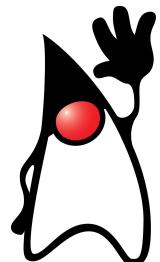
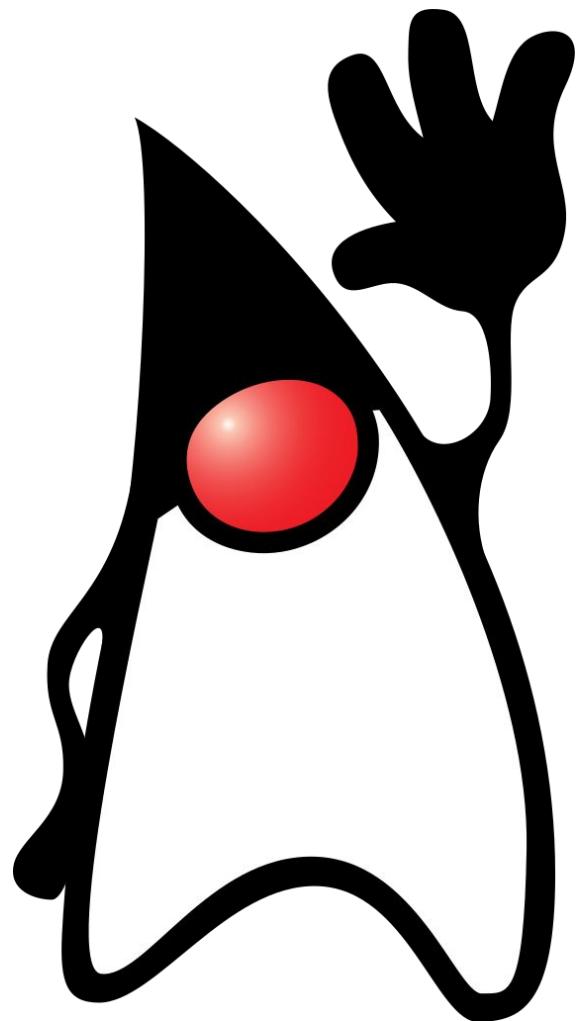


Desenvolvimento com Frameworks e Componentes

Michel Vasconcelos

michel.vasconcelos@gmail.com

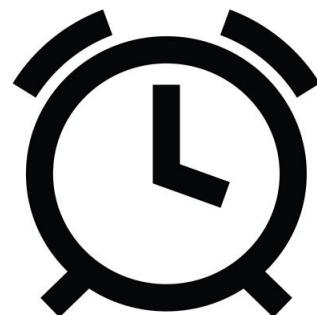




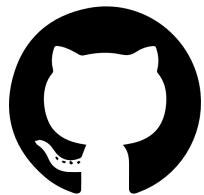
OBJETIVOS

- Entender o conceito de frameworks e componentes
- Conhecer as características da especificação Java EE
 - EJBs
- Conhecer as características do framework Spring

Regras



Contatos

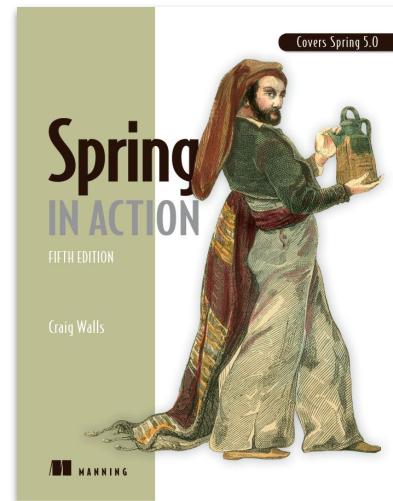
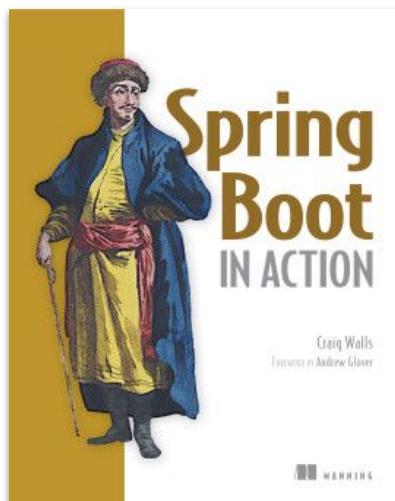
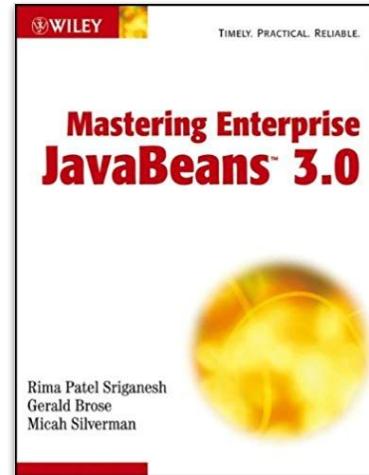
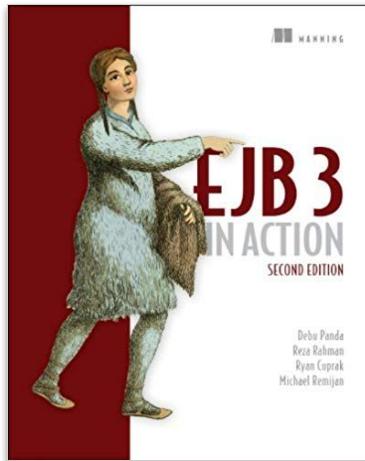


<https://github.com/michelav-uni7>

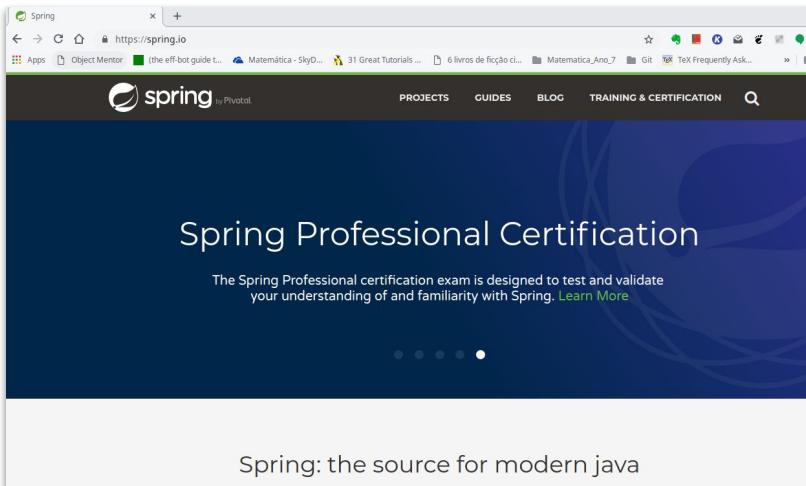


michel.vasconcelos@gmail.com

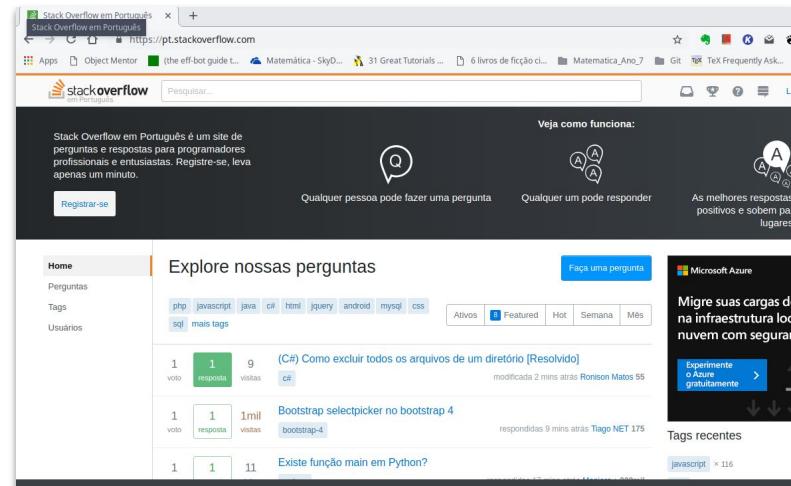
Referências



Referências



The screenshot shows the Spring Professional Certification landing page. At the top, there's a navigation bar with links for 'PROJECTS', 'GUIDES', 'BLOG', 'TRAINING & CERTIFICATION', and a search icon. Below the navigation is a large banner with the title 'Spring Professional Certification'. Underneath the banner, a sub-section titled 'Spring: the source for modern java' is visible. The overall design is clean with a dark blue header and a white main content area.



The screenshot shows the homepage of Stack Overflow in Portuguese. The top navigation bar includes links for 'Perguntas', 'Tags', and 'Usuários'. The main content area features a section titled 'Explore nossas perguntas' (Explore our questions) with three recent posts listed. The first post is about removing files from a directory in C#. The second is about Bootstrap selectpicker. The third is a question about Python's main function. On the right side, there are sections for 'Veja como funciona:' (See how it works), 'Tags recentes' (Recent tags), and an advertisement for Microsoft Azure.

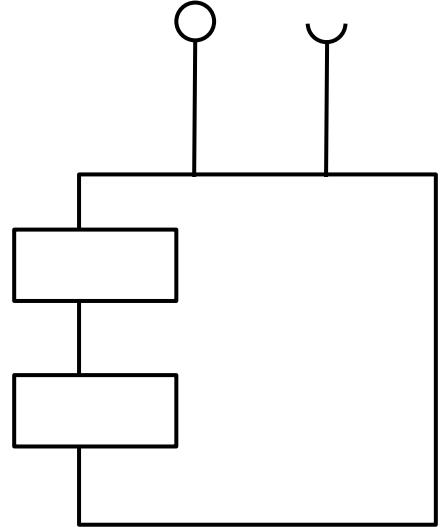
Componentes, Frameworks e Aplicações Corporativas



Fazer do “Zero”

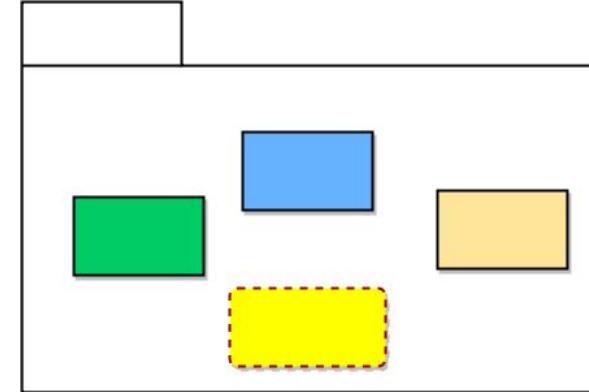
vs

Reuso



Componente de software é um artefato (ou conjunto de artefatos) que forma uma unidade de comportamento e interface bem definidos, podendo ser implantado, testado, composto e reusado de forma independente.

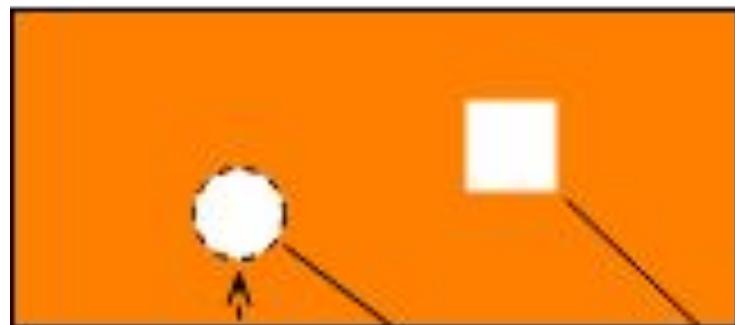
Technical Concepts of Component-Based Software Engineering



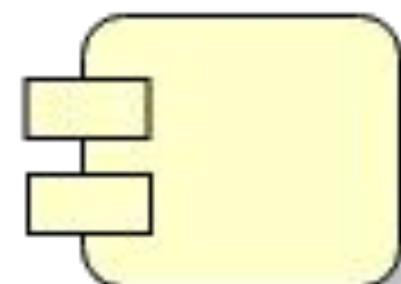
Framework de software é uma abstração formada por um conjunto de classes que auxilia no desenvolvimento de soluções para (geralmente) um domínio específico.

Software Framework - Wikipedia

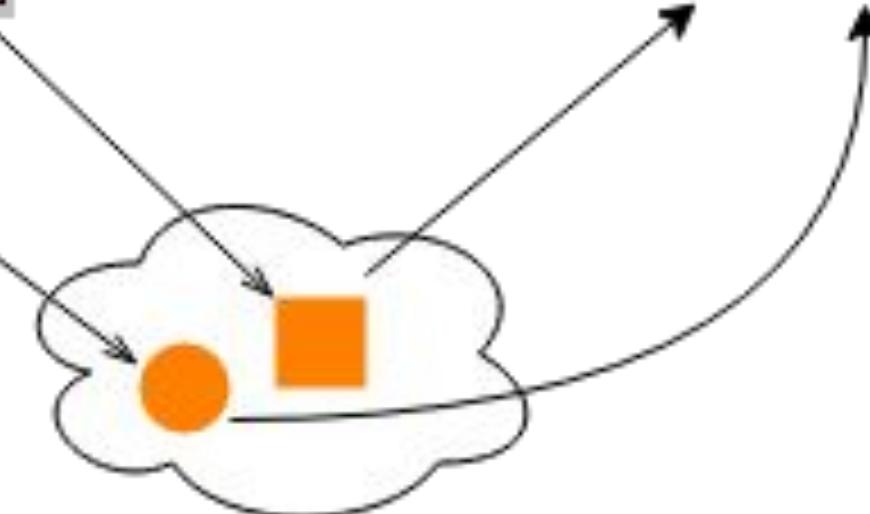
Framework



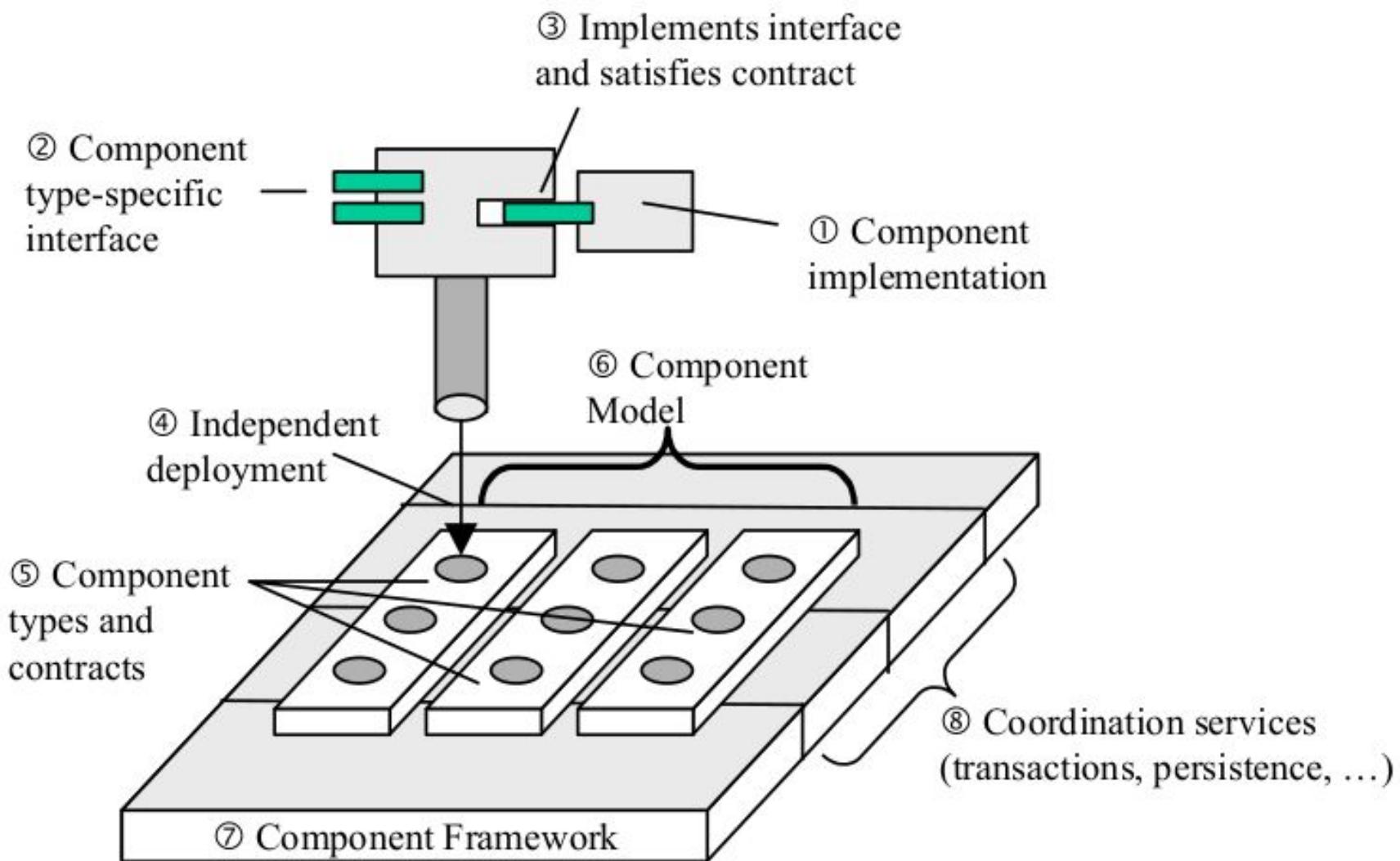
Componente



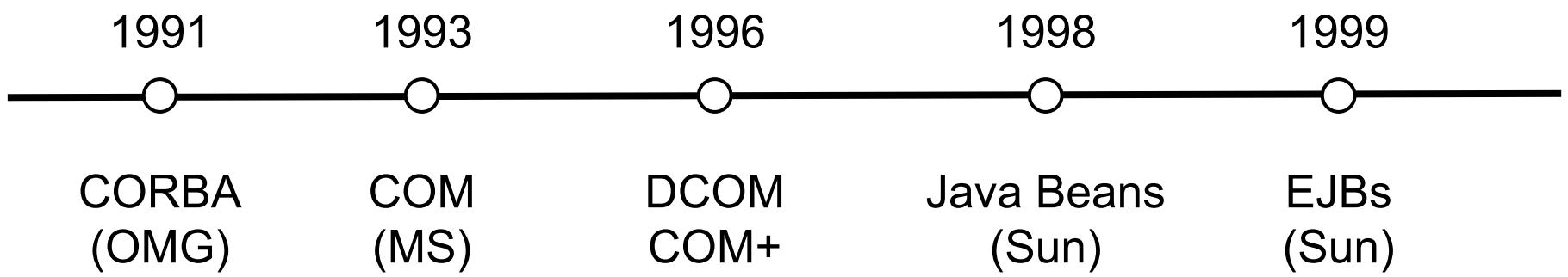
Hot Spot

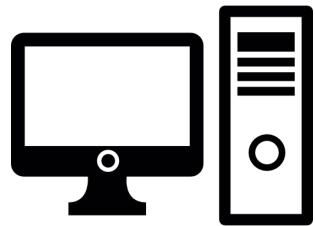


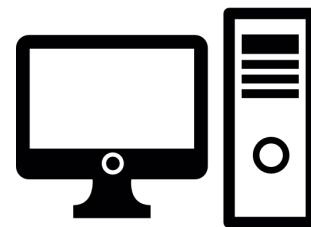
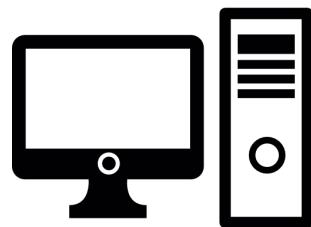
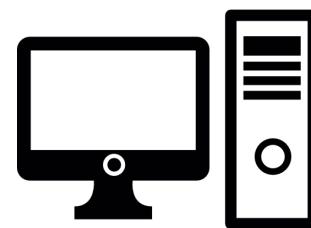
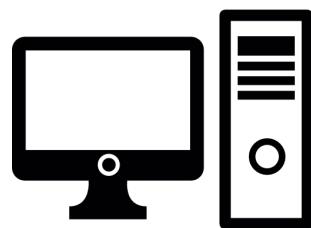
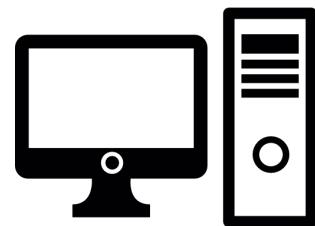
Código

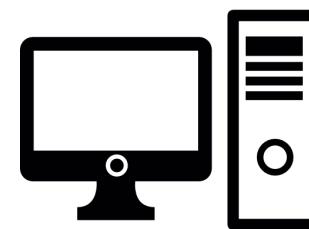
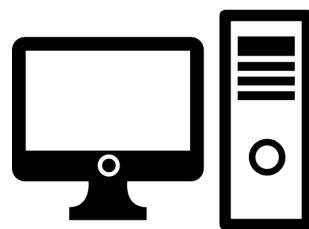
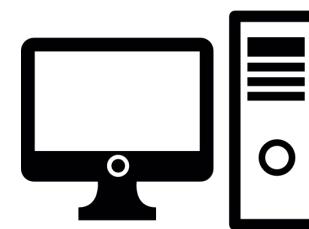
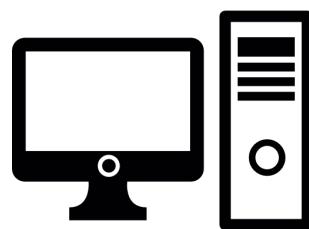


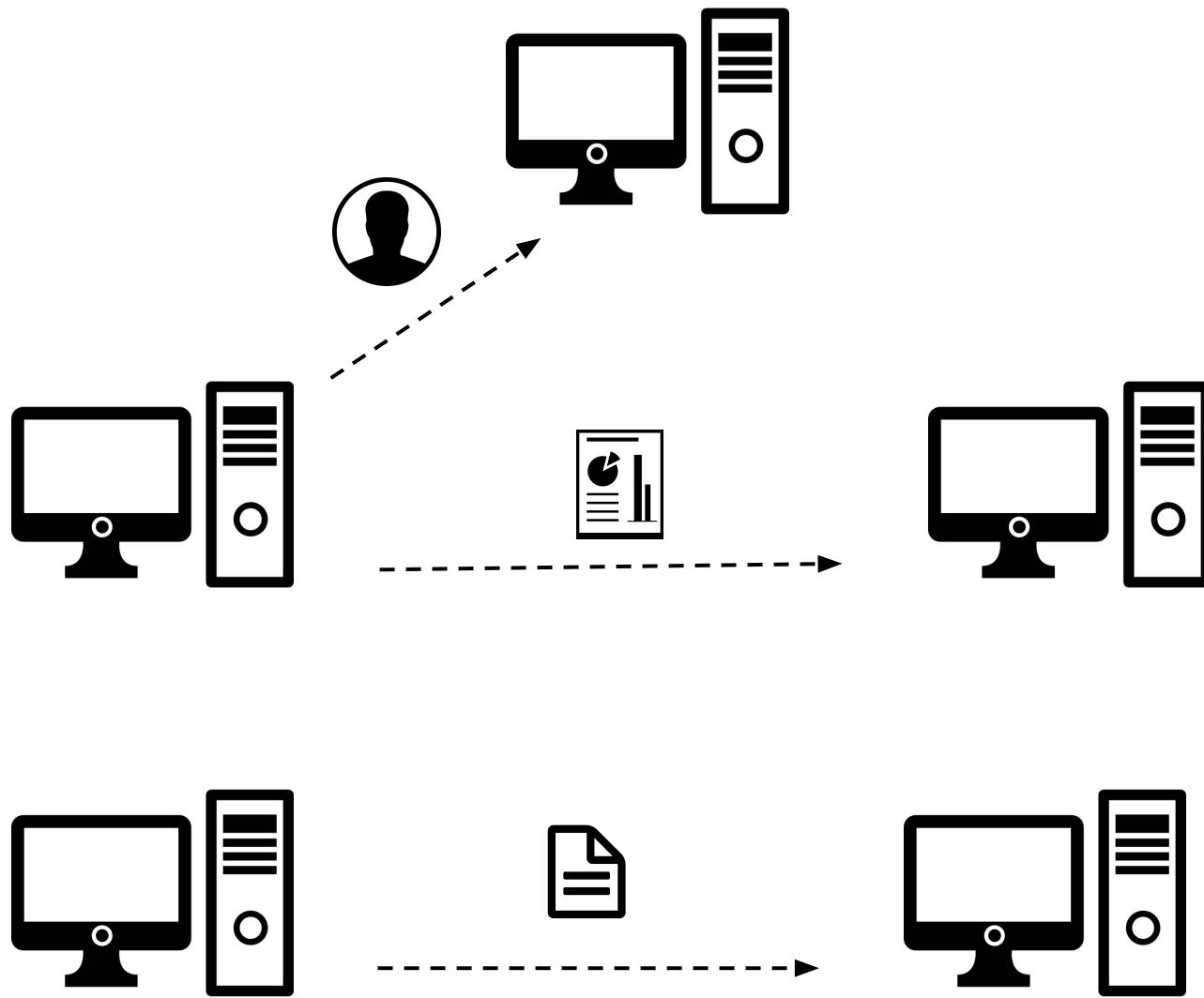
Fonte: Technical Concepts of Component-Based Software Engineering (SEI)











Aplicação Corporativa

(Enterprise Application)

Larga escala

Multicamada

Escalável

Confiável

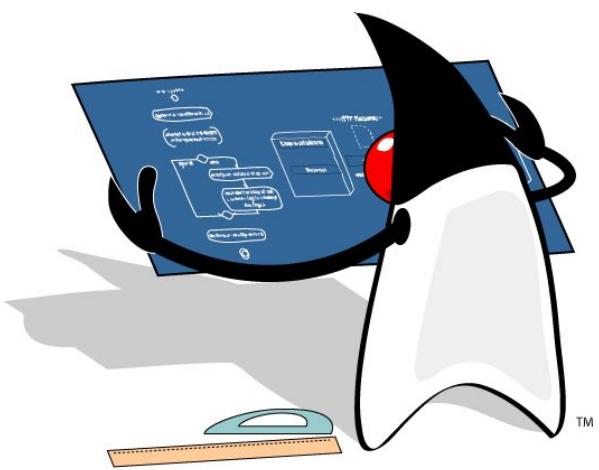
Segura

Distribuída





Plataforma Java EE



Java Enterprise Edition é uma plataforma de auxílio a desenvolvedores na construção de aplicações corporativas.

Java Platform, Enterprise Edition (Java EE) 8

Oracle

Java EE 8



Batch	Dependency Injection	JACC	JAXR	JSTL	Management
Bean Validation	Deployment	JASPIC	JMS	JTA	Servlet
CDI	EJB	JAX-RPC	JSF	JPA	Web Services
Common Annotations	EL	JAX-RS	JSON-P	JavaMail	Web Services Metadata
Concurrency EE	Interceptors	JAX-WS	JSP	Managed Beans	WebSocket
Connector	JSP Debugging	JAXB			
JSON-B	Security				

Fonte: <https://blogs.oracle.com/java/java-ee-8-overview>

JCP vs JSR

The Java Community Process(S) × +

https://jcp.org/en/jsr/detail?id=388

Apps Object Mentor (the eff-bot guide t... Matemática - SkyD... 31 Great Tutorials... 6 livros de ficção ci... Matematica_Ano_7 Git Tex Frequently Ask... Outros favoritos

Press Room | Get Java Here | Search JSRs GO

 Java Community Process Community Development of Java Technology Specifications

JSRs

Search JSRs GO

JSRs by Platform
JSRs by Technology
JSRs by Stage
JSRs by Committee
List of All JSRs

My JCP

Sign-in Register for Site

Use of JCP site is subject to the JCP Terms of Use and the Oracle Privacy Policy

JCP Info

About JCP Get Involved Community Resources Community News FAQ Contact Us

Community Development of Java Technology Specifications

JSR Community Expert Group

Summary | Proposal | Detail (Summary & Proposal) | Nominations

JSRs: Java Specification Requests

JSR 388: Java™ SE 13

Stage Access Start Finish

JSR Review Ballot	View results	18 Dec, 2018	24 Dec, 2018
JSR Review		04 Dec, 2018	17 Dec, 2018

Status: Active
JCP version in use: 2.11
Java Specification Participation Agreement version in use: 2.0

Description:
The JSR for the Java SE 13 Platform. The Reference Implementation of this Specification is the Java Development Kit, version 13.

Expert Group Transparency:
[Issue Tracking](#)

Team

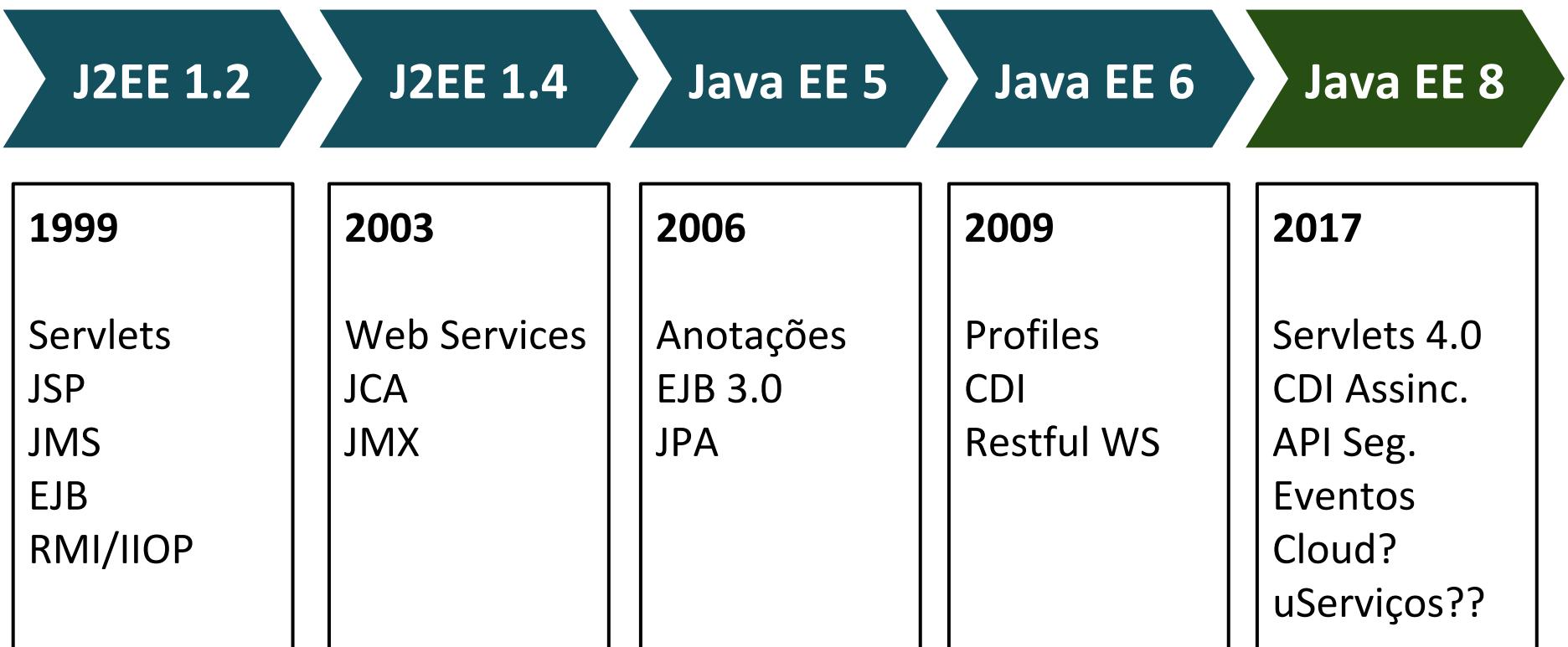
Specification Leads

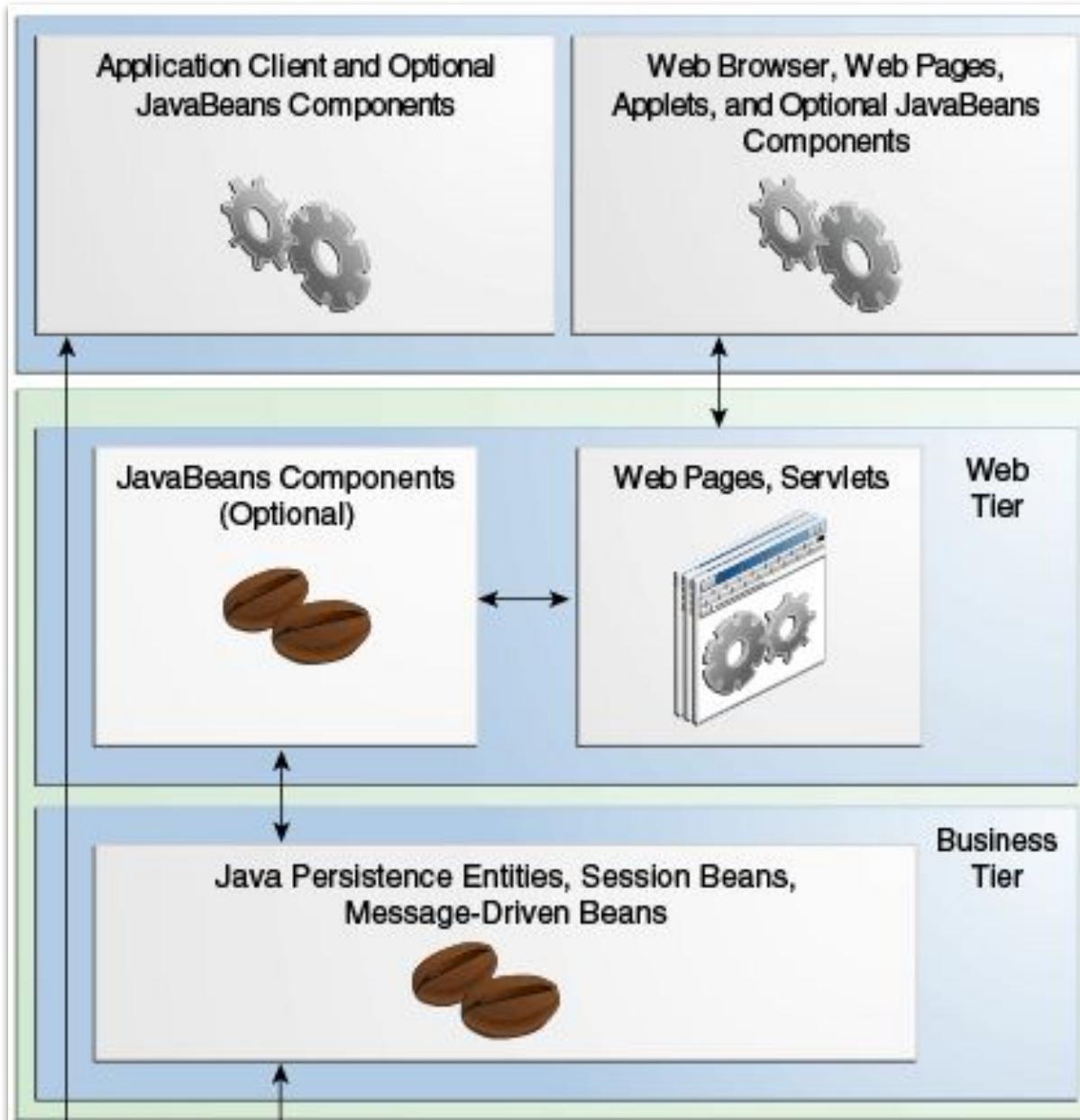
Iris Clark	Oracle
Brian Goetz	Oracle

Expert Group

Azul Systems, Inc. : Simon Ritter	Eclipse Foundation, Inc. : Manoj N Palat	IBM : Tim Ellison
Oracle	Oracle	SAP SE

Evolução

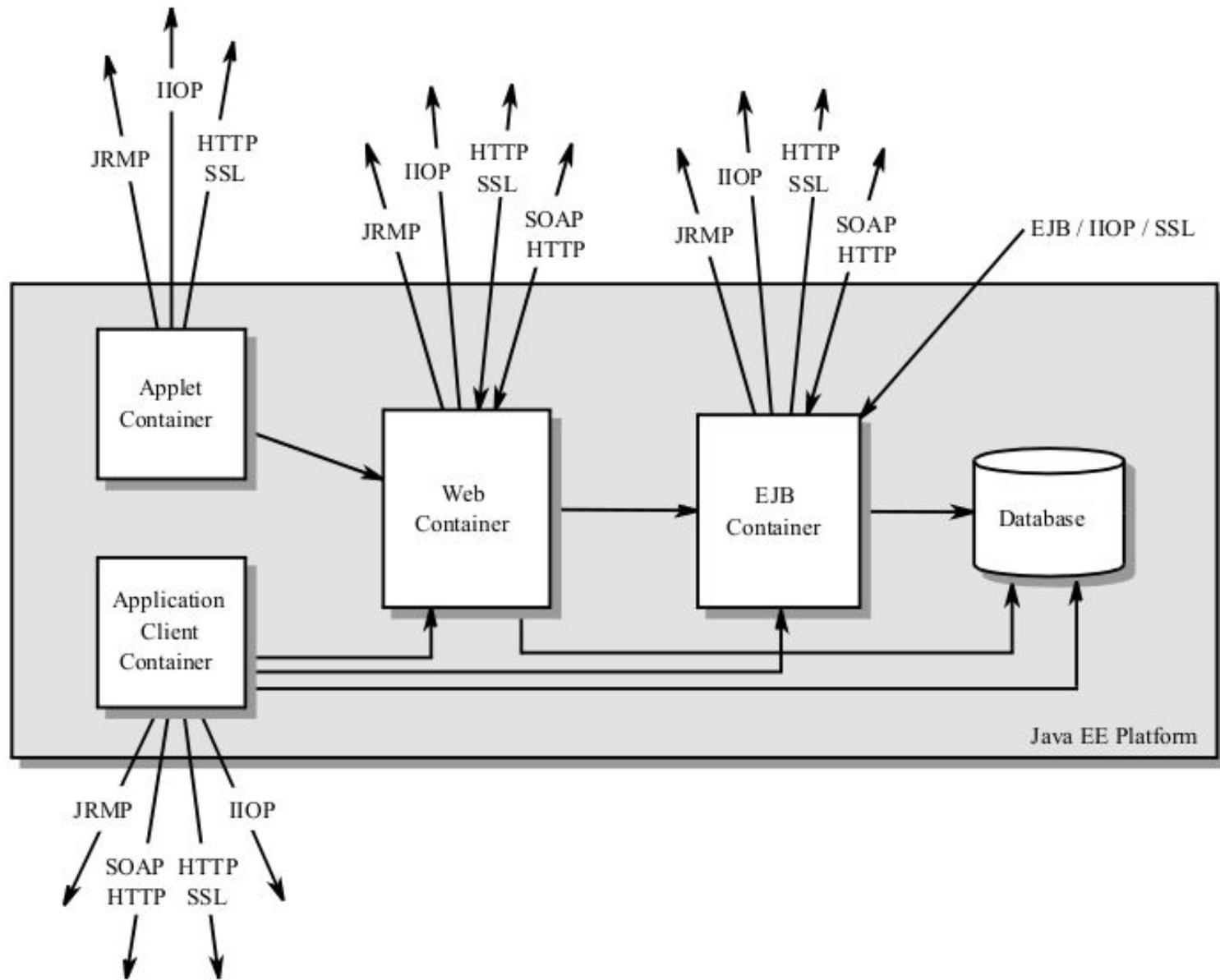




Fonte: Oracle Java EE Tutorial

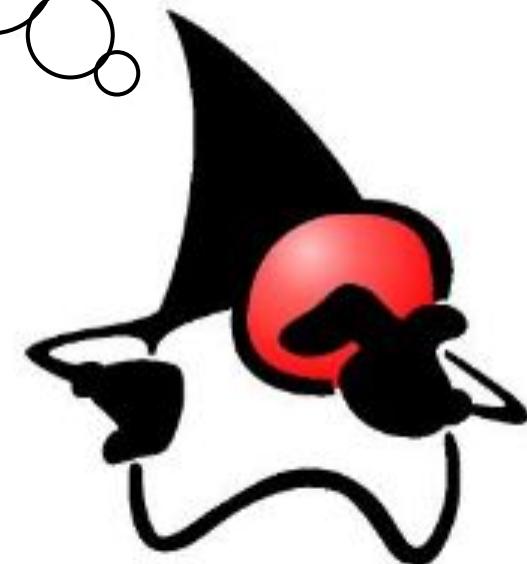
Serviços

- HTTP \ HTTPS
- Java Transaction API (JTA)
- RMI / IIOP
- JDBC API
- Java Persistence API (JPA)
- Java Message Service (JMS)
- Java Naming and Directory Int. (JNDI)
- Java Authn. and Authz Serv. (JAAS)
- Web Services
- XML Processing



Fonte: Java EE Platform Spec

**Quem junta essa “salada”
de padrões e serviços
mesmo?**



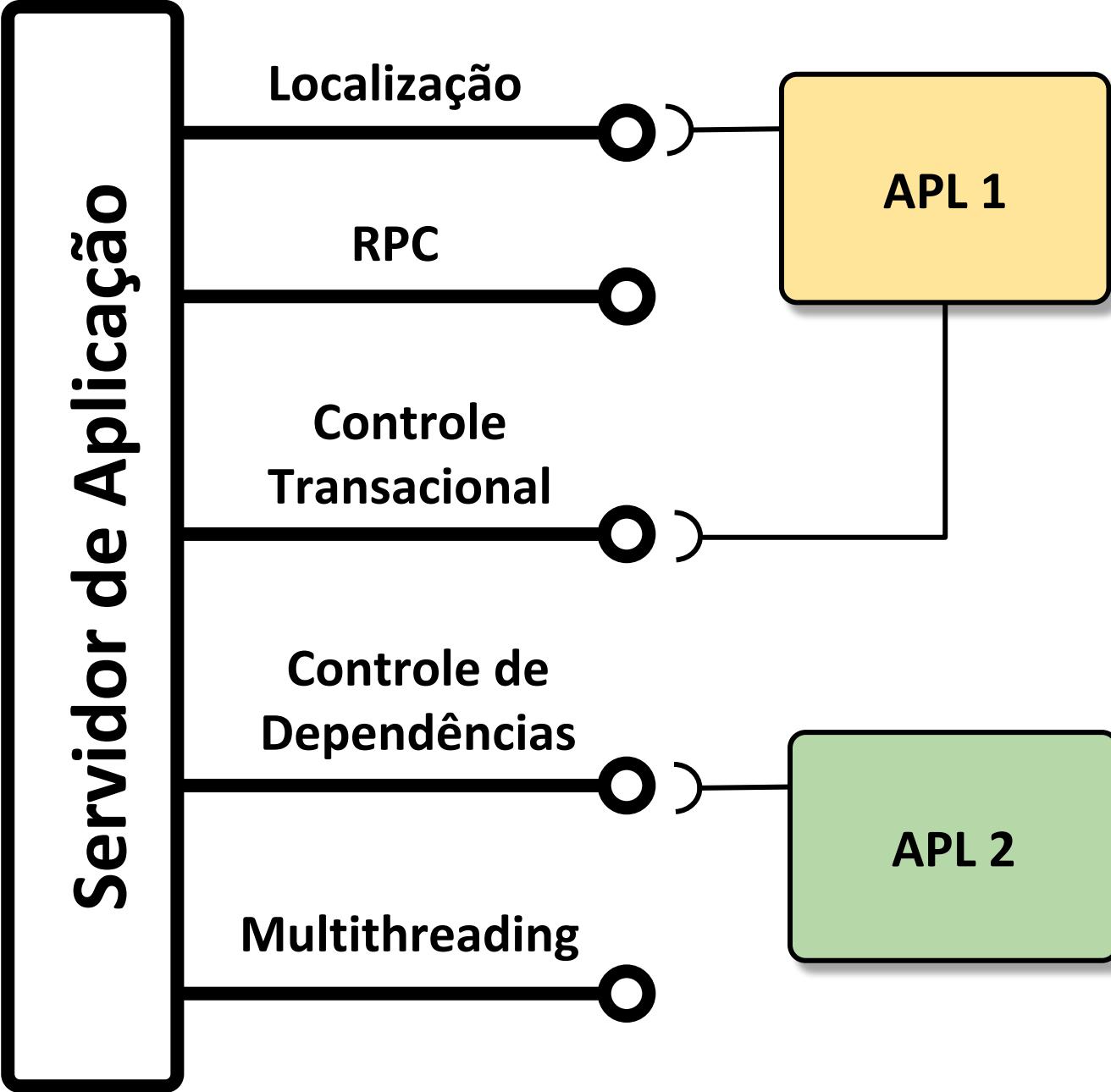
Servidor de Aplicação

Oferece ambiente de execução propício para aplicações

Serviços básicos (chamada remota, localização, etc)

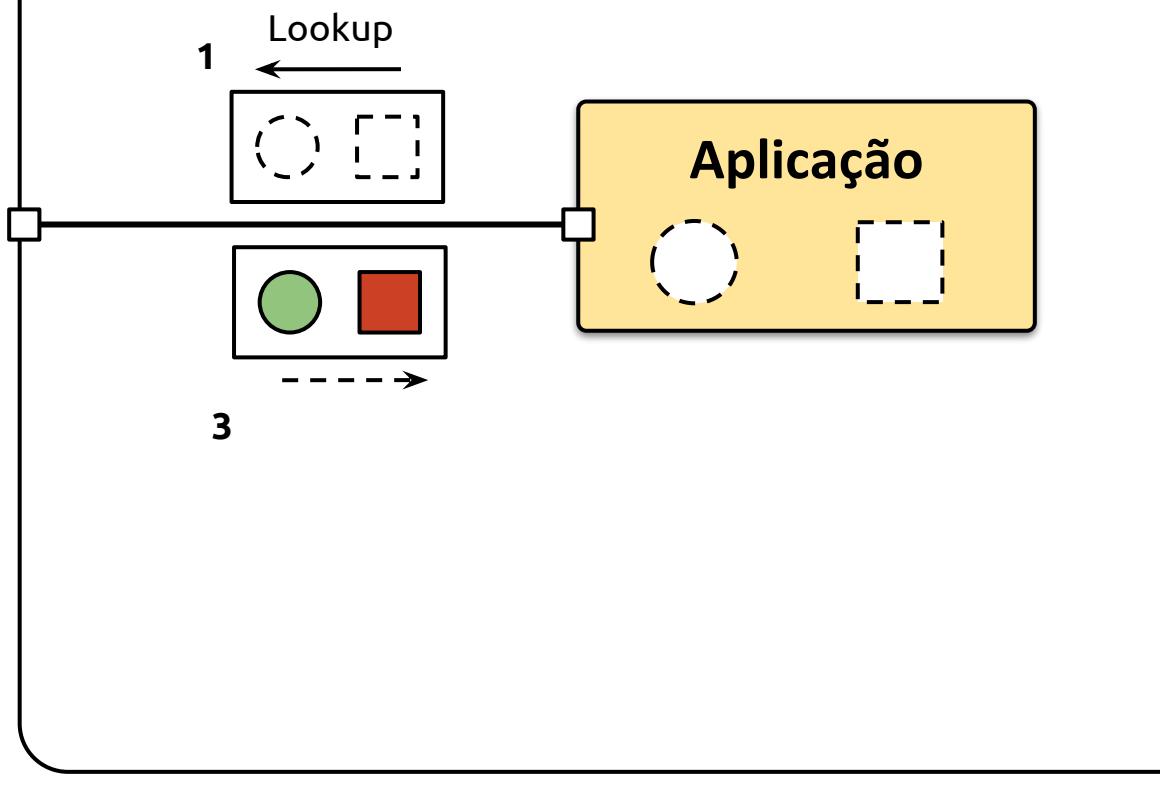
Aplicação foca no negócio e não na infraestrutura

Mais de 20 implementações da especificação

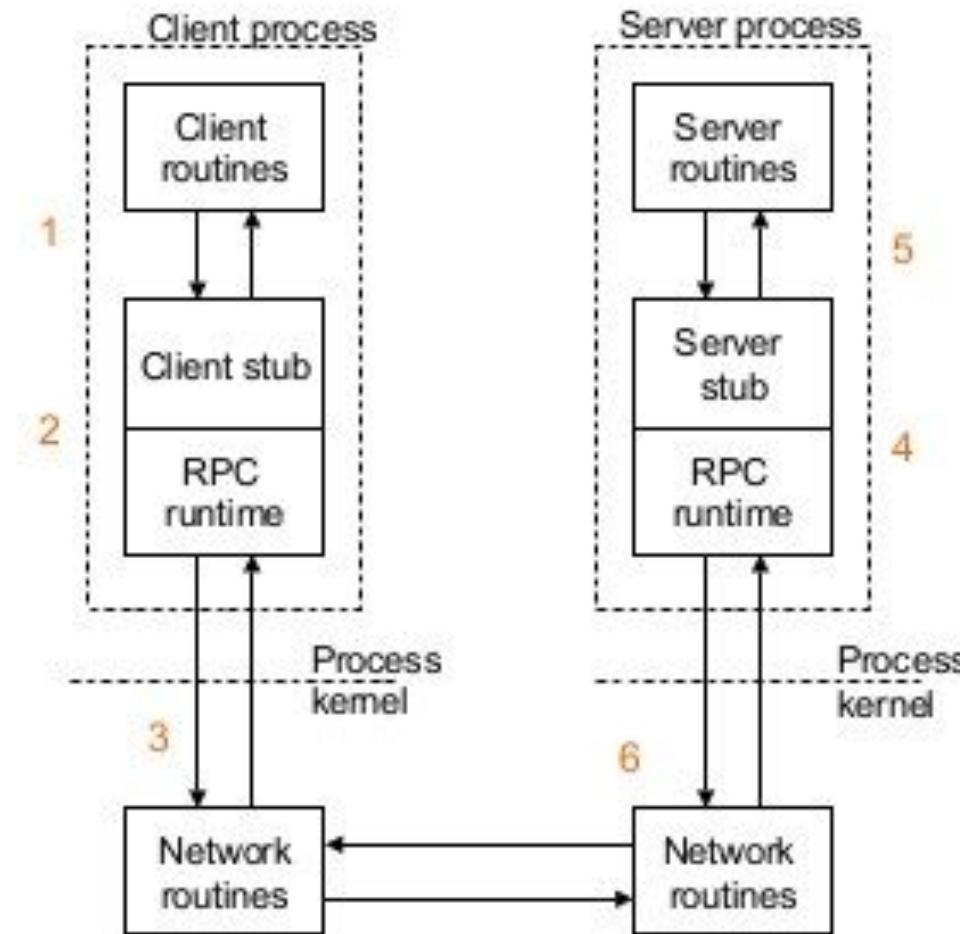


Localização

Servidor de Aplicação

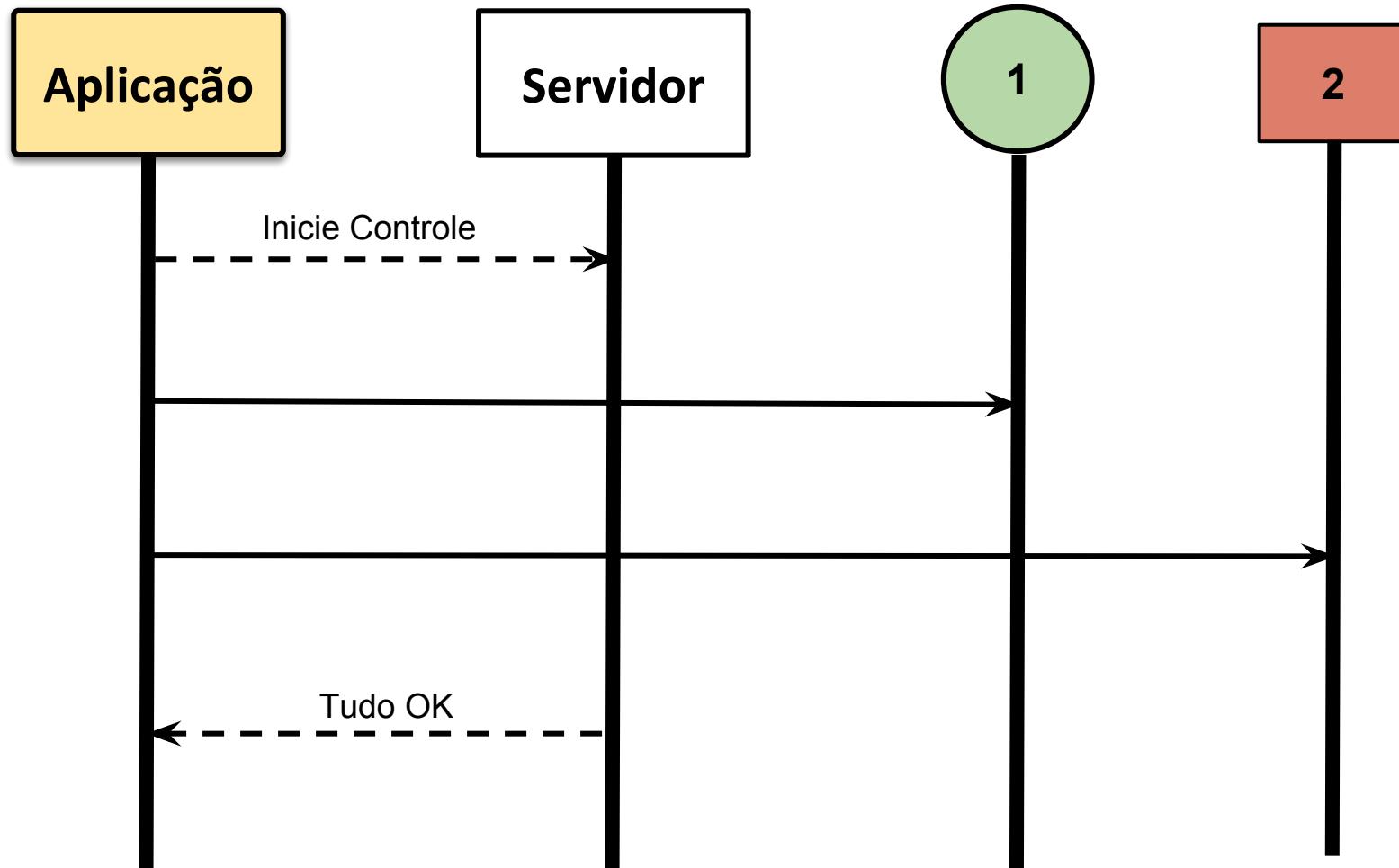


Remote Procedure Call

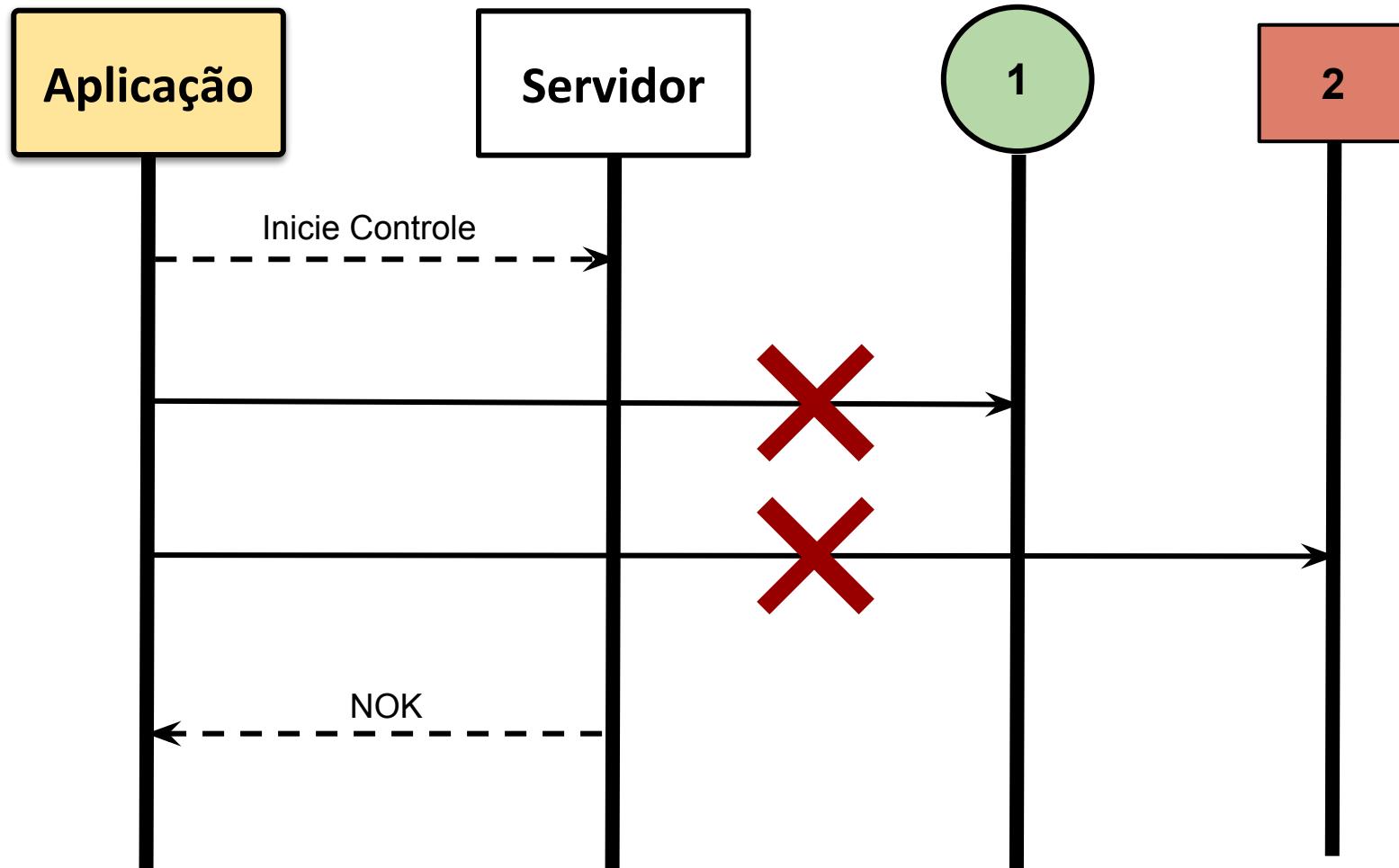


Fonte: Unix Network Programming (Stevens)

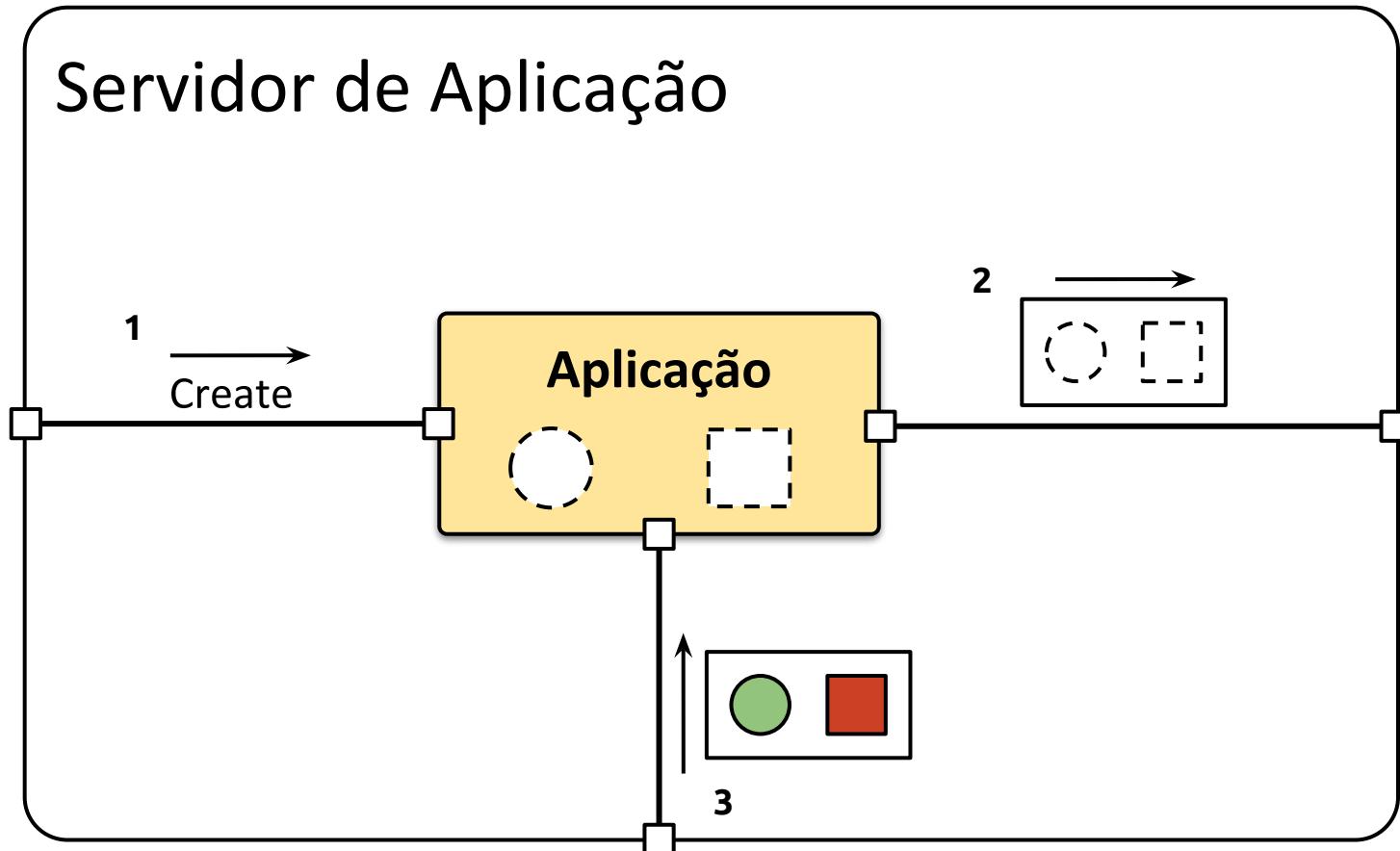
Controle Transacional



Controle Transacional

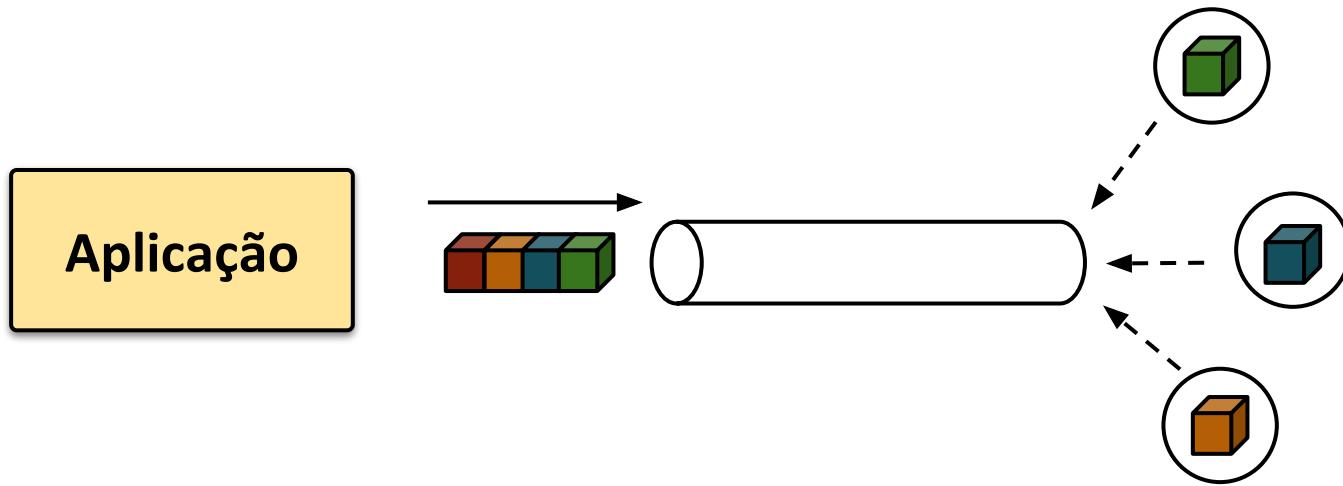


Controle de Dependência



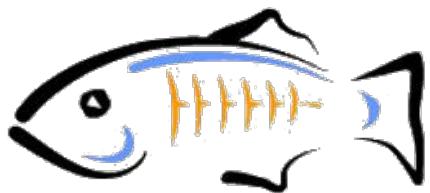
Multithreading

Servidor de Aplicação



Servidores de Aplicação

GlassFish



IBM

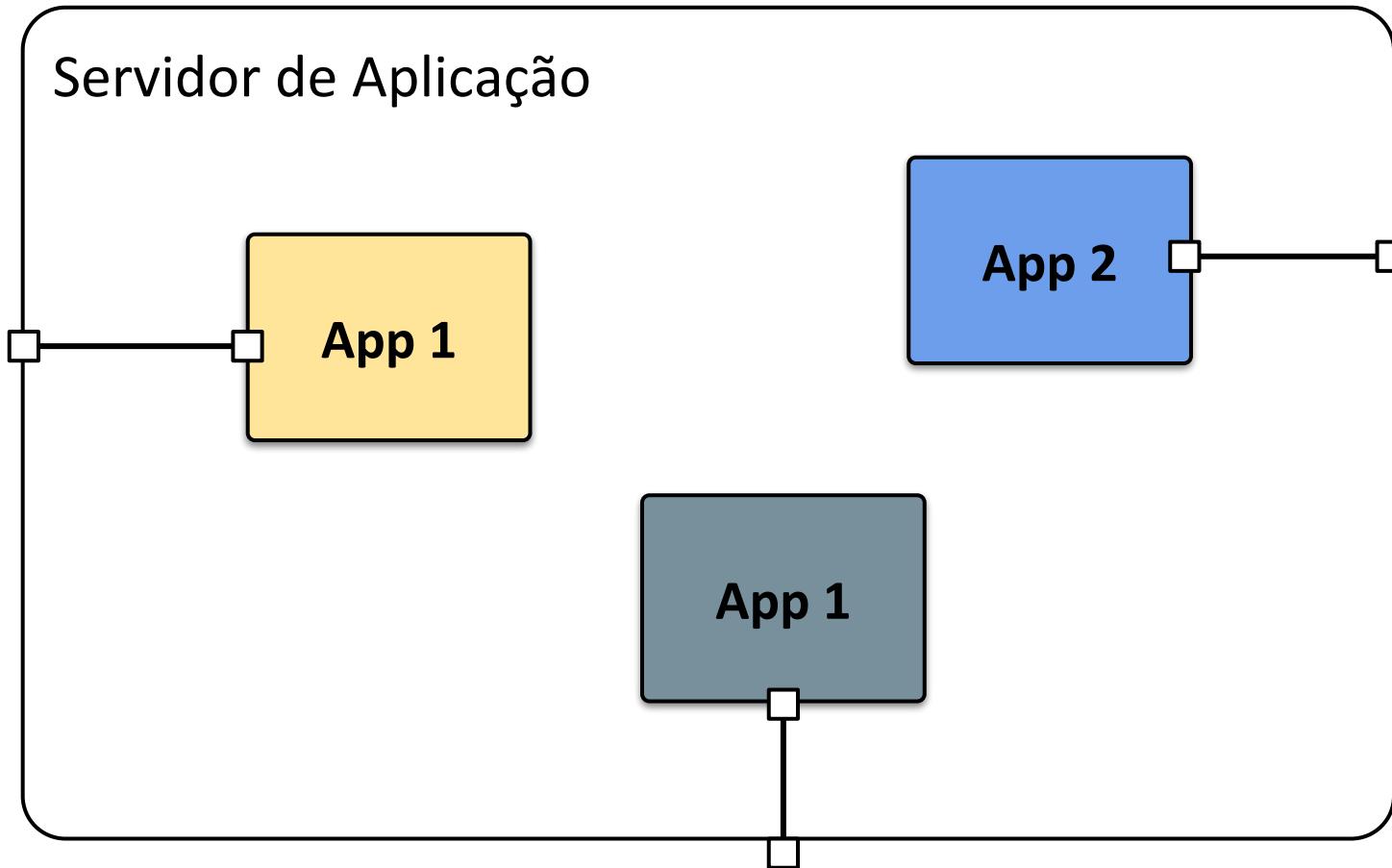
WebSphere



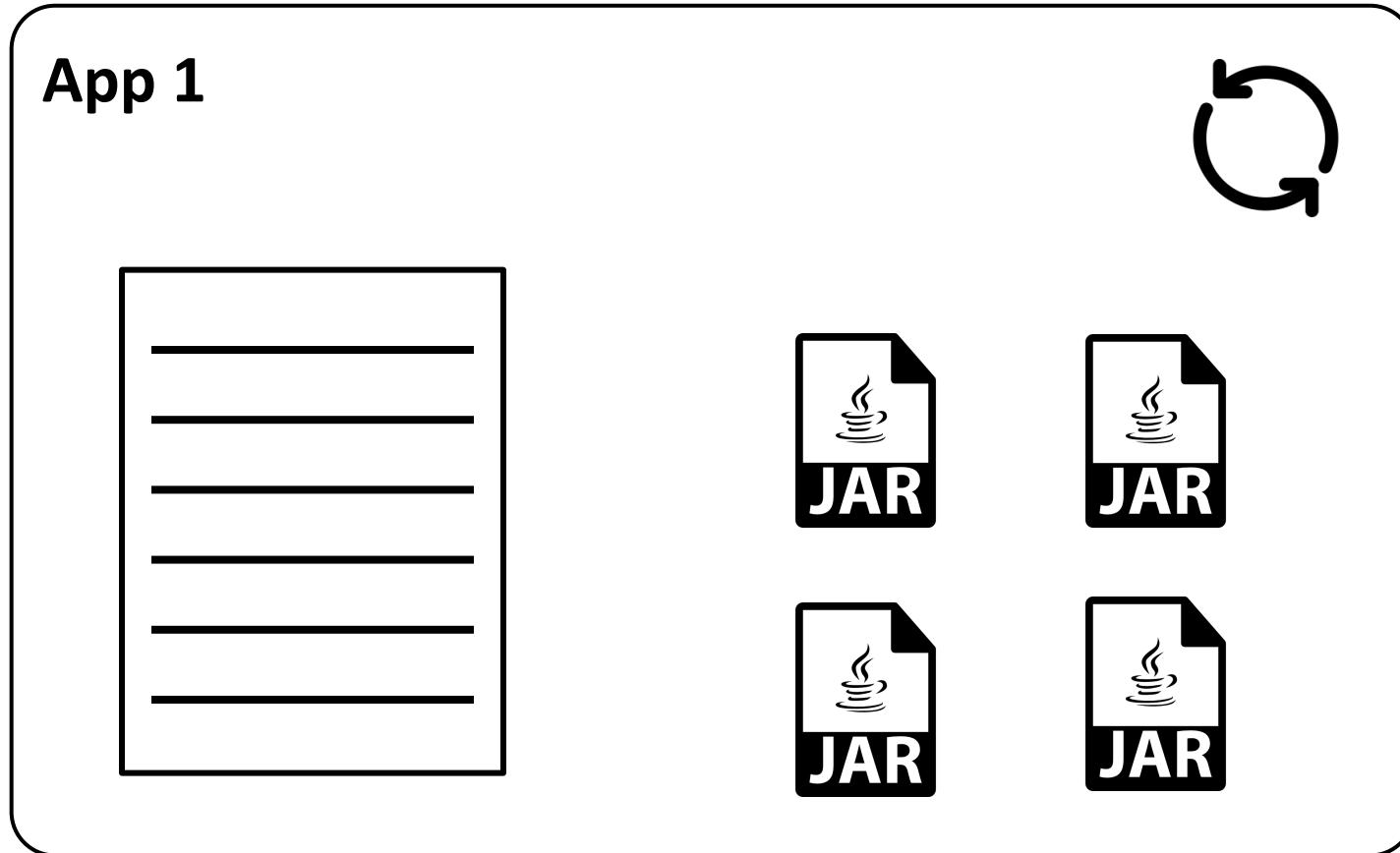
Apache
TomEE



Standalone



Embedded

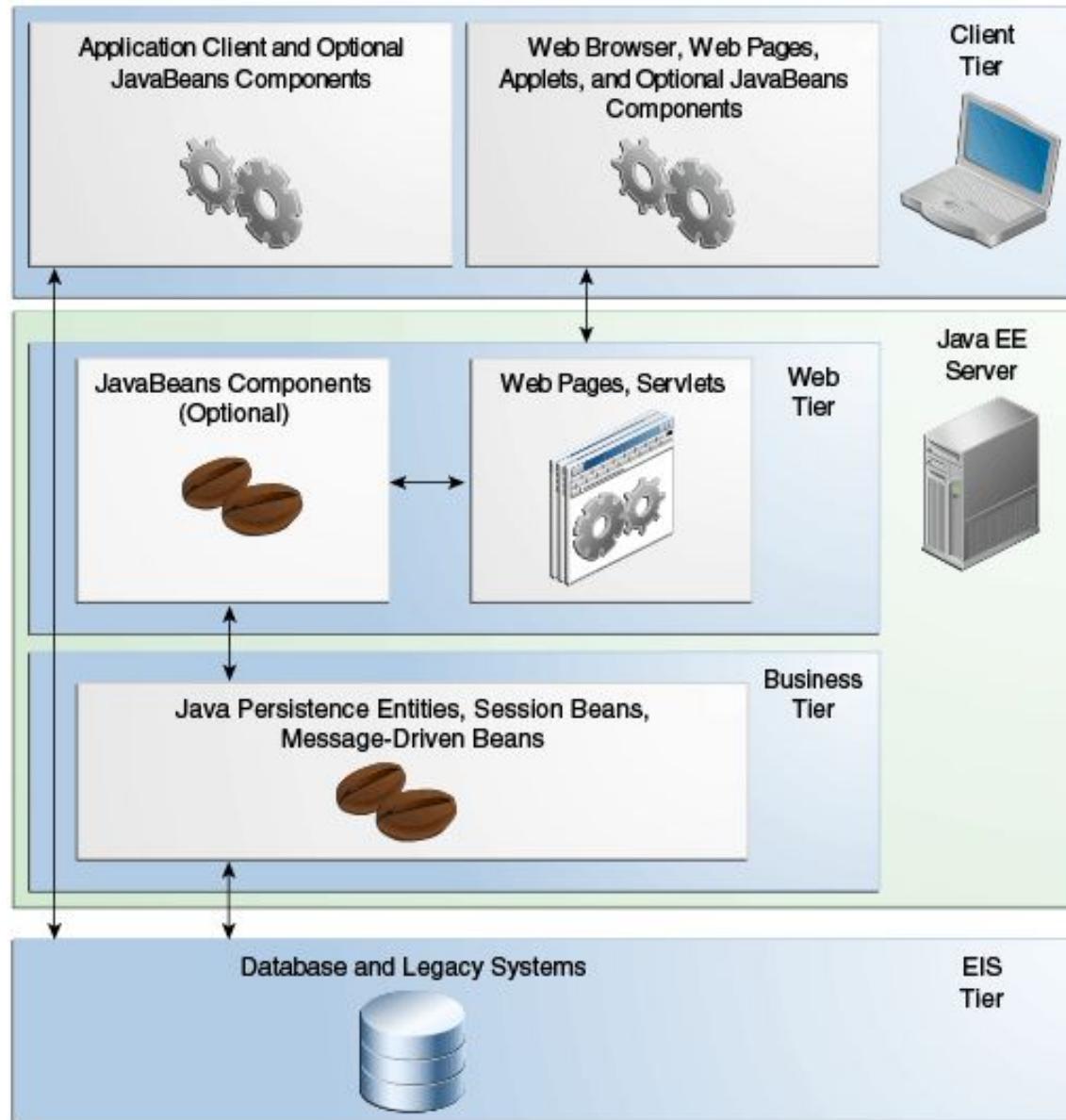


Aplicação Java EE

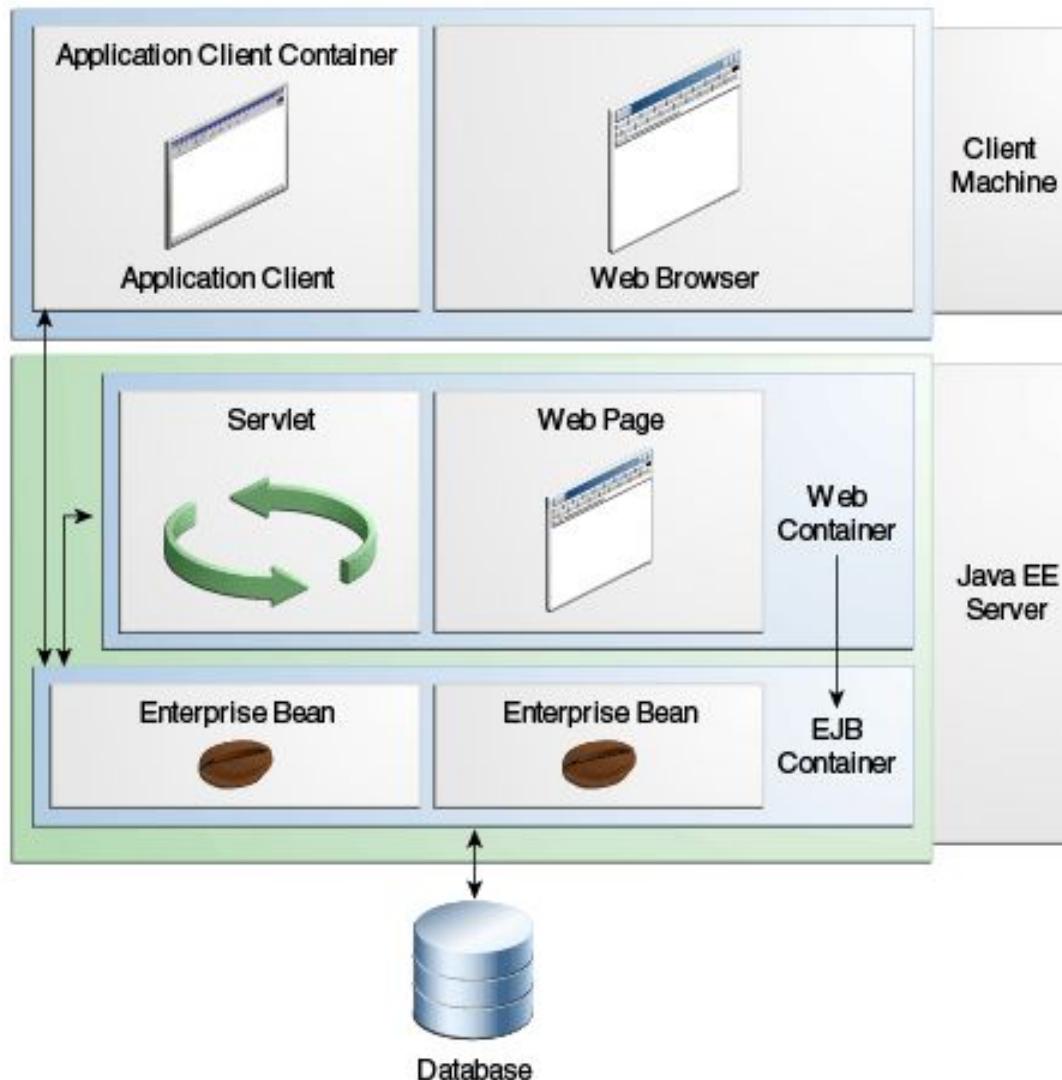
Aplicação distribuída, N-Camadas que segue as restrições impostas pela plataforma e seu respectivo ambiente (servidor de aplicação) para alcançar os requisitos não-funcionais necessários para seu funcionamento

Aplicação Java EE

Aplicação **distribuída**, **N-Camadas** que segue as **restrições impostas pela plataforma** e seu respectivo ambiente (servidor de aplicação) para alcançar os requisitos não-funcionais necessários para seu funcionamento

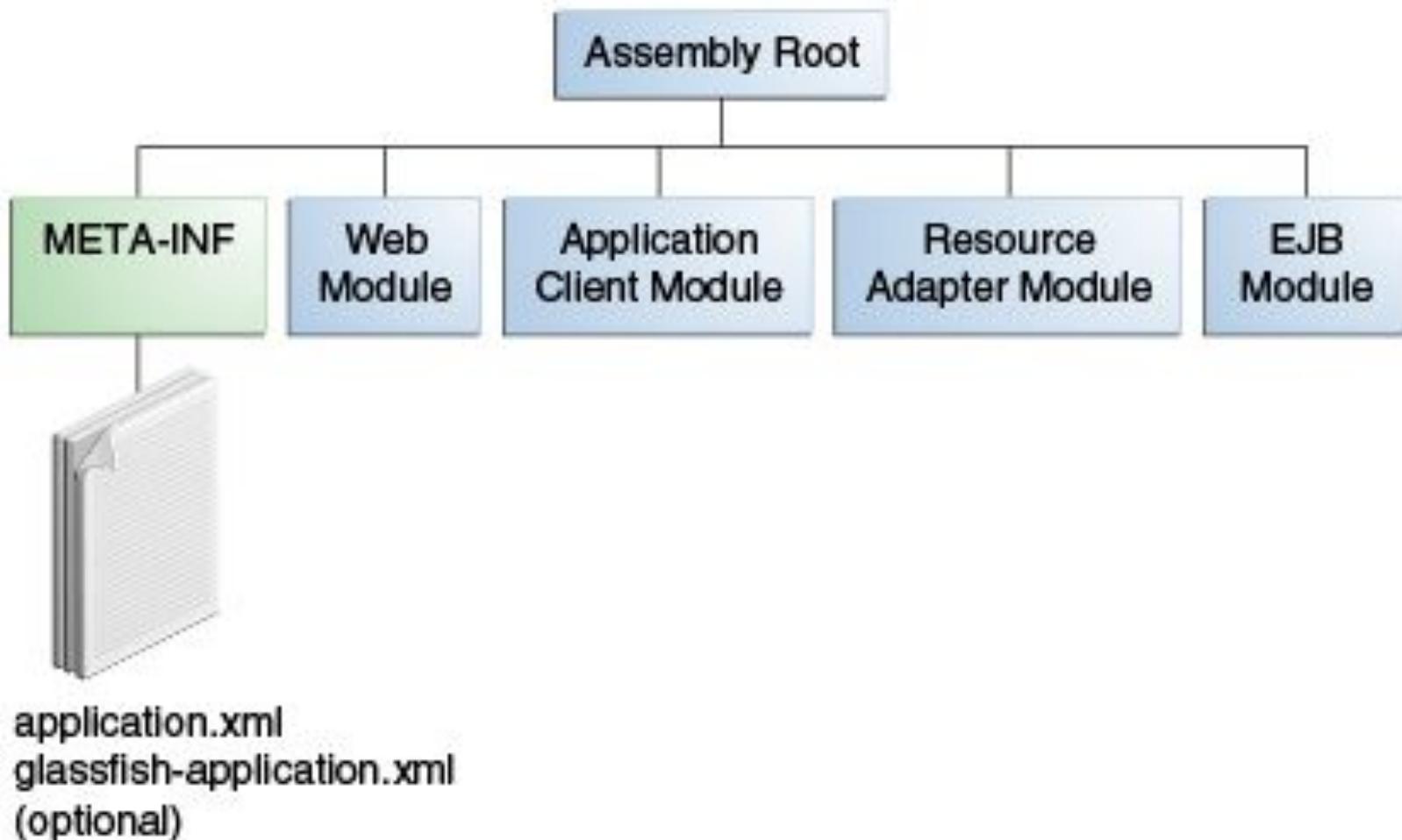


Fonte: Oracle Java EE Tutorial

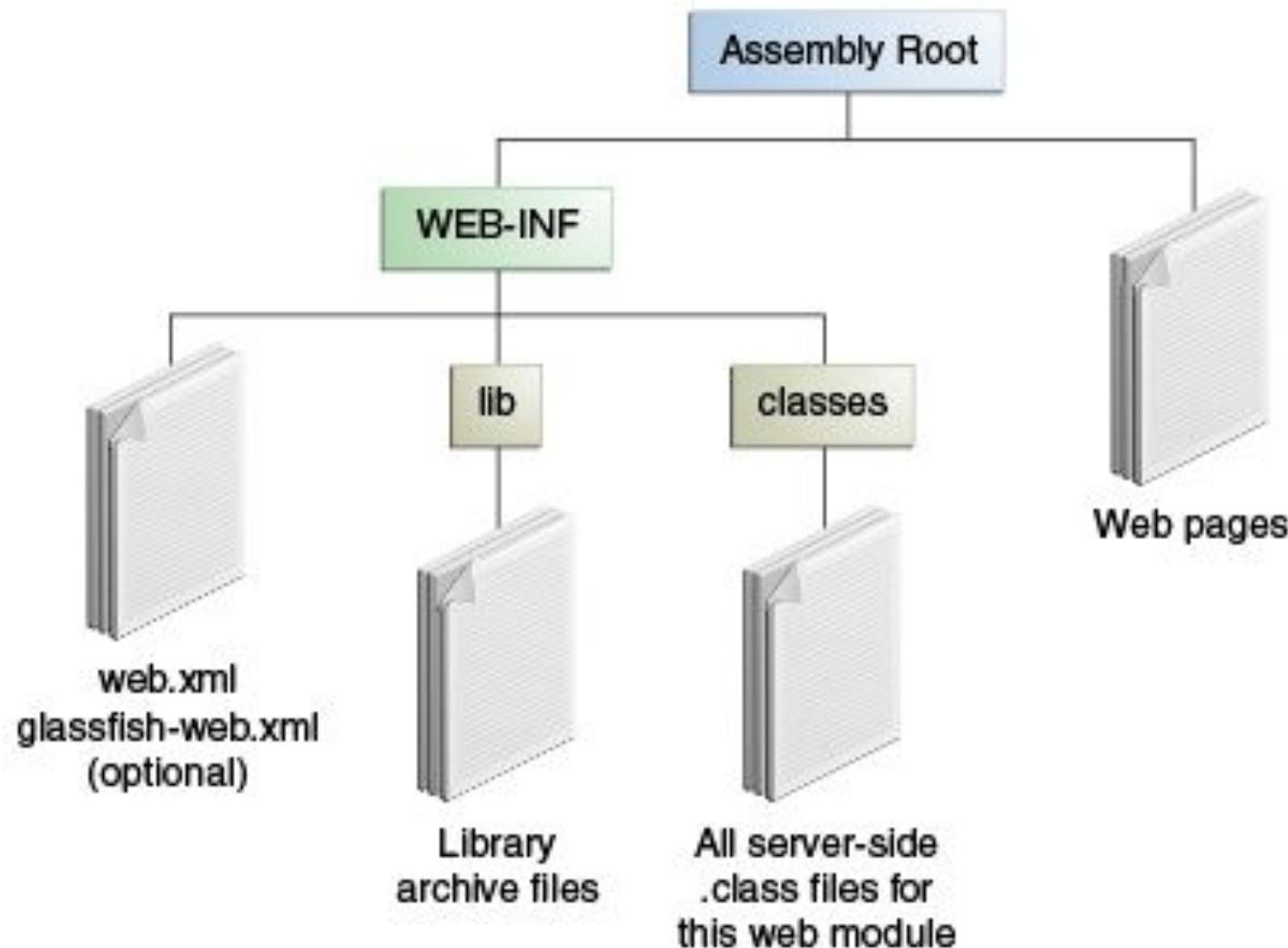


Fonte: Oracle Java EE Tutorial

Empacotamento

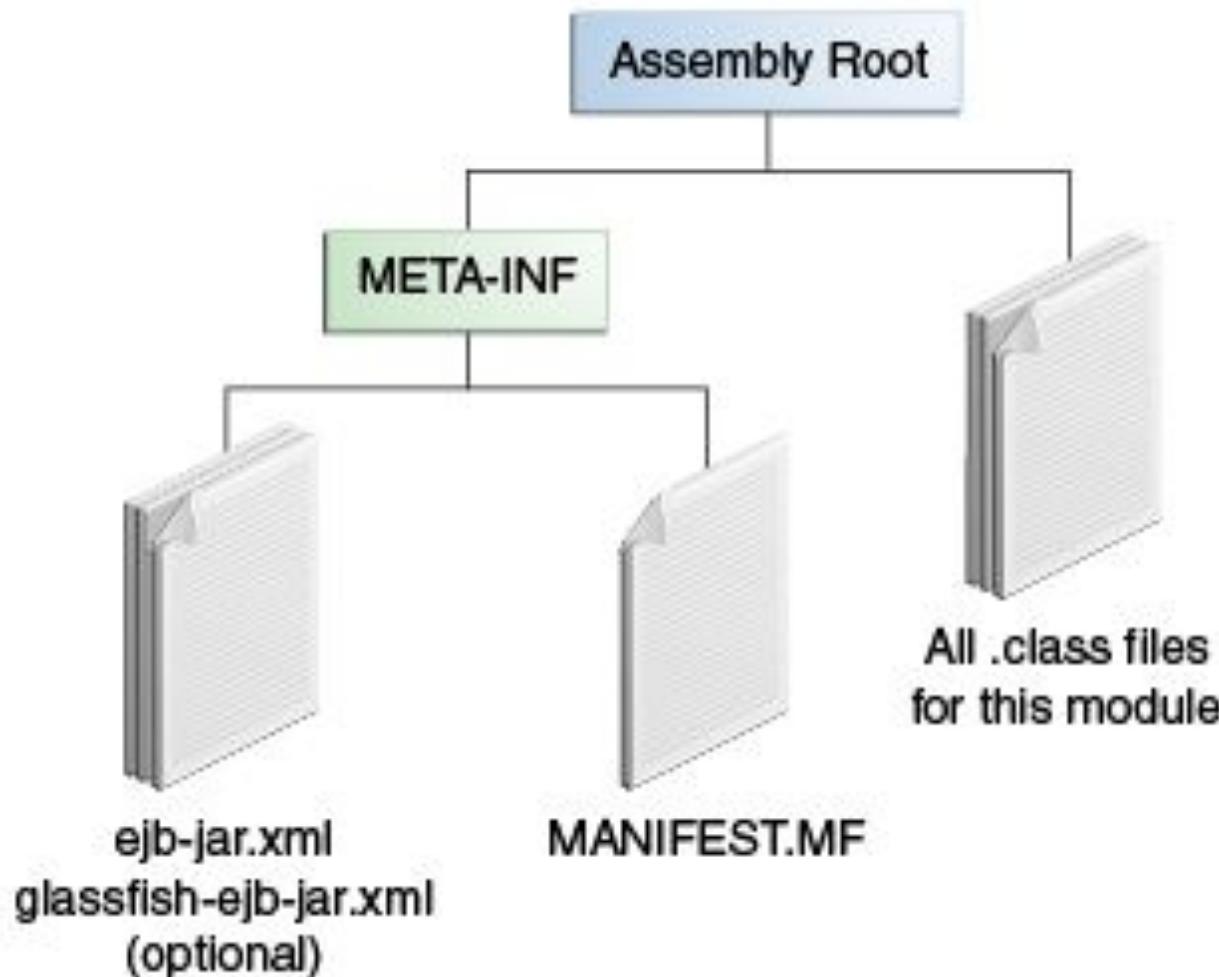


Empacotamento



Fonte: Oracle Java EE Tutorial

Empacotamento



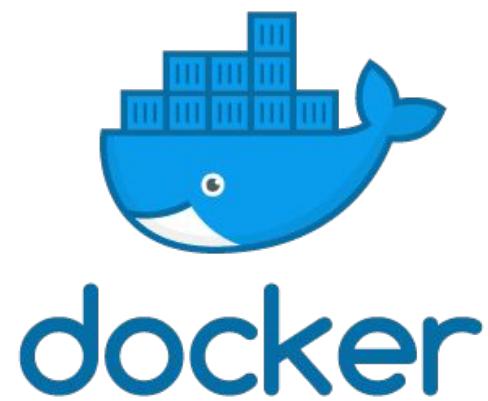
Papéis

- Component Provider
- Application Assembler
- Deployer
- System Administrator

**Sim... Na prática, como
implementamos uma
Aplicação Java EE
mesmo?**



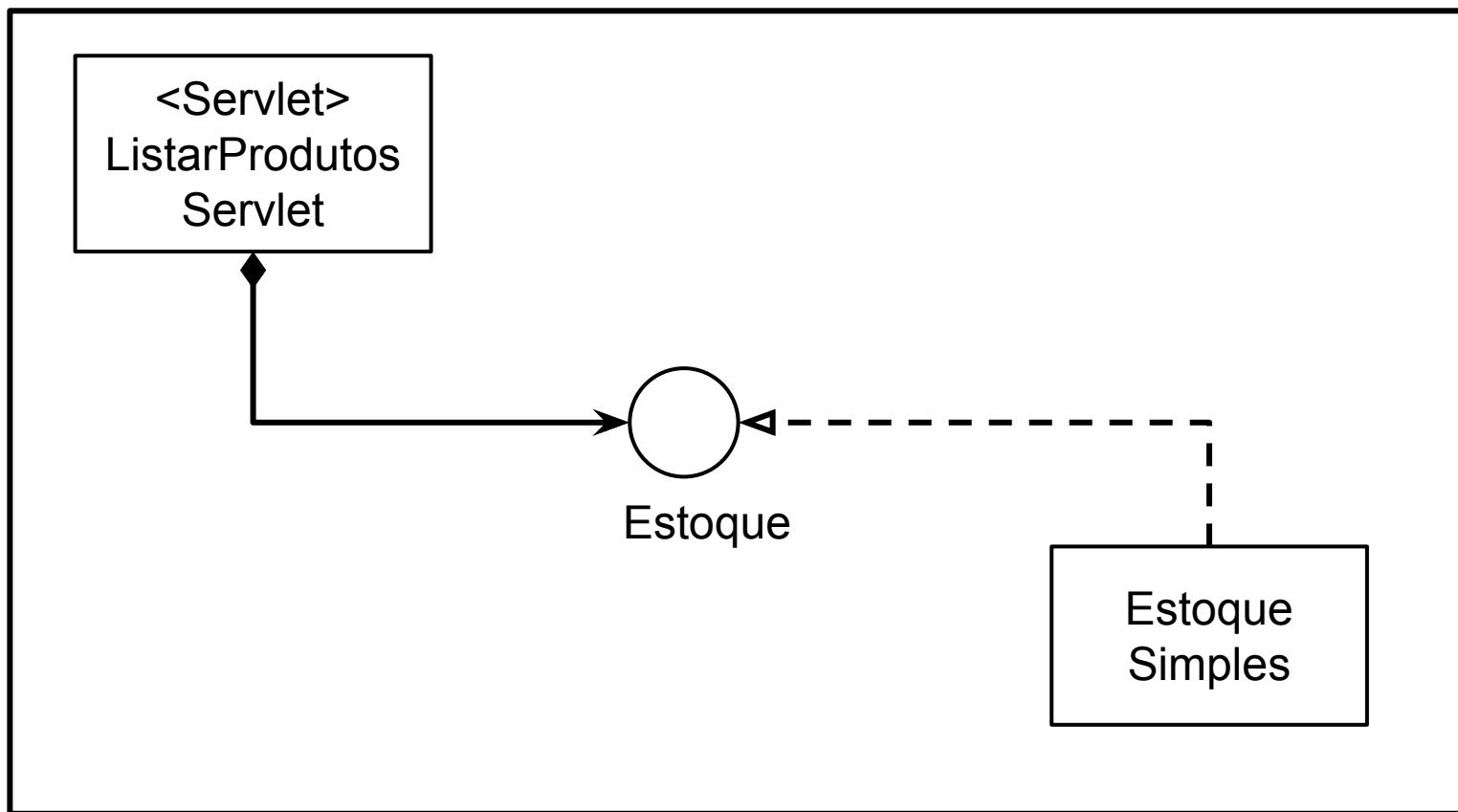
Nós precisaremos de...



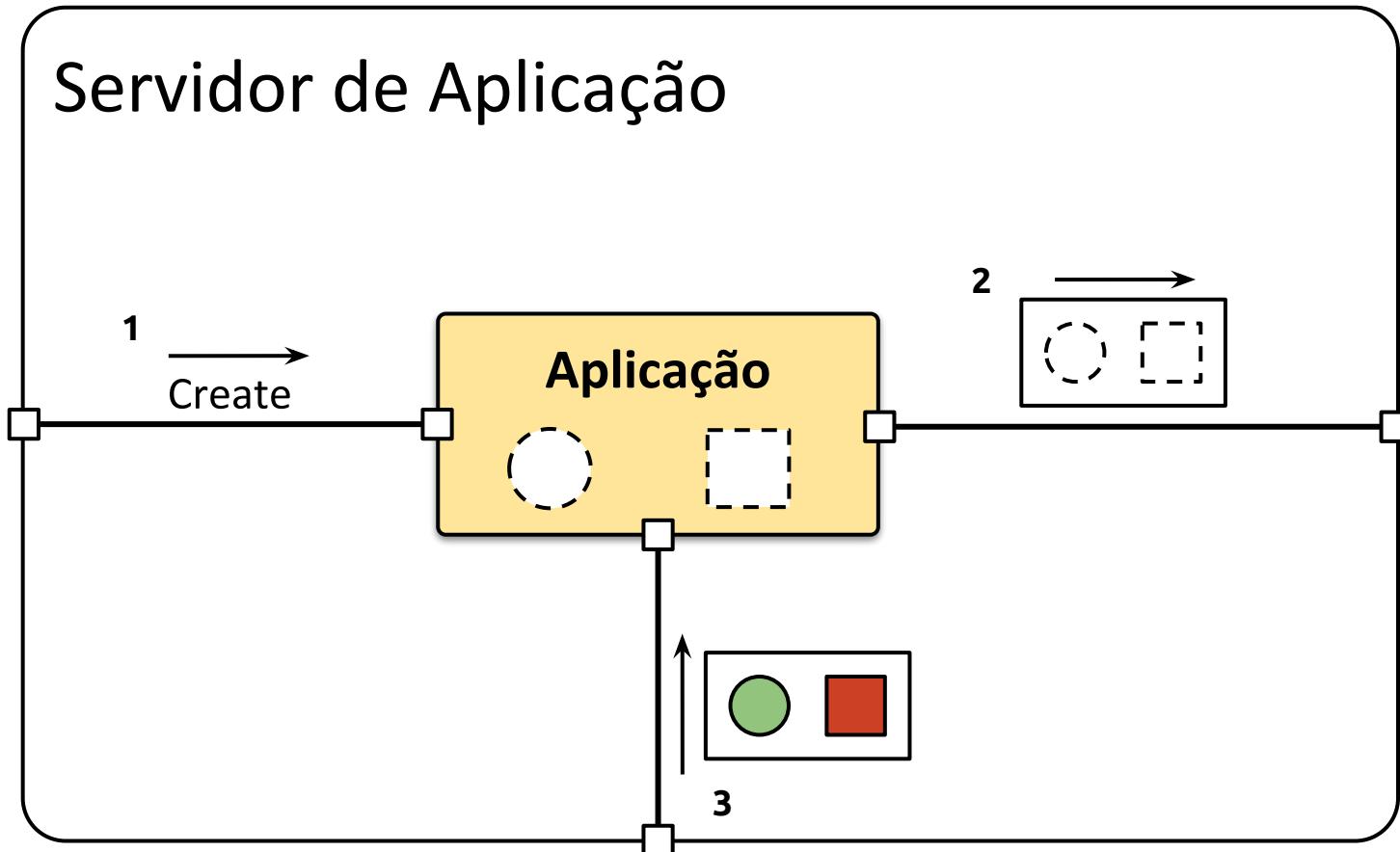
Exercício 01



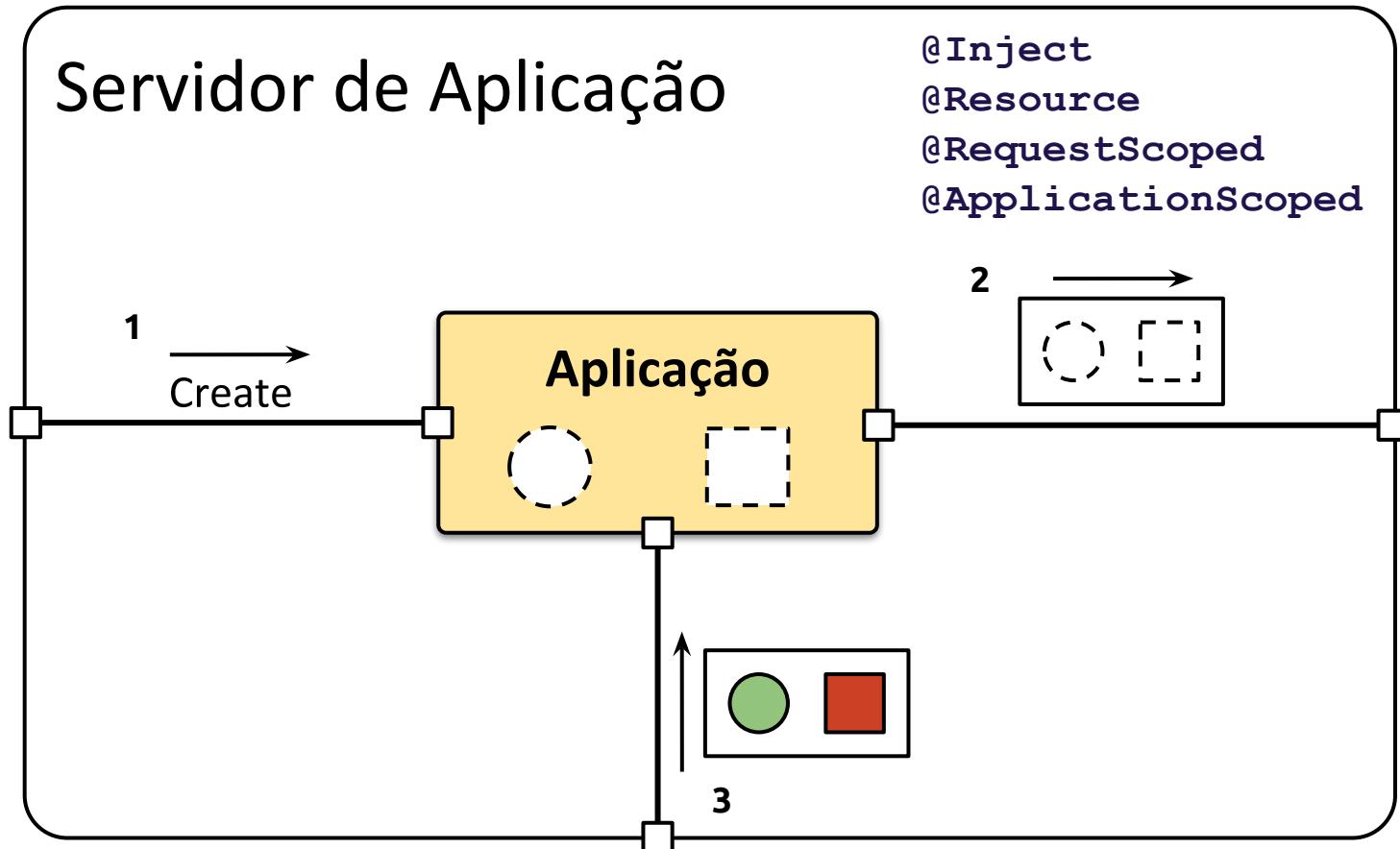
Construir uma aplicação Java EE (apenas na camada Web) que lista os produtos disponíveis em uma loja online. Utilizaremos os recursos do CDI (*Context and Dependency Injection*) para gerenciar os objetos de negócio da nossa aplicação.



Controle de Dependências



Controle de Dependências



Contexts and Dependency Injection

- JSR347 - Contexts and Dependency Injection for the Java EE platform
- Permite a gestão do ciclo de vida de beans e suas dependências
- Tipagem segura, notificações, etc
- **Um novo modelo de desenvolvimento!!**

Contexts and Dependency Injection

Beans

Classes Java com estado e contexto gerenciadas pelo container CDI

Construtor Padrão vs Construtor com parâmetros

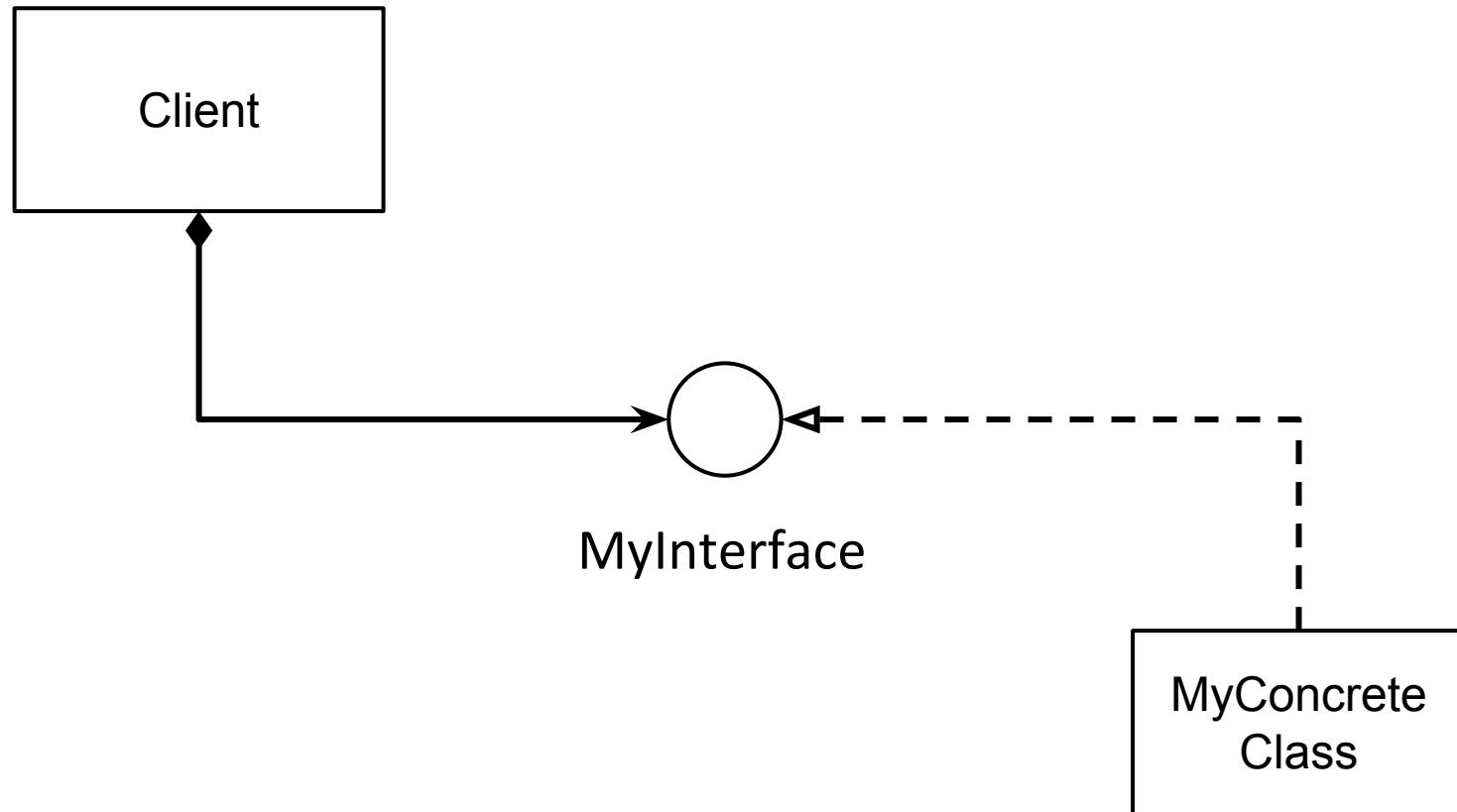
Como eu declaro minhas dependências?

@Inject

Como declaro meu contexto \ escopo?

@RequestScoped, @SessionScoped, @ApplicationScoped

Na prática...



Normalmente...

```
1 public class Client {  
2  
3     MyInterface myObject;  
4  
5     public Client() {  
6         myObject = new MyConcreteObject();  
7     }  
8  
9     public void doSomething() {  
10        myObject.processIt();  
11        // another stuff  
12    }  
13 }
```

Fábricas

```
1 public class ObjectFactory {  
2  
3     private static Map<String, MyInterface> objPool;  
4  
5     public static MyInterface createConcreteObj() {  
6         return objPool.get("concrete");  
7     }  
8 }
```

```
1 public class SmartClient {  
2  
3     MyInterface myObject;  
4  
5     public SmartClient() {  
6         myObject = ObjectFactory.createConcreteObj();  
7     }  
8  
9     public void doSomething() {  
10        myObject.processIt();  
11    }  
12 }
```

CDI

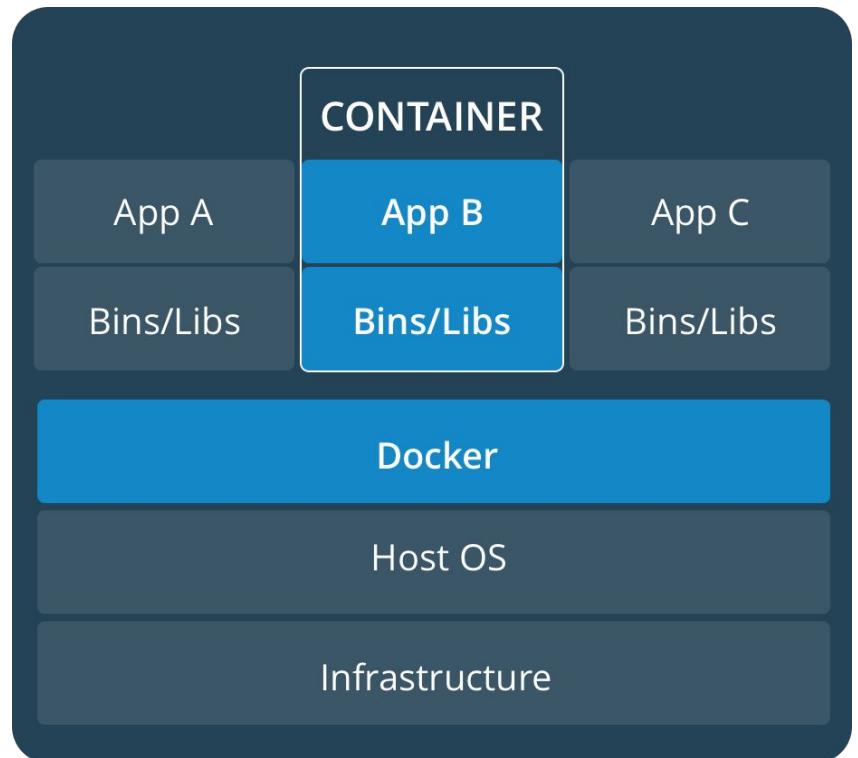
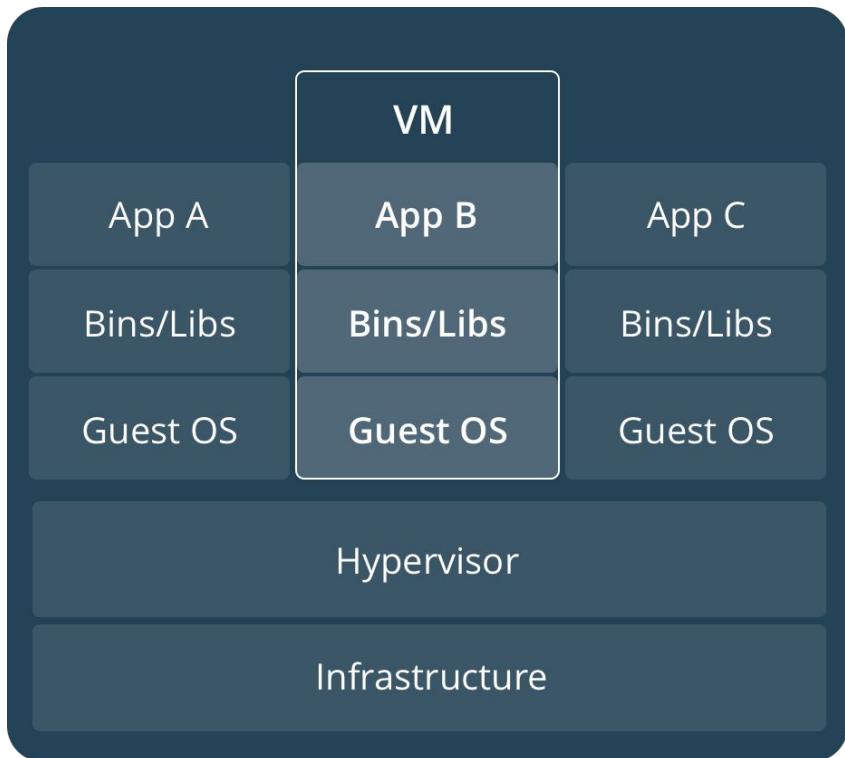
```
1 package myPackage;  
2  
3 public class CDIClient {  
4  
5     @Inject  
6     MyInterface myObject;  
7  
8     public void doSomething() {  
9         myObject.processIt();  
10        // another stuff  
11    }  
12 }
```

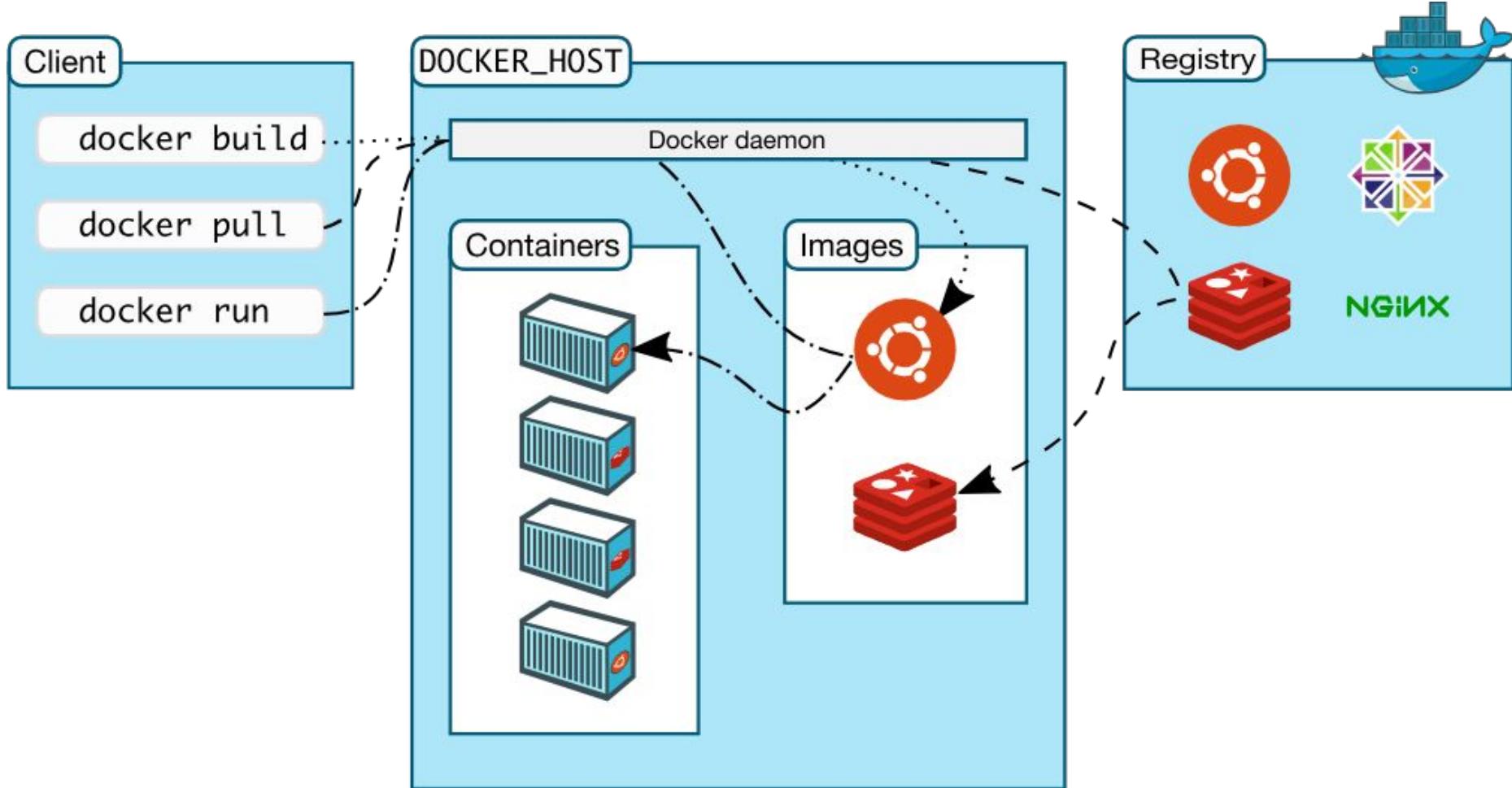
Docker

Plataforma para desenvolver, implantar e executar aplicações em containers

Isola processos, sistema de arquivos, rede, etc

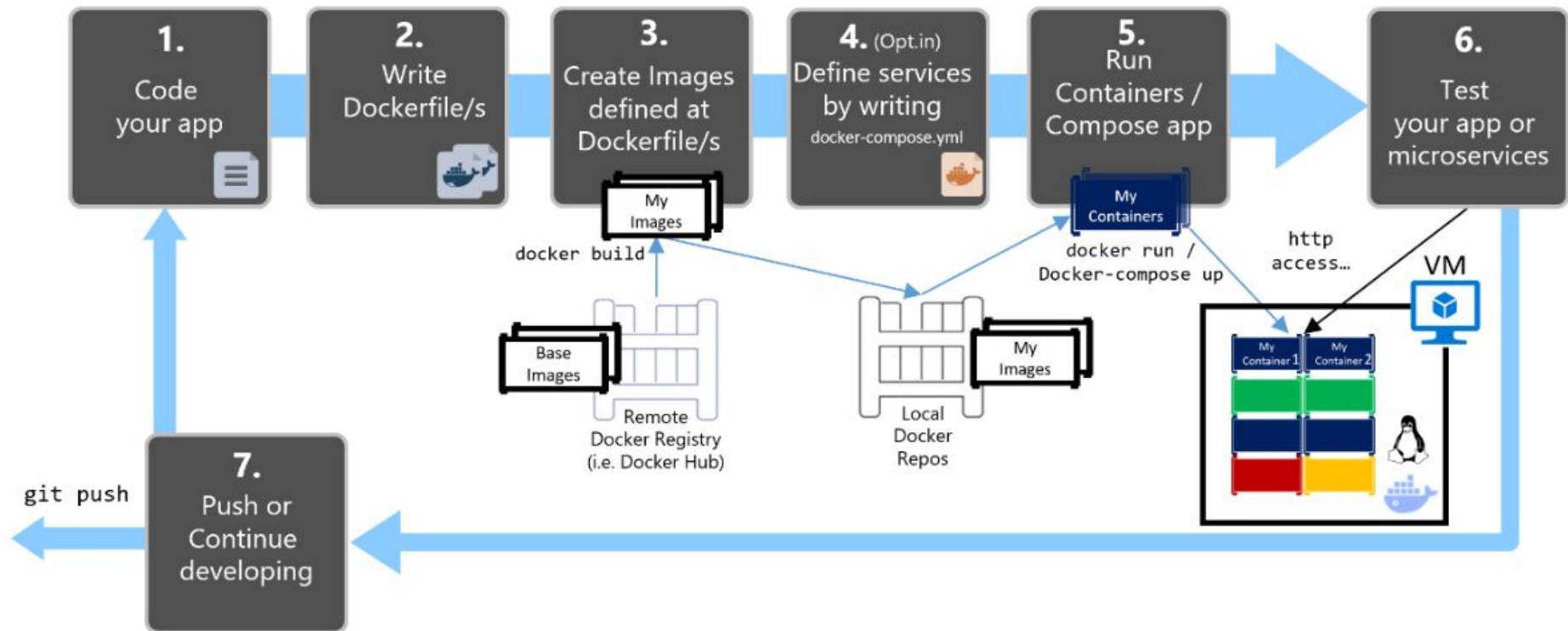
Docker



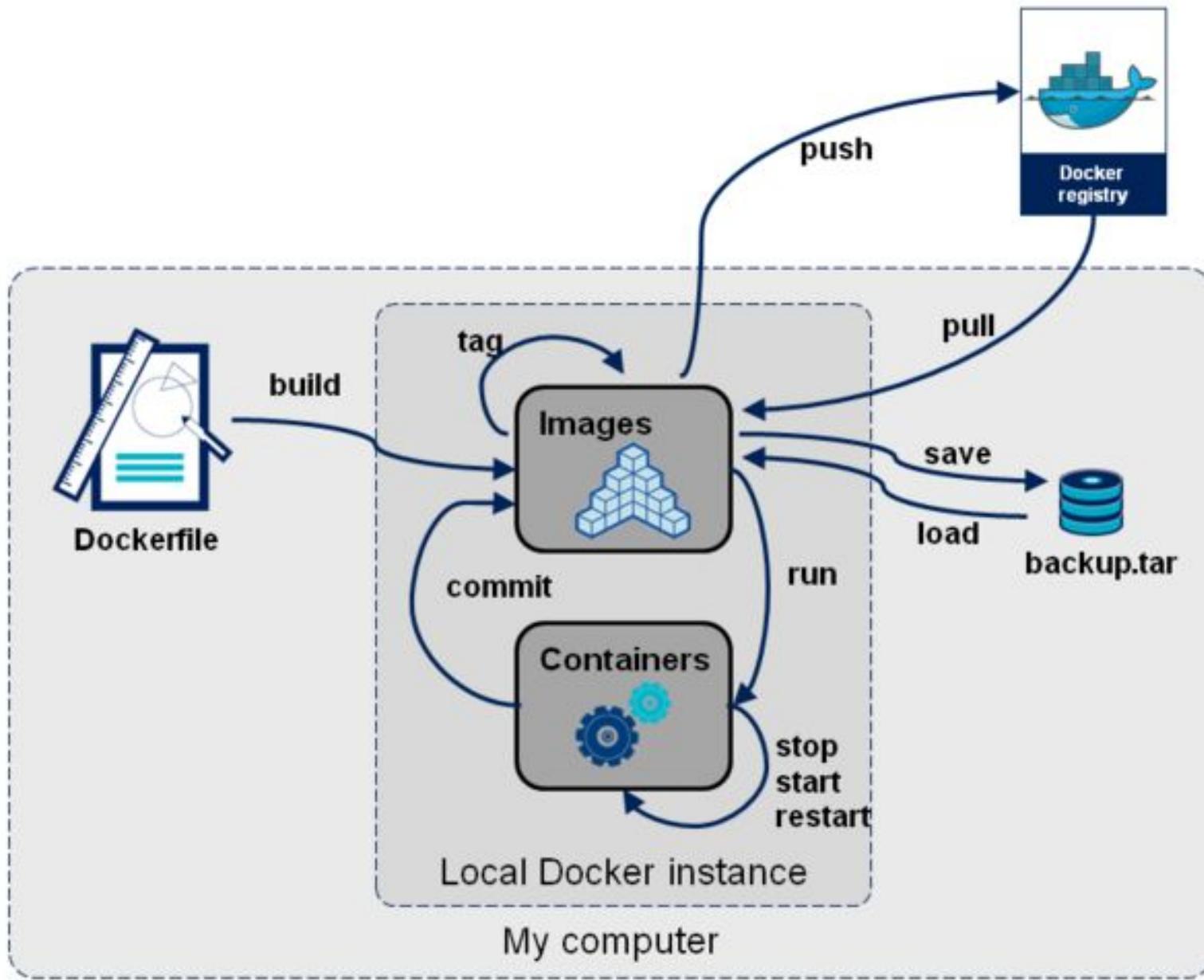


Fonte: Docker overview em docs.docker.com

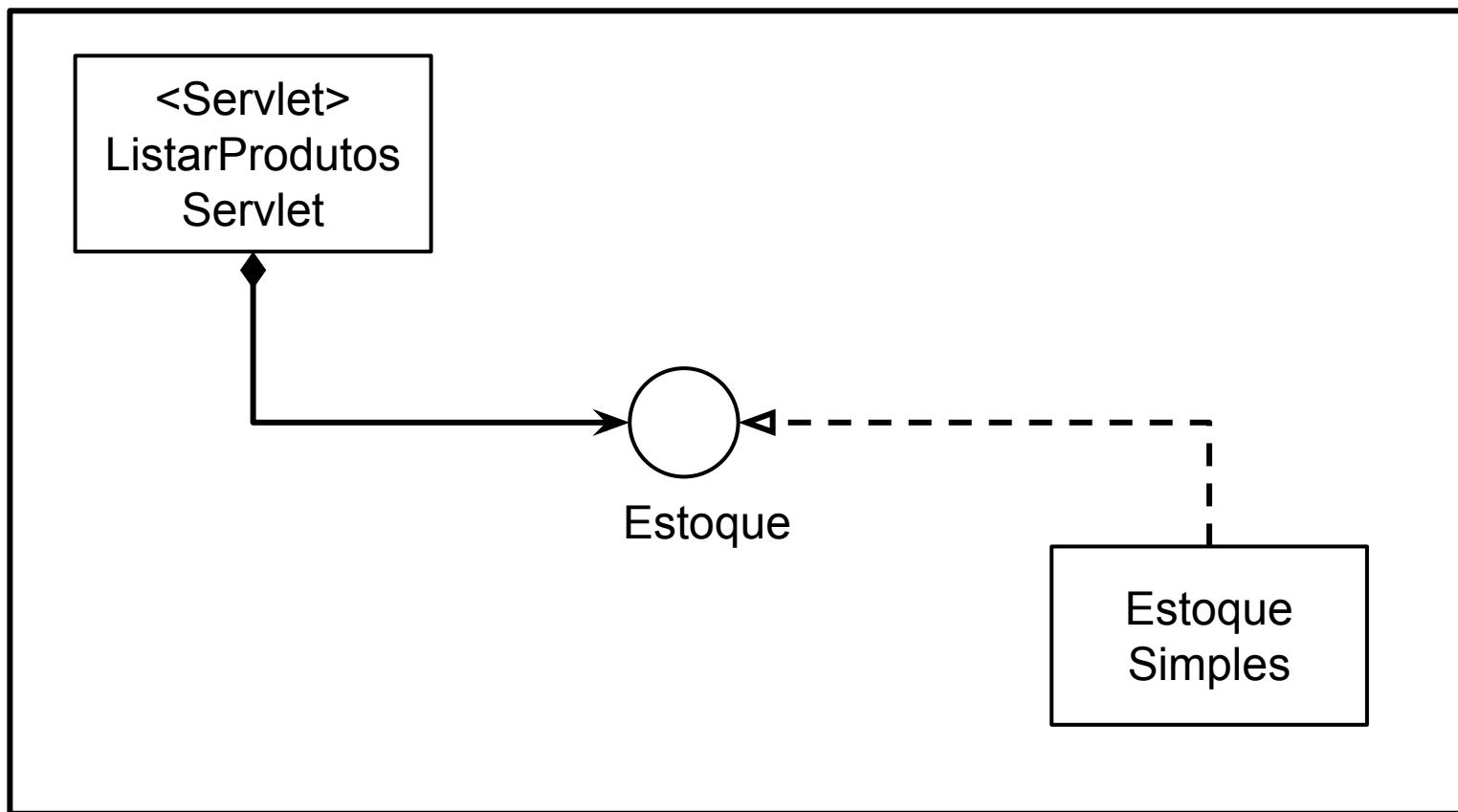
Inner-Loop development workflow for Docker apps



Fonte: MS Development workflow for Docker apps



Construir uma aplicação Java EE (apenas na camada Web) que lista os produtos disponíveis em uma loja online. Utilizaremos os recursos do CDI (*Context and Dependency Injection*) para gerenciar os objetos de negócio da nossa aplicação.



Crie o projeto Maven

```
$ mvn -B archetype:generate  
-DarchetypeGroupId=org.apache.maven.archetypes  
-DgroupId=uni7 -DartifactId=app-web
```

```
$ mvn -N io.takari:maven:0.7.5:wrapper
```

Plugins

- Maven Wrapper
- maven-war-plugin
- maven-compiler-plugin

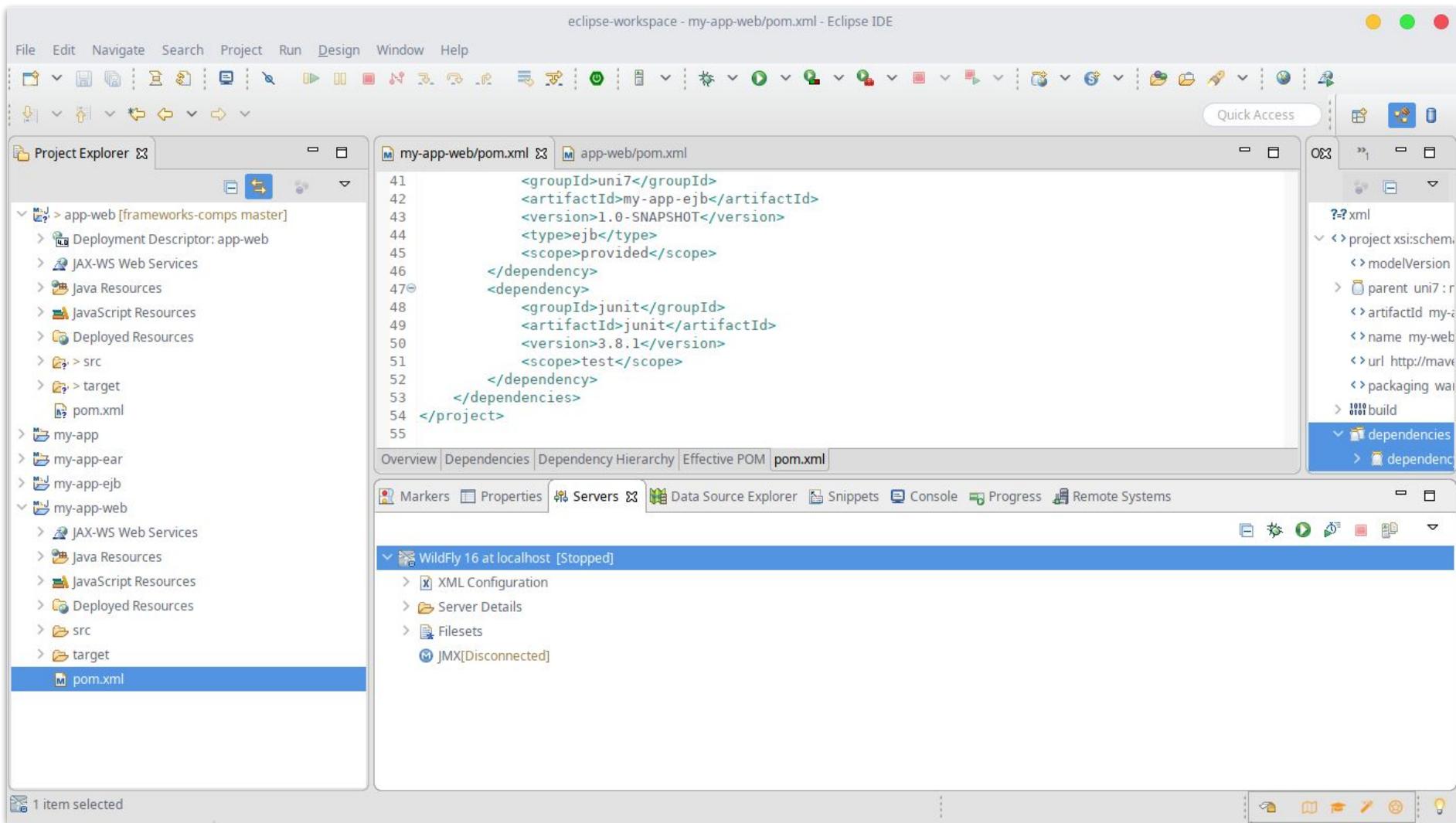
Dependências

- javaee-web-api

```
1 <project ...>
2     <modelVersion>4.0.0</modelVersion>
3     <groupId>uni7</groupId>
4     <artifactId>app-web</artifactId>
5     <packaging>war</packaging>
6     <name>app-web</name>
7     <build>
8         <plugins>
9             <plugin>
10                <groupId>org.apache.maven.plugins</groupId>
11                <artifactId>maven-war-plugin</artifactId>
12                <version>3.2.2</version>
13            </plugin>
14            <plugin>
15                <groupId>org.apache.maven.plugins</groupId>
16                <artifactId>maven-compiler-plugin</artifactId>
17                <version>3.8.0</version>
18                <configuration>
19                    <release>11</release>
20                </configuration>
21            </plugin>
22        </plugins>
23    </build>
24    <dependencies>
25        <dependency>
26            <groupId>javax</groupId>
27            <artifactId>javaee-web-api</artifactId>
28            <version>8.0</version>
29            <scope>provided</scope>
30        </dependency>
31    </dependencies>
32 </project>
```

```
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-war-plugin</artifactId>
    <version>3.2.2</version>
</plugin>
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-compiler-plugin</artifactId>
    <version>3.8.0</version>
    <configuration>
        <release>11</release>
    </configuration>
</plugin>
</plugins>
</build>
<dependencies>
    <dependency>
        <groupId>javax</groupId>
        <artifactId>javaee-web-api</artifactId>
        <version>8.0</version>
        <scope>provided</scope>
    </dependency>
</dependencies>
```

Crie um runtime para testes



Crie um Dockerfile

```
1  FROM adoptopenjdk/openjdk11:alpine as build
2  WORKDIR /workspace/
3
4  COPY mvnw .
5  COPY .mvn .mvn
6  COPY pom.xml .
7  COPY src src
8
9  RUN ./mvnw install -DskipTests
10 # RUN mkdir -p target/dependency && (cd target/dependency; jar -xf ../../*.war)
11
12 FROM michelav/wildfly-alpine
13
14 WORKDIR /opt/wildfly/standalone/deployments/app-web/
15
16 ARG DEPENDENCY=/workspace
17
18 COPY --from=build ${DEPENDENCY}/target/app-web-1.0-SNAPSHOT .
19
20 CMD ["/opt/wildfly/bin/standalone.sh", "-b", "0.0.0.0", "-bmanagement", "0.0.0.0"]
```

Finalmente...

```
$ docker build -t myapp .
```

```
$ docker run -p 8080:8080 -p 9990:9990 -it myapp
```

```
$ curl  
http://localhost:8080/app-web-1.0/listarProdutos
```

Referências

Componentes e Frameworks

<https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=5203>

Java EE (Introdução a plataforma e principais conceitos)

Especialmente os caps. 1, 4 e 5

<https://javaee.github.io/tutorial/toc.html>

Referências

Servidores de aplicação

<http://docs.wildfly.org/16>

Docker

Especialmente *Get Started* e *Develop with Docker*

<https://docs.docker.com/>

<https://medium.com/the-code-review/top-10-docker-commands-you-can-live-without-54fb6377f481>