



Microservices

with Thorntail.io (soon replaced by Quarkus)

Date: 22.04.2020

Author: Ing. Thomas Herzog M.Sc

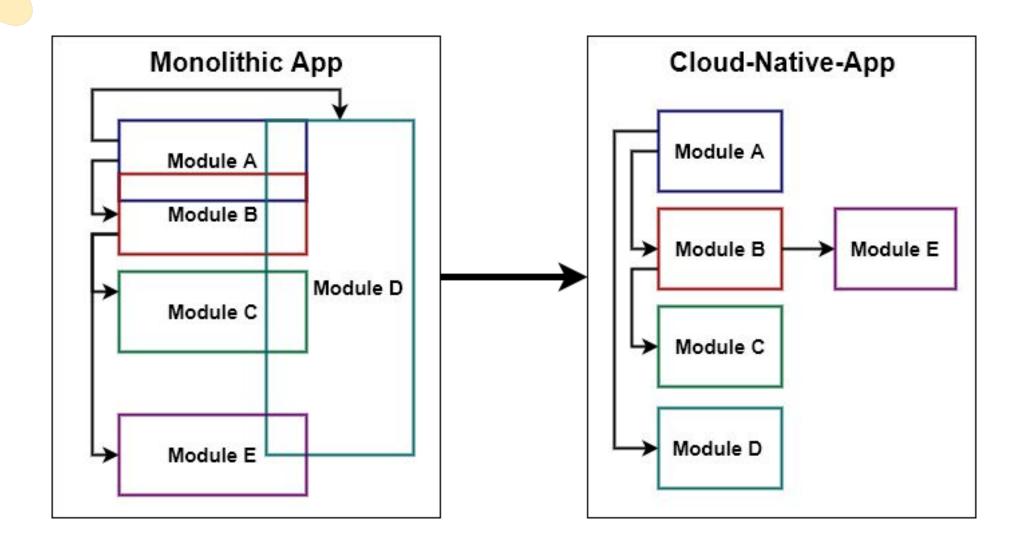


Agenda

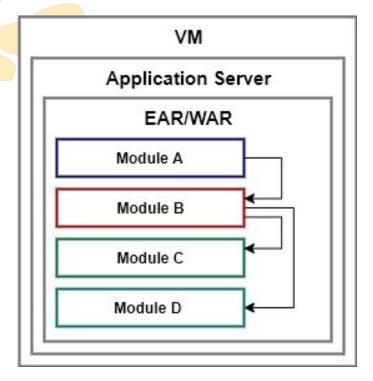
- // Cloud native apps
- // MicroProfile.io
- // Thorntail.io
- // Quarkus.io
- // Hands On
- // Links



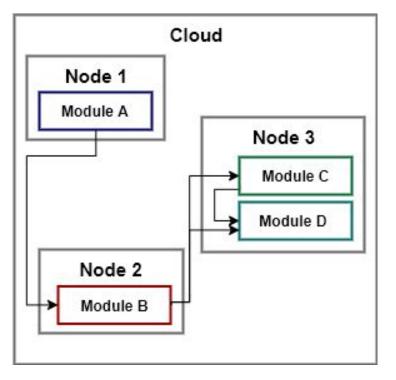
Cloud native apps - Monolith vs Microservice



Cloud native apps - In Process vs. Out of Process

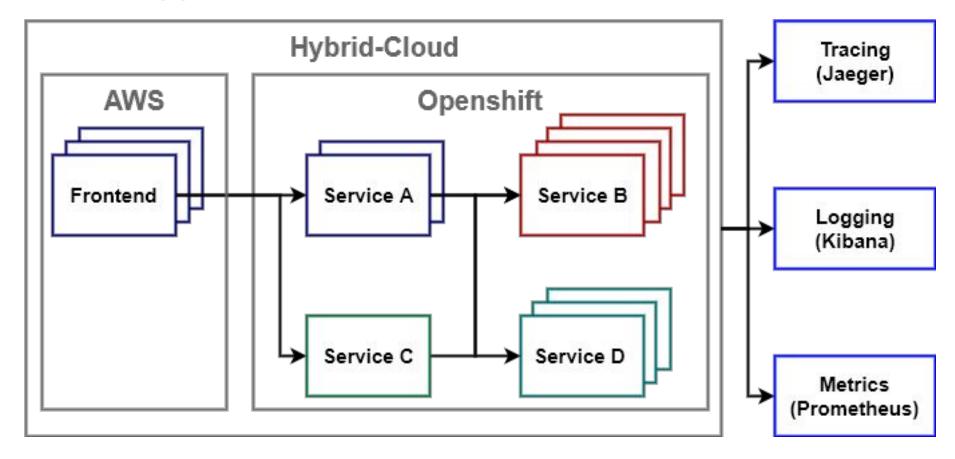


- // No network, no internet (except external access)
- // Communication will never fail
- // Very fast, in-memory, no TCP/IP, no HTTP
- // Logging, Tracing relatively easy



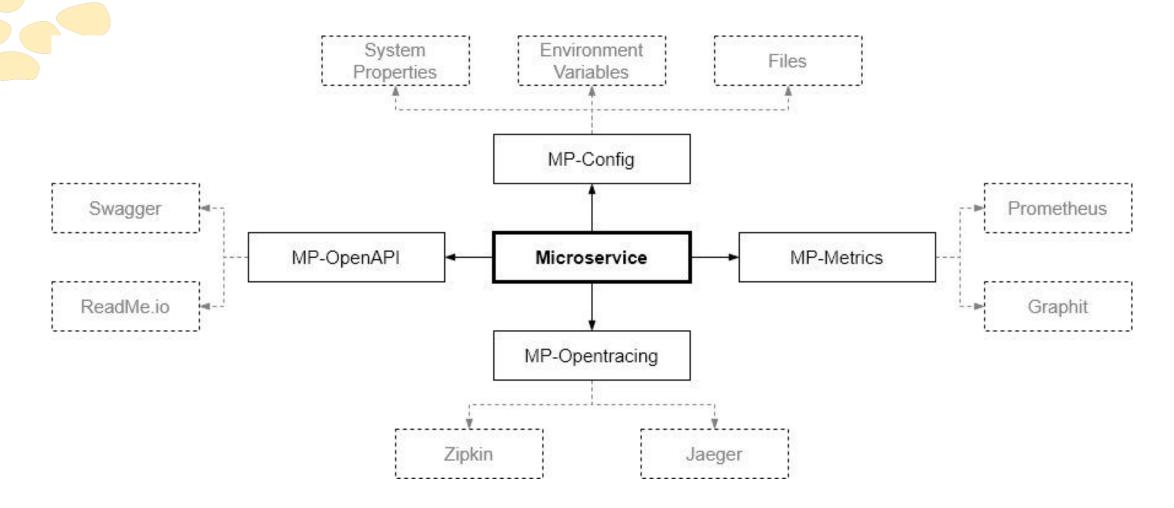
- // Network, Internet involved
- // Communication will fail sometimes
- // Slow, TCP/IP, Http
- // Distributed Logging and Tracing

Cloud native apps - Cloud infrastructure



- // Services are business values, containerized, resilient, scalable and observable
- // See https://12factor.net/

MicroProfile.io





MicroProfile.io

```
// Provide Specifications to solve common problems with microservices
// MicroProfile Configuration
                                  (from Deltaspike Config)
// MicroProfile Fault Tolerance
                                  (from Netflix's Hystrics)
// MicroProfile Opentracing
                                  (from Uber, Jaeger-UI)
// MicroProfile Rest-Client
                                  (from Apache's Resteasy)
// MicroProfile LRA
                                  (from Marayana, JTA)
  (Long-Running-Actions)
```

geparded

Thorntail.io

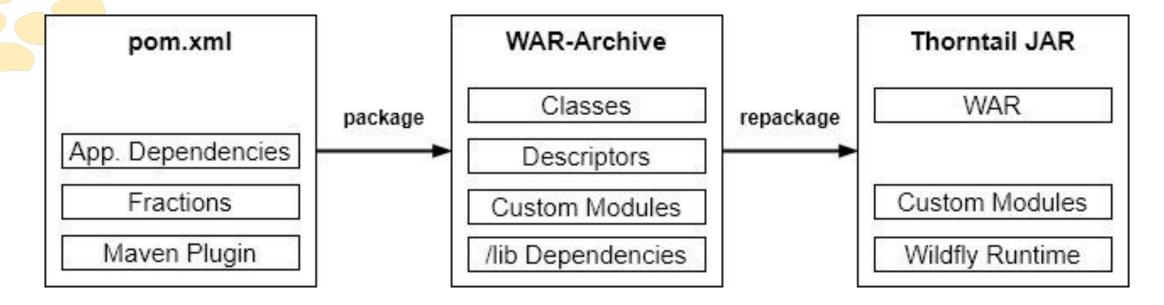
- // JEE compliant MicroService
- // Based on Wildfly
- // Optimized for MicroProfile
- // Modularized by Fractions
- // Produces a standalone JAR



https://microprofile.io

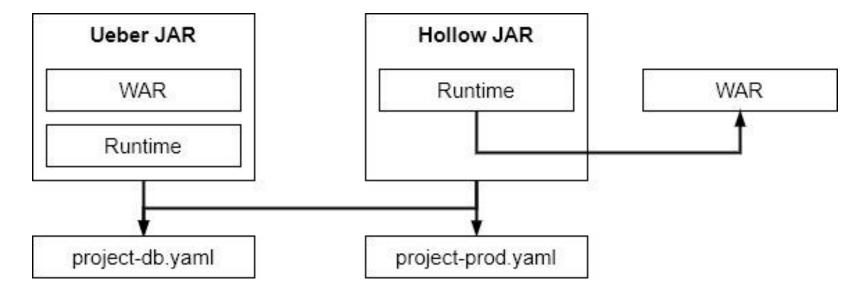


Thorntail.io



- // You feel Thorntail in your pom.xml not your Code
- // Thorntail modularized by Factions (JAX-RS; CDI, Swagger, ...)
- // We still get an ordinary WAR
- // WAR gets repackaged to standalone JAR

Thorntail.io



- // Ueber JAR contains WAR and Runtime components
- // Hollow JAR contains only runtime components
- // Multiple external configuration files possible

Quarkus.io (Idea)

- // Serverless Java
 - Needs very fast bootstrap
- // Immutable runtime
 - Why not? Deterministic bootstraps are faster than dynamic ones
- // Dead Code elimination
 - If not needed, throw it away
- // Compile time boot
 - This is what gives us an immutable/deterministic runtime
- // Cloud/Kubernetes native
 - Now Java integrates with the cloud very well
- // Native binaries

Now Java gets even faster



https://www.graalvm.org/



https://quarkus.io/

Quarkus.io (Why replace Thorntail)

- // Quarkus bootstrap is faster
- // Quarkus supports native binaries
- // Quarkus is Kubernetes native
- // Quarkus integrates common JEE specs.
- // Quarkus focus on developer experience
- // Quarkus is easy to test
- // Quarkus service can be reactive
- // But, still we have our JEE



https://www.graalvm.org/



https://quarkus.io/

Hands on

- // Sample application with microprofiles
- // Runs with Wildfly, Thorntail and Quarkus
- // Still plain Jakarta Enterprise Edition + MicroProfiles



Links

- // https://github.com/Gepardec/GranitTreff-220420
- // https://quarkus.io/
- // https://thorntail.io/
- // https://microprofile.io
- // https://12factor.net/
- // https://github.com/cncf