

Using JMS in a JEE Environment



Grant Little

www.grantlittle.me



JMS Versions

< JEE7

JMS 1.1

JEE7 +

JMS 2.0





WildFly

- Basis for the JBoss Application Server



Configuring WildFly for JMS



<http://wildfly.org/downloads/>



Scripts to Use

Windows

Use *.bat

***nix**

Use *.sh



“Typical” installation of
WildFly does not enable JMS
by default



Adding Queues & Topics To WildFly



Creating a JMS 1.1 MessageSender EJB





JEE Container

- Caches/Pools
 - Connections
 - Sessions
 - Consumers
 - Producers

Sending Messages Using JMS 1.1



Creating a JMS 2.0 MessageSender EJB



JMSContext replaces the separate Connection and Session objects in the classic API with a single object.



Java EE 7 introduces
a *platform default JMS
connection factory*. This is a
built-in connection factory
that connects to the
application server's built-in
JMS provider.



JMSProducer Simplifies Send Operations

```
Destination destination;
```

```
JMSProducer producer = context.createProducer();
```

```
producer.send(destination, "Some text to send");
```

```
producer.send(destination, byteArray);
```



JMSProducer Fluid API

```
JMSProducer producer =  
    context.createProducer()  
        .send(destination1, "Some text")  
        .send(destination2, someByteArray)
```



JMSContext Application Managed

```
@Resource
```

```
ConnectionFactory cf;
```

```
JMSContext context = cf.createContext();
```

```
// Do whatever
```

```
context.close();
```



JMSContext Container Managed

```
@Inject
```

```
JMSContext context;
```

```
// Do whatever
```



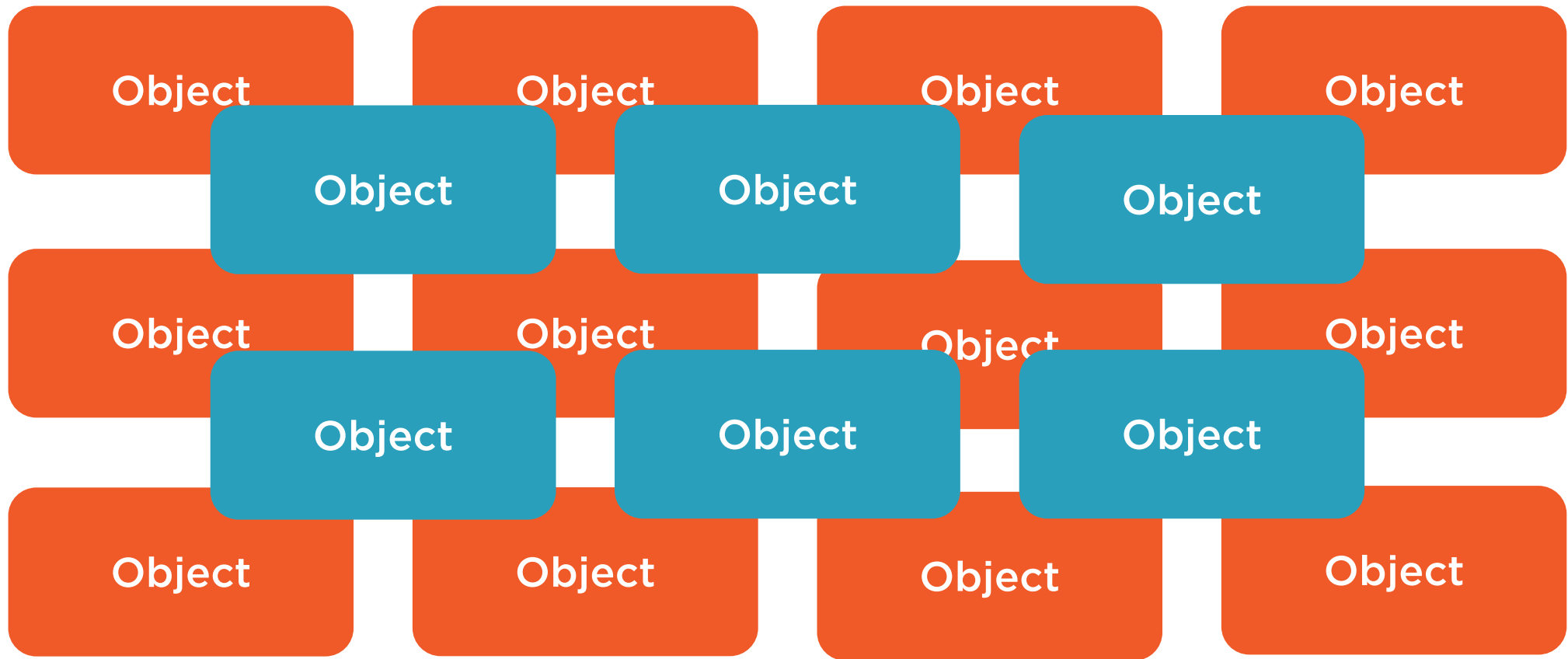
Receiving Messages



Creating a Custom Message Converter



Serialization Issues





How Do We Represent Business Objects on a Queue?

- Member variables as JMS Properties?
- Member variables in the message body?
- Whole message as a binary object?
- Versioning?
- Backwards compatibility?

Message Converter



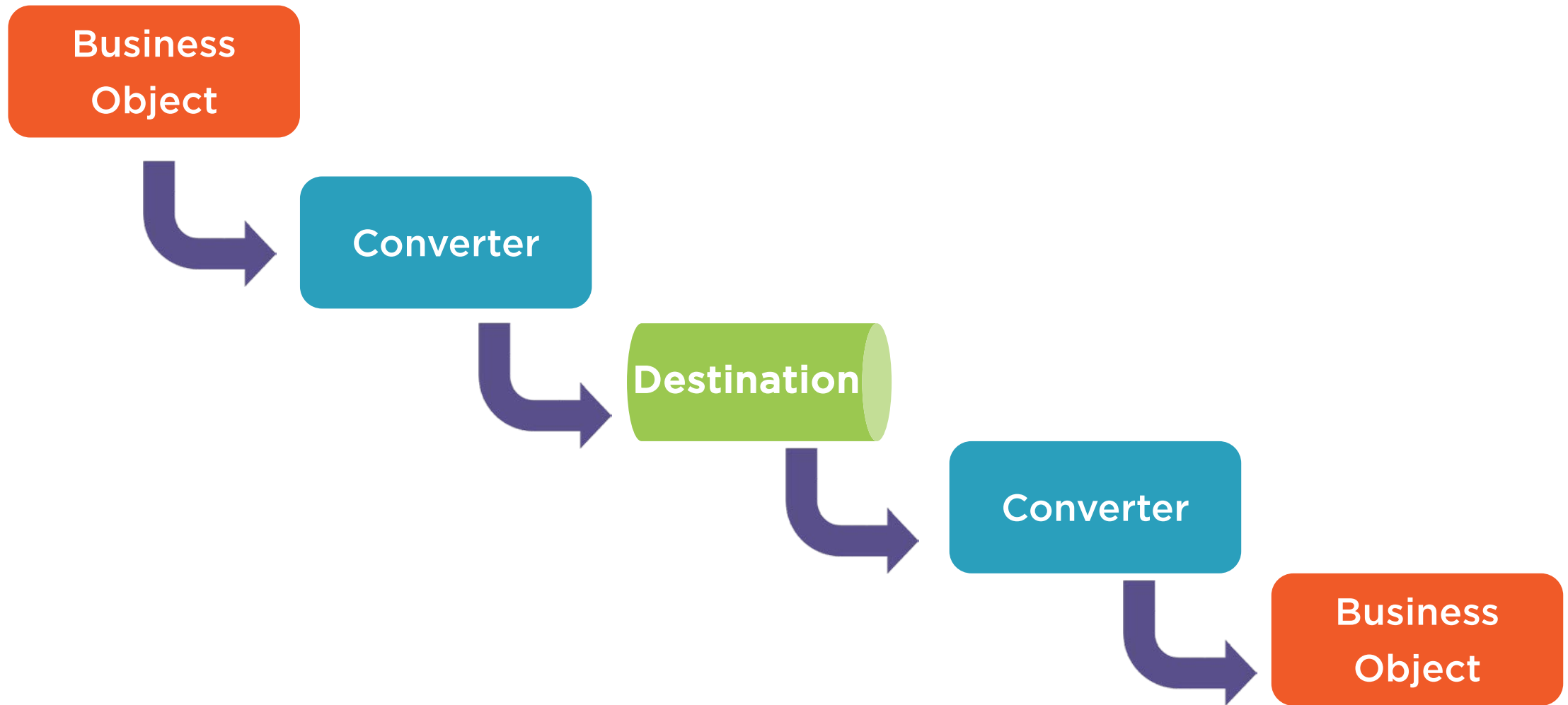
```
public class MyObject implements Serializable {}  
  
ObjectMessage message = context.createObjectMessage(new  
MyObject());
```

Java Serialization & Backward Compatibility

There isn't a good concept of versioning in standard Java serialization that supports backwards compatibility or working with 2 versions of the same object (possibly with different member variables)



For a Lot of Business Applications



spring-jms-xx.xx.jar



Module Summary



**Configuring WildFly Application Server
for JMS**

Defining Queues & Topics

Generic MessageSender EJB

Sending JMS Messages

**Receiving Messages Asynchronously
Using MDBs**

Implement Custom Message Converter

