Designing Your JMS Solution for Production

Grant Little

www.grantlittle.me

Overview

Efficient Use of Resources

High Availability & Throughput

Message Ordering

Error & Exception Handling

Message Selectors

Synchronous Messaging

Dead Letter Queues

JEE and some frameworks like Spring can take care or help with some of these concerns

Efficient Use of Resources

Caching of Resources

Cache Resources Where Appropriate

Connections

Sessions

Consumers

Producers

Cache Resources

```
private Connection conn;

private Session session;

ConnectionFactory connFactory = ….

conn = connFactory.createConnection();

session = conn.createSession(false,
   Session.AUTO_ACKNOWLEDGE)
```

Reuse Sessions

```
private Session session;


session = connection.createSession(true,
    Session.AUTO_ACKNOWLEDGE)

MessageConsumer consumer = session.createConsumer(…)

MessageProducer producer = session.createProducer(…)
```

Reuse Consumers/Producers

```
MessageConsumer consumer = session.createConsumer(…);
Message msg1 = consumer.receive();
Message msg2 = consumer.receive();


MessageProducer producer = session.createProducer(…);
producer.send(textMessage1);
producer.send(textMessage2);
```

Consider Pooling Libraries
like
Apache Commons Pool

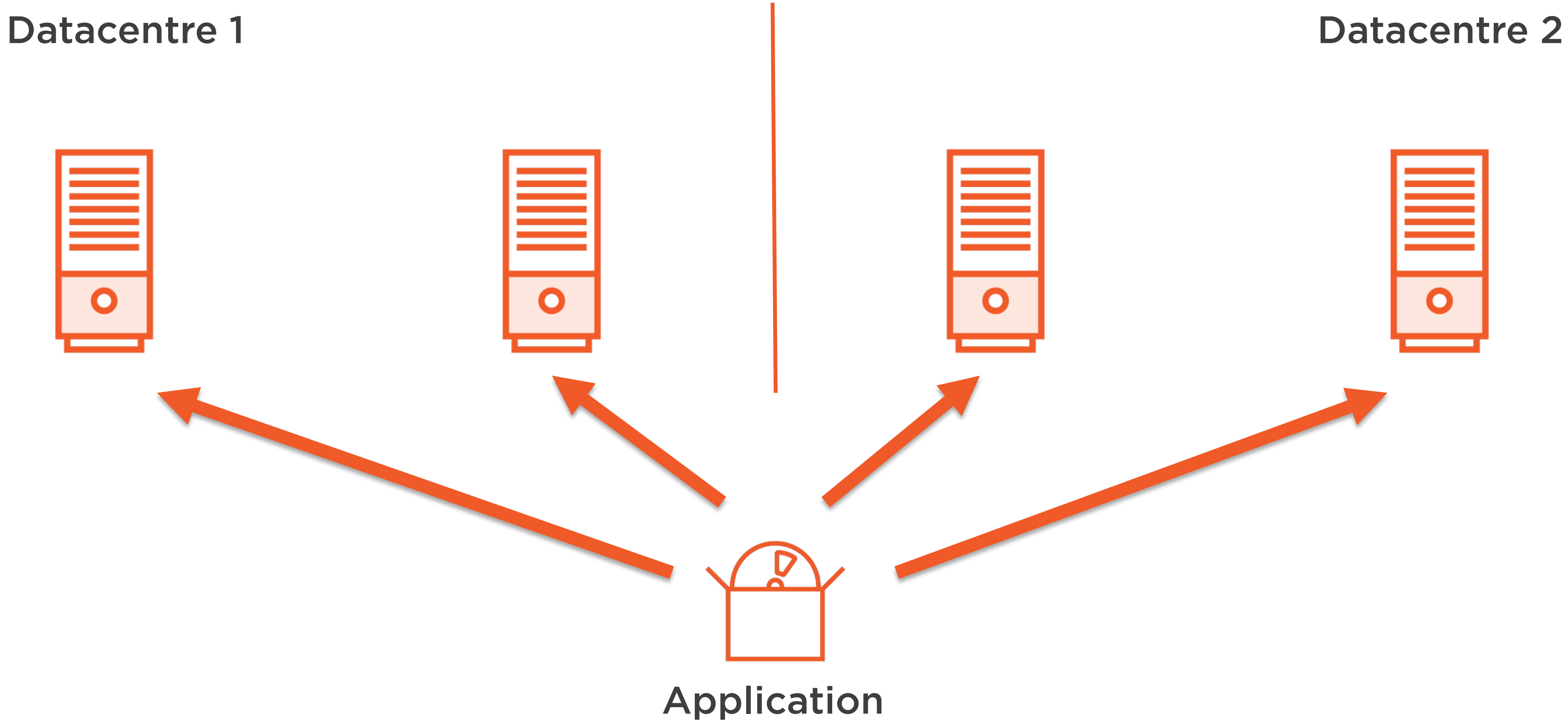
High Availability and Throughput

Good HA Architecture

Clustered

Reliable & Fast Failover

Architecture



Pros & Cons

- Load Balanced
- Scalable
- Failover

- Possible Waste of Some Resources
- Still Need to Consider Deployment

To save on resources, it’s possible to use multiple consumers per JVM

Spring DefaultMessageListenerContainer

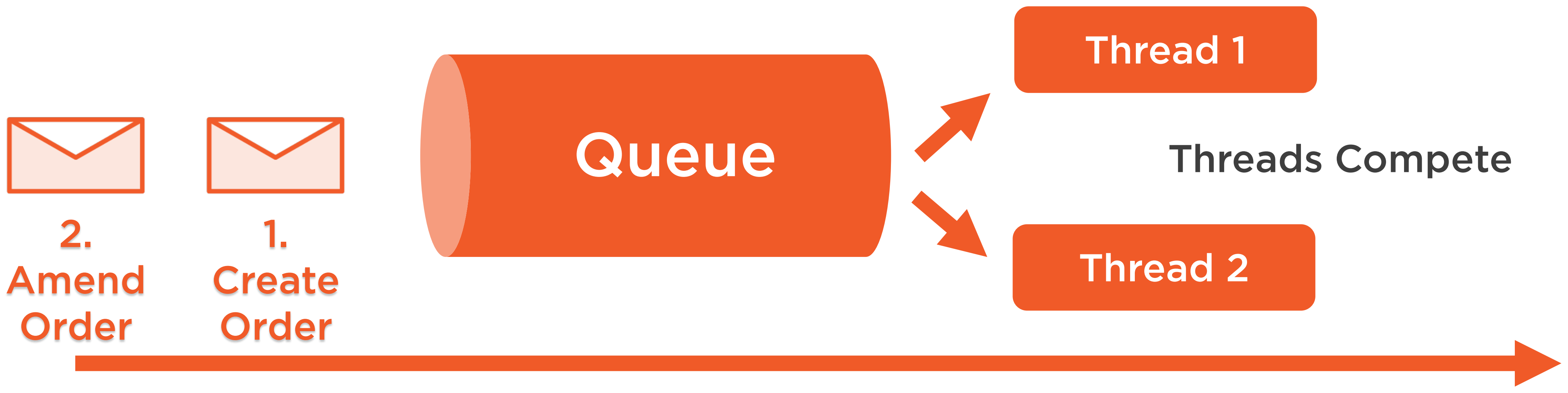
Resource Caching

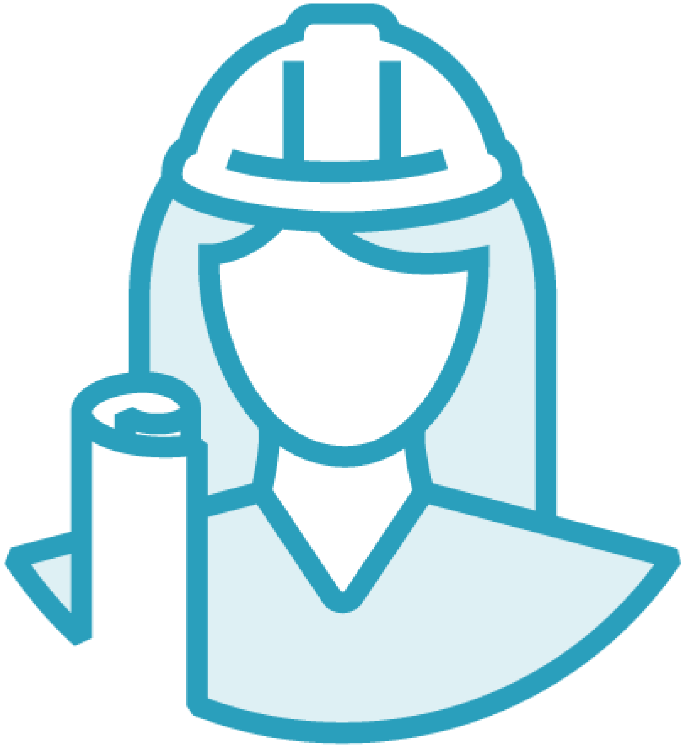
Dynamic Multiple Consumers

Automatic Reconnection

Message Ordering

Message Ordering





BEST: Code Should Deal with Messages Out of Order

Place Unexpected Messages Back at End of Queue (only partially solves the problem)

Message Priorities (only partially solves the problem)

JMSXGroupID

JMSXGroupID

✓

Guarantees Order based on JMSXGroupID property

Still allows failover should consumer “die”

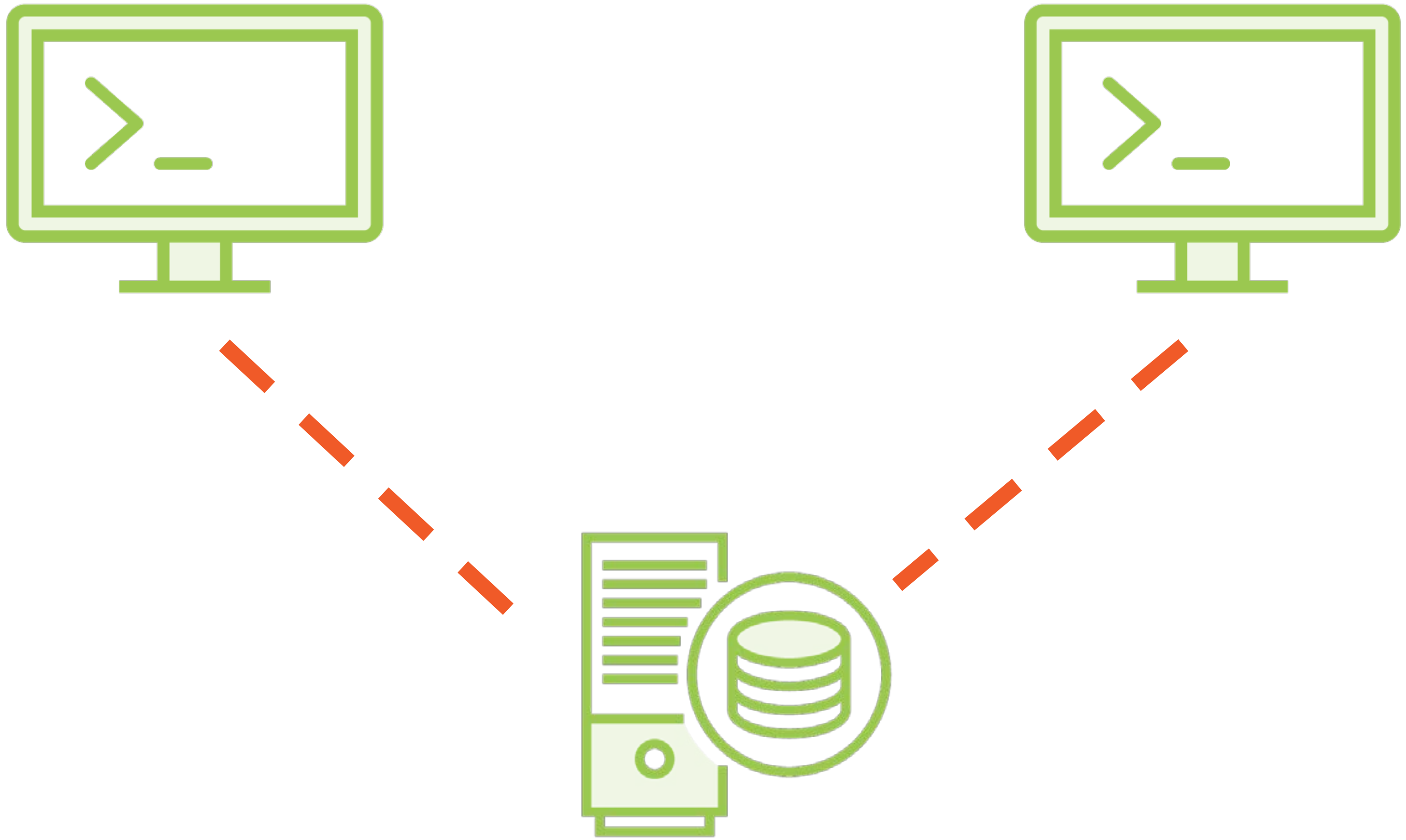
✗

Only 1 consumer per JMSXGroupID property value

Set JMSXGroupID to lowest common denominator eg account number

Error & Exception Handling

Automatic Reconnection

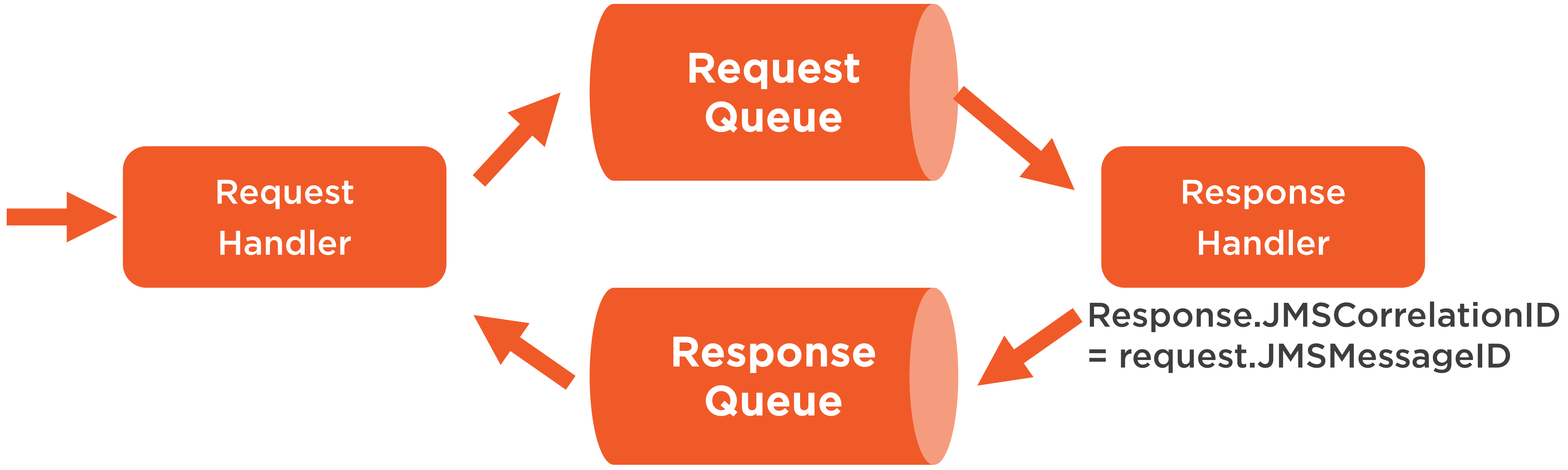


Message Selectors

Possible to consume messages based on property evaluation

Request/Response (Synchronous) Message Using Message Selectors

Synchronous Messaging



Response.JMSCorrelationID = request.JMSMessageID

Consumer Receive with Timeout

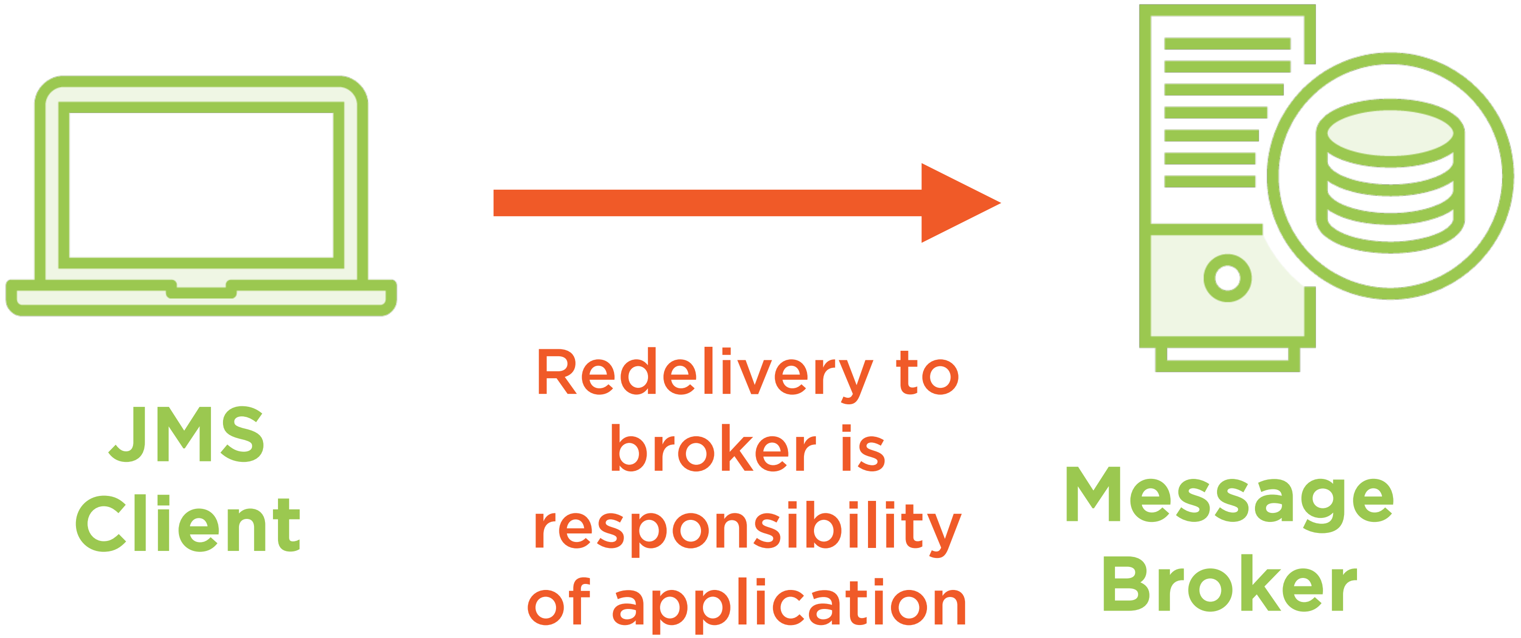
```
MessageConsumer consumer = session.createConsumer(…);

//Timeout in milliseconds
Message msg = consumer.receive(30000);
```

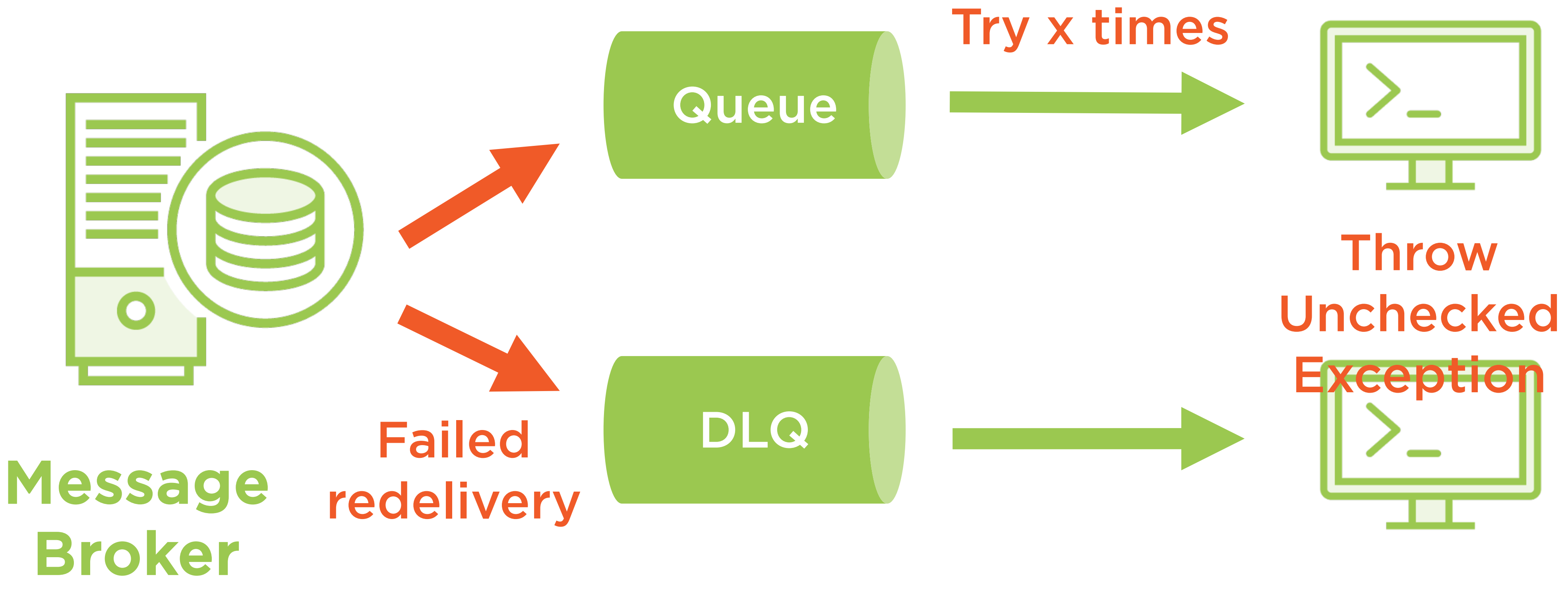
Messages with a finite lifespan should have a TTL (time to live) defined

Dead Letter Queues

Message Producer



Message Consumers



Summary

Efficient Use of Resources

High Availability & Throughput

Message Ordering

Error & Exception Handling

Message Selectors

Synchronous Messaging

Dead Letter Queues