



UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ



Enterprise Java Beans

Alexandre L'Erario
alerario@utfpr.edu.br



UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ



Message-Driven Beans

alerario@utfpr.edu.br

Message-Driven Java Beans

- Listener para aplicações assíncronas
- Atua com uma estrutura JMS
- O usuário não cria interface de acesso
- A estrutura de acesso é configurada no container JEE
- Um MDSB pode ser processado por vários clientes

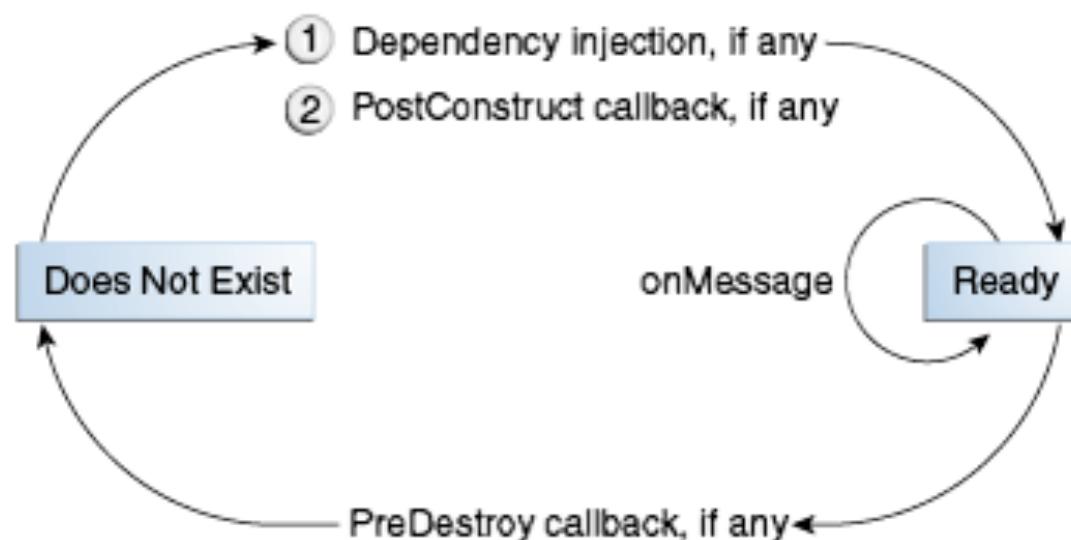


Características

- Eles executam após o recebimento de uma mensagem de cliente único
- Eles são chamados de forma assíncrona
- Eles são tem relativamente vida curta
- Eles não representam os dados diretamente no banco de dados, mas eles podem acessar e atualizar dados
- Eles podem ser transaction-aware (reconhecer transações)
- Eles não têm estado



Ciclo de vida

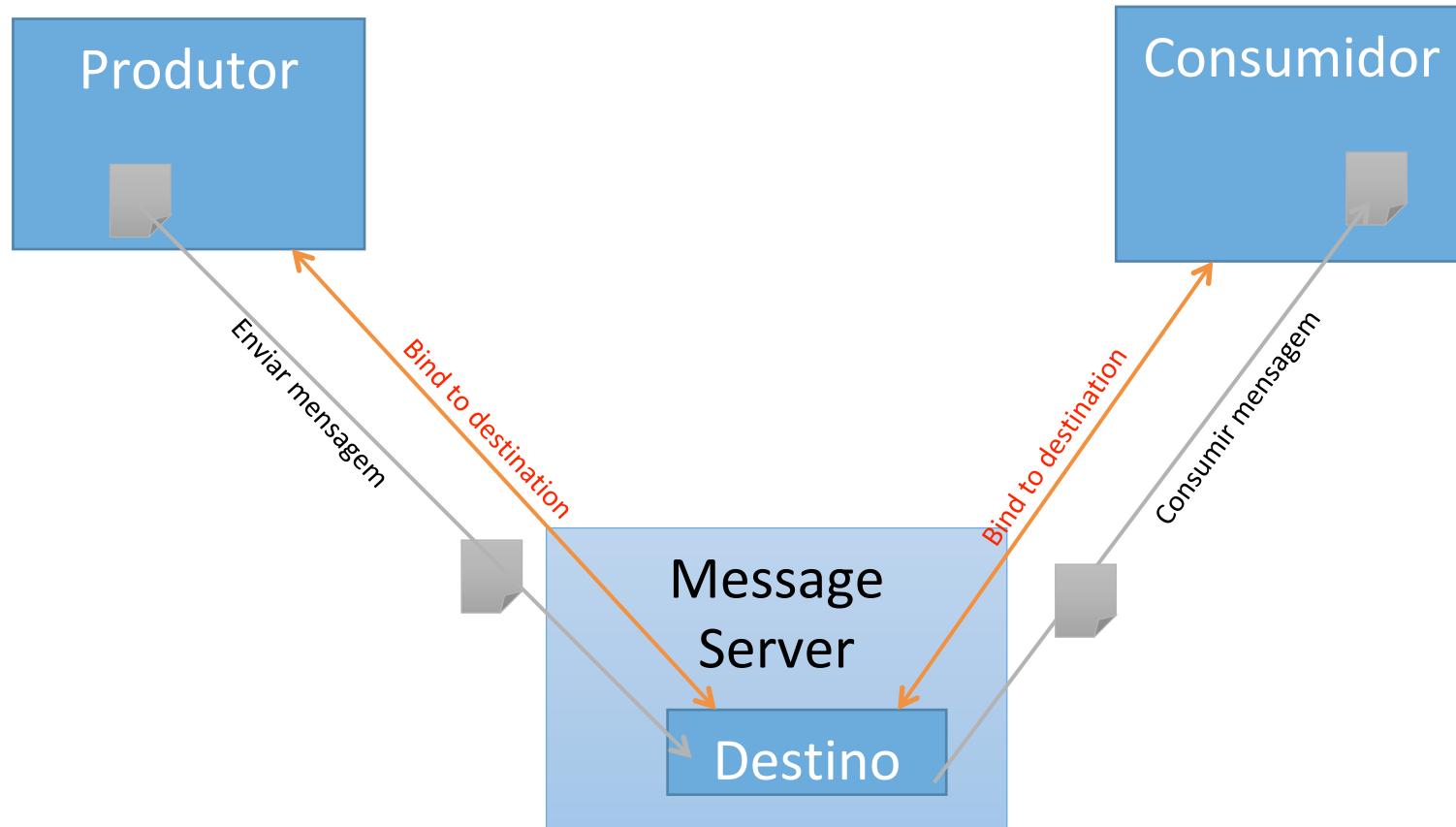


Tipos de mensagens JMS

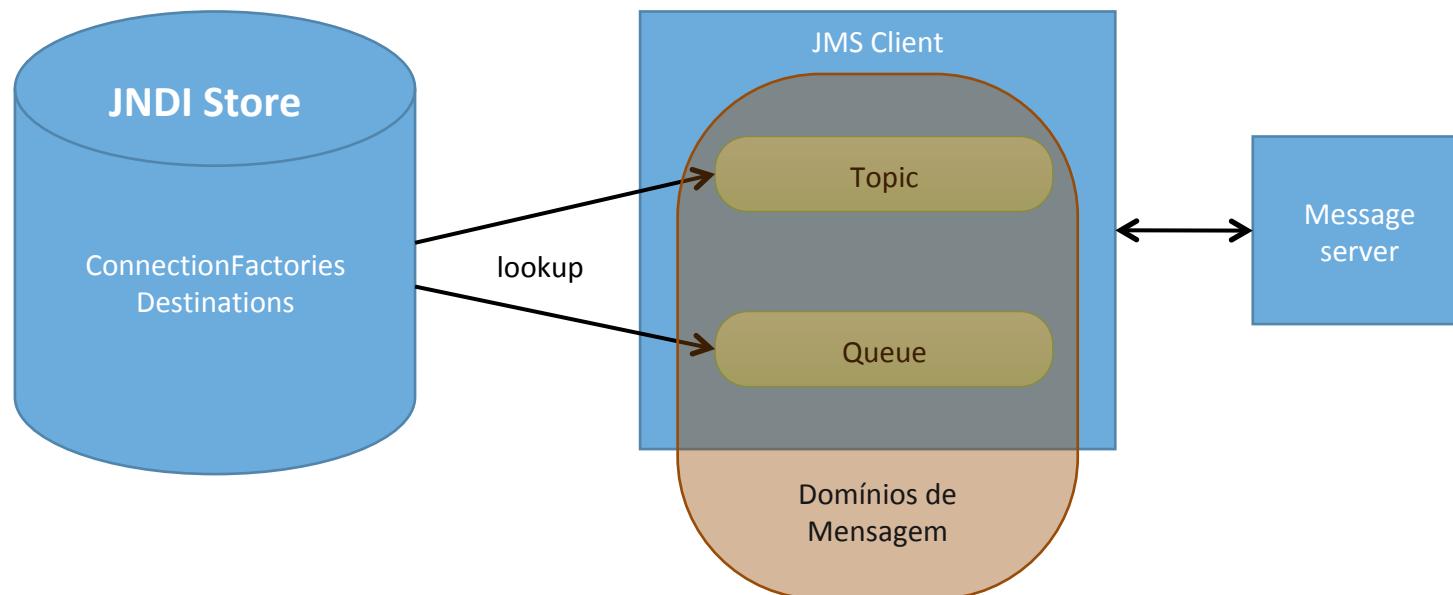
Tipo da mensagem	Corpo
Message	Sem corpo
TextMessage	String padrão java
ObjectMessage	Um objeto serializável
MapMessage	Um conjunto de pares nomes/valores utilizados em primitivas java
StreamMessage	Uma stream de Java
BytesMessage	Uma stream de bytes



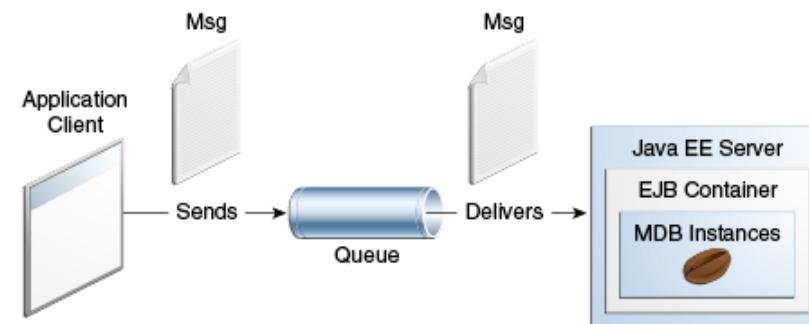
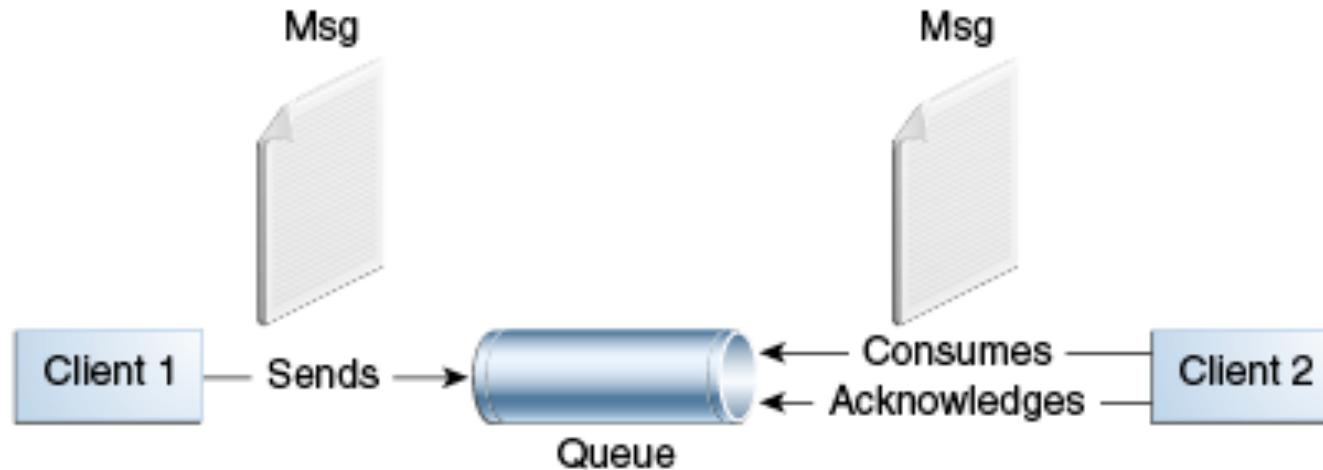
Produtores, consumidores e destinos



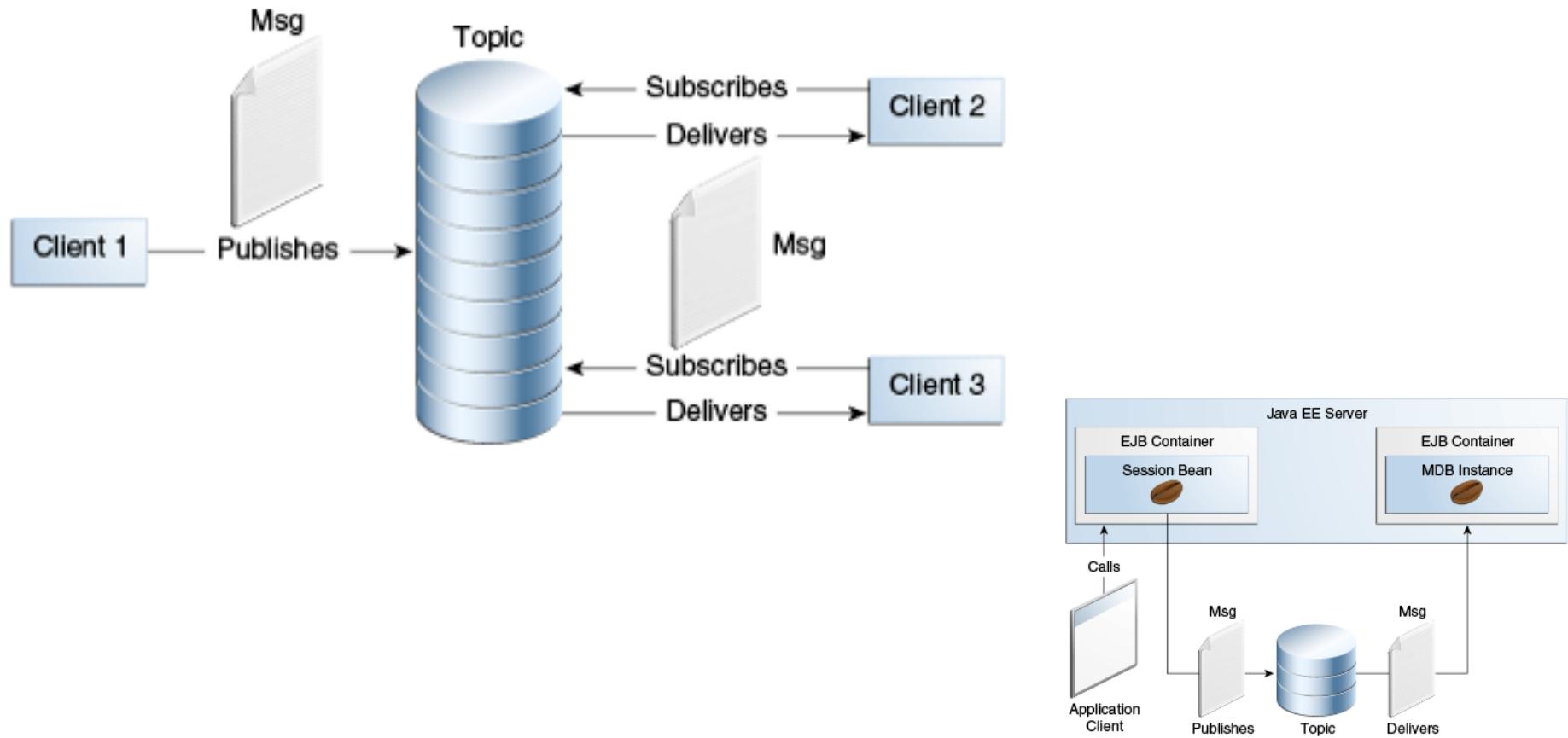
Connection Factories



Point-to-Point Messaging Domain



Publish/Subscribe Messaging Domain



Implementação

- Ao criar o message-driven, o netbeans seleciona ou cria uma nova estrutura no conteiner JEE
- É necessário codificar o método onMessage
- No cliente são criados dois métodos
 - Cria a mensagem
 - Envia a mensagem



Consumidor

- Deve implementar a interface MessageListener e iniciar com a anotação @MessageDriven
- @MessageDriven deve conter as configurações de ativação

```
@MessageDriven(activationConfig = {  
    @ActivationConfigProperty(propertyName = "destinationLookup",  
        propertyValue = "jms/testeDestino"),  
    @ActivationConfigProperty(propertyName = "destinationType",  
        propertyValue = "javax.jms.Queue")  
})
```

Esta configuração deve ser criada no servidor



@ActivationConfigProperty Settings

Property Name	Description
acknowledgeMode	Acknowledgment mode, used only for bean-managed transactions; the default is Auto-acknowledge (Dups-ok-acknowledge is also permitted)
destinationLookup	The lookup name of the queue or topic from which the bean will receive messages
destinationType	Either javax.jms.Queue or javax.jms.Topic
subscriptionDurability	For durable subscriptions, set the value to Durable; see Creating Durable Subscriptions for more information
clientId	For durable subscriptions, the client ID for the connection (optional)
subscriptionName	For durable subscriptions, the name of the subscription
messageSelector	A string that filters messages; see JMS Message Selectors for information
connectionFactoryLookup	The lookup name of the connection factory to be used to connect to the JMS provider from which the bean will receive messages



Implementação do produtor

```
@Resource(lookup = "java:comp/DefaultJMSSConnectionFactory")
    private ConnectionFactory connectionFactory;
@Resource(lookup = "jms/testeDestino")
    private Queue queue;
=====
try{
    JMSContext context = connectionFactory.createContext();
    context.createProducer().send(queue, "conteudo");
}catch(Exception e){ }
```

