

THE DEVELOPER'S CONFERENCE



Criando Microservices com Java EE 7

George Gastaldi

Principal Software Engineer – Red Hat



Quem sou eu?



THE
DEVELOPER'S
CONFERENCE

- Engenheiro Software – Red Hat/JBoss
- Líder Projeto - JBoss Forge
 - <http://forge.jboss.org>
- Contribuidor do projeto WildFly Swarm
- E de outros projetos open-source...
 - Hibernate Validator, Weld, etc...
- Membro do GUJavaSC



Por quê Microservices?



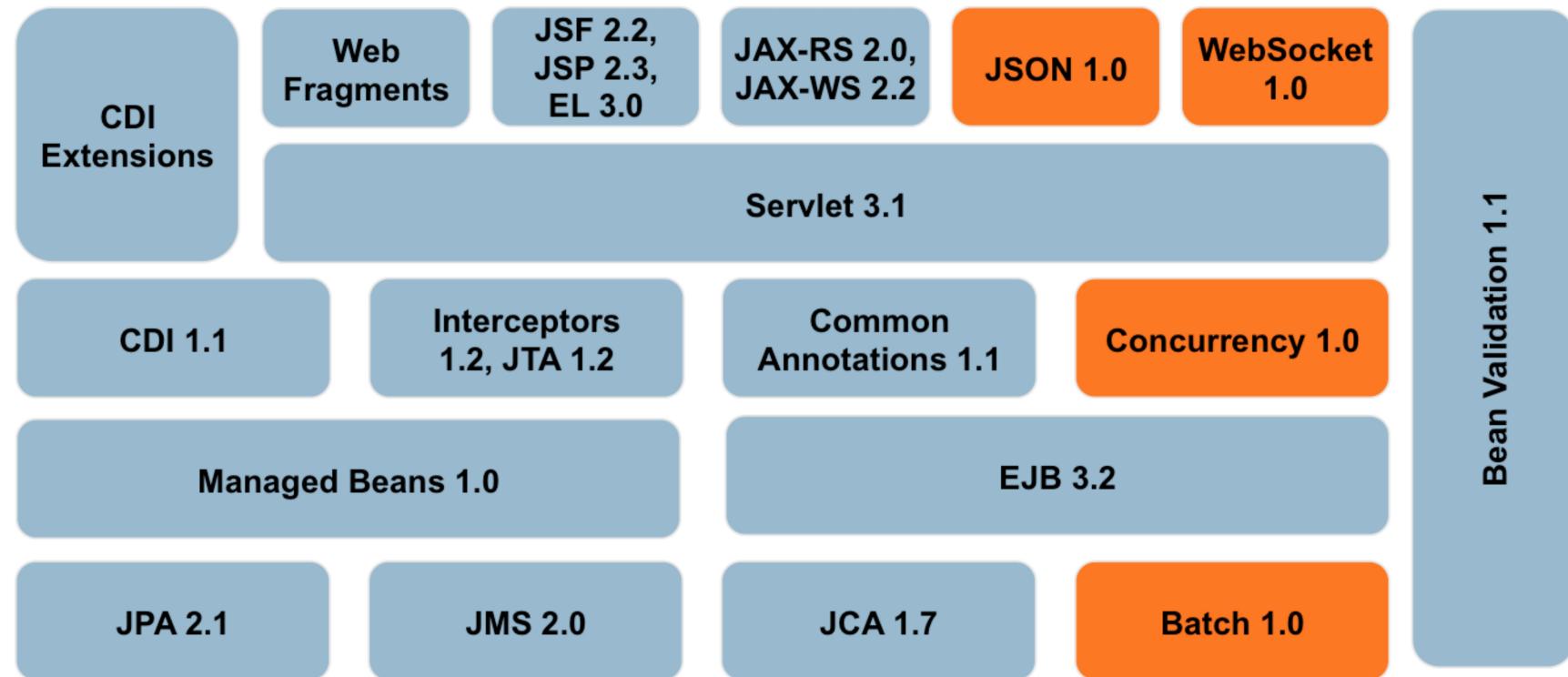
THE
DEVELOPER'S
CONFERENCE

- Desacoplados
- Ciclos de liberação independentes
- Cada serviço faz uma coisa bem, com limites bem-definidos
- Escalar independentemente
- Se costumava ser uma biblioteca, é muito provável que venha a ser um serviço

Java EE 7



THE
DEVELOPER'S
CONFERENCE





THE
DEVELOPER'S
CONFERENCE

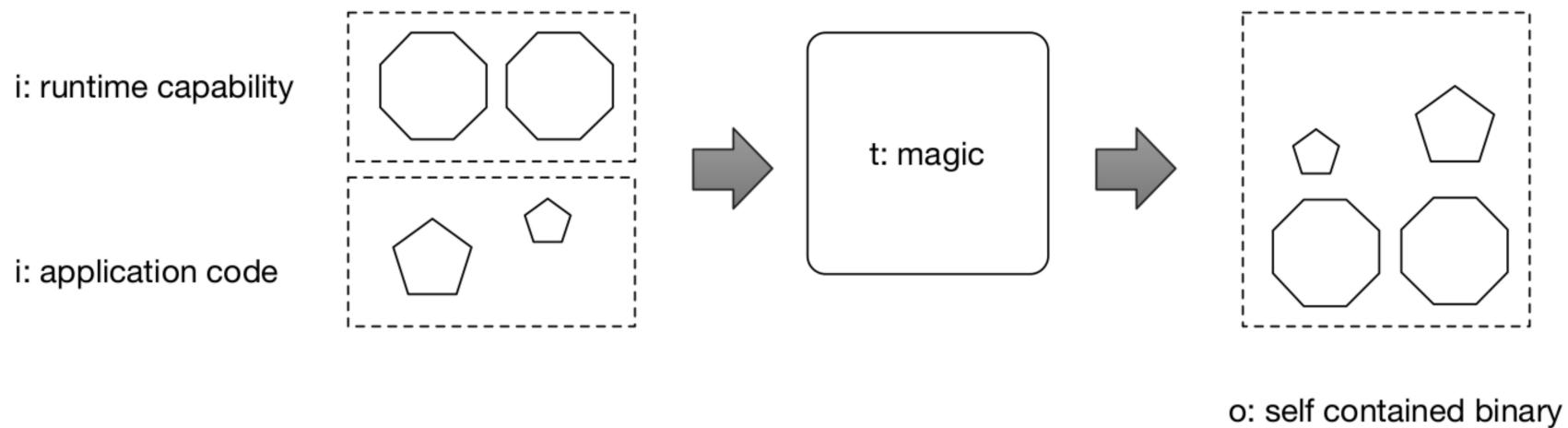
WildFly Swarm

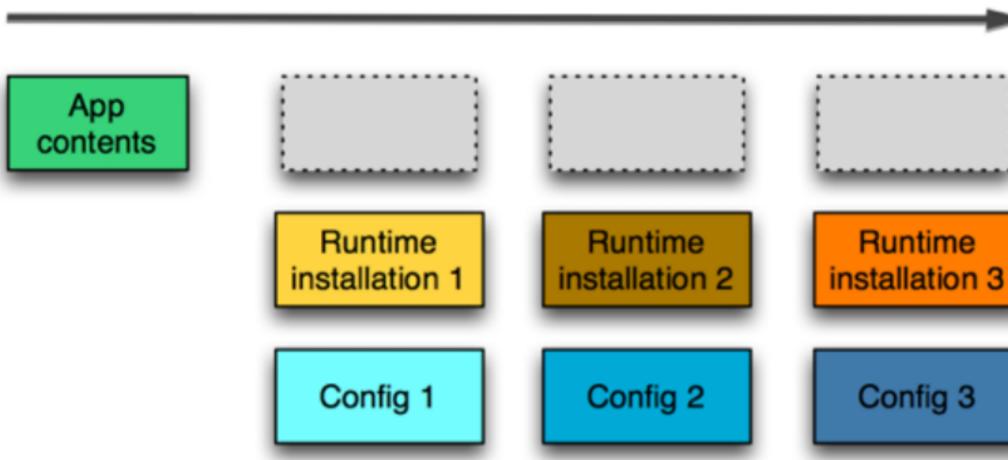
permite que você empacote
somente o necessário de um
servidor de aplicação Java EE

**juntamente com a sua
aplicação**

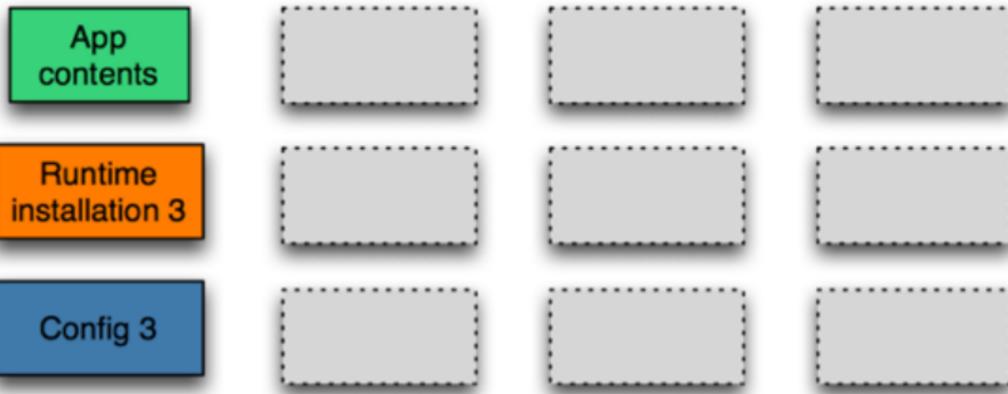


THE
DEVELOPER'S
CONFERENCE



Develop Test Staging Production

- Dev / Ops separation
- App / Runtime separation
- Configuration drift



- DevOps unification
- Self-containment
- Reproducibility

WildFly



THE
DEVELOPER'S
CONFERENCE

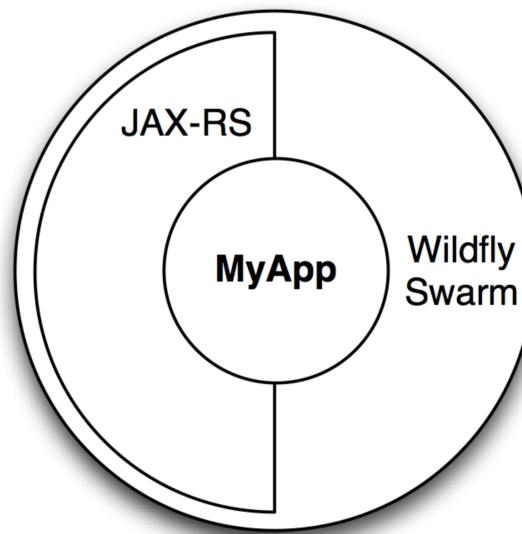
MyApp				
JAX-RS	EJB3	Transactions	CORBA	Batch
Wildfly				



WildFly Swarm



- Um simples JAR contendo a sua aplicação





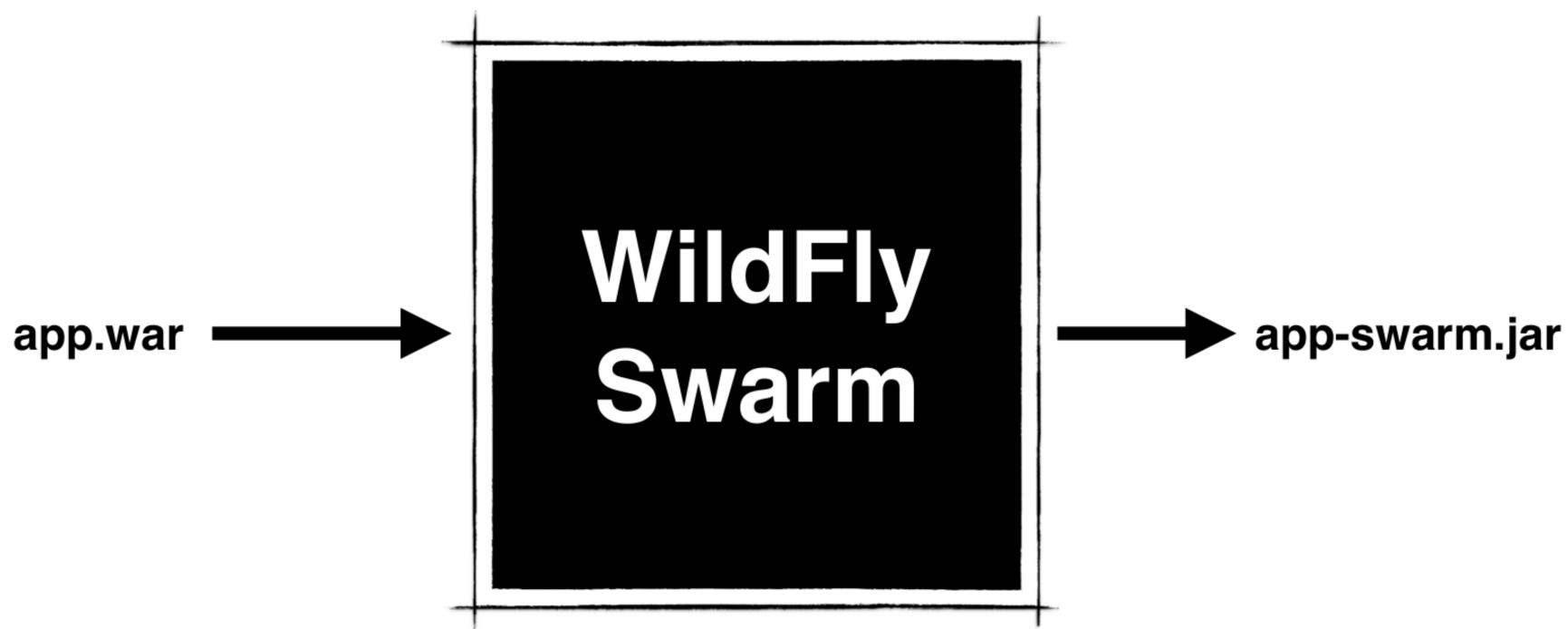
THE
DEVELOPER'S
CONFERENCE

Como configurar qualquer projeto Java EE 7 para usar WildFly Swarm?

```
<plugin>
  <groupId>org.wildfly.swarm</groupId>
  <artifactId>wildfly-swarm-plugin</artifactId>
  <version>${version.wildfly-swarm}</version>
  <executions>
    <execution>
      <id>package</id>
      <goals>
        <goal>package</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```



THE
DEVELOPER'S
CONFERENCE



Execute!



THE
DEVELOPER'S
CONFERENCE

```
$ java -jar myapp-swarm.jar
```



THE
DEVELOPER'S
CONFERENCE

Hello World JAX-RS WildFly Swarm



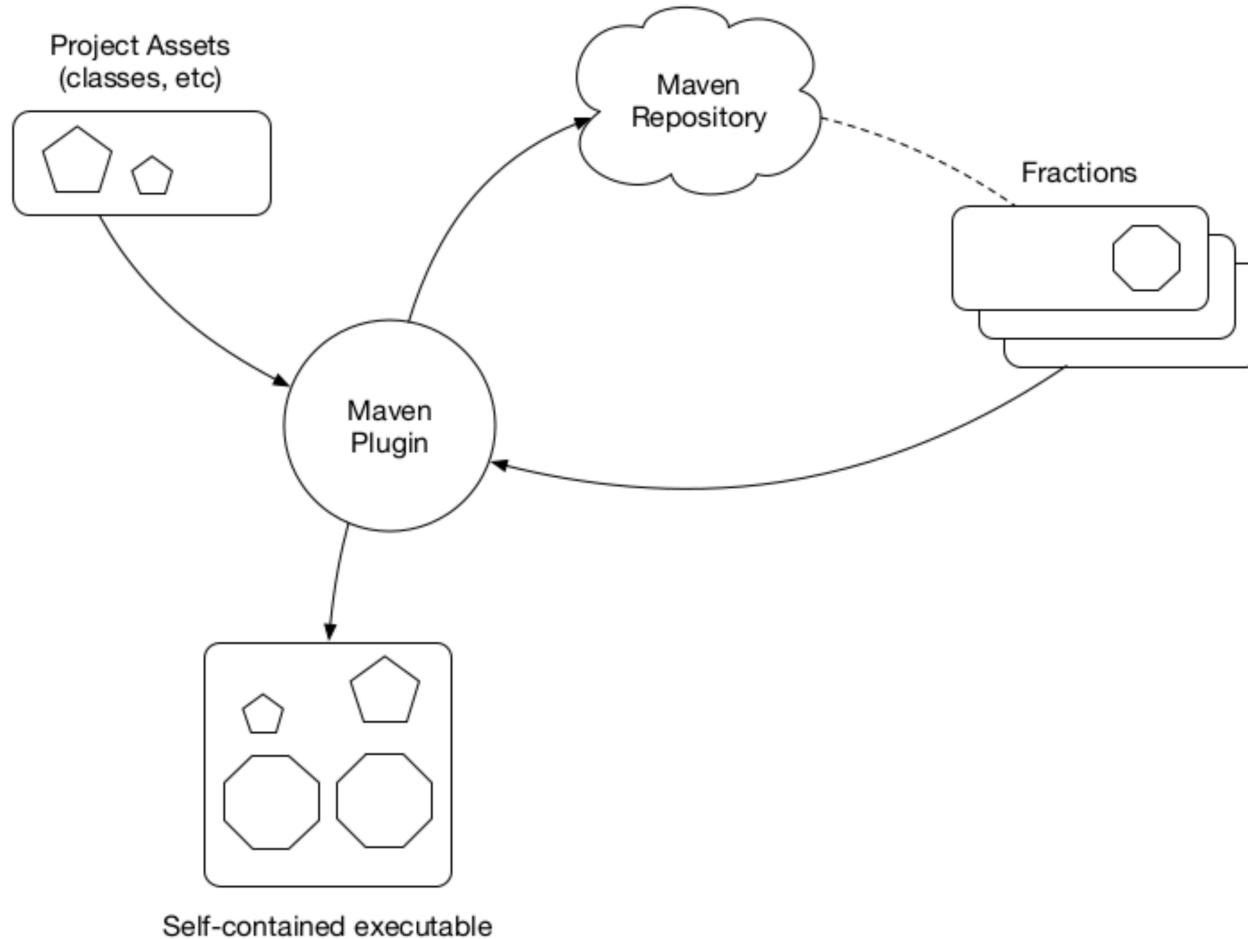
THE
DEVELOPER'S
CONFERENCE

Fractions

Fractions



THE
DEVELOPER'S
CONFERENCE



Fractions



- Suportam configurações explícitas ou implícitas
 - Na maioria dos casos, não é necessário configurar nada
- Podem ser detectadas ou declaradas
 - O caso mais simples é um projeto <war>, somente com o plugin maven
- Toda a especificação Java EE 7 é suportada:
 - JPA, JAX-RS, CDI, EJB, JMS, Batch,...
- E também além...

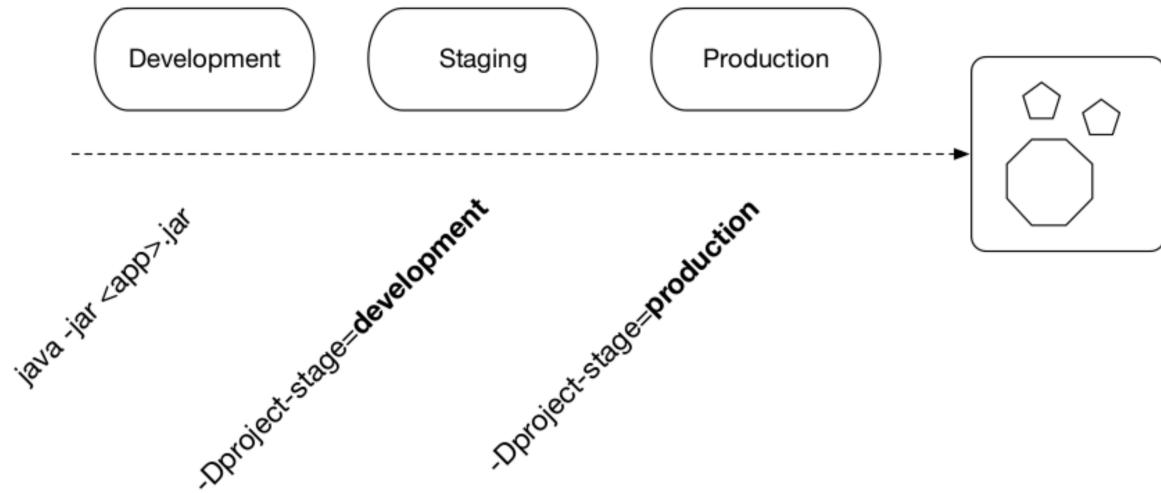
Configuração/Ambiente



THE
DEVELOPER'S
CONFERENCE

```
logger:  
  level: DEBUG  
database:  
  jdbc:  
    url: foo  
---  
project:  
  stage: development  
logger:  
  level: DEBUG  
database:  
  jdbc:  
    url: bar  
---  
project:  
  stage: production  
logger:  
  level: INFO  
database:  
  jdbc:  
    url: somethingElse
```

project-stages.yml

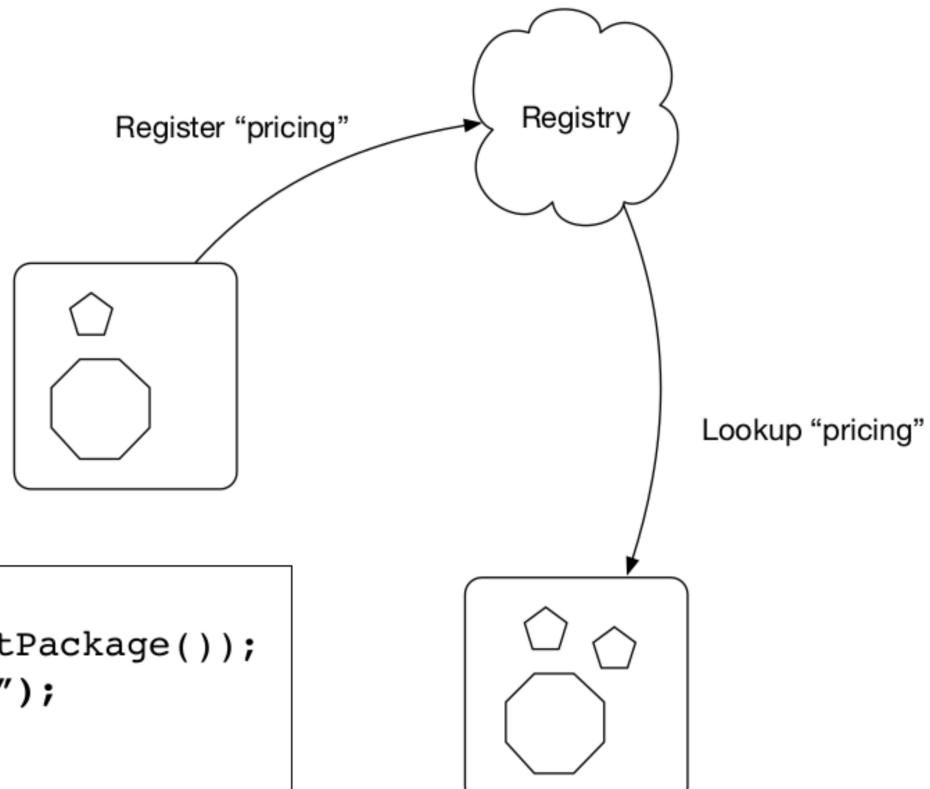


Registro Serviços



THE
DEVELOPER'S
CONFERENCE

```
<dependency>
<groupId>org.wildfly.swarm</groupId>
<artifactId>topology-consul</artifactId>
</dependency>
```

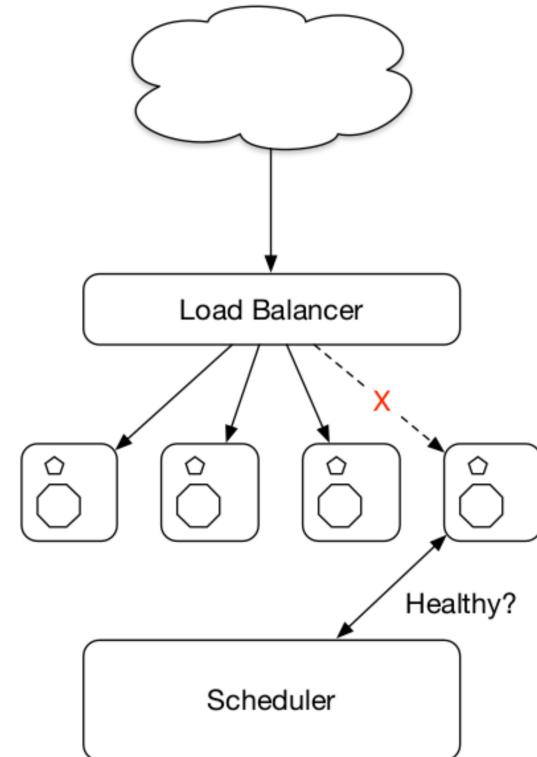


Verificação de Disponibilidade



```
<dependency>
<groupId>org.wildfly.swarm</groupId>
<artifactId>monitor</artifactId>
</dependency>
```

```
@GET
@Path( "/other" )
@Health
public HealthStatus checkSomethingElse() {
    return HealthStatus.up();
}
```

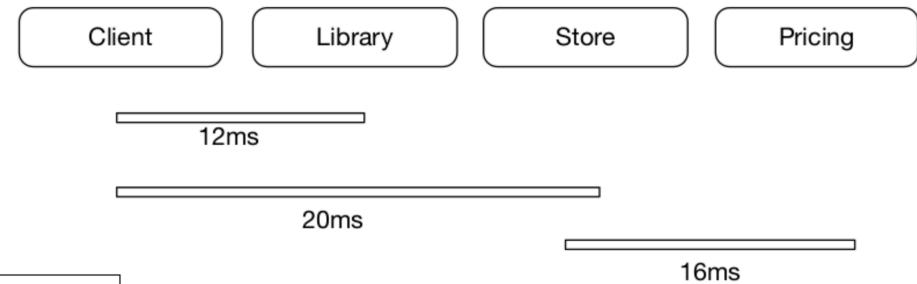


Rastreamento Distribuído



```
<dependency>
  <groupId>org.wildfly.swarm</groupId>
  <artifactId>zipkin</artifactId>
</dependency>
```

```
container.fraction(
  new ZipkinFraction("pricing")
    .reportAsync("http://localhost:9411/api/v1/spans")
    .sampleRate(0.1f) // keep 10%
);
```



Resiliência

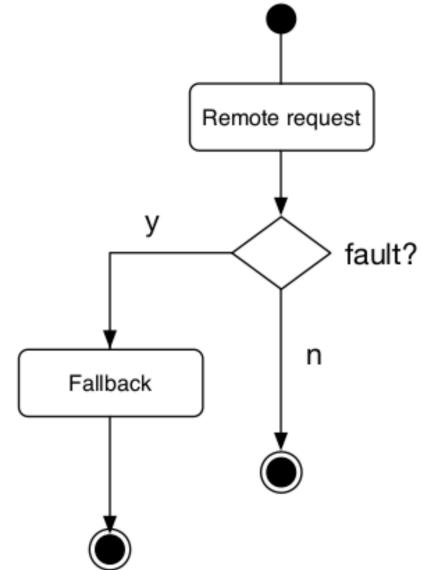


```
<dependency>
<groupId>org.wildfly.swarm</groupId>
<artifactId>hystrix</artifactId>
</dependency>
```

```
@CircuitBreaker(fallbackMethod = "myFallback")
public String isolatedCommand() {
    Client client = ClientBuilder.newClient();
    WebTarget target = client.target("pricing");

    Response response = target.get();
    return response.readEntity(String.class);
}

public String myFallback() {
    return ..; // cached values
}
```



Fractions



THE
DEVELOPER'S
CONFERENCE

WildFly Subsystems

DataSource	EJB	JAX-RS	JCA	Transactions
Remoting	JPA	Keycloak	JSF	Messaging
Undertow	CDI	BeanVal	JMX	Mail
Clustering	Infinispan	Hawkular	Validator	Batch
JGroups	Webservice	Logging	Hibernate Search	

Non-WildFly

Ribbon	Hystrix	Logstash	Camel
RxJava	RxNetty	Hashicorp Consul	Swagger



Eclipse Microprofile



THE
DEVELOPER'S
CONFERENCE

- 1.0 – Setembro 2016
- 1.1 – Planejado para segundo semestre 2017
 - Config API
 - Health Check
 - Interoperabilidade JWT (JSON Web Tokens)





THE
DEVELOPER'S
CONFERENCE

DEMO TIME !

