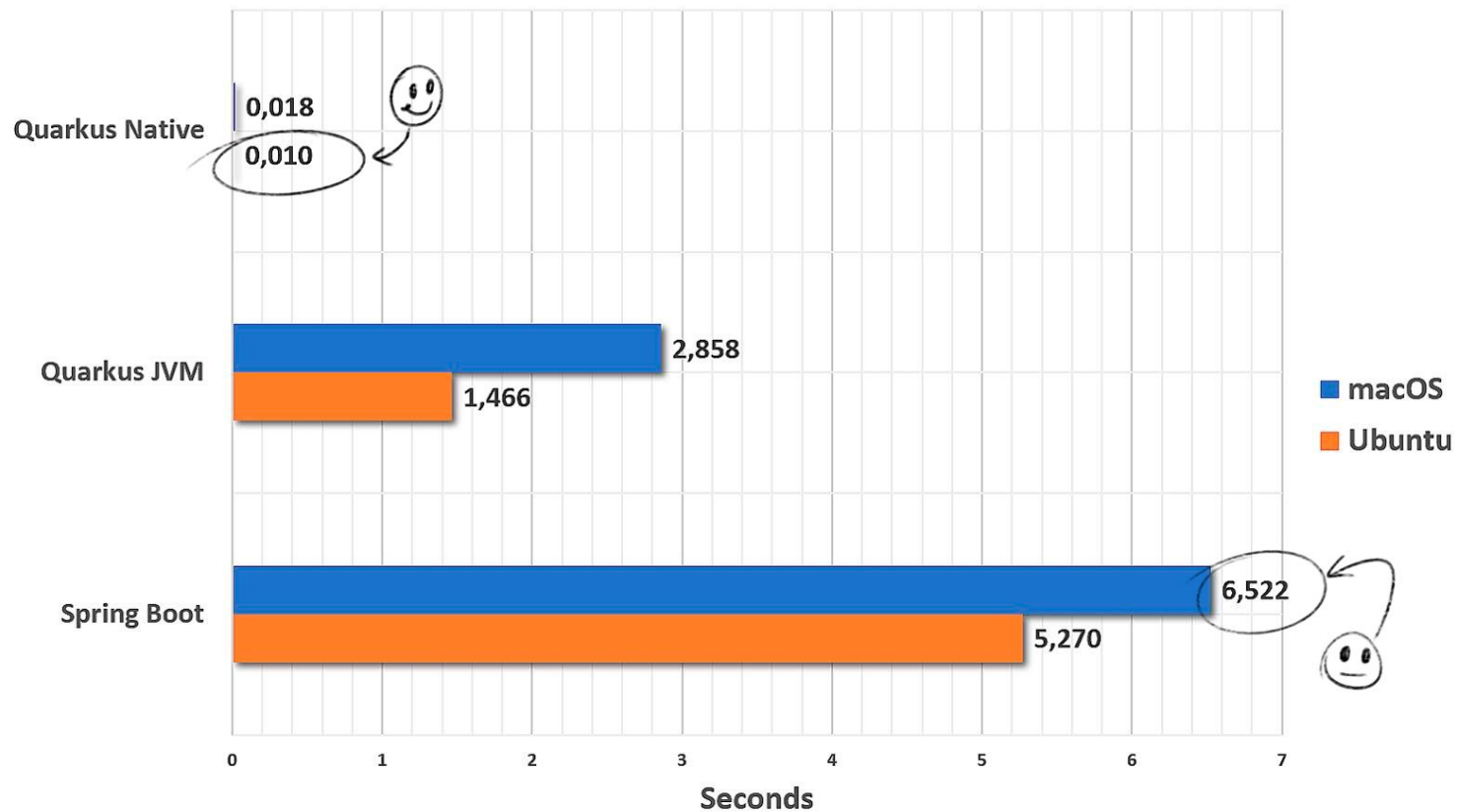


- ***let's dive in...***

Quarkus

- **Benchmarks**

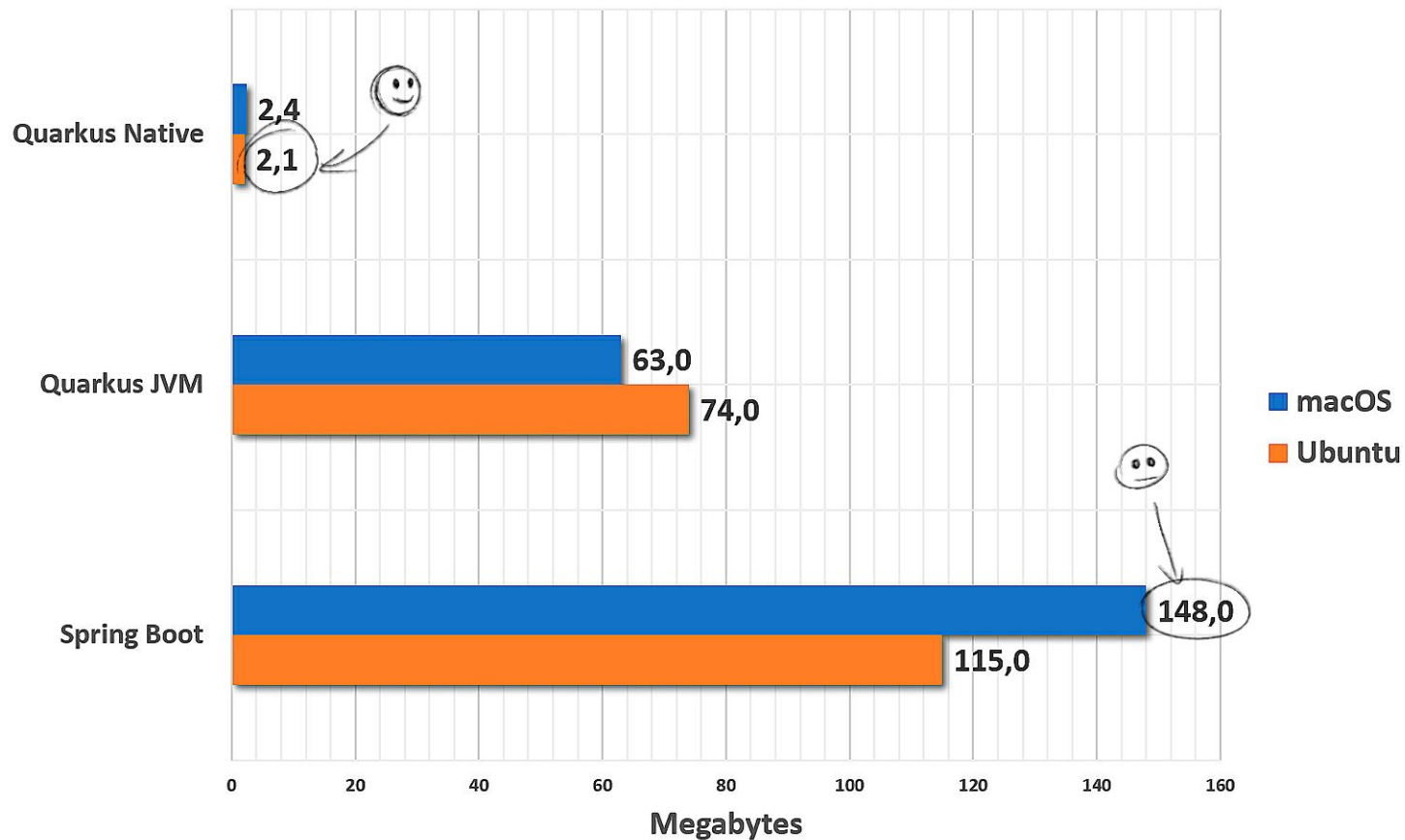
- Startup time



Quarkus

- **Benchmarks**

- Memory footprint





Q/A