

Jakarta/Enterprise Java Beans Conceitos

Prof. Dr. Alexandre L'Erario



Enterprise Java Beans

- Um tipo especial de classe que reside dentro de um container
- São componentes implementados em Java, que utilizam a especificação EJB e rodam sobre um servidor JEE que implementa um container EJB
- Embora transparente para o desenvolvedor, o container EJB provém um conjunto de serviços como transações e segurança.

Enterprise Java Beans

- O que é um EJB
 - Escrito em Java um componente enterprise java beans é um componente *server-side* que encapsula lógica de negócios de uma aplicação. O código da lógica de negócio é do ponto de vista do cliente o que agrega valor ao software
- Benefícios do EJB
 - Plataforma para desenvolvimento de sistema distribuído
 - Plataforma para o desenvolvimento de regras de negócio
 - Componentes portáteis
- Quando utilizar EJB
 - Quando a aplicação abrange os seguintes requisitos:
 - Escalabilidade
 - Controle de transações
 - Variedade e grande quantidade de clientes

Desenvolvimento de EJB

- Desenvolvimento orientado a POJOS

(Plain Old Java Objects)

- Uso de anotações:

- @Stateless
- @Stateful
- @Remote, @Local
- @MessageDriven
- @PreCreate, @PostCreate, @PrePassivate, @Destroy
- @Interceptors
- @AroundInvoke

Tipos de EJB

- Sessão

- Executa uma tarefa para um cliente, eventualmente pode implementar um web service

- Mensagem

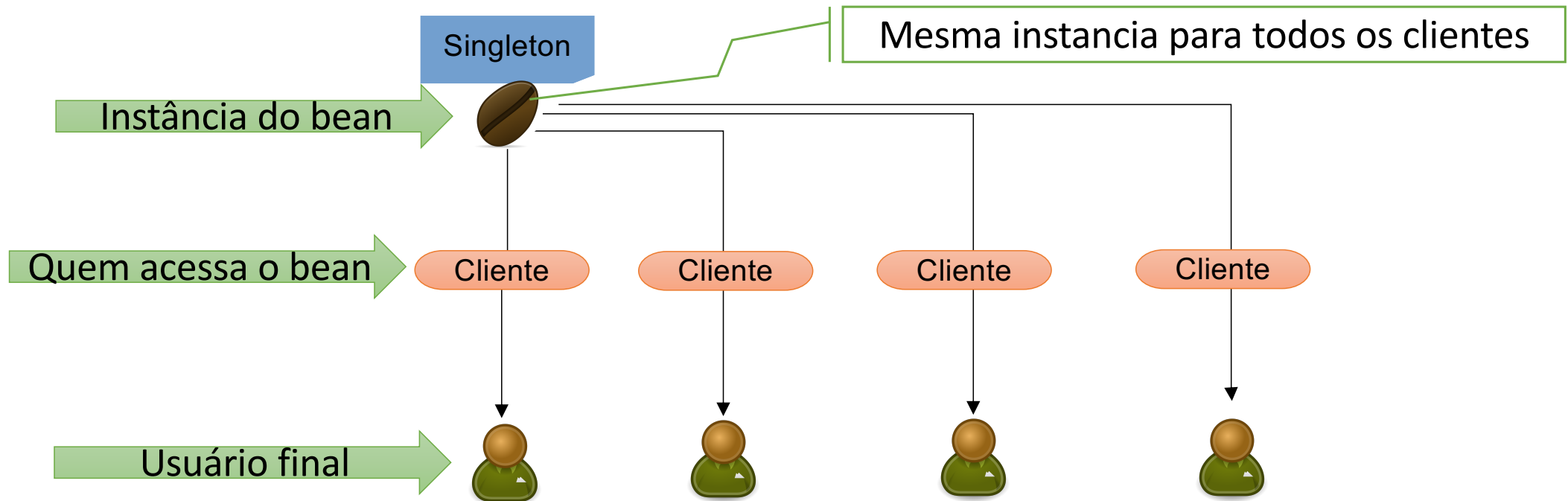
- Age como um listener de uma aplicação assíncrona, pode utilizar JMS por exemplo

Tipos de Enterprise Java Beans

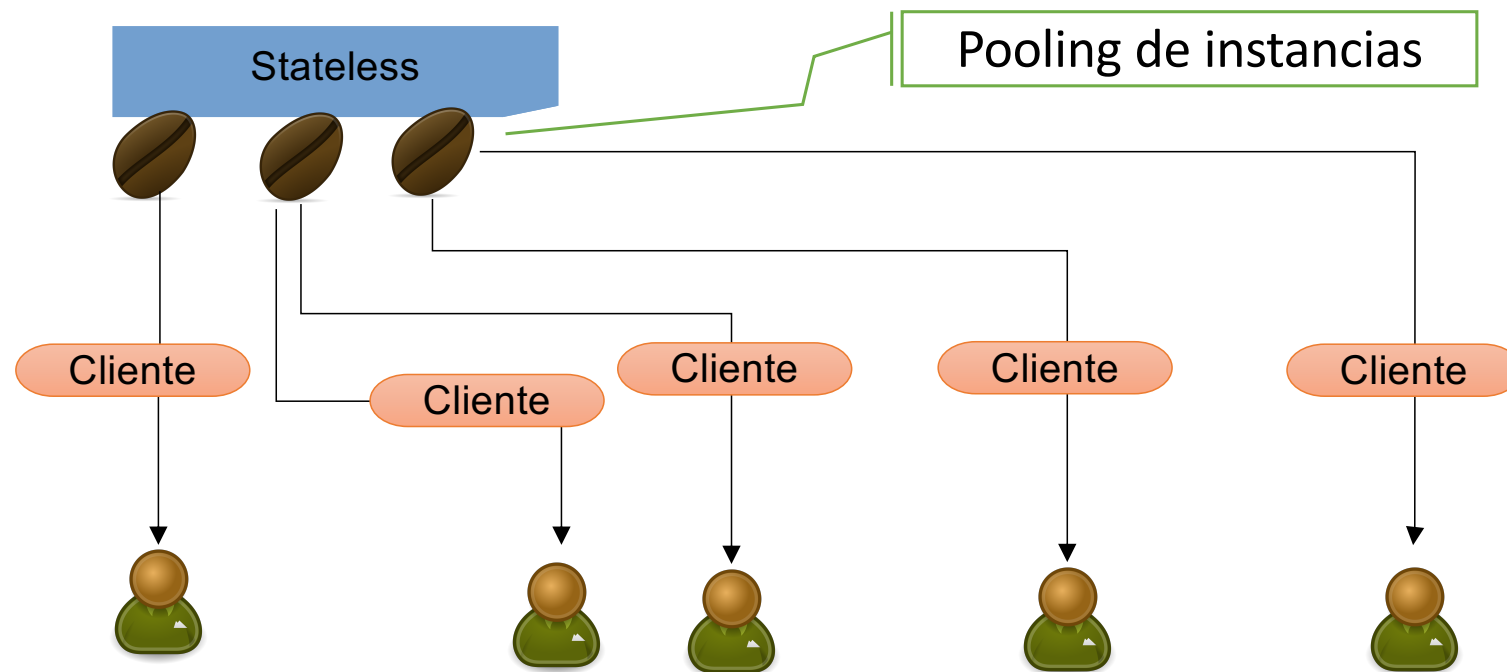
Tipo	Tipo	Descrição
Session Beans	Singleton	= Stateless + mantém o estado (1 instância/aplicação)
	Stateless	não guarda o estado de suas variáveis de instância (atributos) entre as requisições, ou seja, terminada a requisição (execução de um método), o bean é desassociado do cliente.
	Stateful	guarda o estado de suas variáveis de instância, lembrando o conceito de uma sessão web. O bean é desassociado do cliente quando este solicita a remoção do bean ou quando termina sua execução, deixando a remoção do bean a cargo do container.
Message-Driven	Age como um listener de uma aplicação assíncrona	

Singleton Session Bean

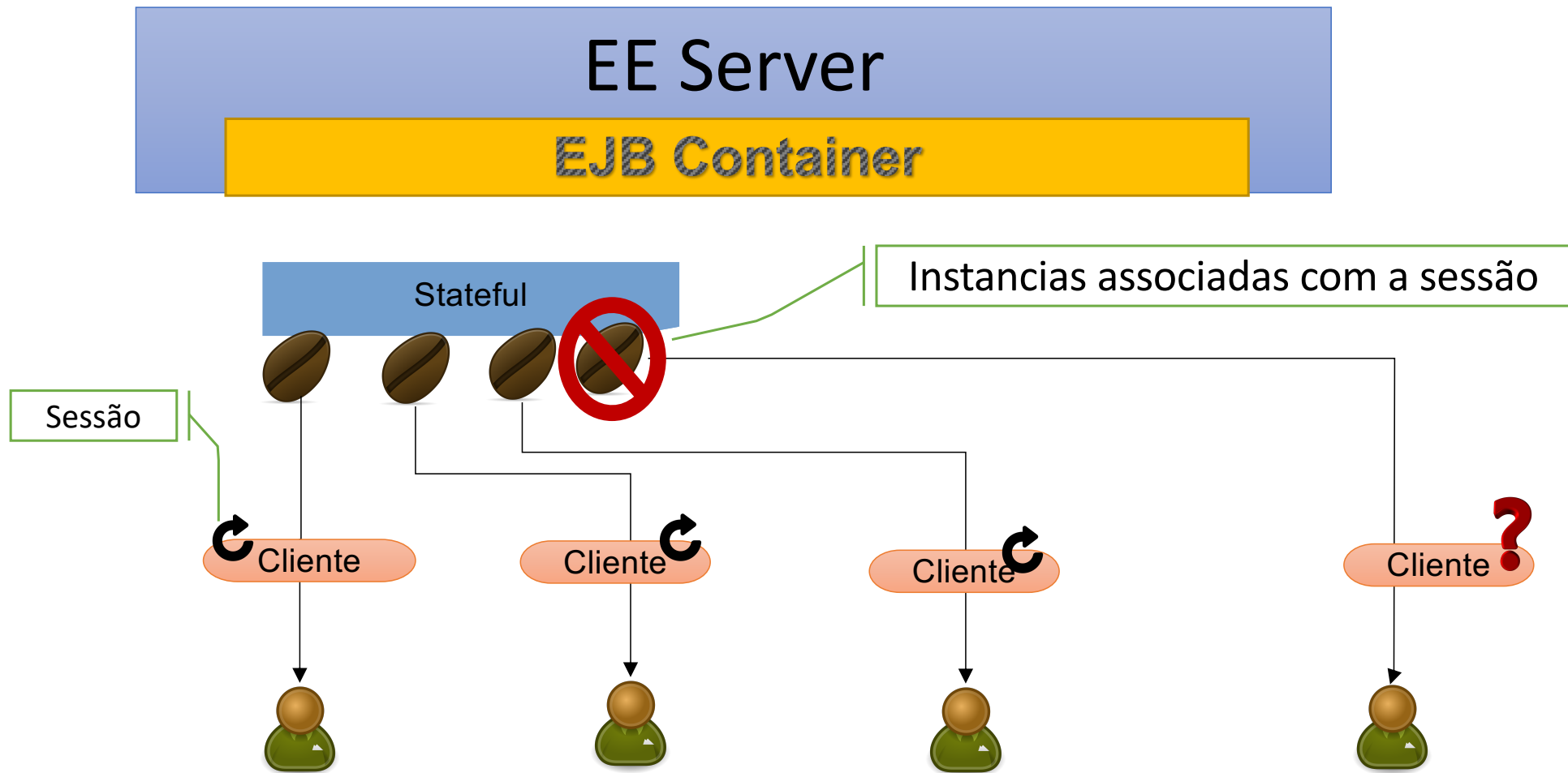
EE Server
EJB Container



Stateless Session Bean



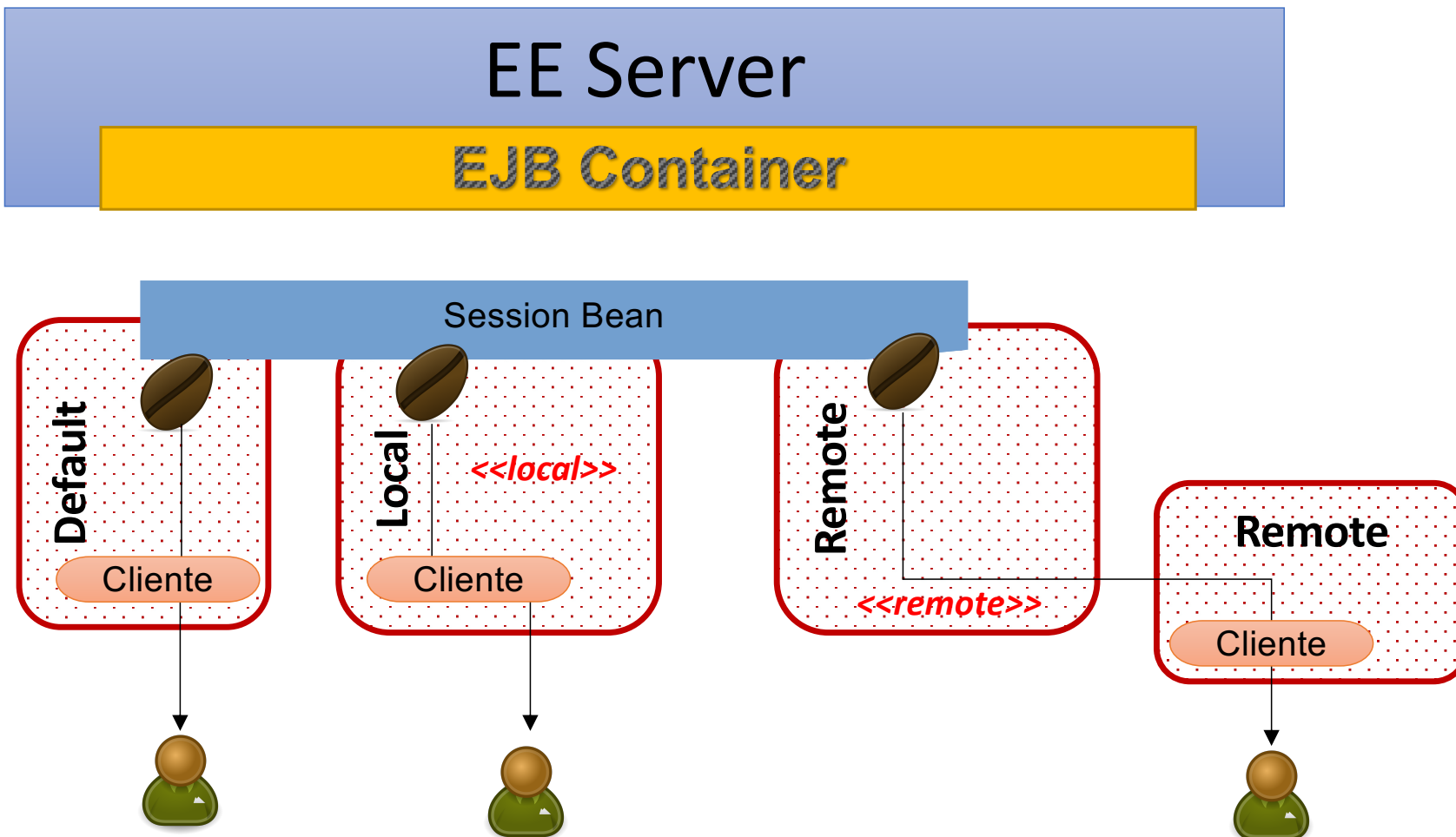
Stateful Session Bean



Interfaces de acesso

- Utiliza JNDI
- Tipos de acesso:
 - Local
 - Remote
 - Web Service
- A decisão de qual utilizar está centrada na arquitetura /demanda desejada
 - Acoplamento entre tiers
 - Tipo dos clientes
 - Distribuição dos componentes
 - Performance

Interfaces de acesso

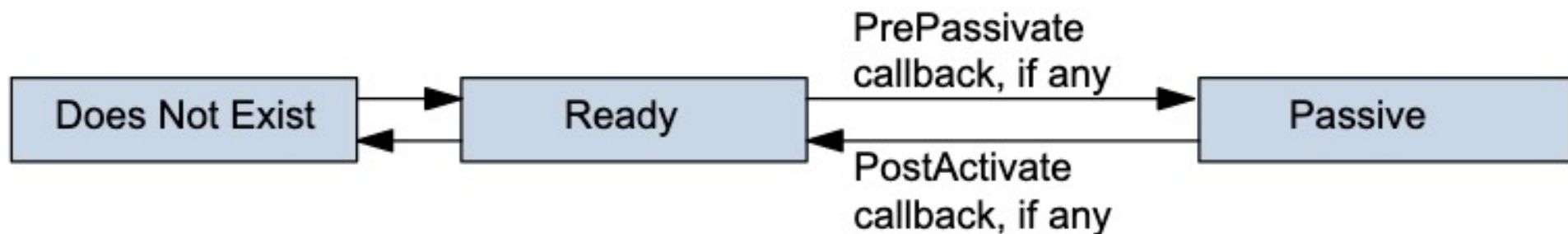


Quando utilizar Session beans

- Singleton:
 - Acesso compartilhado a mesma informação
- Stateless:
 - Uma interação resolve o problema
 - Demanda vazão e performance
- Stateful:
 - Várias interações para resolver um problema
 - Uma instancia por cliente

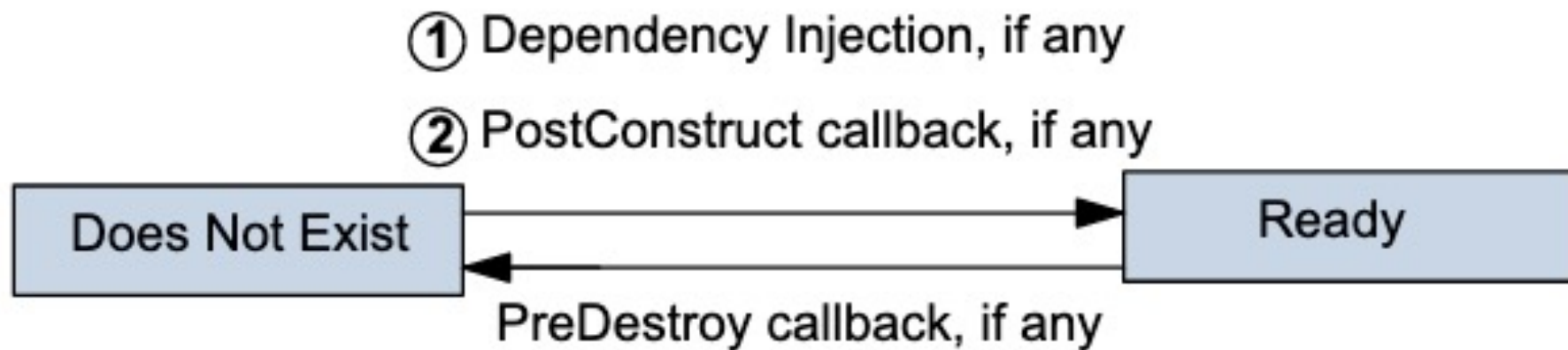
Ciclo de vida do EJB - Stateful

- ① Create
- ② Dependency Injection, if any
- ③ PostConstruct callback, if any
- ④ Init method, or ejbCreate<METHOD>, if any

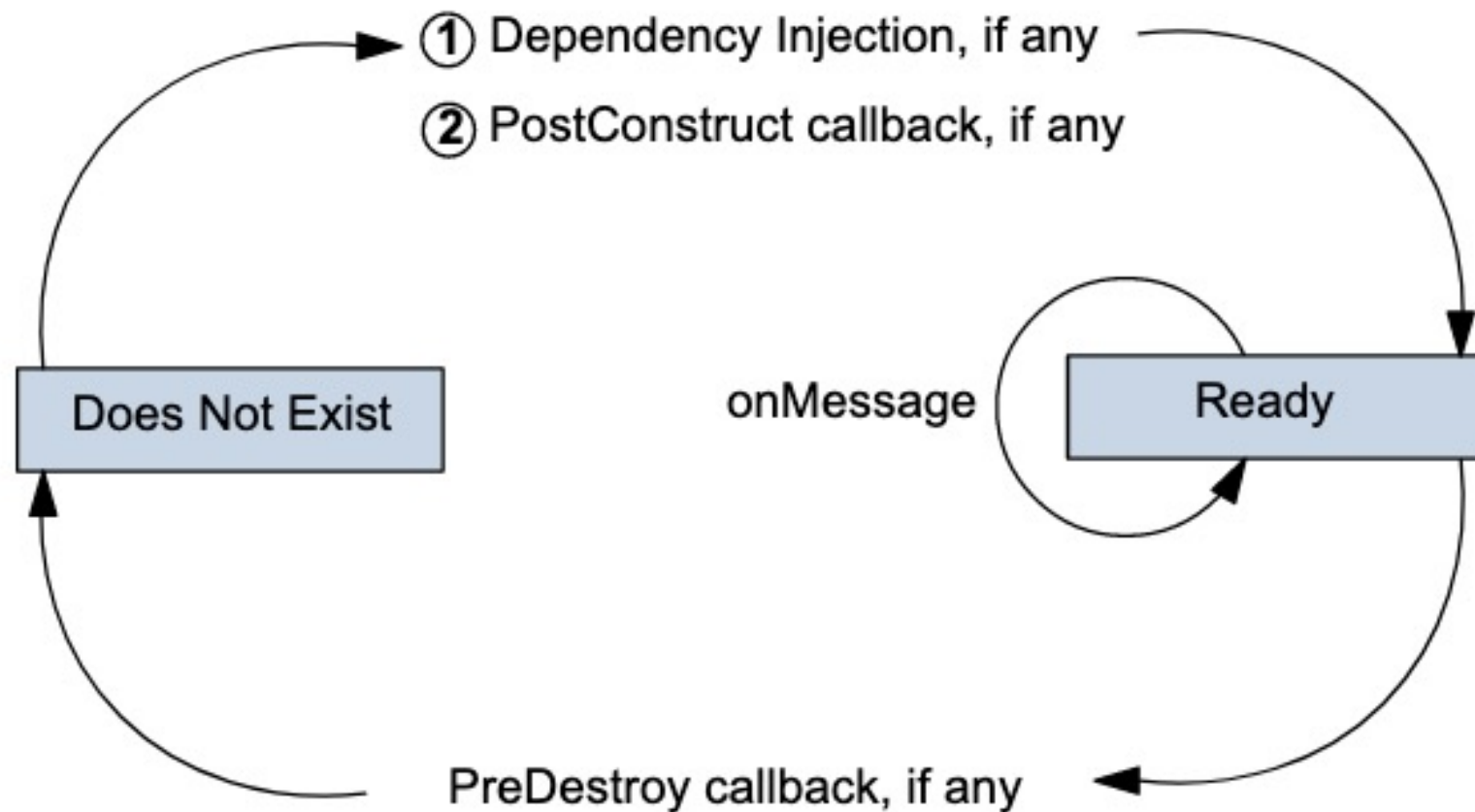


- ① Remove
- ② PreDestroy callback, if any

Ciclo de vida do EJB – Stateless e Singleton



Ciclo de vida do EJB – Message-driven



Alguns servidores EE

