

Plataforma JEE

Plataforma JEE

O que é?

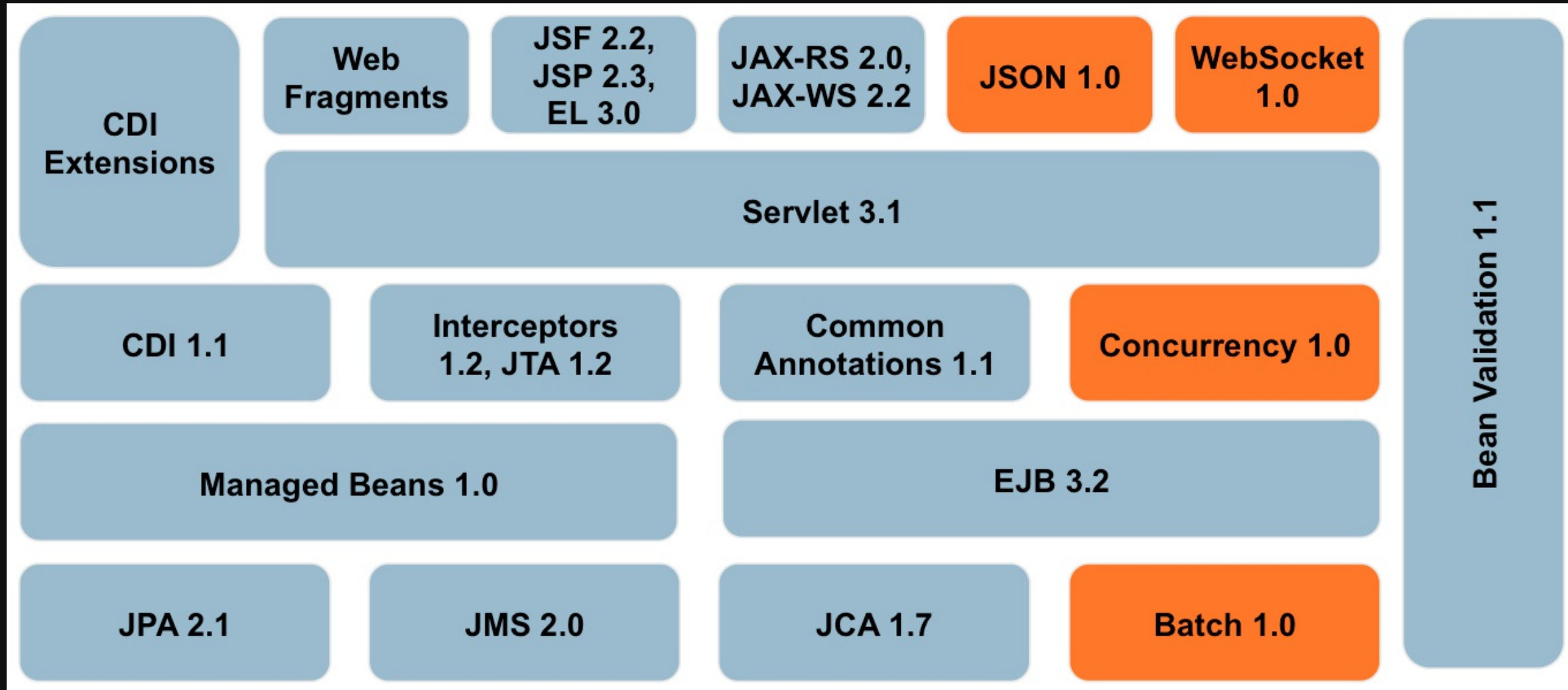
Plataforma JEE

Conjunto de especificações concebidas pela comunidade (JCP) e indústria para fornecer um ambiente onde componentes distribuídos colaboram entre si.

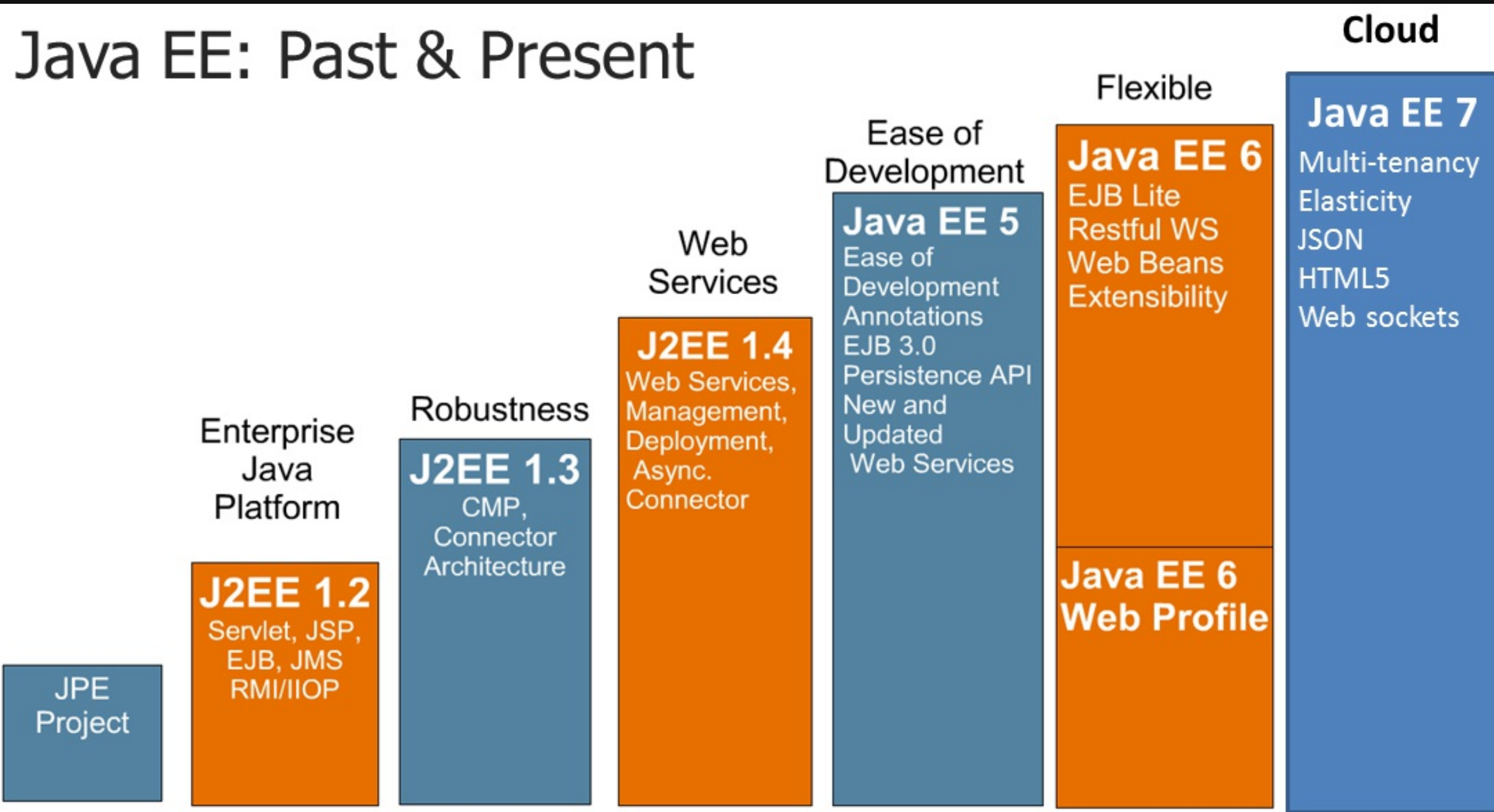
Plataforma JEE

O objetivo da plataforma é reduzir o tempo de desenvolvimento e complexidade das aplicações enquanto maximiza o seu desempenho.

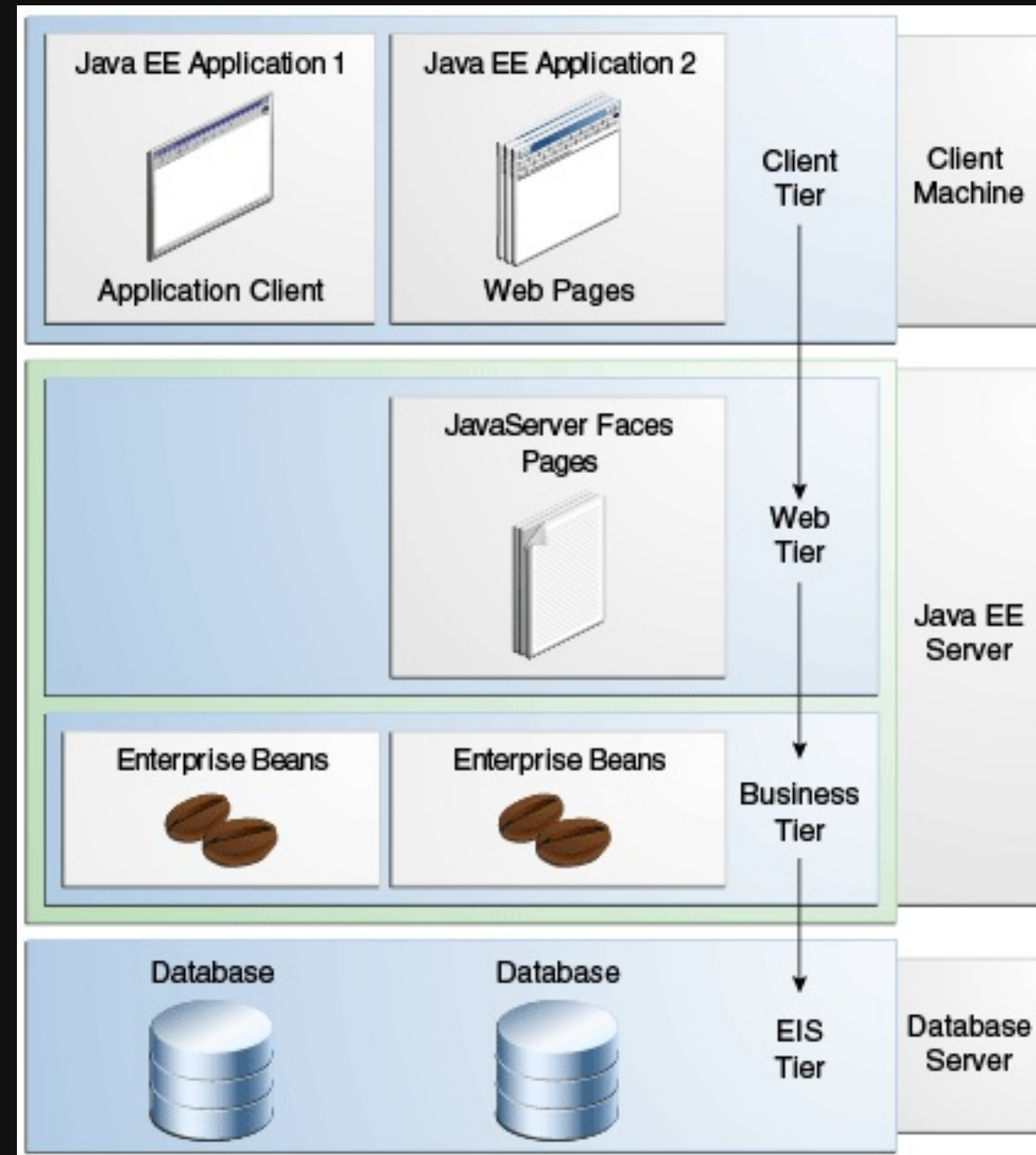
Especificações



Histórico



Camadas

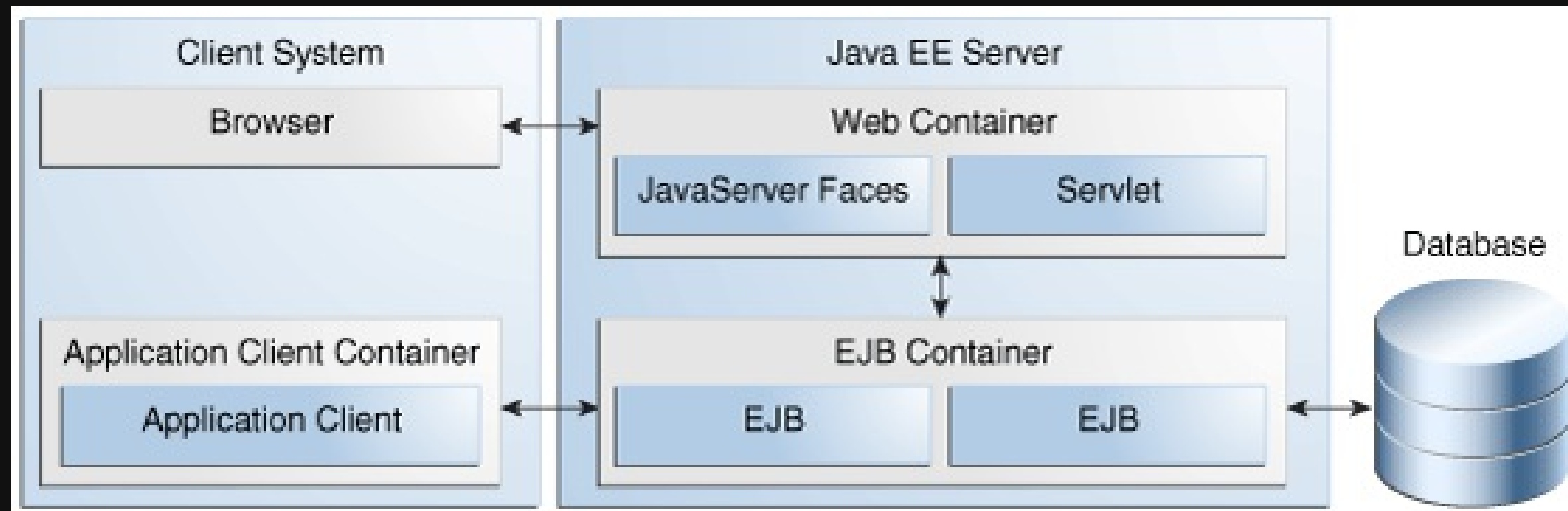


Na prática, quem junta essa "salada" de especificações?

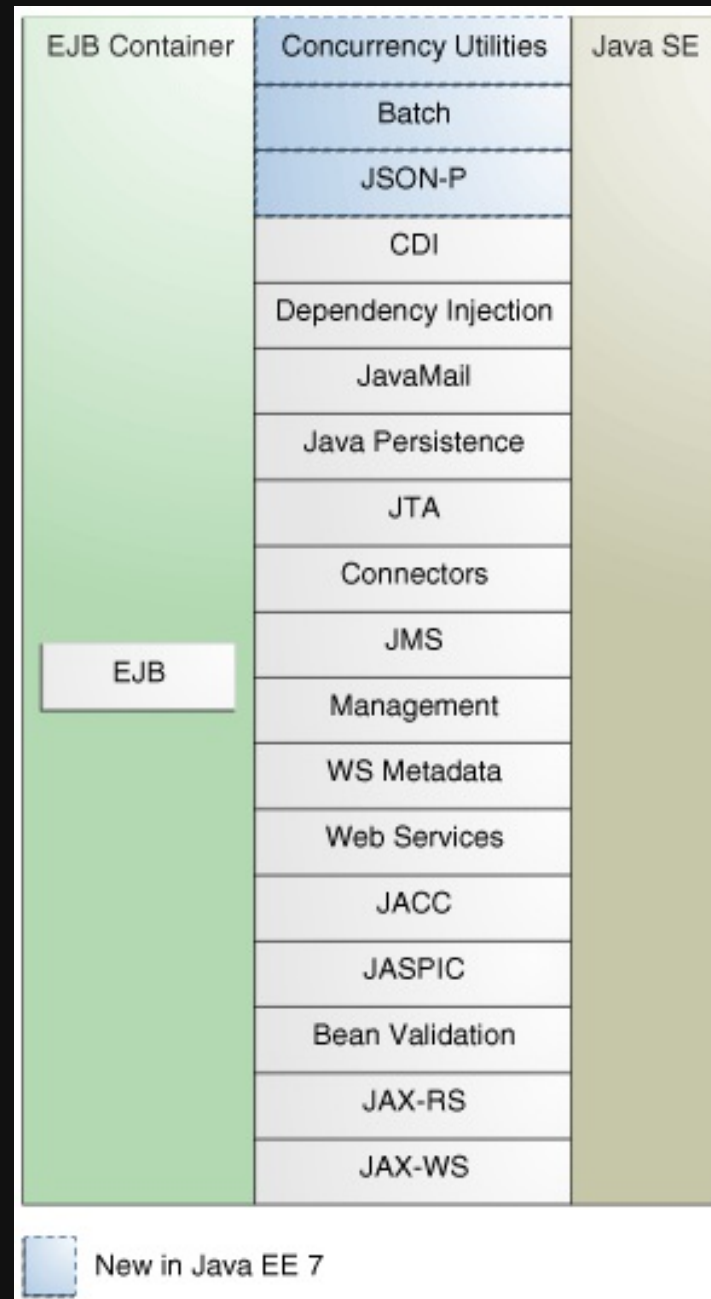
Container

- **Ambiente propício para a execução de aplicações JEE.**
- **Conjunto de serviços básicos (operacionais) para as aplicações**
- **Componentes focam na implementação da lógica de negócio**

Organização



APIs



Alguns Fornecedores



Glassfish

The screenshot displays the GlassFish Console web interface in a browser window. The browser's address bar shows the URL `localhost:4848/common/index.jsf`. The page title is "GlassFish Console - Common Tasks". The interface includes a navigation sidebar on the left with a tree view of the system hierarchy: Domain, server (Admin Server), Clusters, Standalone Instances, Nodes, Applications, Lifecycle Modules, Monitoring Data, Resources (expanded), and Configurations. The main content area is titled "GlassFish Console - Common Tasks" and is divided into several sections: "GlassFish News" with a link to "GlassFish News"; "Deployment" with links to "List Deployed Applications" and "Deploy an Application"; "Administration" with links to "Change Administrator Password" and "List Password Aliases"; "Monitoring" with a link to "Monitoring Data"; "Documentation" with links to "Open Source Edition Documentation Set", "Quick Start Guide", "Administration Guide", "Application Development Guide", and "Application Deployment Guide"; "Update Center" with links to "Installed Components", "Available Updates", and "Available Add-Ons"; and "Resources" with links to "Create New JDBC Resource" and "Create New JDBC Connection Pool". The top of the page features a header with "Home", "About...", and "Help" links, and a status bar indicating the user is "admin" on "domain1" at "localhost".

GlassFish Console - Common Tasks

GlassFish News

- GlassFish News

Deployment

- List Deployed Applications
- Deploy an Application

Administration

- Change Administrator Password
- List Password Aliases

Monitoring

- Monitoring Data

Documentation

- Open Source Edition Documentation Set
- Quick Start Guide
- Administration Guide
- Application Development Guide
- Application Deployment Guide

Update Center

- Installed Components
- Available Updates
- Available Add-Ons

Resources

- Create New JDBC Resource
- Create New JDBC Connection Pool

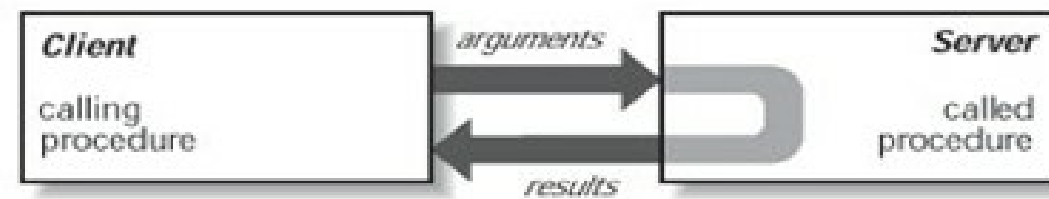
Serviços Básicos

Invocação Remota

Uma aplicação distribuída necessita disparar execuções em componentes que não estão disponíveis localmente.

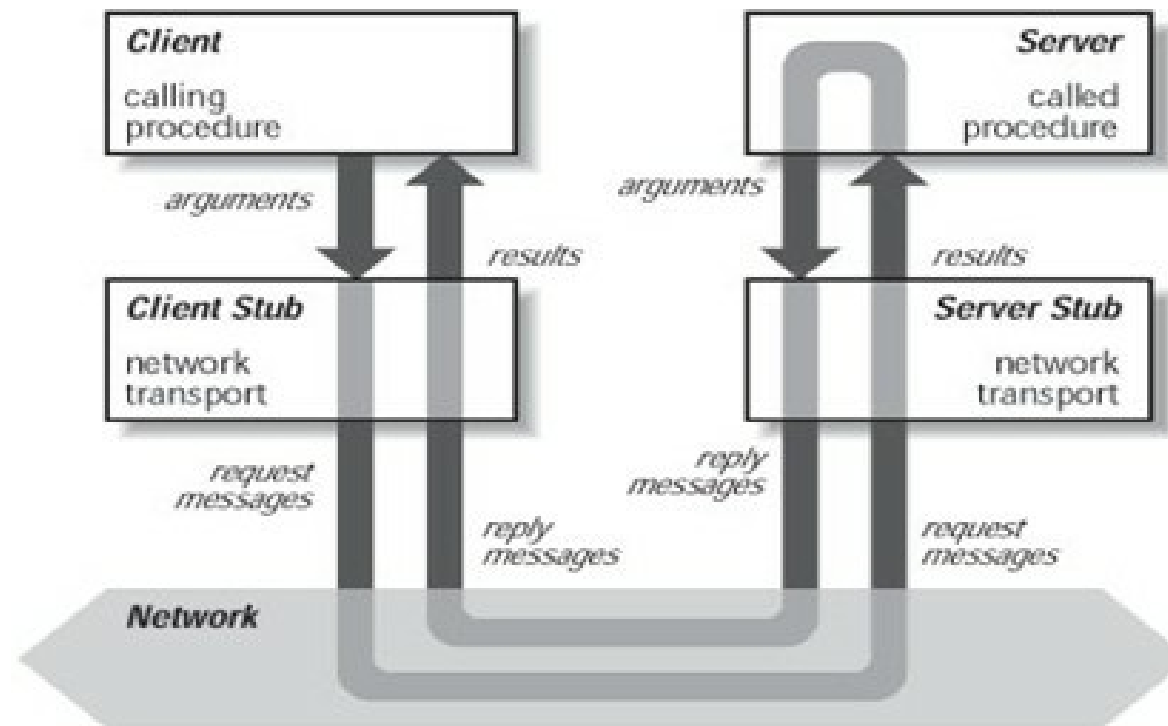
Invocação Remota

RPC



In a local procedure call, a calling process executes a procedure in its own address space.

Local Procedure Call



In a remote procedure call, the client and server run as two separate processes. It is not necessary for them to run on the same machine.

The two processes communicate through stubs, one each for the client and server. These stubs are pieces of code that contain functions to map local procedure calls into a series of network RPC function calls.

Remote Procedure Call

Invocação Remota

RMI/IIOP

- Protocolo que permite comunicação remota entre objetos
- Objeto remoto acessado via proxy
- Permite interoperar Java e Corba

Localização

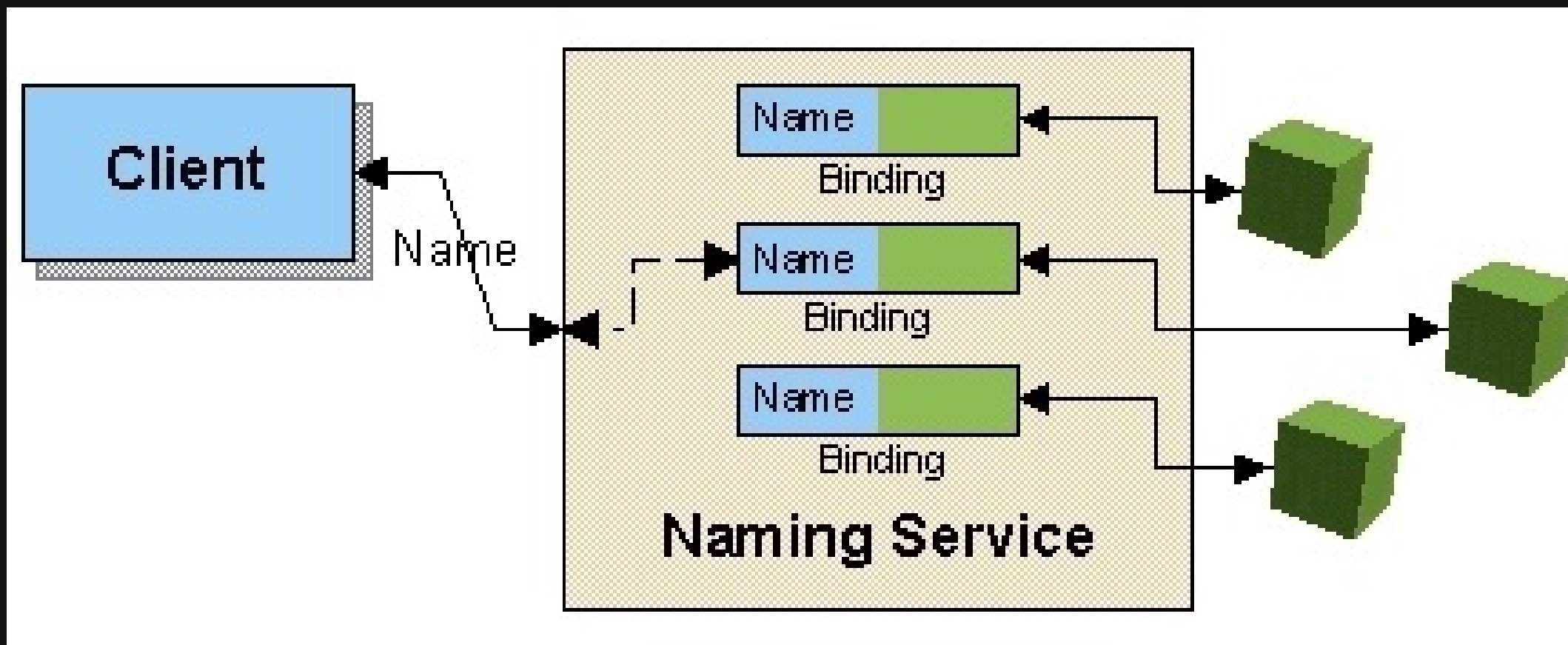
Aplicações distribuídas necessitam acessar recursos e componentes para executar suas ações.

Localização

A aplicação pode estar diretamente "ligada" ao recurso ou acessá-lo de outra forma.

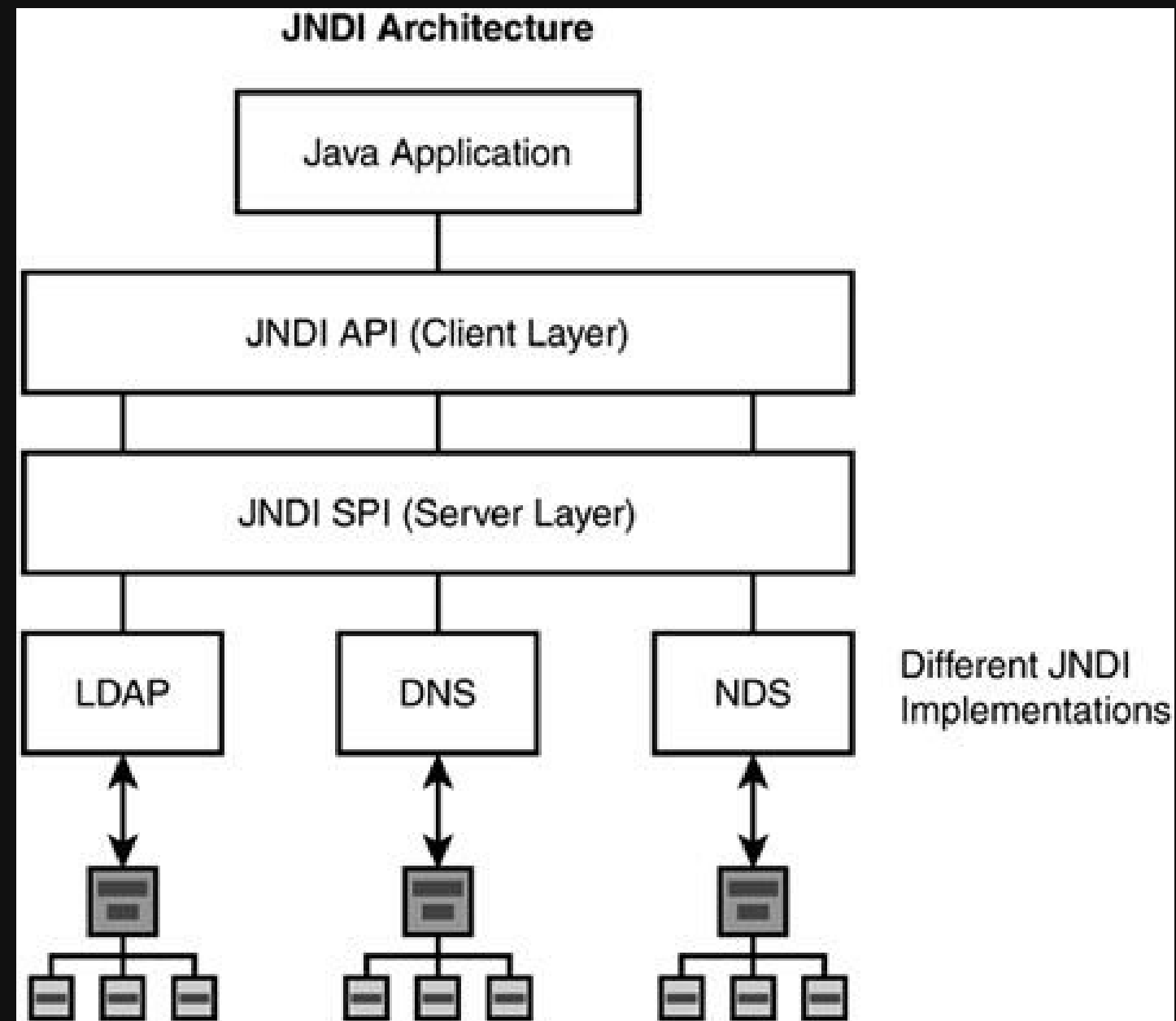
Localização

Serviço de Nomes



Localização

JNDI



Localização

Localização

```
// Lookup the remote object via JNDI
Context ctx = new InitialContext(System.getProperties());
Object remoteObject = ctx.lookup("MyObj");
// Cast the remote object, RMI-IIOP style
MyRemoteObj generator = (MyRemoteObj)
    PortableRemoteObject.narrow(remoteObject, MyRemoteObj.class);
generator.doSomething();
```

Recursos e Dependência

Uma aplicação pode localizar ou construir diretamente o recurso que necessita

```
protected void doGet(...) {  
    String mercadoria = req.getParameter("mercadoria");  
    ControladorEstoque control = new ControladorEstoque();  
    String status = control.reservaMercadoria(mercadoria);  
    resp.getWriter().append("Status da Reserva: ").append(status);  
}
```

Recursos e Dependência

Problemas da abordagem

- Código de infraestrutura misturado ao código útil (negócio).
- Aplicação conhece como construir o recurso. Logo, está acoplado a ele.

Recursos e Dependência

Uma alternativa a essa abordagem, é inverter a ordem das coisas...

Recursos e Dependência

Injeção de Recurso

- Permite injetar recursos em um componente
- Ambos devem ser gerenciados pelo container
- O recurso é identificado por um nome
 - `@Resource`

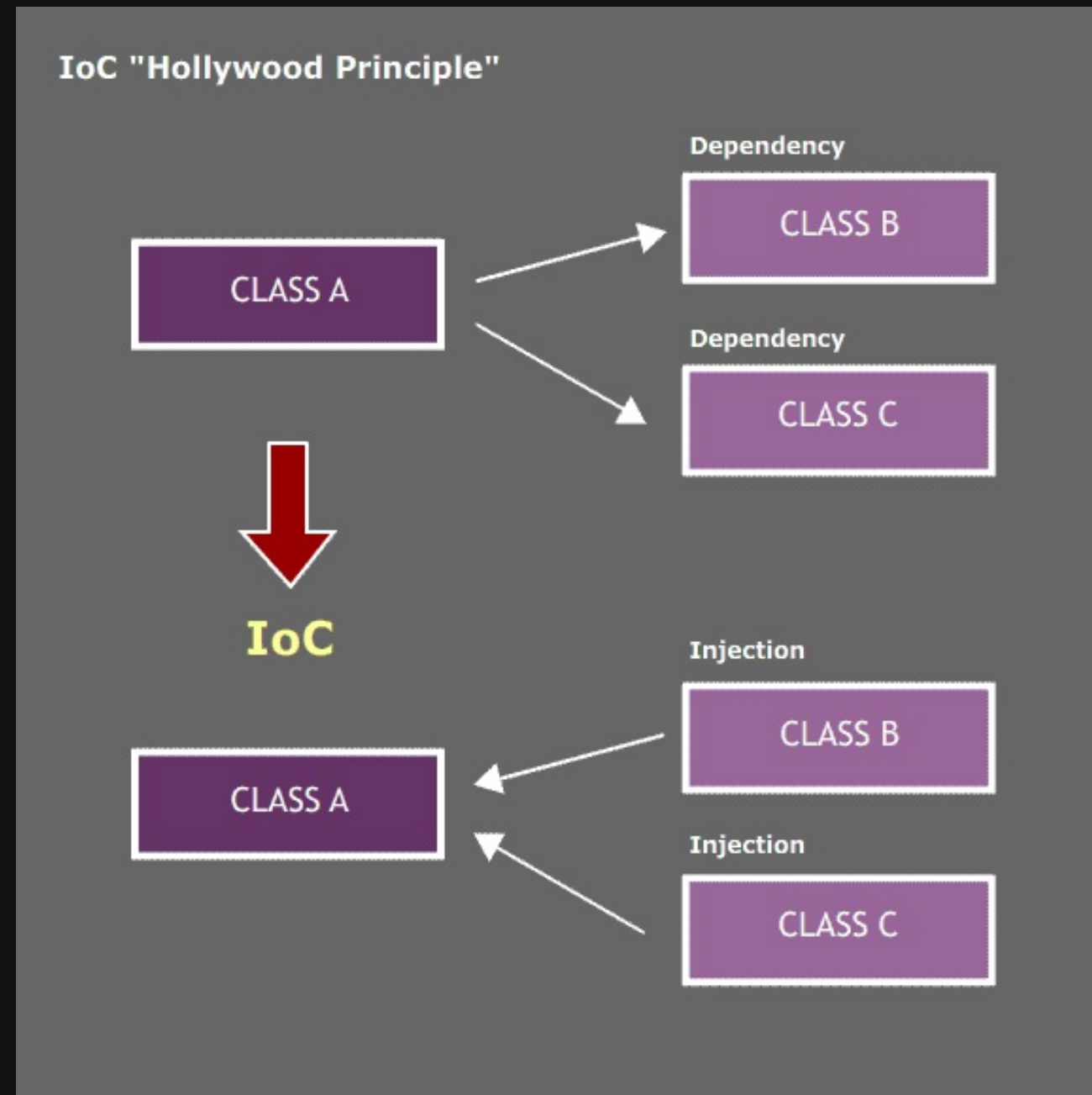
Recursos e Dependência

Exemplo

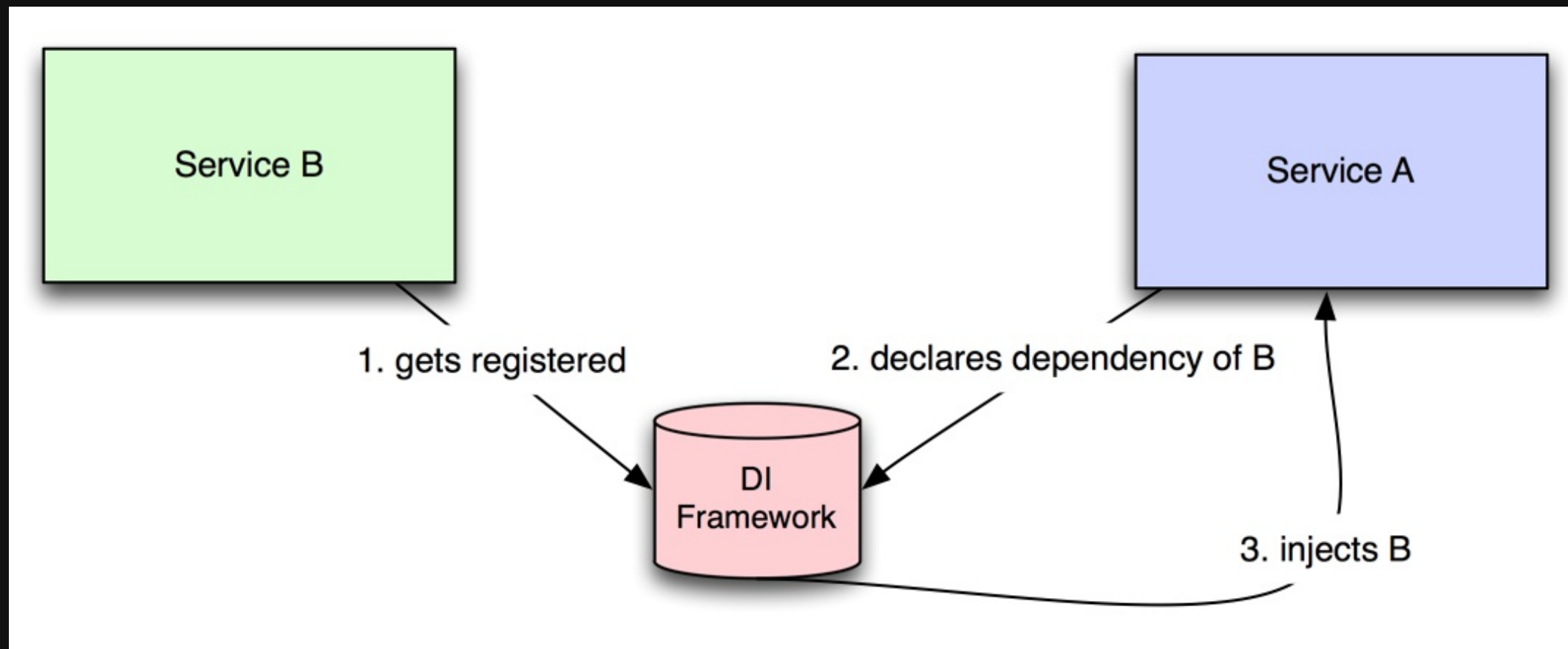
```
public class LibraryServlet extends HttpServlet {  
    @Resource(lookup="java:comp/SQLiteDataSource")  
    private javax.sql.DataSource dsc;  
    ...  
}
```

Recursos e Dependência

Inversão de Controle



Recursos e Dependência



Recursos e Dependência

Considerações

- O componente e o objeto devem ser criados pelo container
- O componente é um objeto gerenciado pelo container
 - Servlets, Managed Beans, EJBs, etc
- As dependências são declaradas por anotações
 - `@Inject`, `@EJB`, etc
- O objeto inserido possui um escopo
 - `@RequestScoped`, `@SessionScoped`, `@ApplicationScoped`, etc

Recursos e Dependência

Objeto

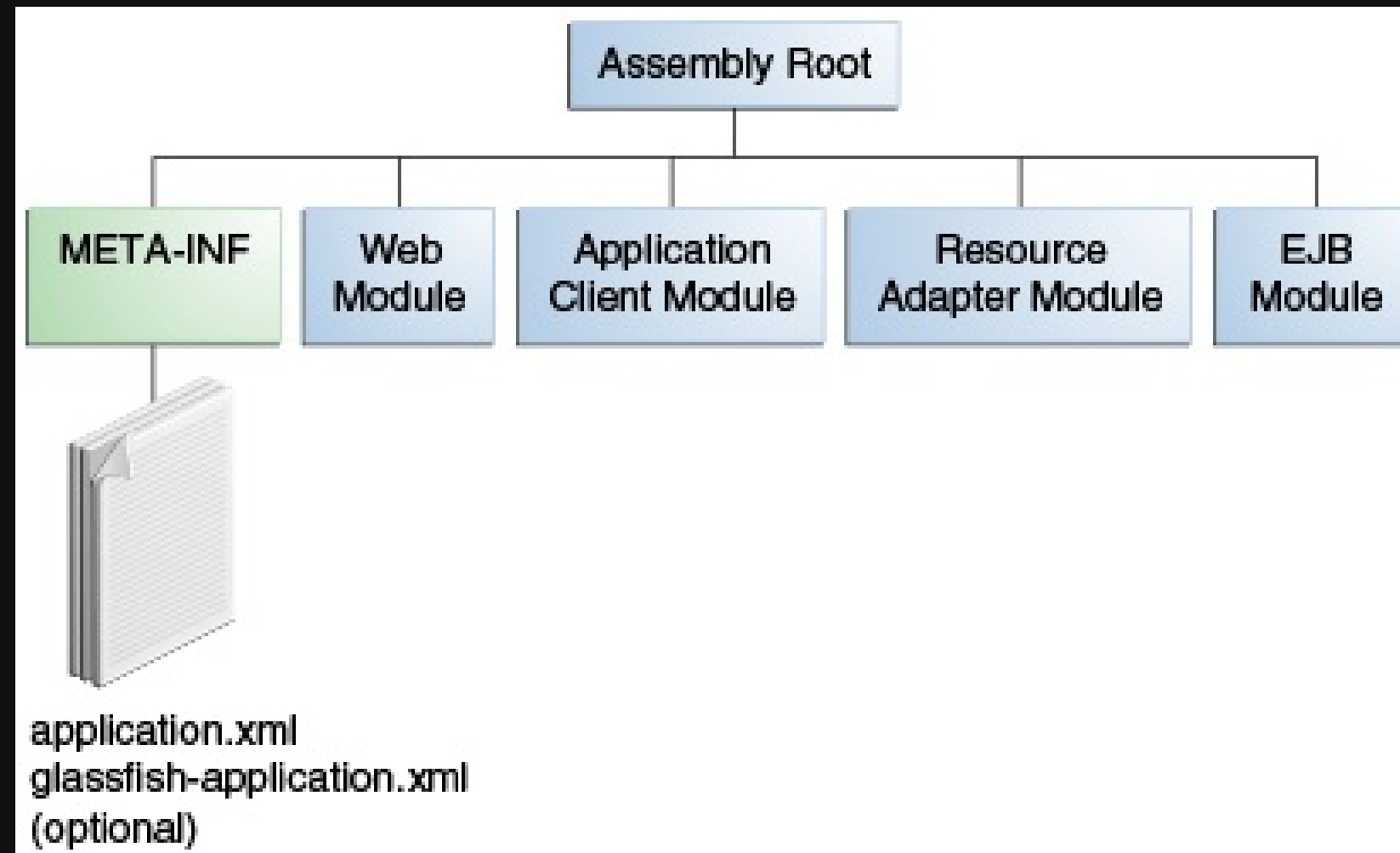
```
@RequestScoped  
public class EstoqueSimples implements Estoque {  
    ...  
}
```

Componente

```
public class ReservaServlet extends HttpServlet {  
    @Inject private ControladorEstoque controle;  
    ...  
    protected void doGet(...) {  
        String merc = request.getParameter("mercadoria");  
        controle.avaliarPedido(merc);  
    }  
}
```

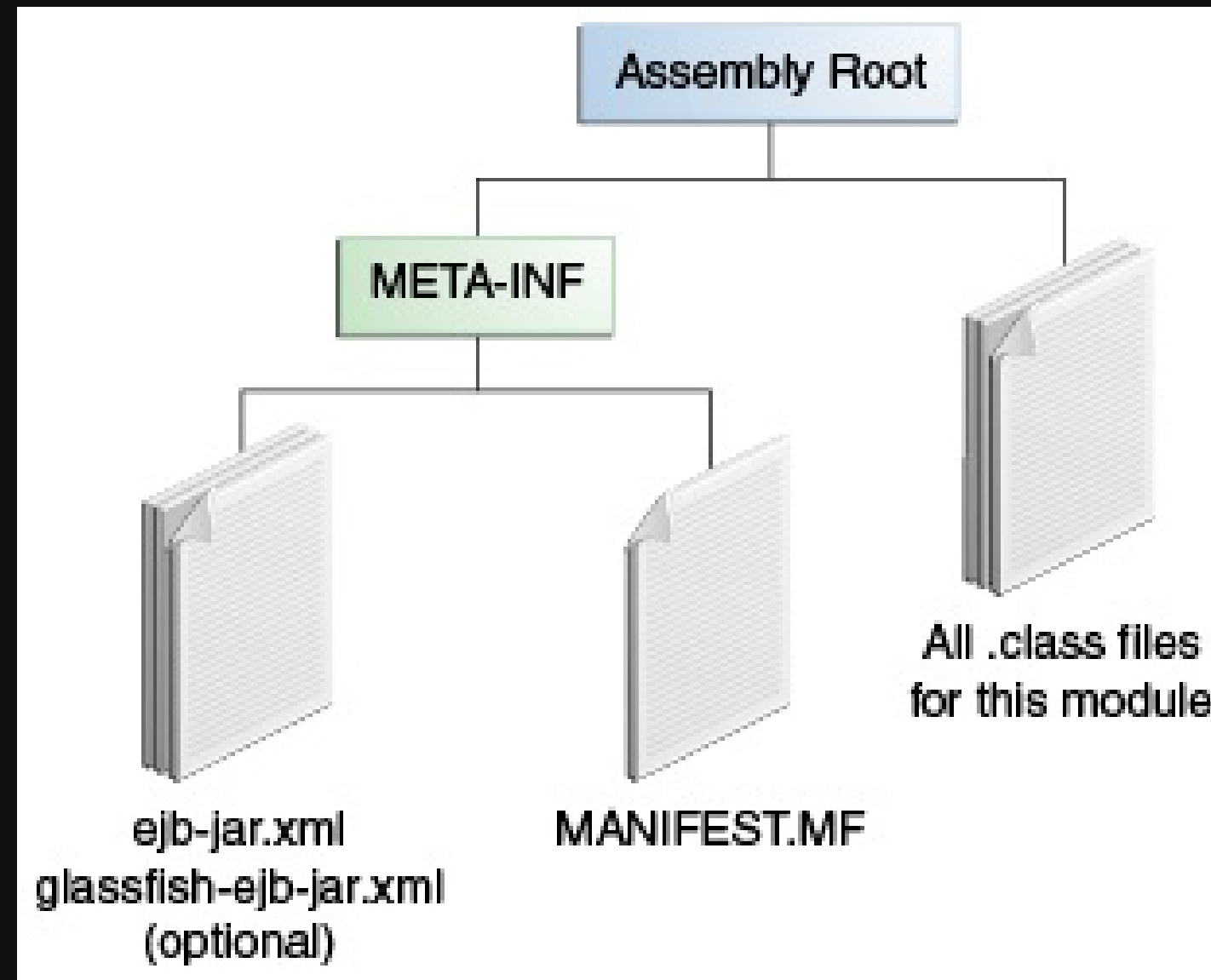
Empacotamento

Estrutura de uma aplicação



Empacotamento

Estrutura de um componente



Exercício 1

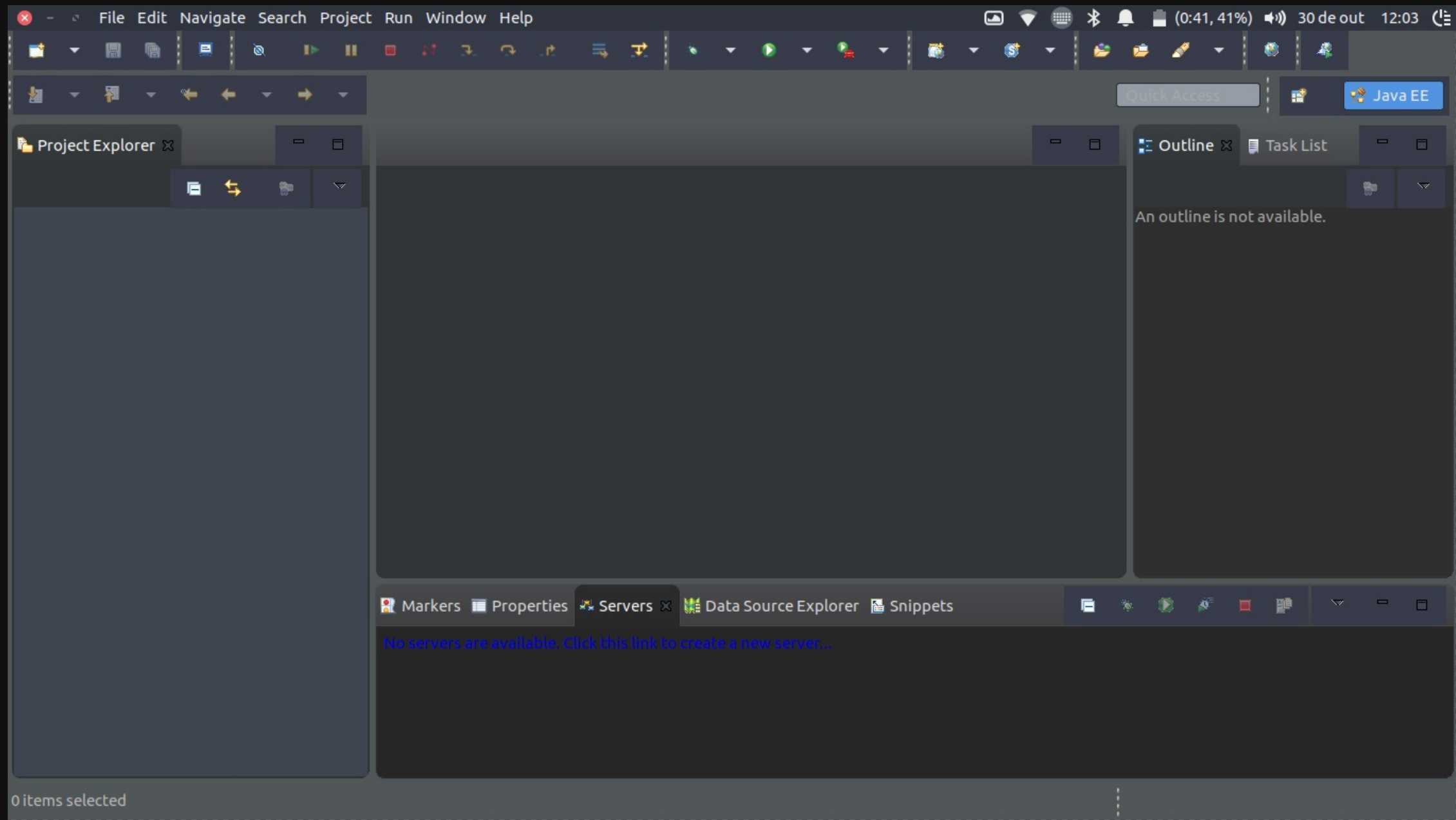
Uma aplicação corporativa reserva de mercadorias por meio de um servlet `ReservaMercadoriaServlet` e um objeto do tipo `ControleEstoque`.

Adaptar o servlet e o objeto para que a referência ao objeto seja obtida por meio de injeção simples com o escopo de Requisição.

Dicas

1. Crie o servidor Glassfish
2. Importe o projeto distribuidora que se encontra na pasta de exercícios
3. Use as anotações `@Inject` e `RequestScoped`


Criando o servidor



New Server

Define a New Server

Click Next to download and install




☒ Show downloadable server adapters


Select the server type:

type filter text

▶ ObjectWeb

▼ Oracle

 GlassFish Tools

 Oracle Cloud Tools

Tools for developing applications for GlassFish.

?

< Back


Next >

Cancel

Finish

New Server

Feature License



Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and


b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program; where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include

☒ I accept the terms of the license agreement

☐ I do not accept the terms of the license agreement



< Back

Next >

Cancel

Finish

New Server

Define a New Server

Choose the type of server to create

Show downloadable server adapters

Select the server type:

type filter text

▼ GlassFish

GlassFish 3.1

GlassFish 4

Reference implementation of Java EE 7.

Server's host name:

localhost

Server name:

GlassFish 4 at localhost

?

< Back

Next >


Cancel

Finish

New Server

GlassFish 4

Define GlassFish runtime properties.



Name:


†

ClassFish 4

ClassFish location:

†


/home/michel/apps/glassfish4/glassfish




Java location:

†

/usr/lib/jvm/java-8-oracle





< Back

Next >

Cancel

Finish

42 / 42