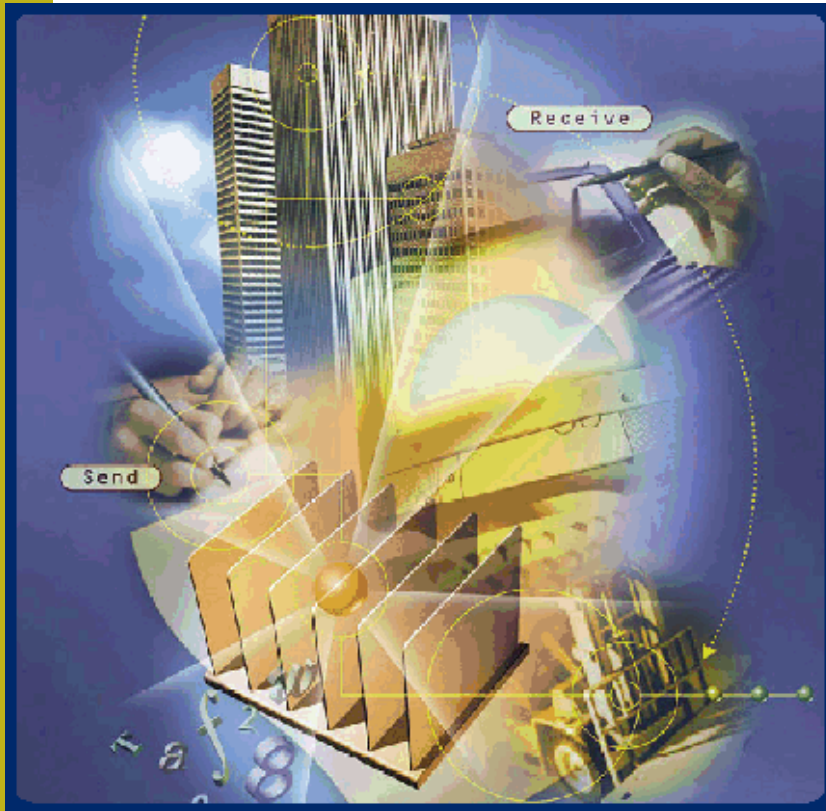
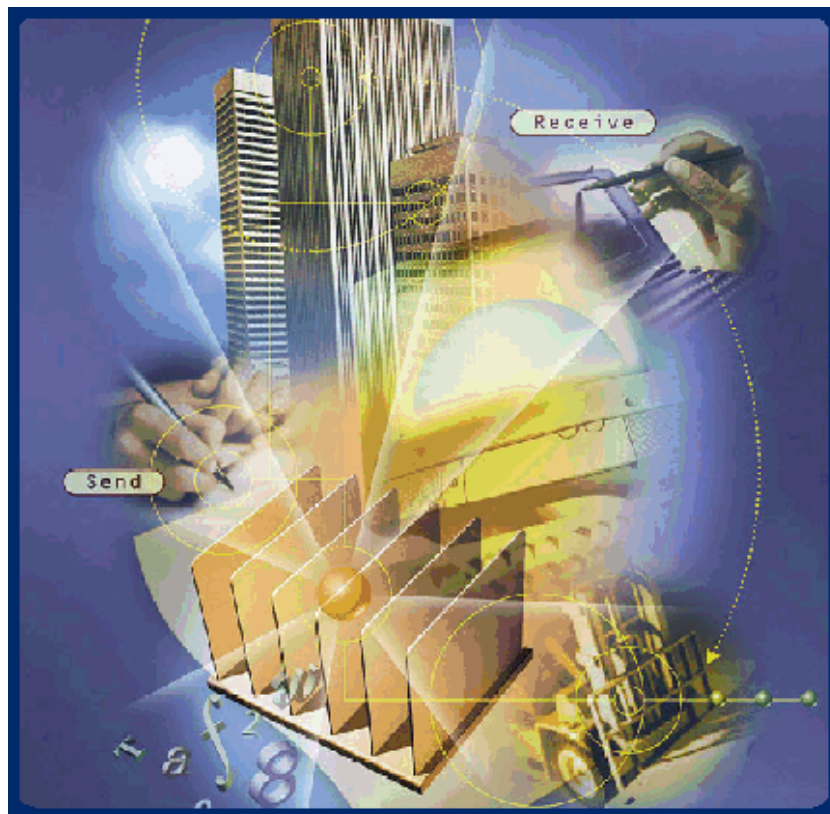


## Overview

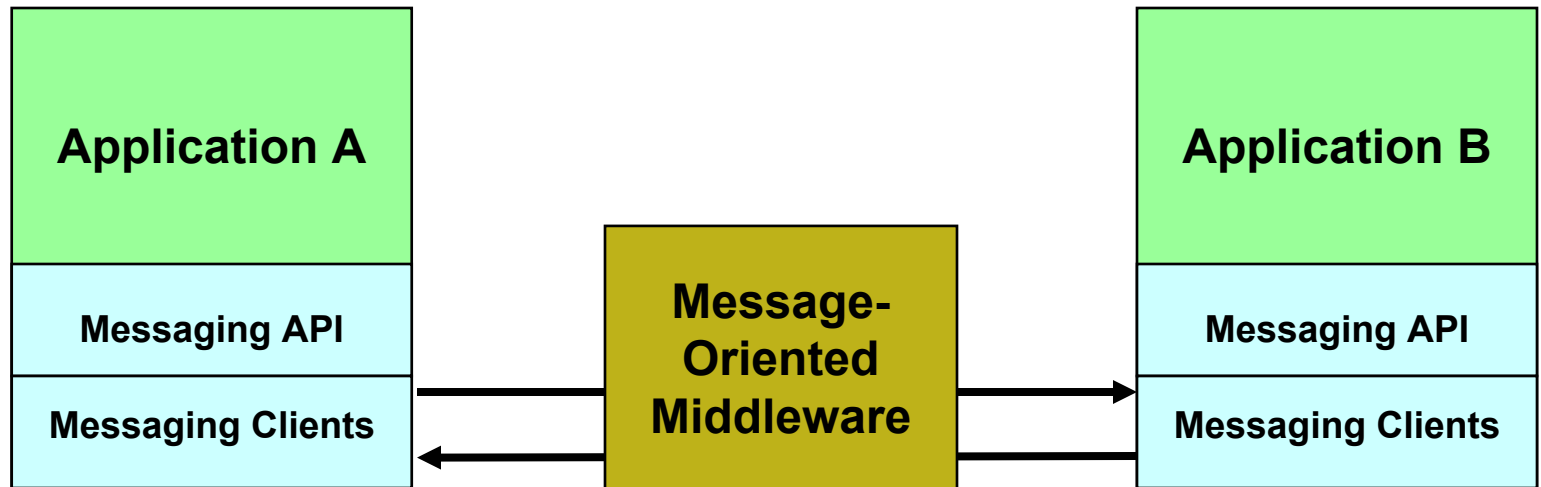


- **Describe the concepts and some practical applications of messaging.**
- **Describe the concepts and basic structure of JMS.**
- **Write simple JMS messaging code using the publish and subscribe and point-to-point domains.**
- **Discuss advanced JMS issues such as guaranteed messaging, transactions, reliability, and recovery.**

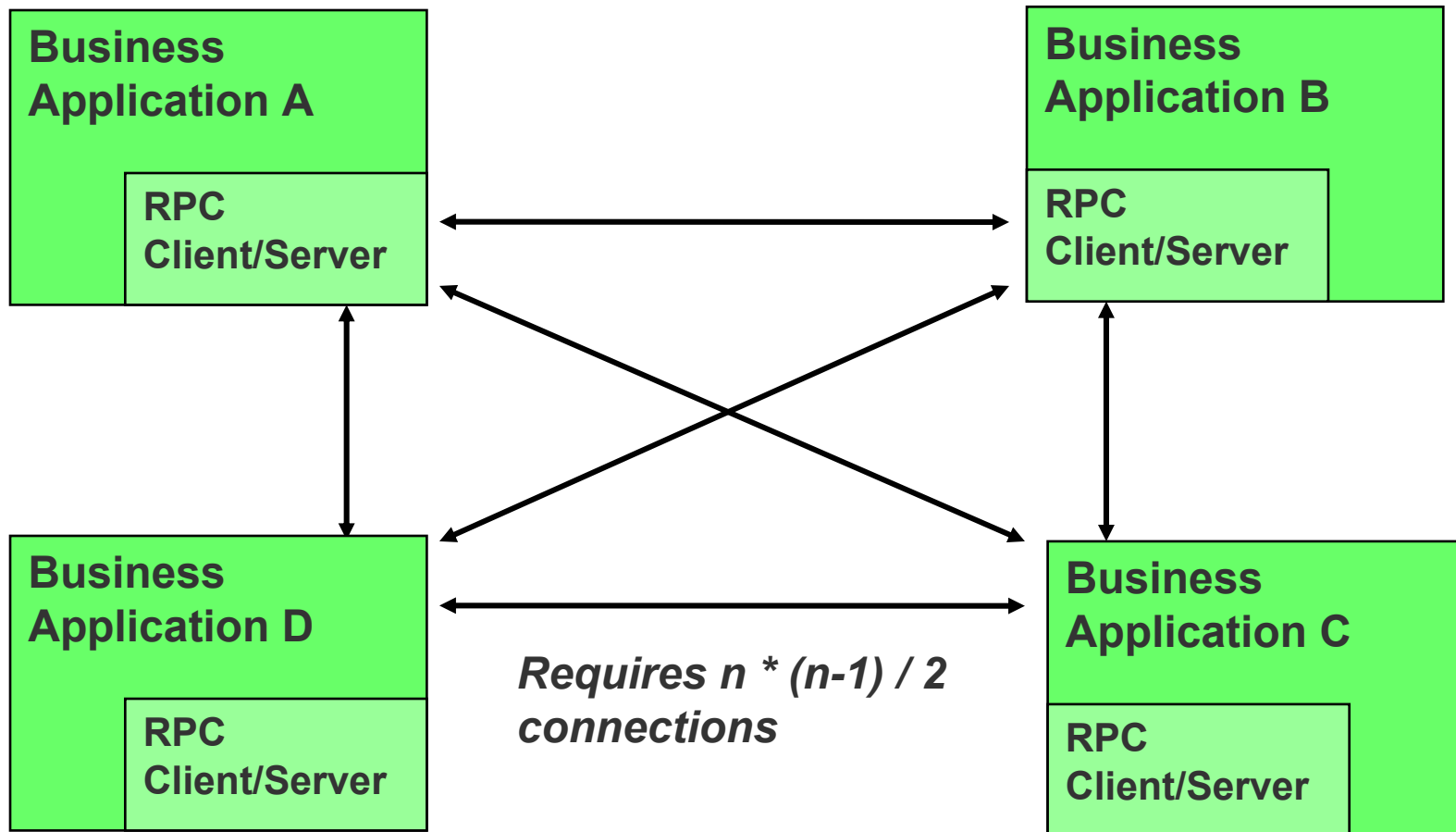
# Part 1: Enterprise Messaging and JMS



# Enterprise messaging

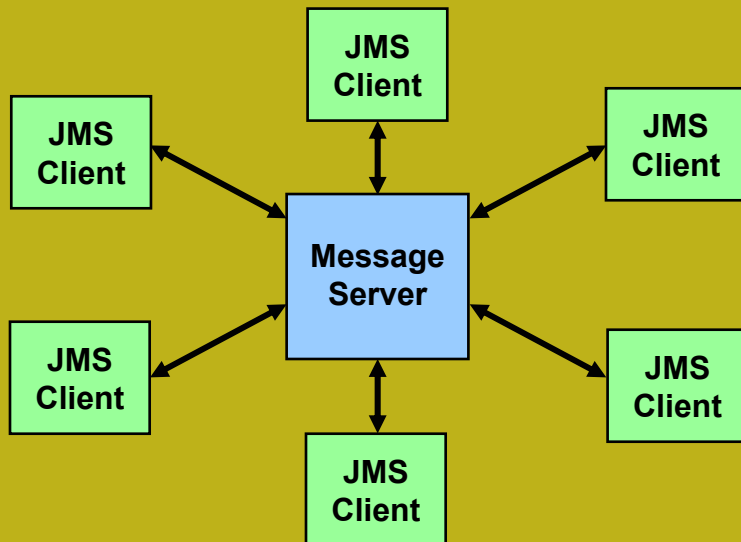


## Tightly Coupled RPC

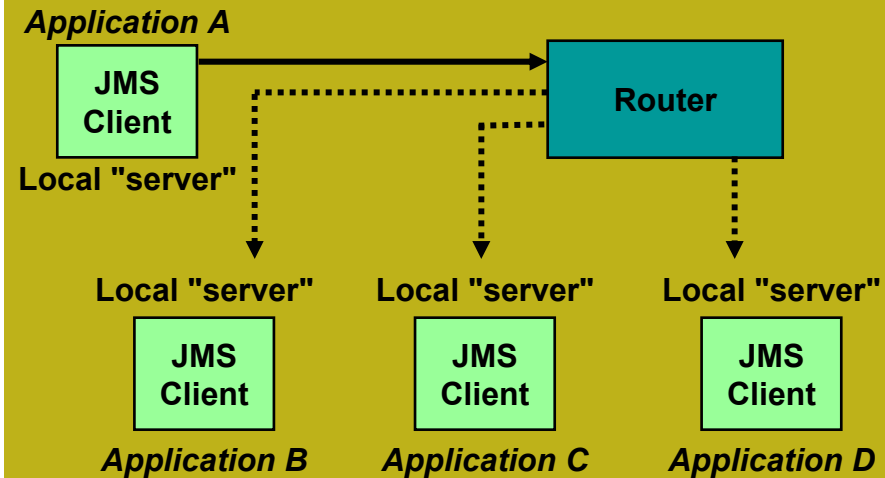


# JMS messaging

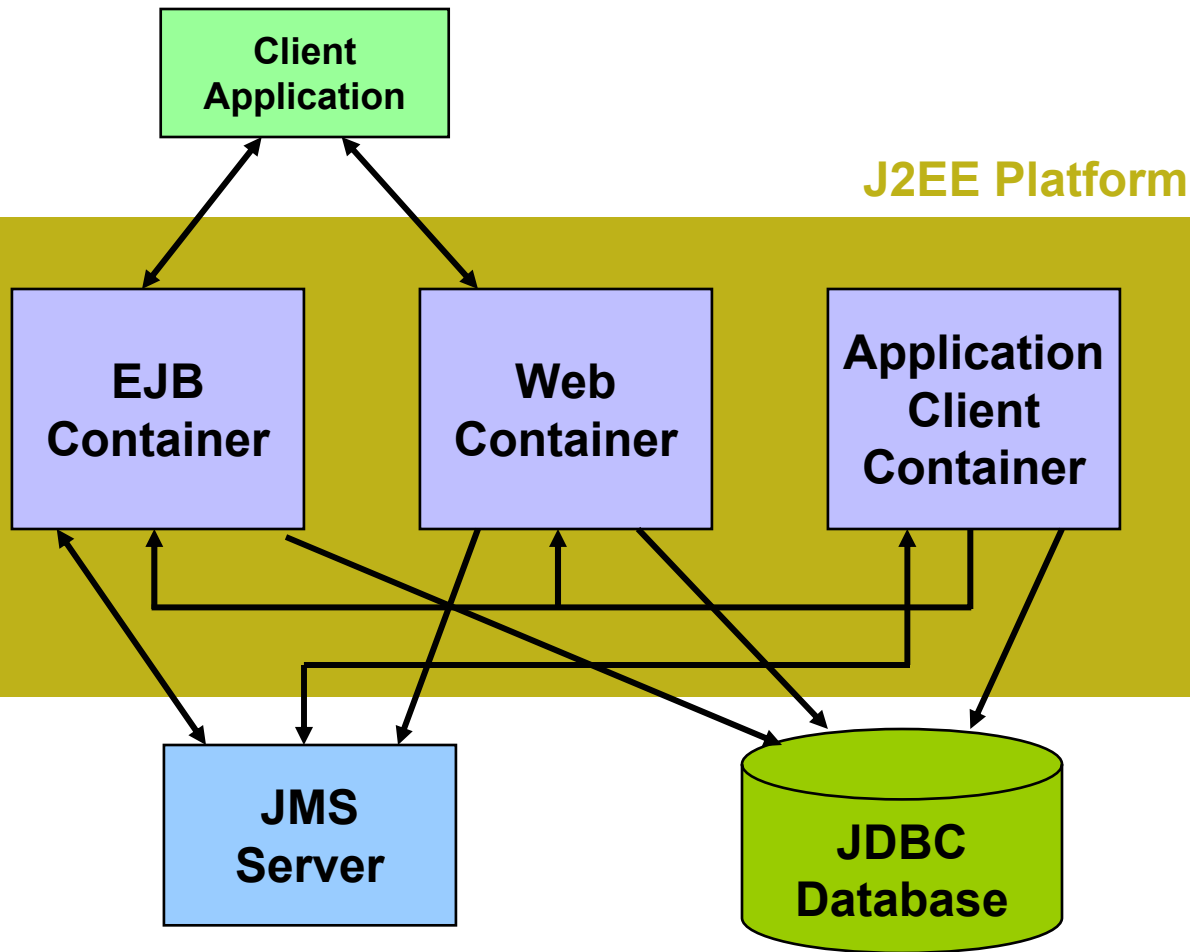
## Centralized message server (hub and spoke)



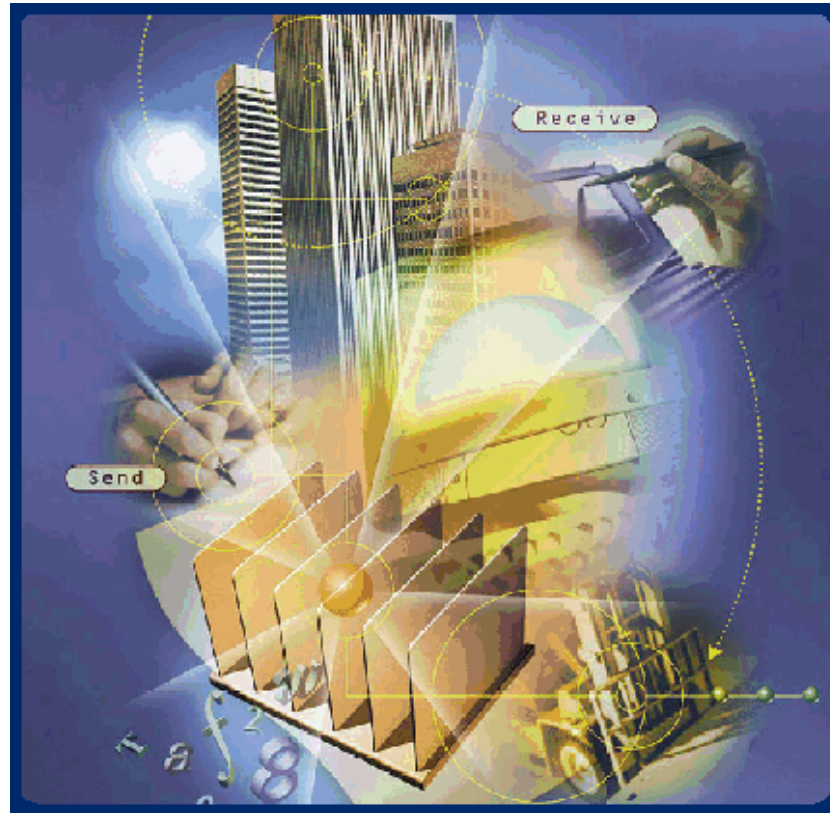
## Decentralized message server (IP multicast)



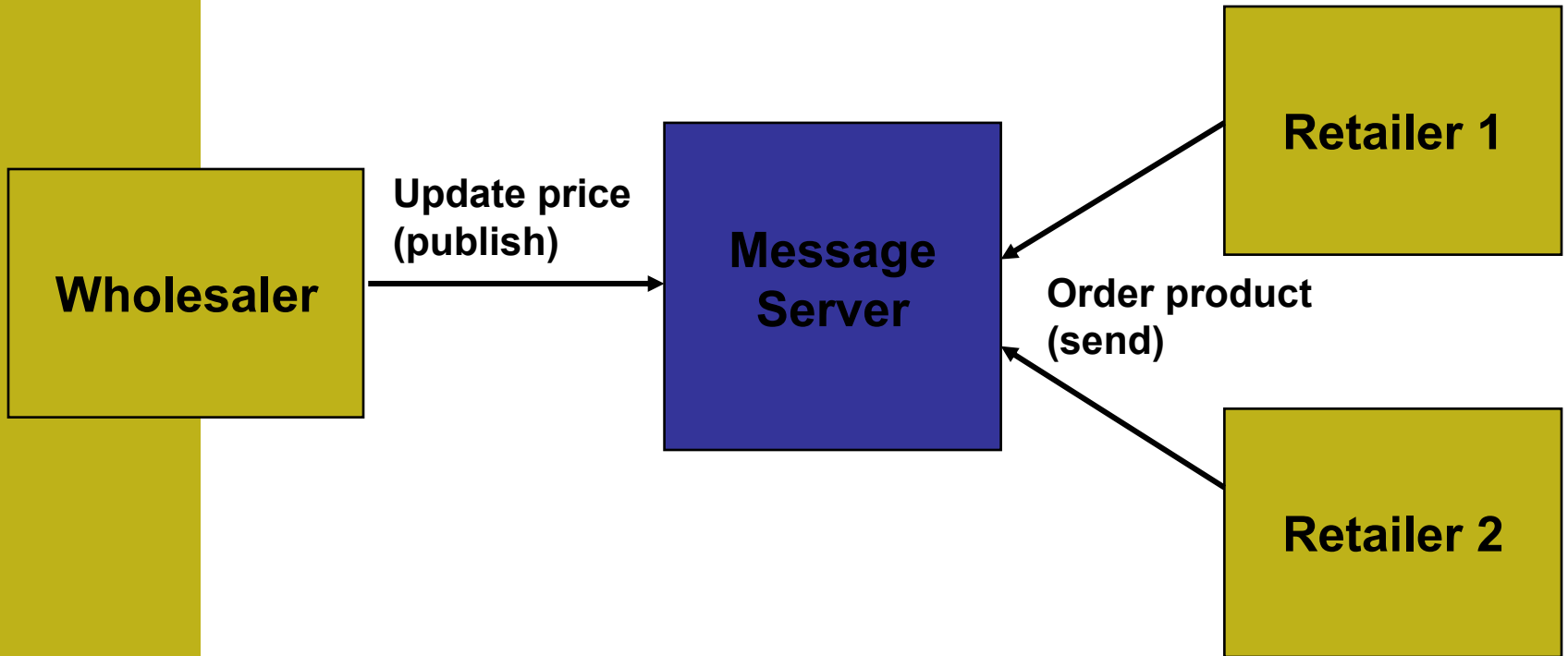
# The J2EE platform



# Part 2: JMS Concepts and Coding



# Wholesaler and retailer





## JMS features

- Standard Java API
- Message delivery modes
- Two messaging models
- Transactions
- Reliability levels
- Persistent messaging

## JMS messaging domains

### Publish and subscribe

- Many consumers per message

### Point-to-point

- One consumer per message

# Publish and subscribe: Supply chain management

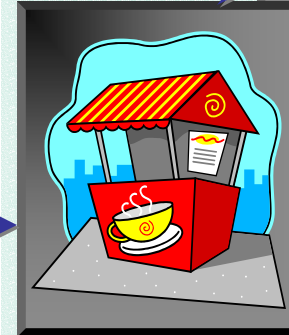
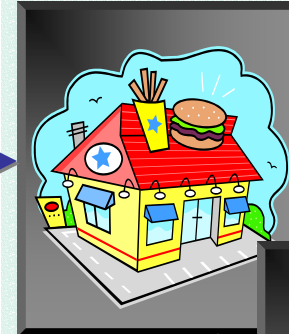
**Publisher**



**Topic**



**Subscriber**



# Point-to-point: Order and fulfillment

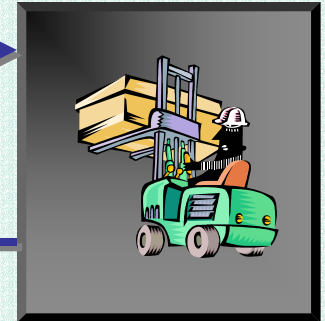
**Sender**



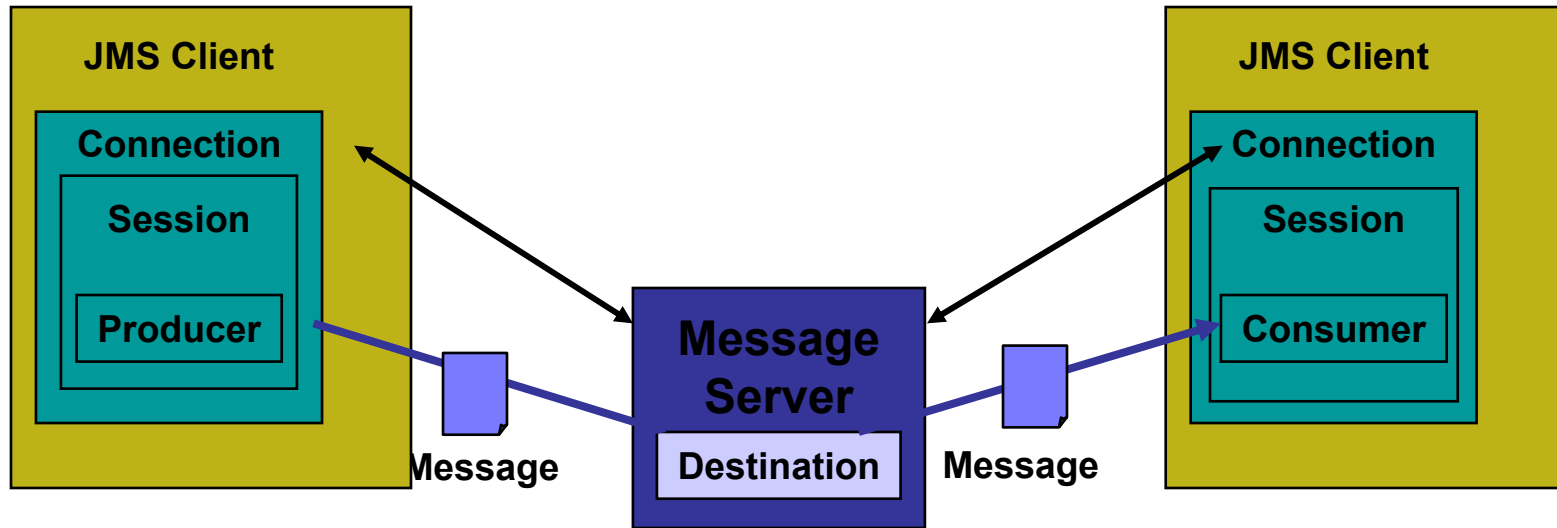
**Queue**



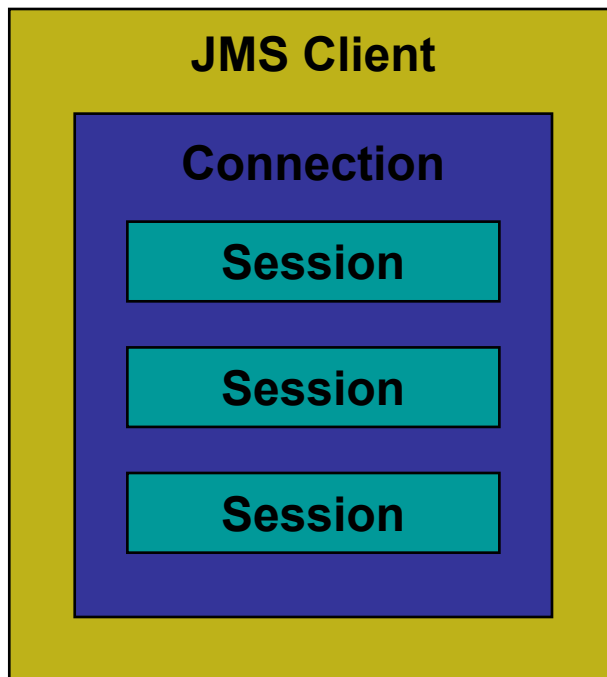
**Receiver**



# JMS components

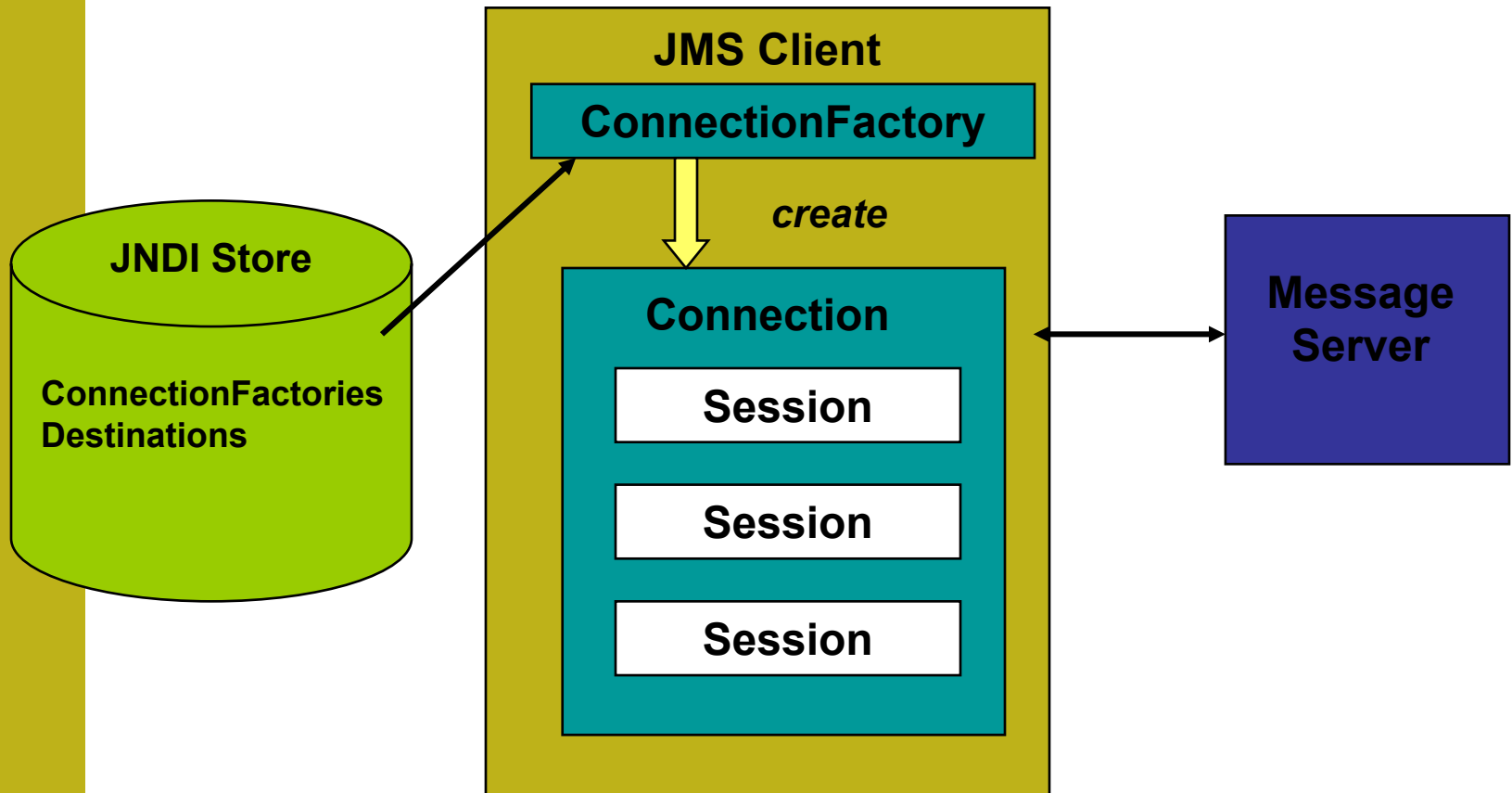


## Connections and sessions



- A connection connects to a message server.
- You can create one or more sessions within a connection.

## Creating connections and sessions

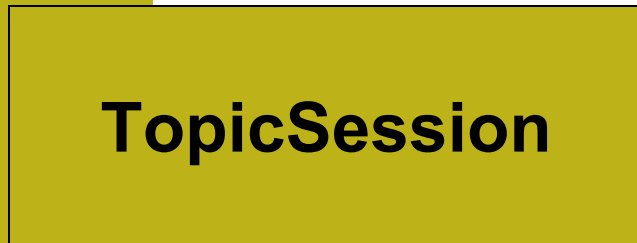


## JMS message types

Message type	Message body
Message	No body
TextMessage	A standard Java string
ObjectMessage	A serializable Java object
MapMessage	A set of name/value pairs where values are Java primitives
StreamMessage	A stream of Java primitives
BytesMessage	A stream of uninterpreted bytes



## Creating a message



`createStreamMessage ( );`



`createTextMessage ( );`



## JMS message headers

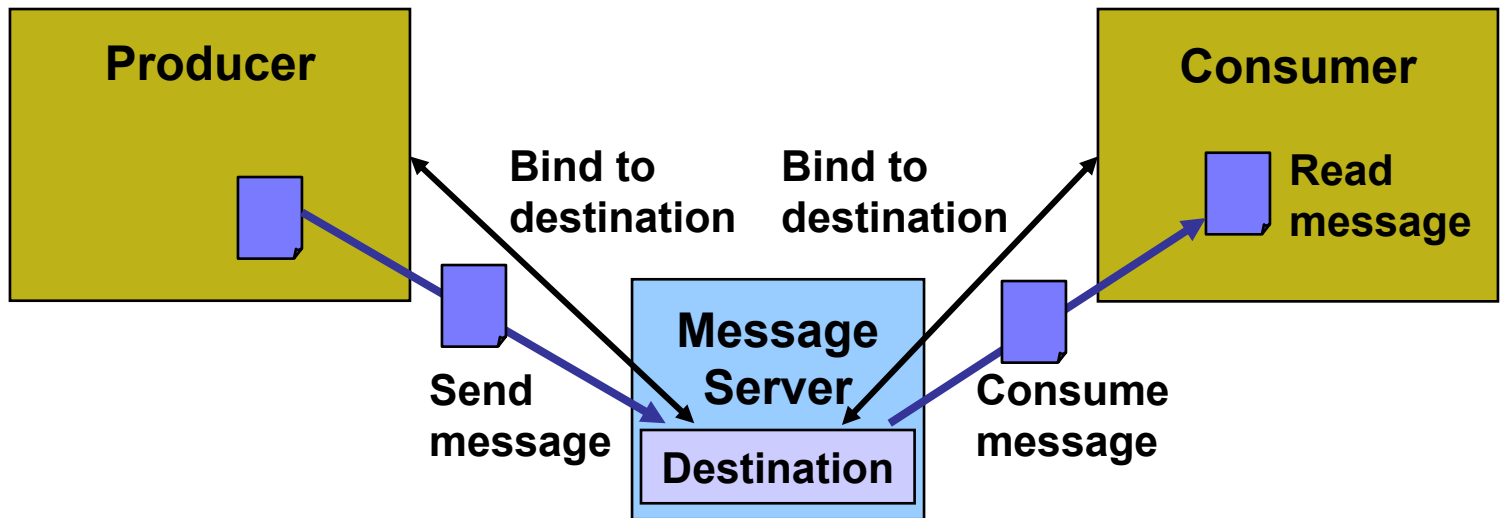
### Automatically assigned headers

- JMSDestination
- JMSDeliveryMode
- JMSMessageID
- JMSTimestamp
- JMSExpiration
- JMSRedelivered
- JMSPriority

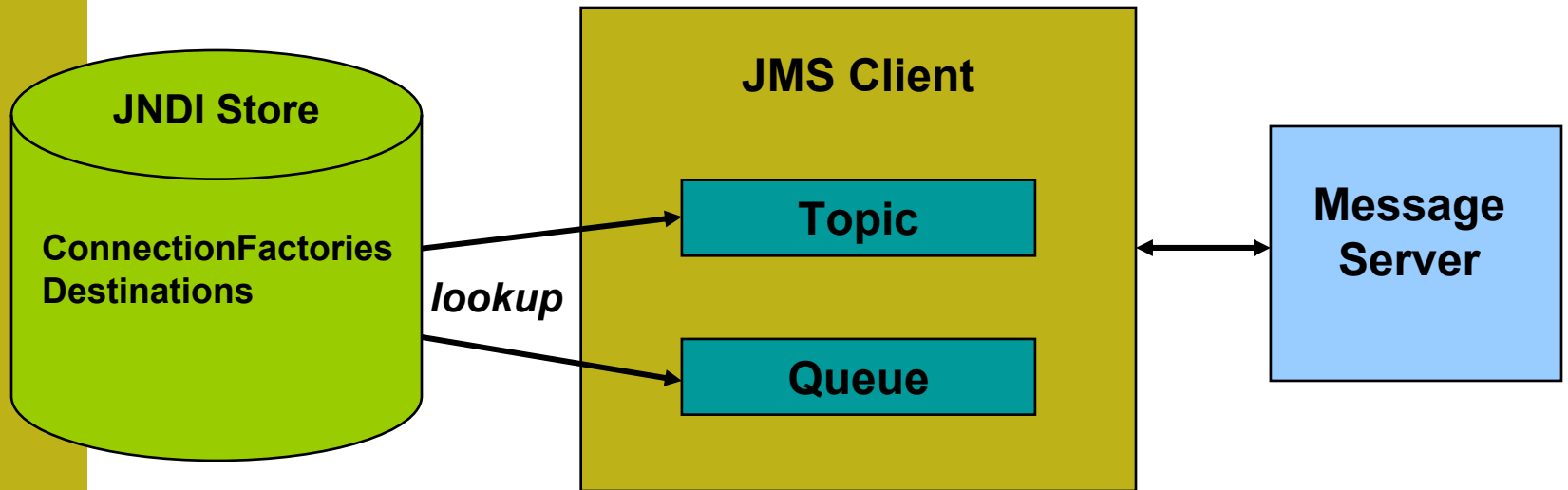
### Developer-assigned headers

- JMSReplyTo
- JMSCorrelationID
- JMSType

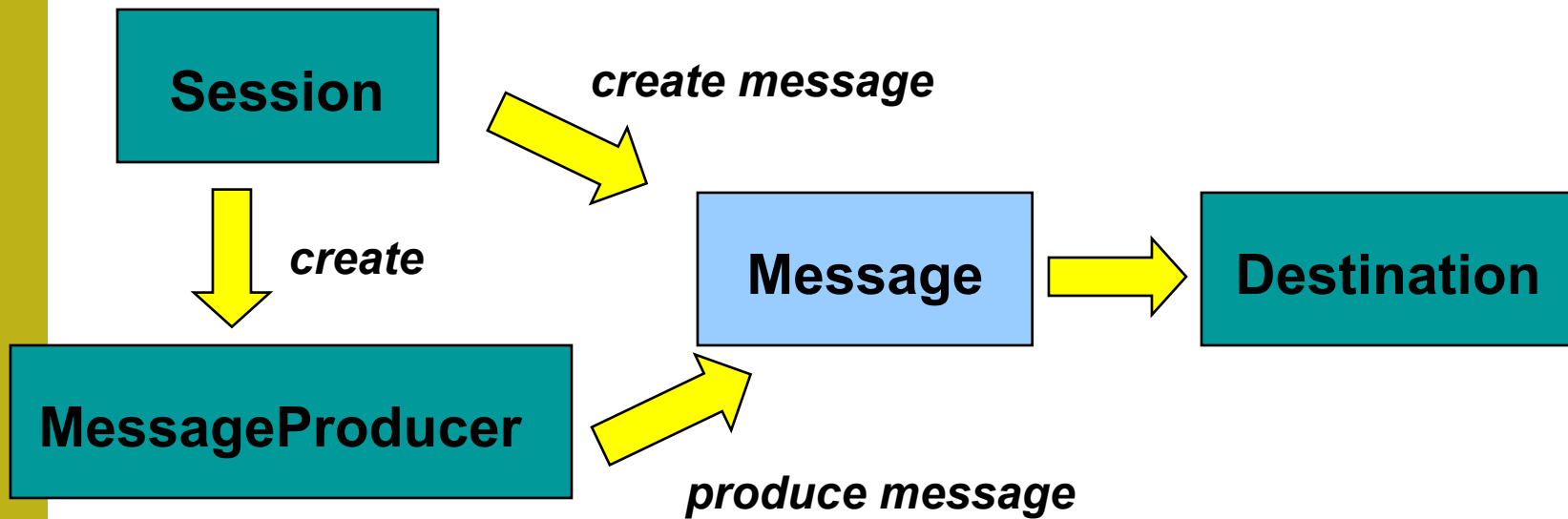
## Producers, consumers, and destinations



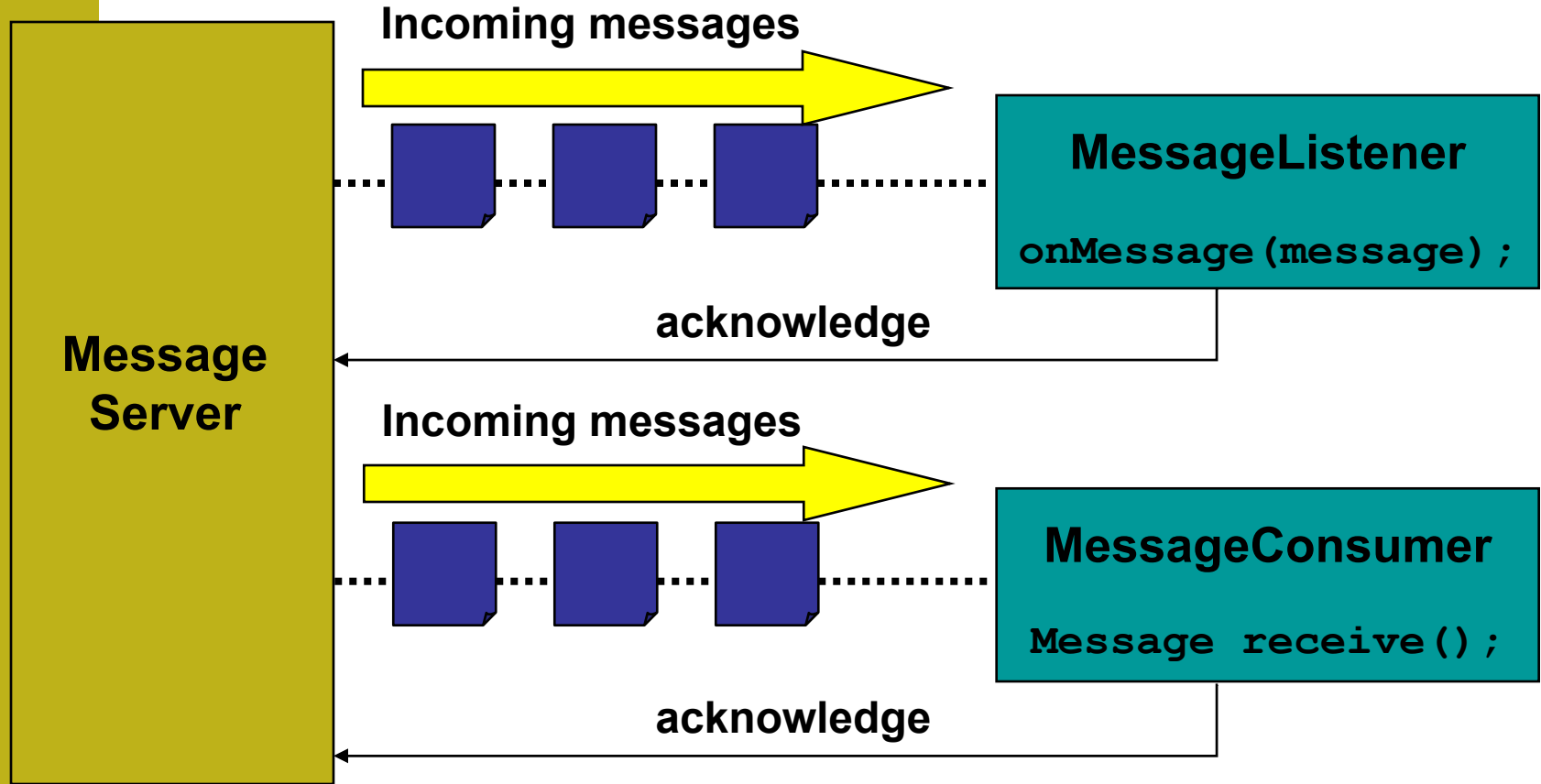
## Creating destinations



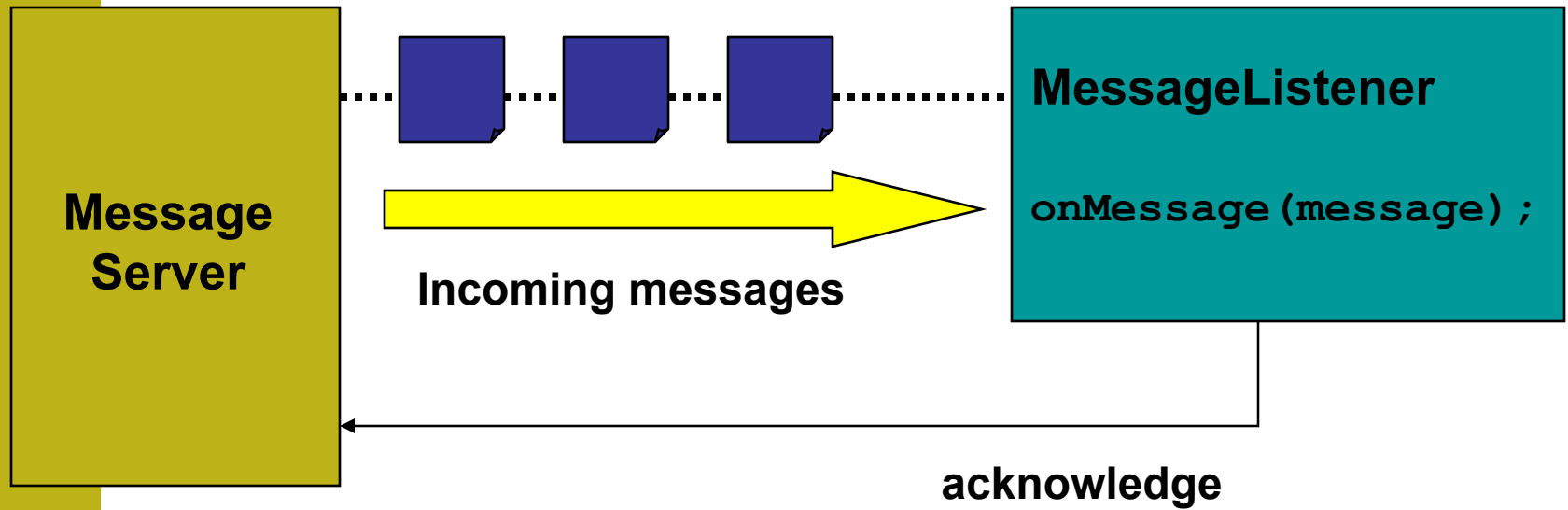
## Producing a message



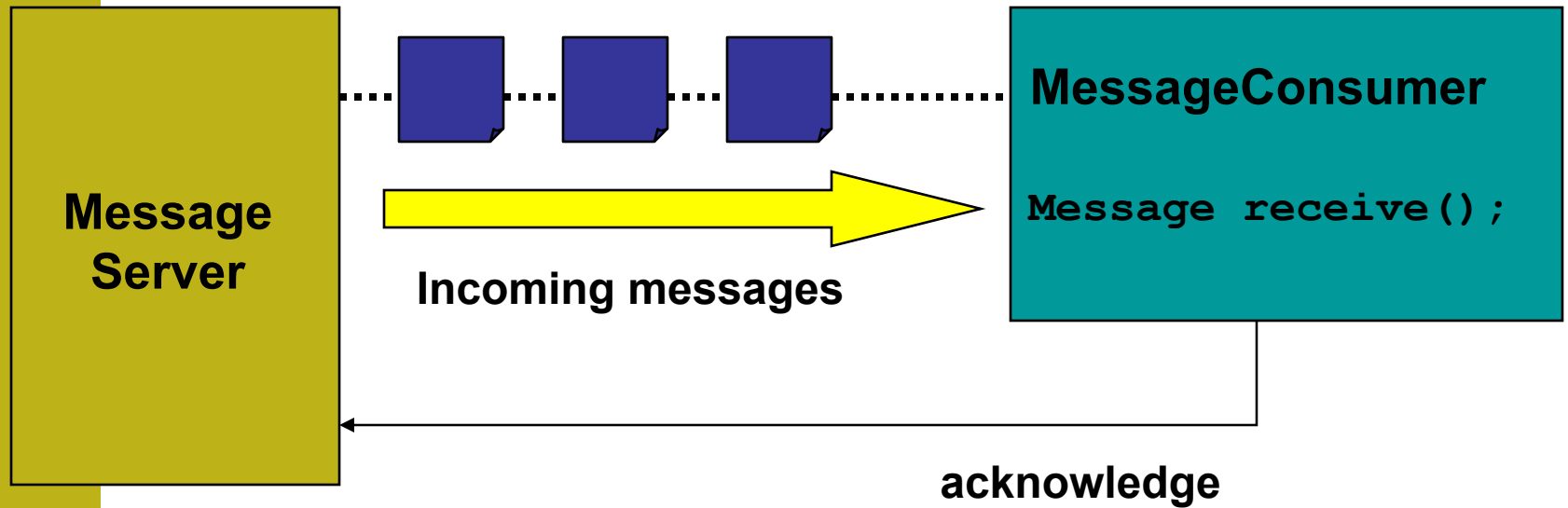
# Consuming messages



## Asynchronous message delivery

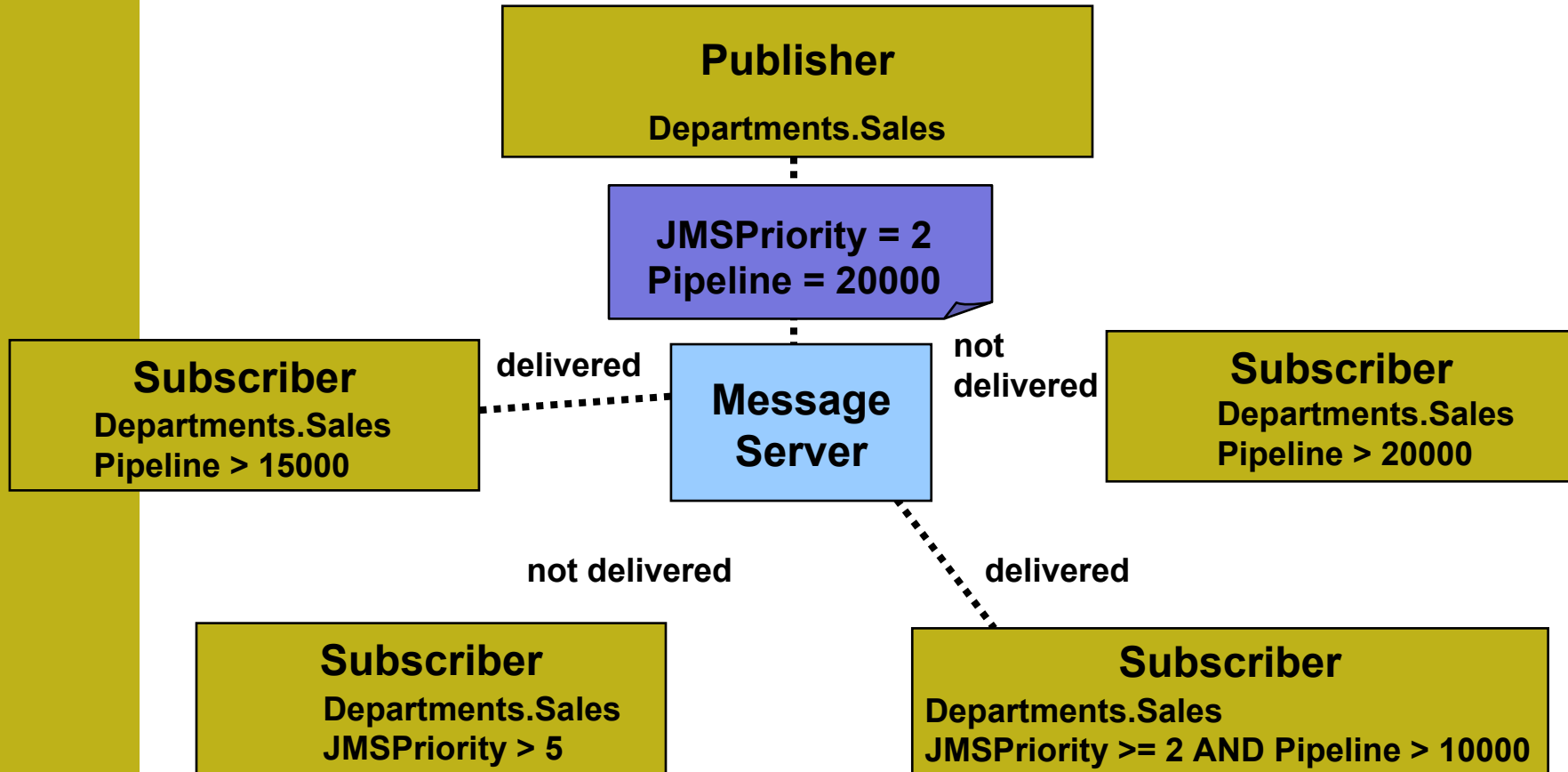


Using receive()

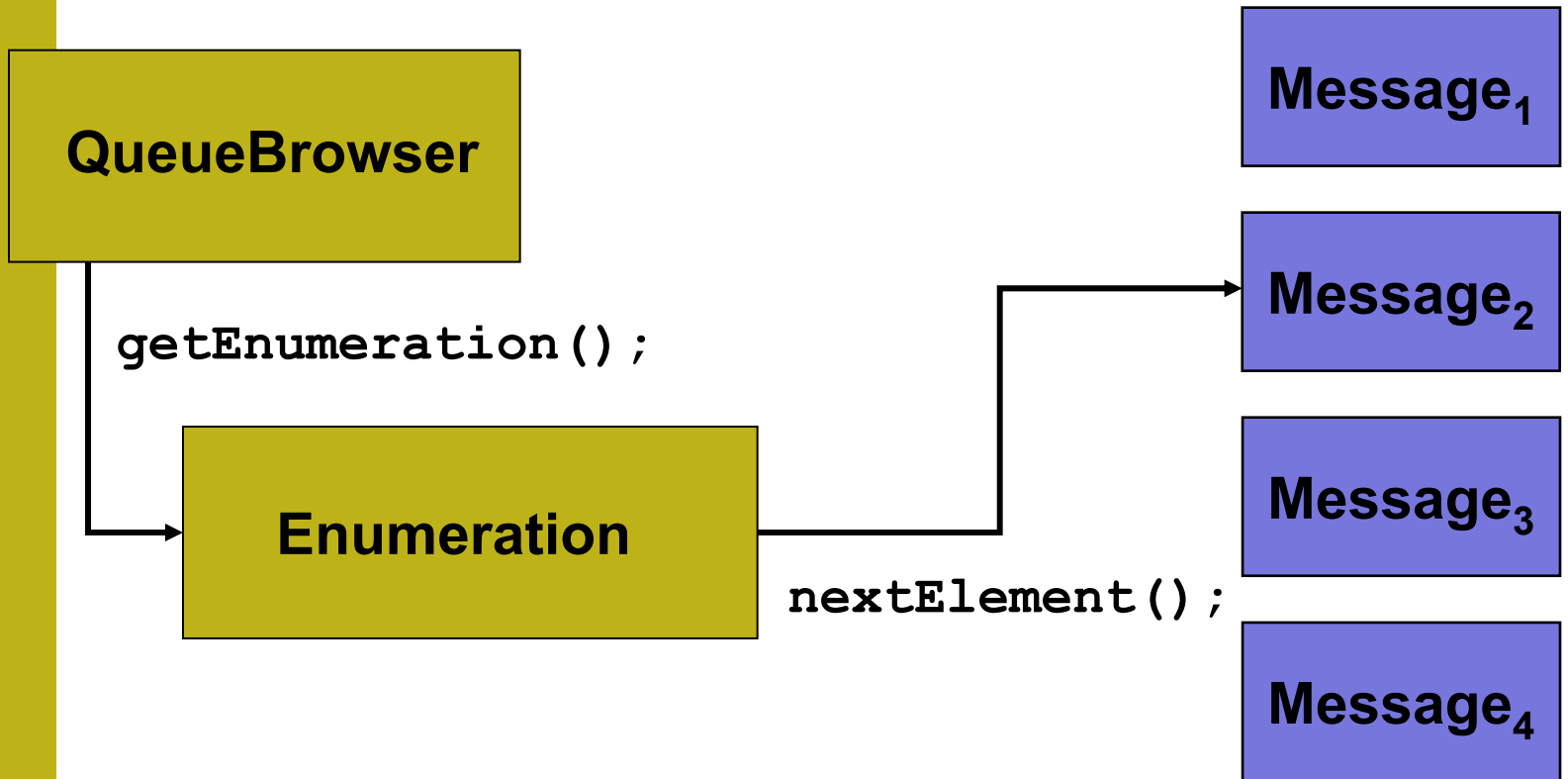




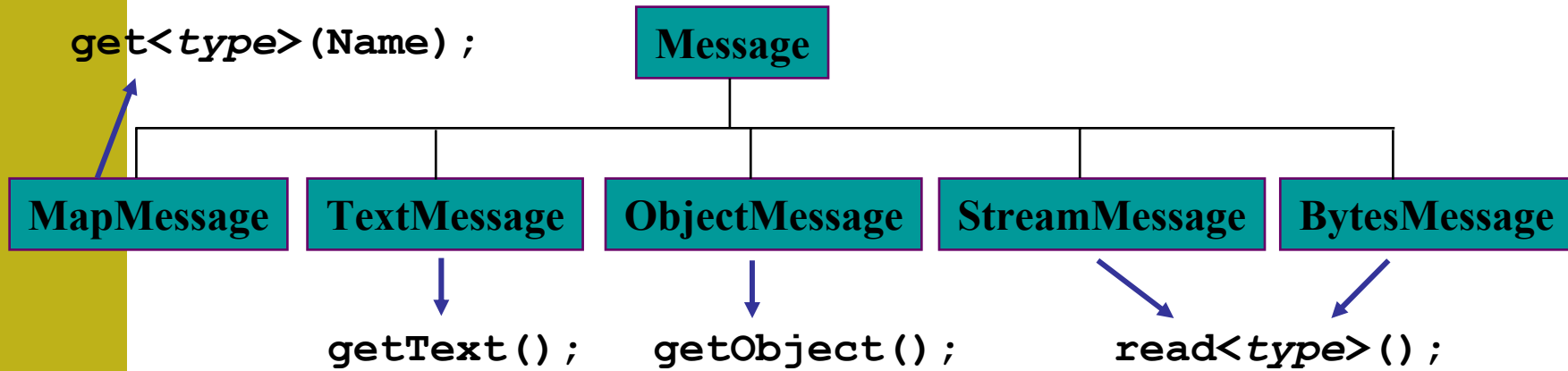
# Filtering with message selector



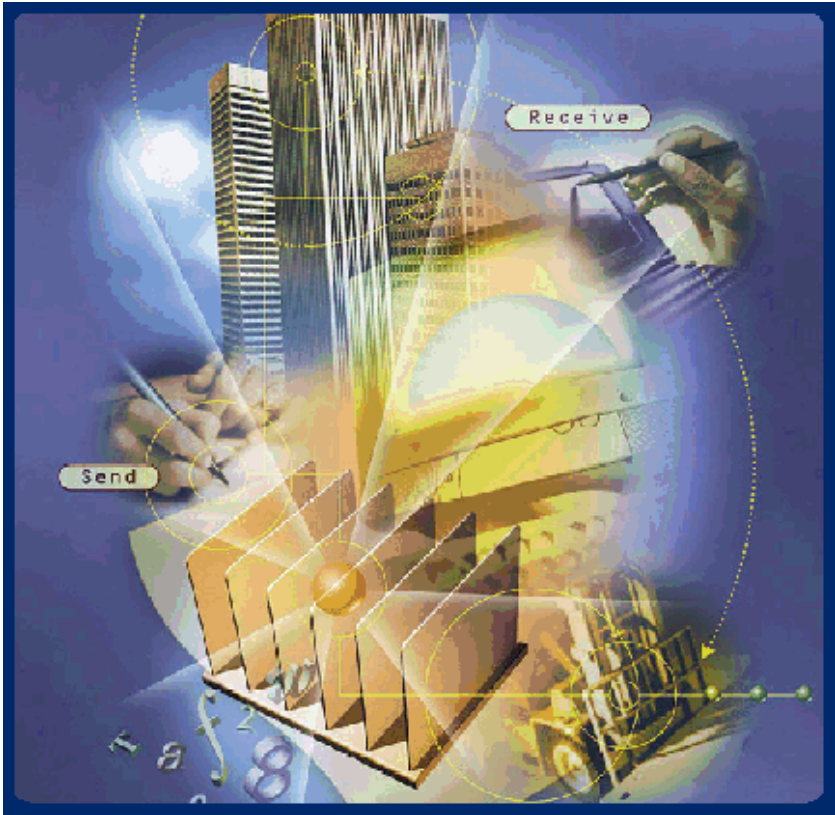
## Browsing a queue



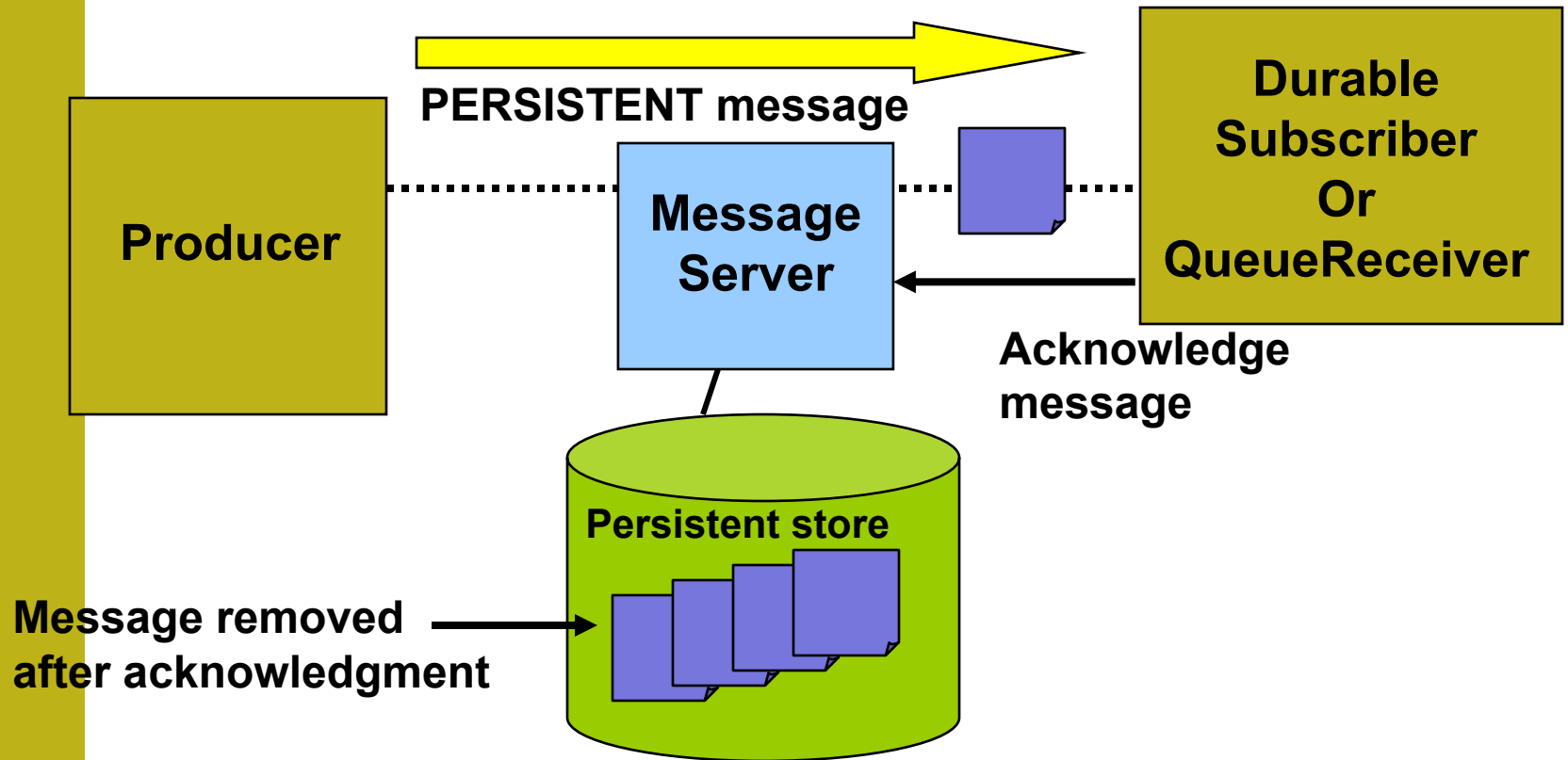
## Accessing message content



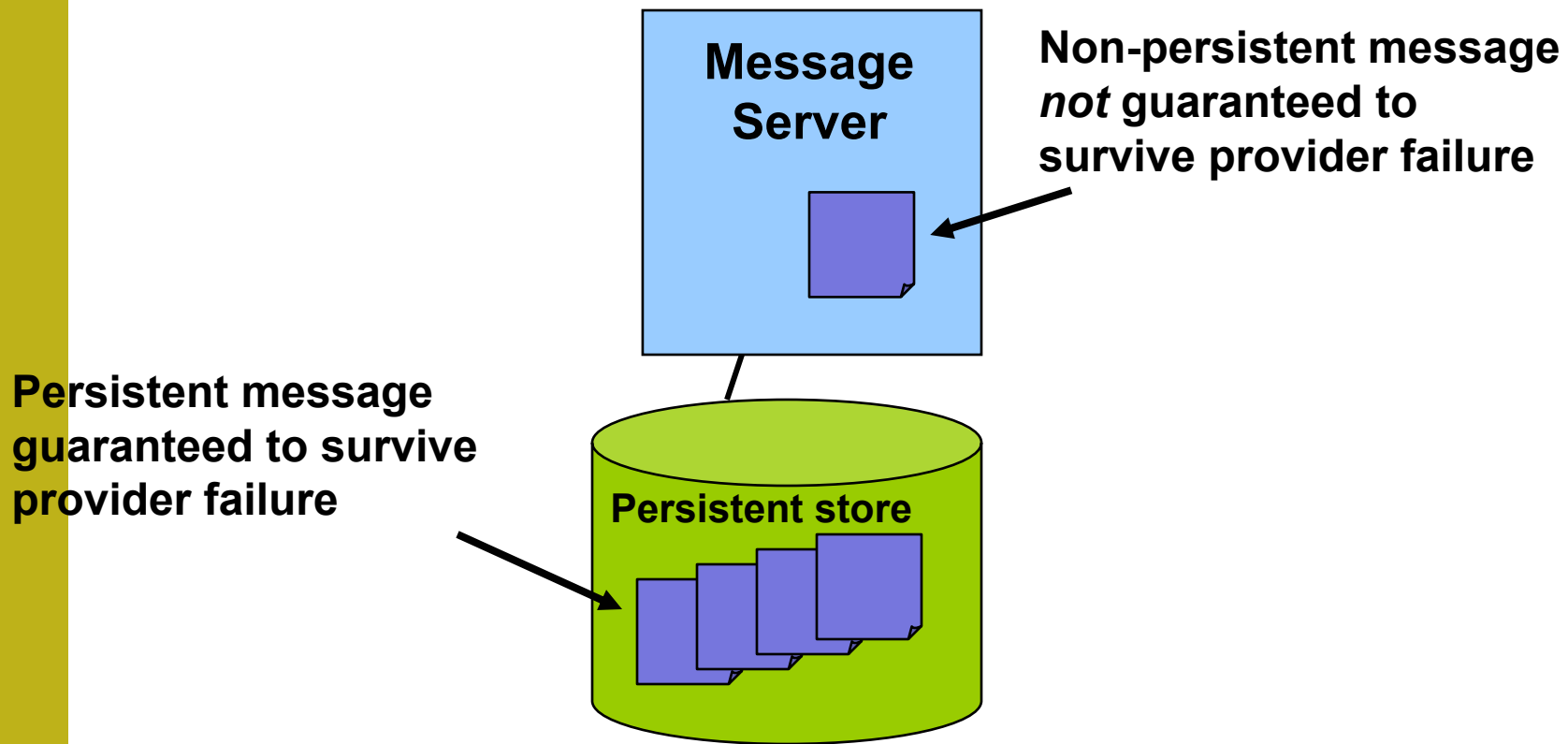
# Part 3: Guaranteed messaging



# Guaranteeing message delivery



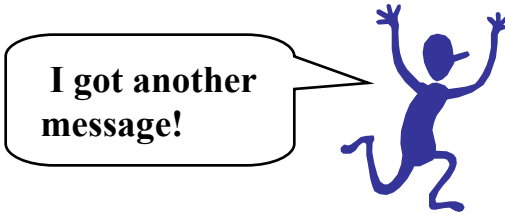
## Persistent and non-persistent messages



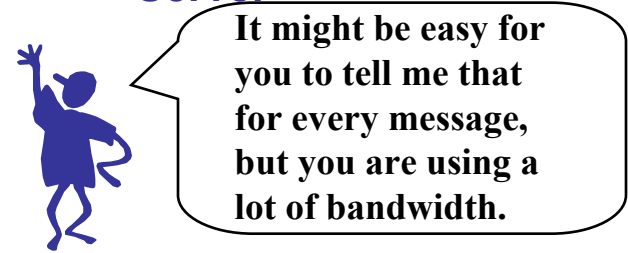
# Acknowledgement modes

**AUTO**

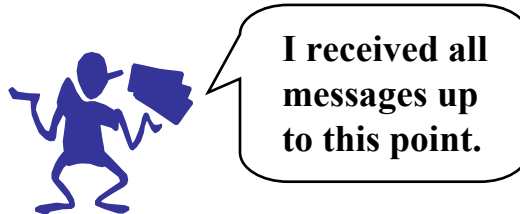
**Consumer**



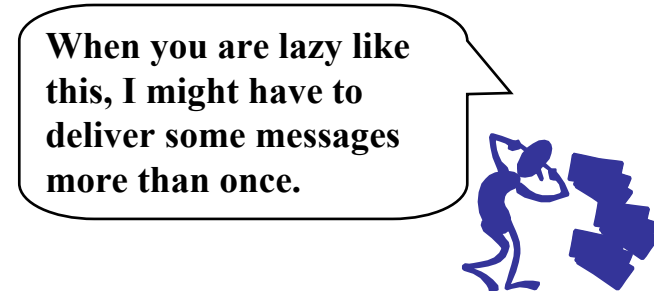
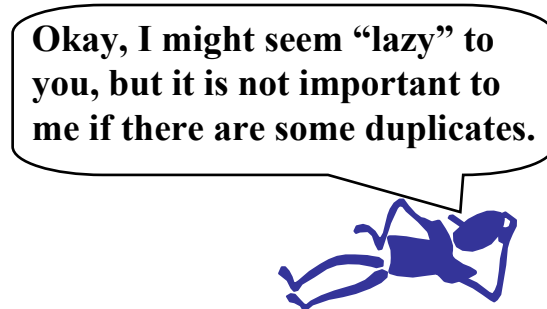
**Server**



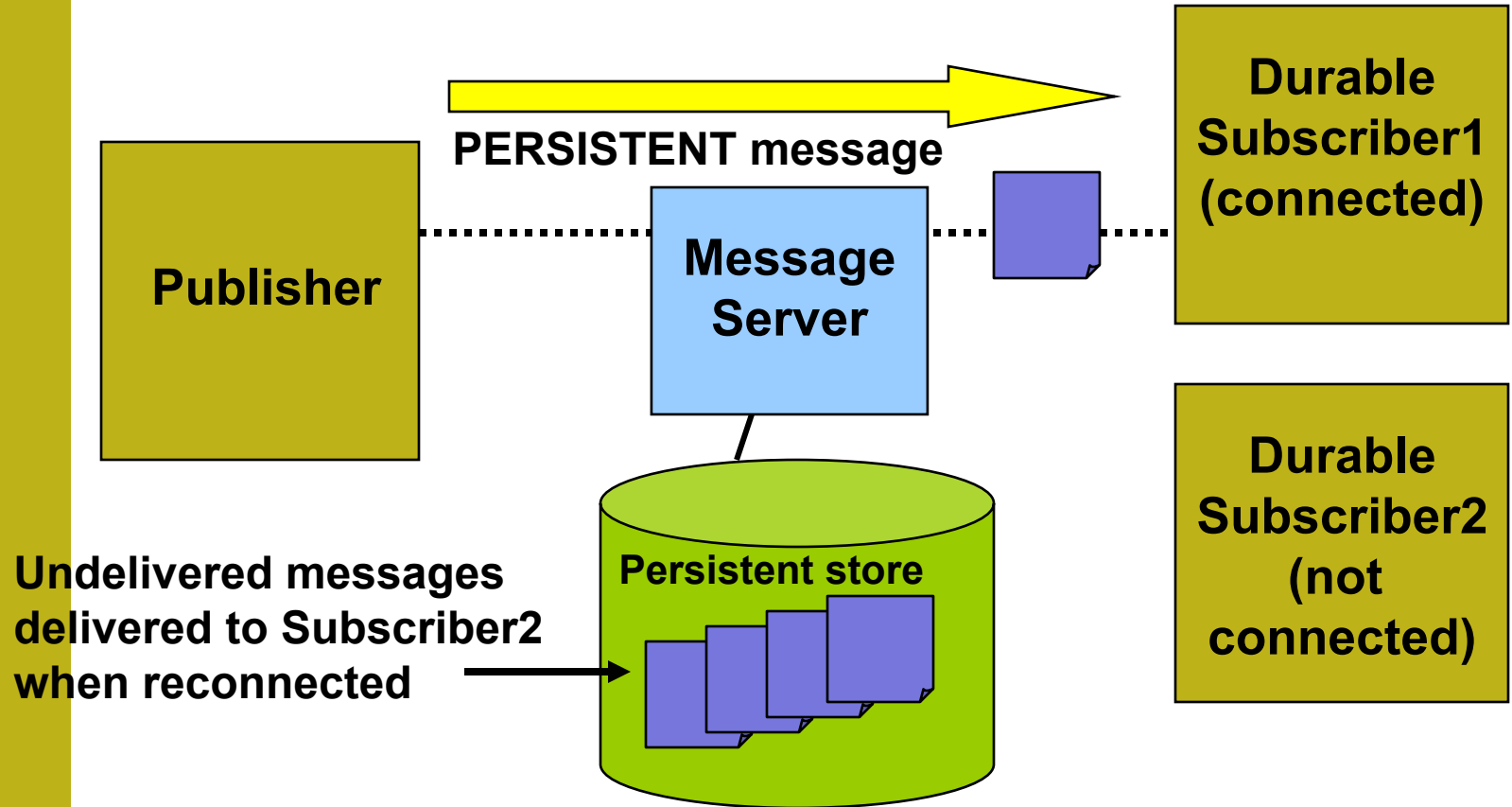
**CLIENT**



**DUPS\_OK**



## Durable subscribers





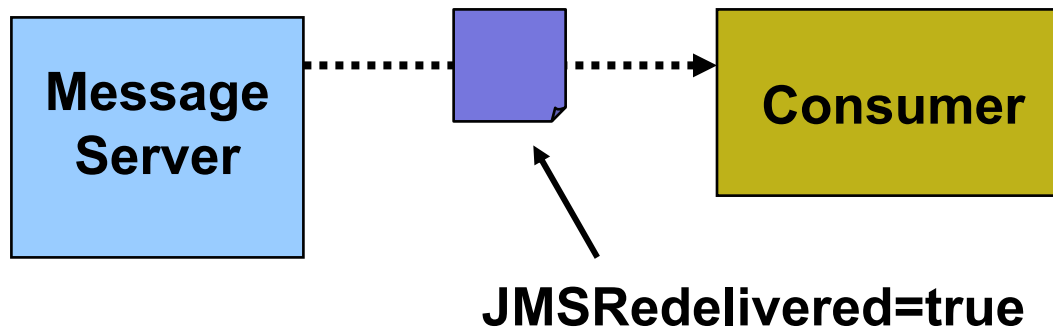
## Redelivered messages

**Before message server receives acknowledgement:**

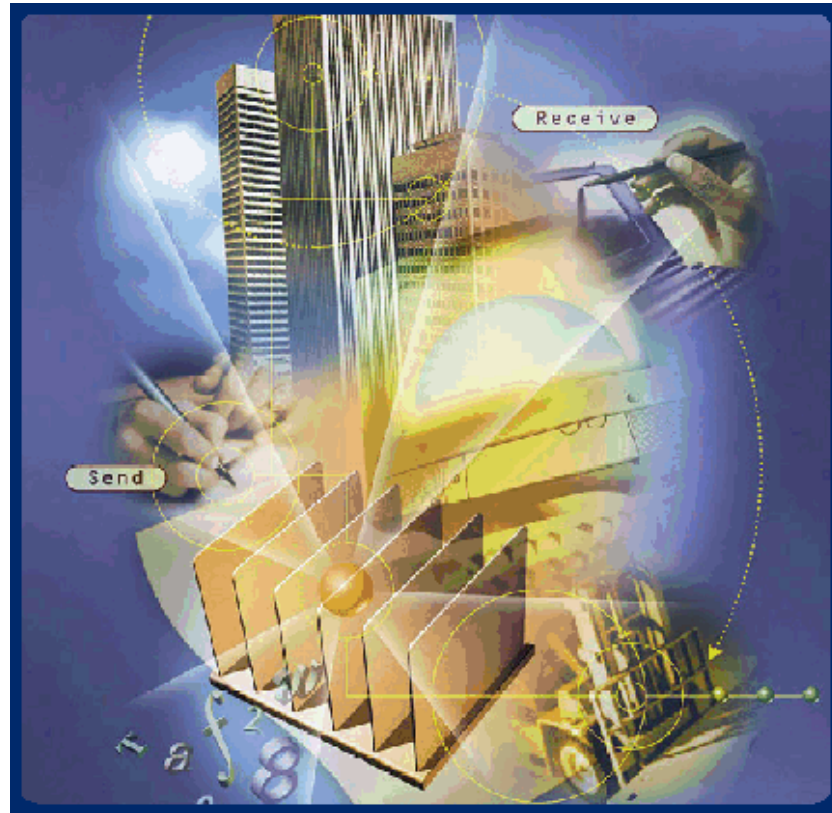
➤ Message server has failed and restarted

OR

➤ Consumer has failed and restarted

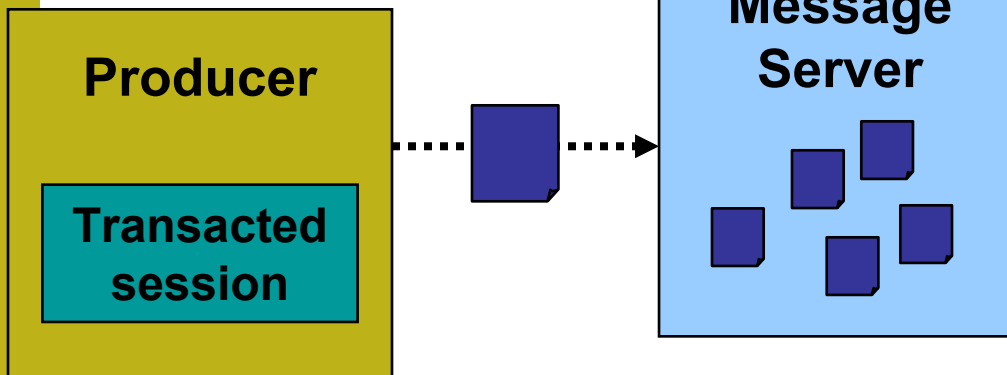


# Part 4: Transactions and recovery

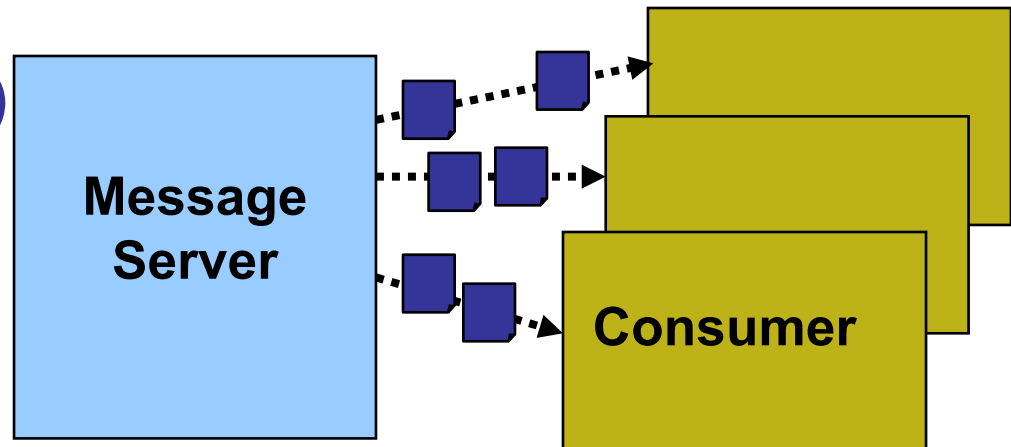


## Transacted session: Producer

**Before commit( )**

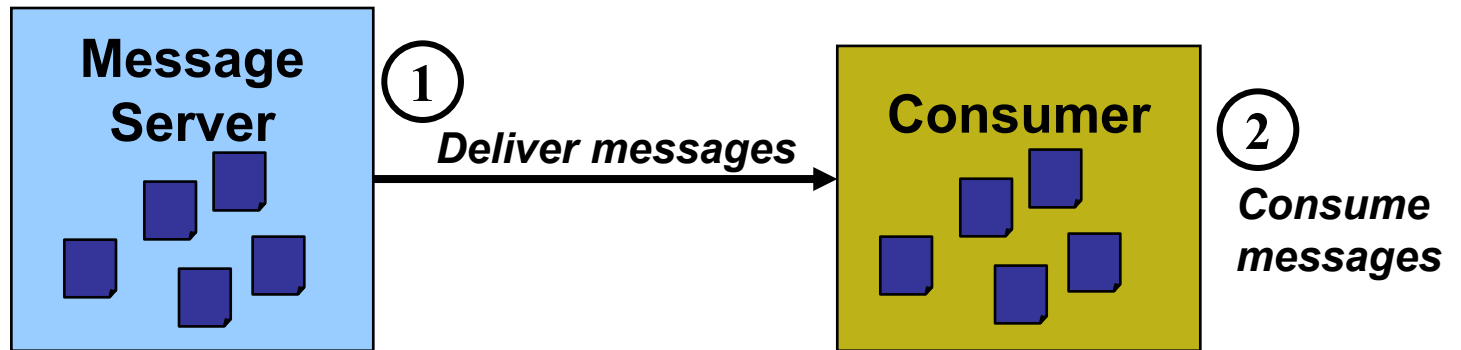


**Upon commit( )**

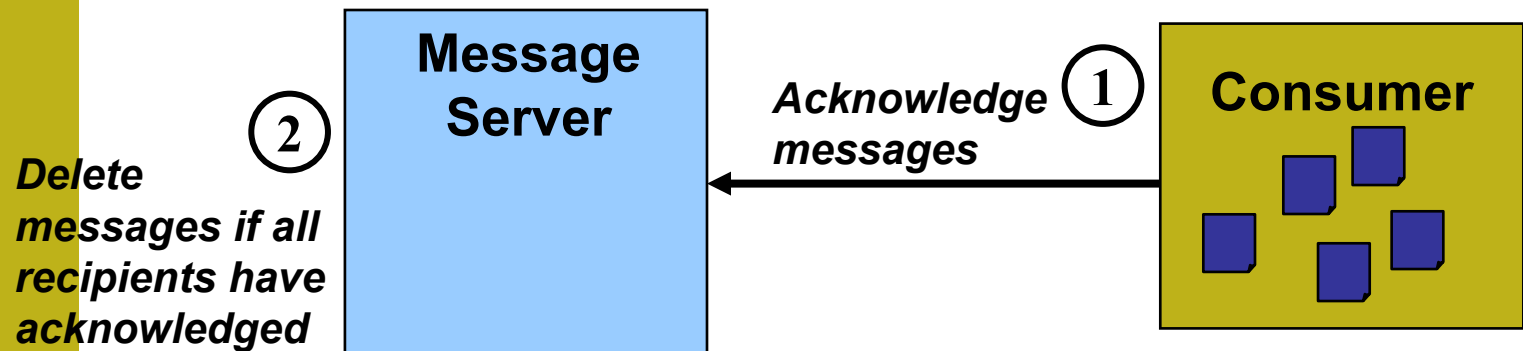


## Transacted session: Consumer

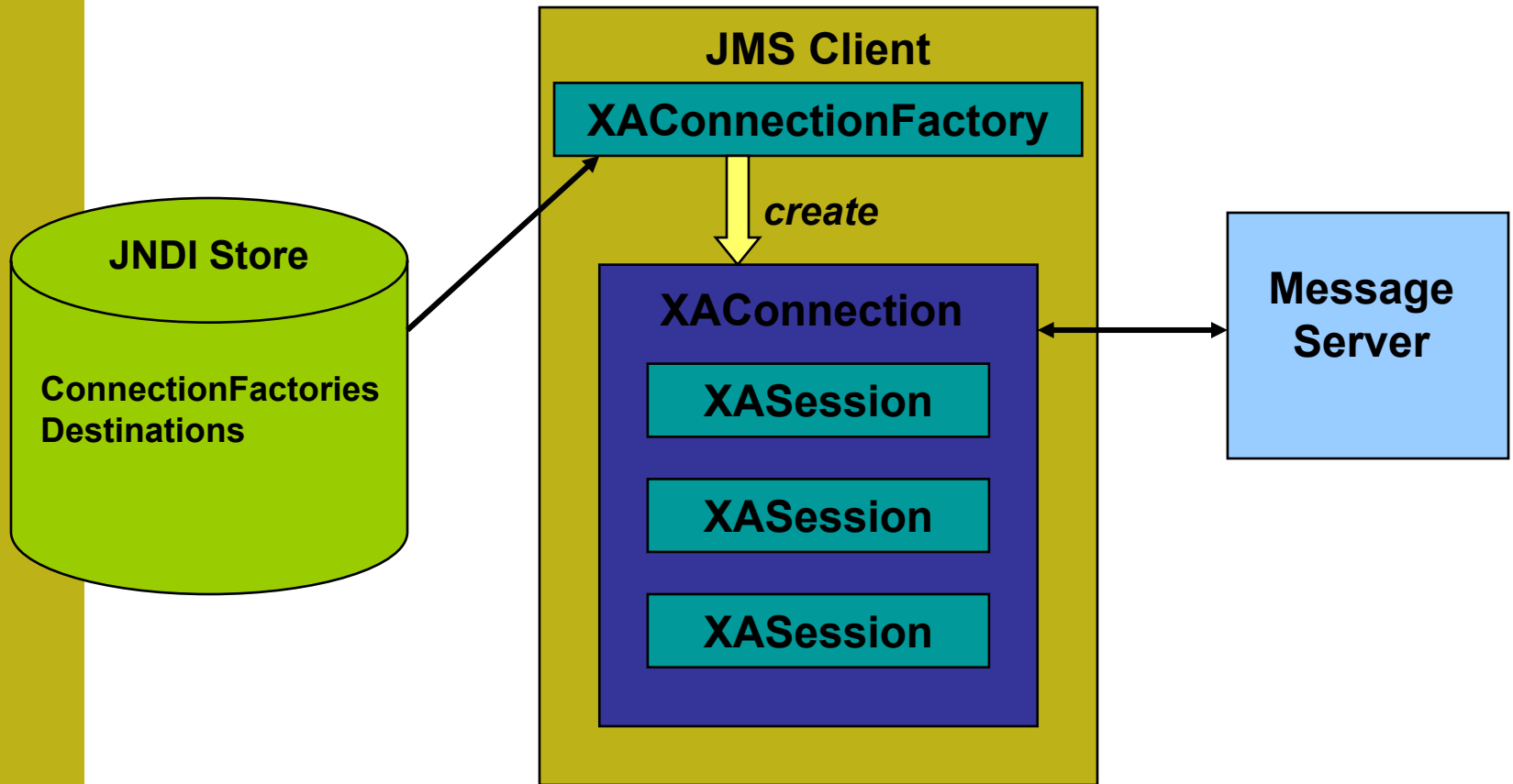
### Before commit()



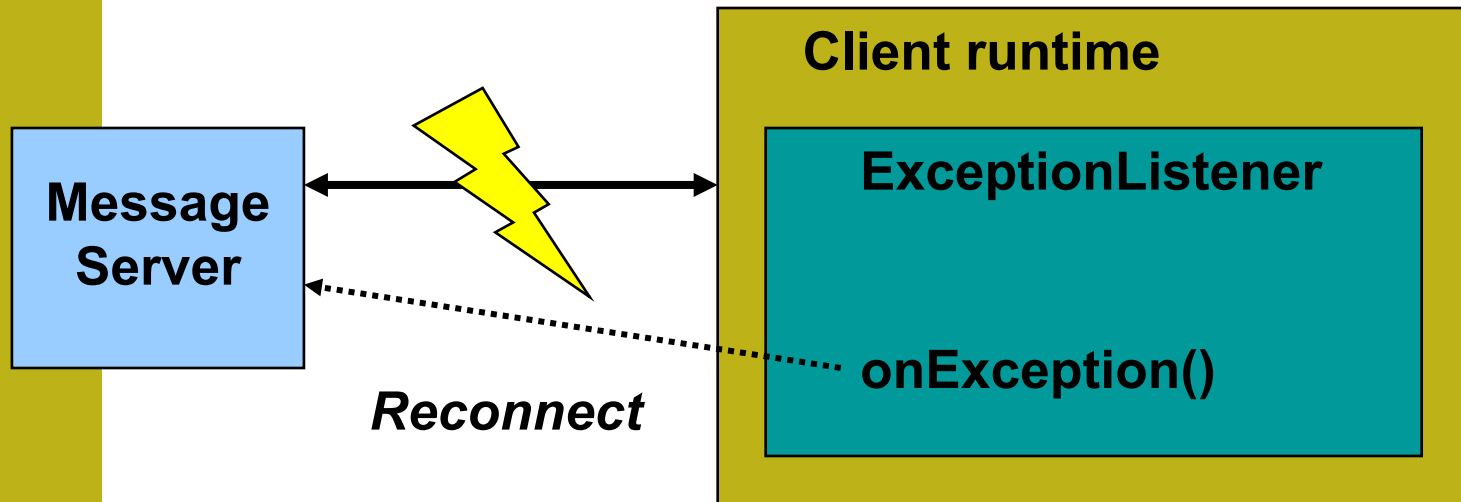
### Upon commit()



## Distributed transaction support



# Recovering from a lost connection



# Understanding the Java Message Service: Conclusion

