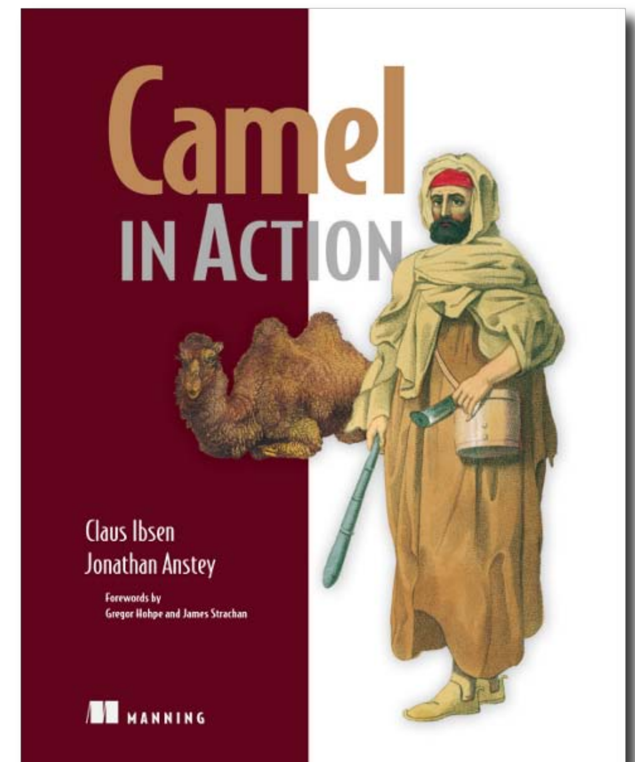
Getting the most out of your ServiceMix deployment of Camel

Jonathan Anstey
Principal Engineer
FuseSource

FuseSource
A Progress Software Company

Your Presenter is: Jonathan Anstey

- Principal Software Engineer at FuseSource
  http://fusesource.com
- Apache Camel PMC member, Apache ActiveMQ committer
- Co-author of Camel in Action



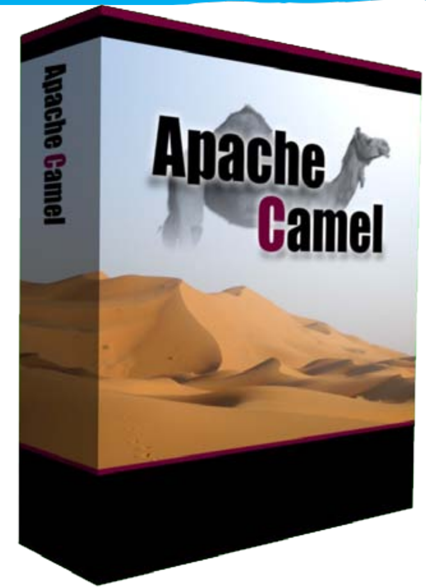
Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential
FuseSource
A Progress Software Company

Agenda

- What is Apache Camel?
- What is Apache ServiceMix?
- Rider Auto Parts example
- Best practices for deploying into ServiceMix

Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential
FuseSource
A Progress Software Company

What is Camel?



- Apache Camel focuses on making integration easier by having:
  - Implementations of the Enterprise Integration Patterns (EIPs)
  - Connectivity to many transports and APIs
  - Easy to use Domain Specific Language (DSL) to wire EIPs and transports together
  - No container dependency

Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved. FuseSource Confidential

Known Deployment Containers

- Apache ServiceMix / Fuse ESB
- Apache Karaf
- Apache ActiveMQ / Fuse MB
- Apache Tomcat
- Jetty
- JBoss
- OpenESB
- Etc…
- Sonic ESB
- IBM WebSphere
- Oracle WebLogic
- Oracle OC4j
- Google App Engine
- Amazon EC2
- Etc…

Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential
FuseSource
A Progress Software Company

What is ServiceMix?

- Open source container useful for integration and SOA – an ESB.
  - EIP-style integration flows
  - SOAP & REST web services
  - Business processes
  - Reliable messaging

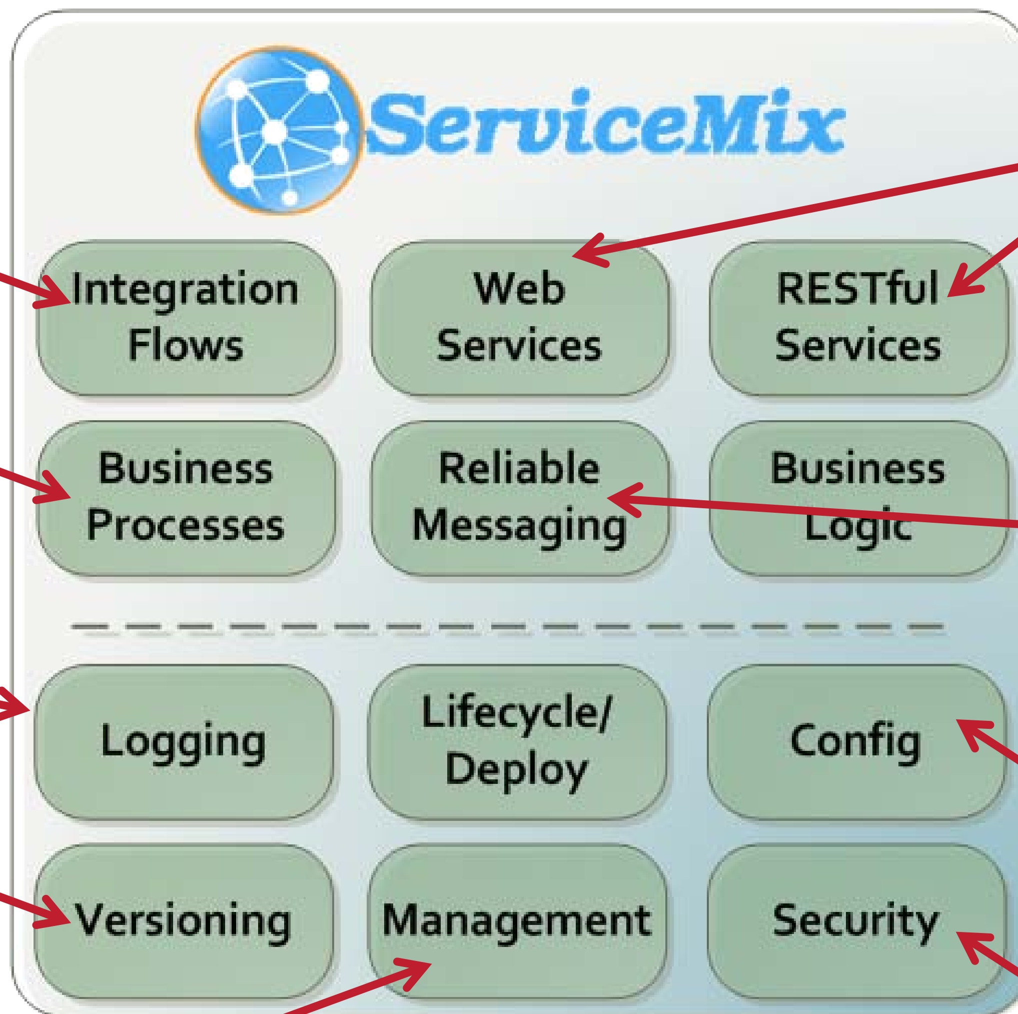
Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential

What is ServiceMix?

- Support for various cross-functional concerns
  - Logging
  - Lifecycle and deployment
  - Configuration
  - Versioning & Dependency Mgmt
  - Management
  - Security
  - Transactions

Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential
FuseSource
A Progress Software Company

ServiceMix – the Technologies



EIP via Camel

BPEL via ODE

OSGi, JCL, JUL, Log4j, Slf4j

OSGi

JMX, web console, SSH

JAX-WS and JAX-RS via CXF

JMS via ActiveMQ

OSGi Config Admin

JAAS, SSH, HTTPS, TLS, etc

Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential
FuseSource
A Progress Software Company

Modular Deployment with OSGi bundles

- ServiceMix will deploy almost anything
  - OSGi bundles, JBI, WARS, Spring , JARs, etc
- Prefer to create Java modules as OSGi bundles
  - Precise control over classloading
  - Builtin versioning support
  - Lifecycle: load, start, stop bundles
  - Dependency management
  - Highly dynamic: upgrade your application without bringing the whole app server down
- OSGi bundles are just a JAR + MANIFEST enties

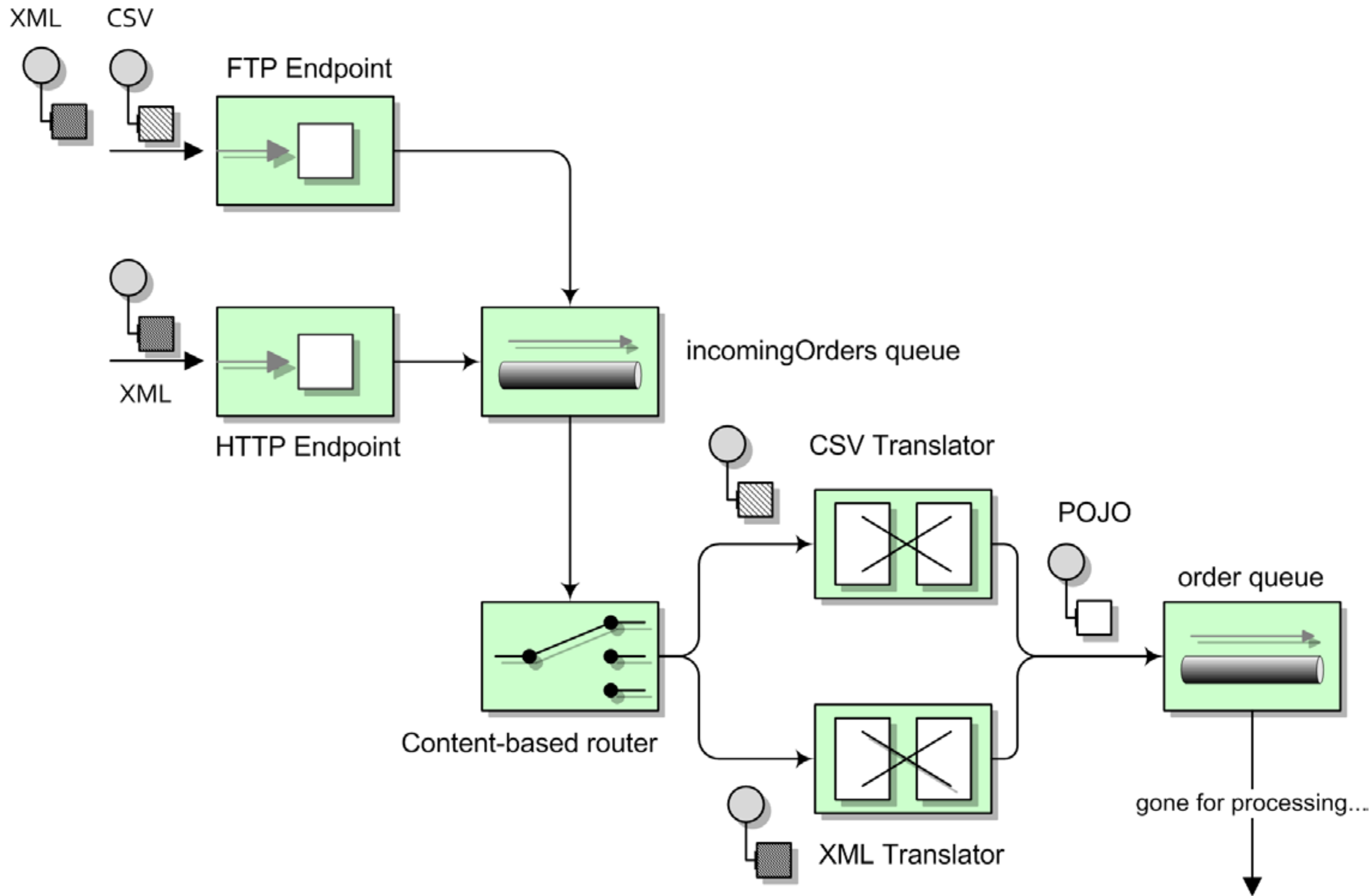
Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.

FuseSource Confidential

Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential
FuseSource
A Progress Software Company

Rider Auto Parts Example

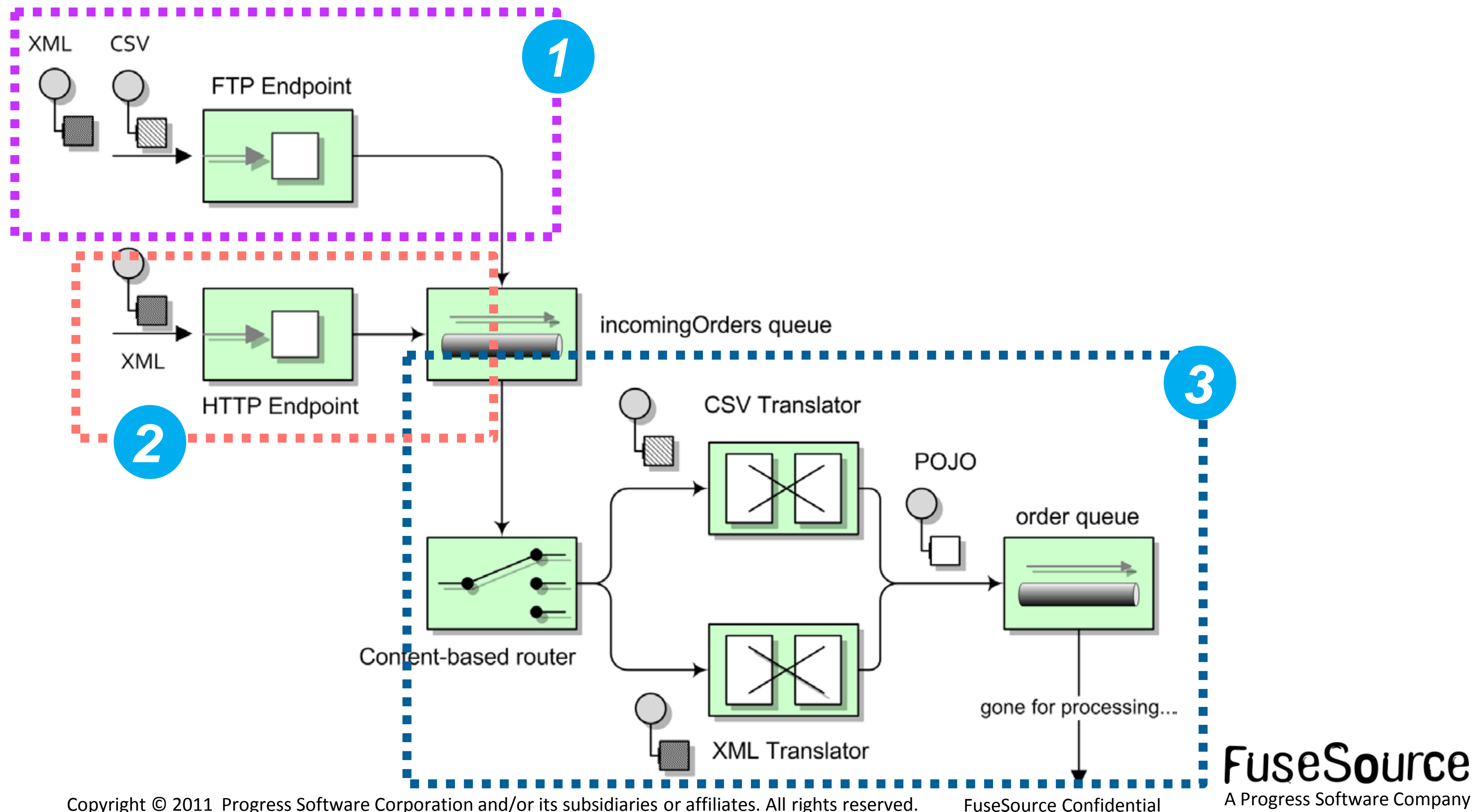
- http://java.dzone.com/articles/open-source-integration-apache



Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential
FuseSource
A Progress Software Company

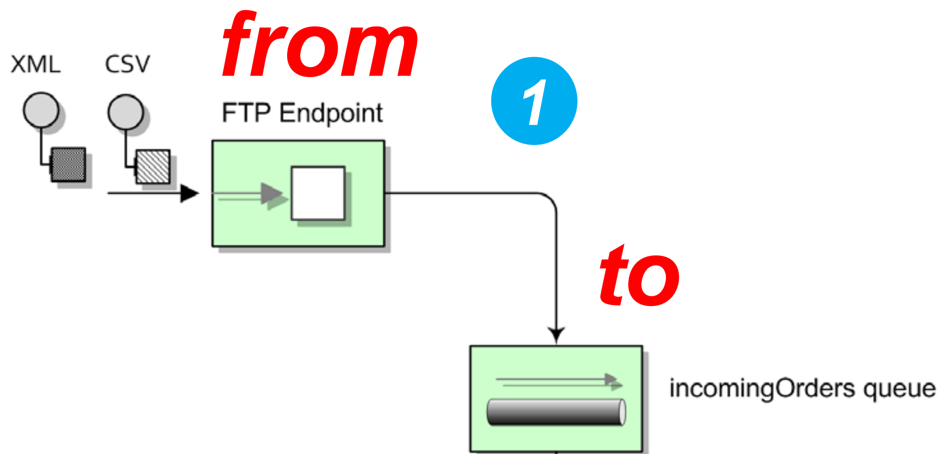
Rider Auto Parts Example

- Rider Auto Parts Example - 3 Routes



Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential
FuseSource
A Progress Software Company

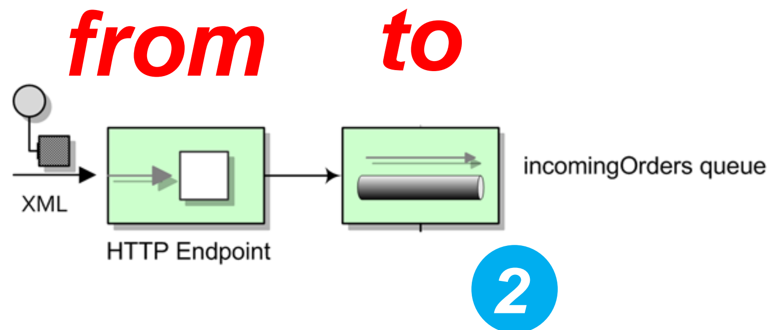
Rider Auto Parts Example




```
<route id="FileToJMS">
    <from uri="file:target/placeorder"/>
    <to uri="jms:incomingOrders"/>
</route>
```


Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved. FuseSource Confidential

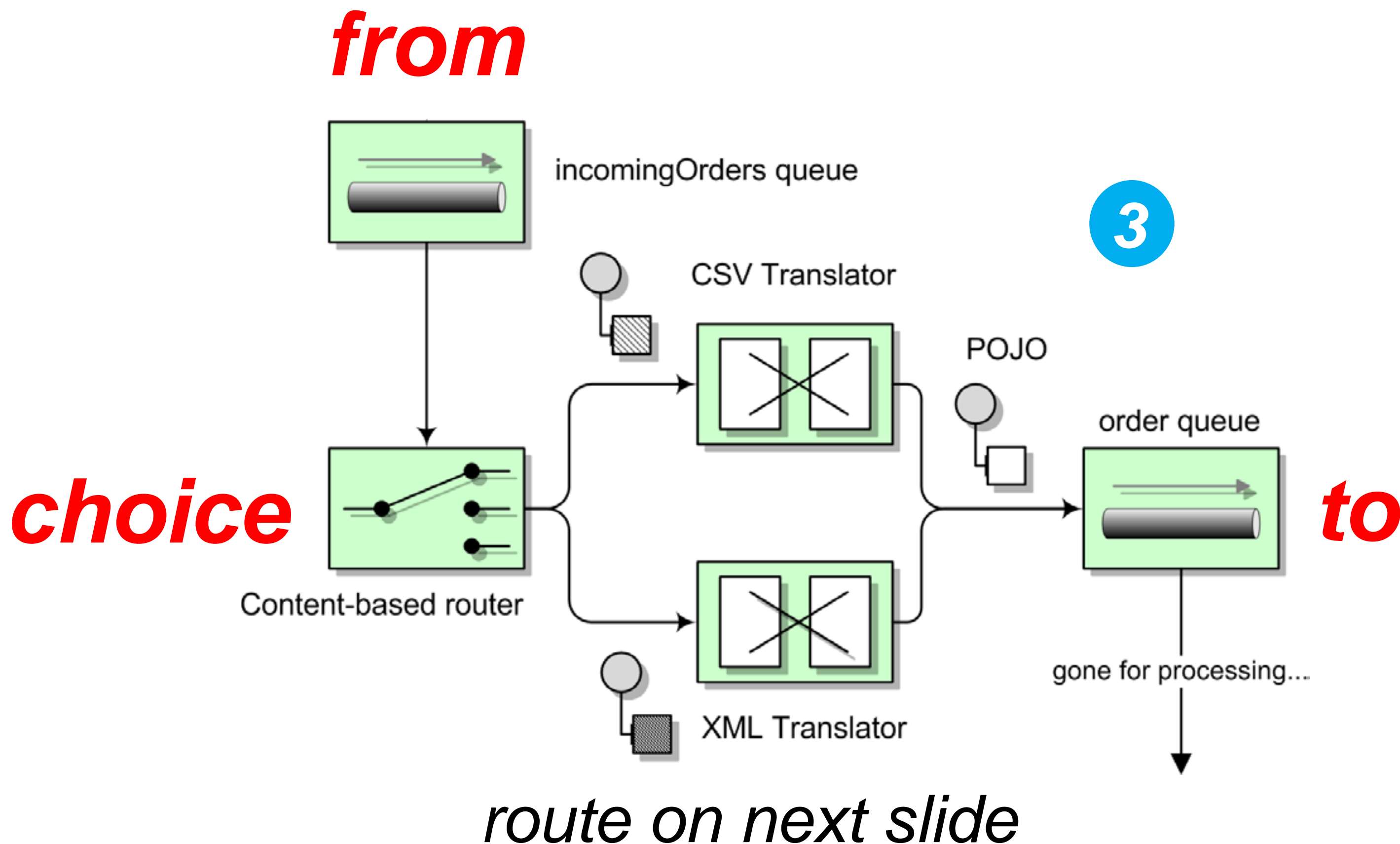
Rider Auto Parts Example




```
<route id="HTTPtoJMS">
    <from uri="jetty:http://0.0.0.0:8888/placeorder"/>
    <inOnly uri="jms:incomingOrders"/>
    <transform>
        <constant>OK</constant>
    </transform>
</route>
```


Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential
FuseSource
A Progress Software Company

Rider Auto Parts Example



Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential
FuseSource
A Progress Software Company

Rider Auto Parts Example

```xml
<route id="NormalizeMessageData">
  <from uri="jms:incomingOrders" />
  <convertBodyTo type="java.lang.String" />
  <choice>
    <when>
      <simple>${body} contains '?xml'</simple>
      <unmarshal>
        <jaxb contextPath="org.fusesource.camel.model" />
      </unmarshal>
    </when>
    <otherwise>
      <unmarshal>
        <bindy packages="org.fusesource.camel.model" type="Csv" />
      </unmarshal>
    </otherwise>
  </choice>
  <to uri="jms:orders" />
</route>
```


Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential
FuseSource
A Progress Software Company

Rider Auto Parts Example

- org.fusesource.examples.rider-auto-spring
  - src/main/java
    - org.fusesource.camel
      - Order.java
  - src/main/resources
    - META-INF
      - spring
        - camel-context.xml
    - org
    - log4j.properties
  - src/test/java
  - JRE System Library [J2SE-1.5]
  - Maven Dependencies
  - src
  - target
  - pom.xml
  - ReadMe.txt

JAXB & Bindy annotated POJO (Order.java)

Spring CamelContext (camel-context.xml)

Maven-based build (pom.xml)

Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved. FuseSource Confidential
FuseSource
A Progress Software Company

Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential

OSGi-ifying the example

- Change Maven POM packaging type
  - <packaging>bundle</packaging>
- Use the maven-bundle-plugin to generate OSGi entries in the JAR's MANIFEST

```
<plugin>
  <groupId>org.apache.felix</groupId>
  <artifactId>maven-bundle-plugin</artifactId>
  <extensions>true</extensions>
</plugin>
```

- This will automatically import and export the necessary packages.

Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential
FuseSource
A Progress Software Company

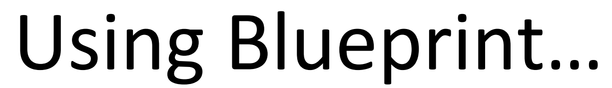
OSGi-ifying the example

- Usually best to NOT export implementation packages however.
- If we had Java DSL routes in org.fusesource.camel.impl, we could tell the bundle plugin to not export those routes.

```
<plugin>
  <groupId>org.apache.felix</groupId>
  <artifactId>maven-bundle-plugin</artifactId>
  <extensions>true</extensions>
  <configuration>
    <instructions>
      <Import-Package>*</Import-Package>
      <Export-Package>
        org.fusesource.camel.model,
        !org.fusesource.camel.impl
      </Export-Package>
    </instructions>
  </configuration>
</plugin>
```


Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential

FuseSource
A Progress Software Company



Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential

Using Blueprint

- Spring-DM is an OSGi add-on to Spring
- Blueprint from the Apache Aries project is a standardized version of Spring-DM
- Better integration with OSGi
  - Ability to wait on bundles with custom namespaces prior to starting
  - Don't need tons of schema imports
  - Version of Camel namespace used determined at runtime

Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential
FuseSource
A Progress Software Company

Using Blueprint

- Blueprint file should be placed in OSGI-INF/blueprint
- Syntax within the camelContext is identical to plain Spring deployment

```
<blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0">

  <camelContext id="rider-auto-orders" xmlns="http://camel.apache.org/schema/blueprint">
    <route id="HTTPtoJMS">
      <from uri="{{httpEndpoint}}" />
      <inOnly uri="jms:incomingOrders" />
      <transform>
        <constant>OK</constant>
      </transform>
    </route>
```


Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential
FuseSource
A Progress Software Company

Using Blueprint – Caveats

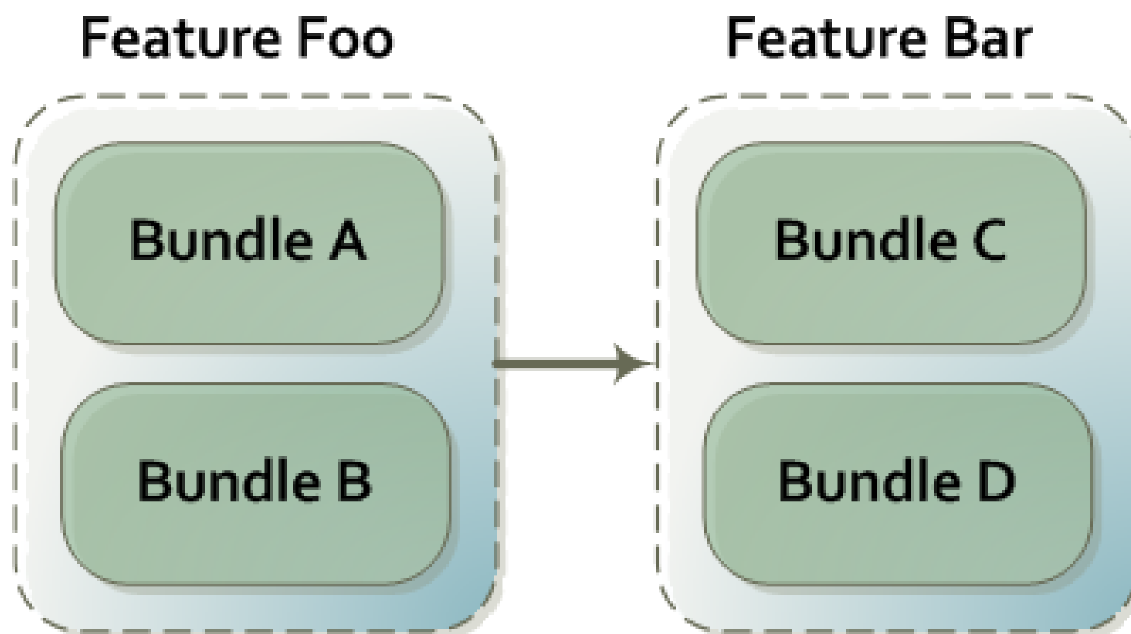
- Blueprint support across all components in Camel is not complete in current releases
  - Ex. Camel-cxf component will start having Blueprint support in Camel 2.8
- Best practice
  - Keep Spring-DM in place for already developed projects
  - Consider Blueprint for new projects

Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential

Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential

Deploy with features

- Features group bundles into a logical unit of deployment
- Installing feature "Foo" would install bundles A, B, C and D



Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.

FuseSource Confidential

FuseSource
A Progress Software Company

Deploy with features

- You should specify existing features in ServiceMix to depend on rather than individual bundles.

```
<features name="rider-auto-osgi"
  xmlns="http://karaf.apache.org/xmlns/features/v1.0.0">
    <feature version="${project.version}" name="rider-auto-osgi">
        <feature>camel-core</feature>
        <feature>camel-blueprint</feature>
        <feature>camel-activemq</feature>
        <feature>camel-jaxb</feature>
        <feature>camel-bindy</feature>
        <feature>camel-jetty</feature>
        <bundle>mvn:org.fusesource.examples/rider-auto-osgi/${project.version}</bundle>
    </feature>
</features>
```


Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.

FuseSource Confidential

FuseSource
A Progress Software Company

Deploy with features

- You can then SSH into ServiceMix and use the features shell to install the feature.

```
janstey@duffman:~$ ssh -p 8101 smx@192.168.1.101
smx@192.168.1.101's password:
 ____                  _          __  __ _
/ ___|  ___ _ ____   _(_) ___ ___|  \/  (_)_  __
\___ \ / _ \ '__\ \ / / |/ __/ _ \ |\/| | \ \/ /
 ___) |  __/ |   \ V /| | (_|  __/ |  | | |>  <
|____/ \___|_|    \_/ |_|\___\___|_|  |_|_/_/\_\

  Apache ServiceMix (4.4.0-fuse-SNAPSHOT)

Hit '<tab>' for a list of available commands
and '[cmd] --help' for help on a specific command.

smx@root> features:addurl file:/tmp/features.xml
smx@root> features:install rider-auto-osgi
```


Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential
FuseSource
A Progress Software Company

Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.

FuseSource Confidential

FuseSource
A Progress Software Company

Take advantage of OSGi Config Admin

- The Config Admin service provides an easy way of getting configuration into your bundle

```
<cm:property-placeholder persistent-id="org.fusesource.camel-config">
  <cm:default-properties>
    <cm:property name="httpEndpoint" value="jetty:http://0.0.0.0:8888/placeorder" />
    <cm:property name="fileEndpoint" value="file:target/placeorder" />
  </cm:default-properties>
</cm:property-placeholder>
```

- You can then use these properties in your routes

```
<camelContext id="rider-auto-orders" xmlns="http://camel.apache.org/schema/blueprint">
  <route id="HTTPtoJMS">
    <from uri="{{httpEndpoint}}" />
    <inOnly uri="jms:incomingOrders" />
    <transform>
      <constant>OK</constant>
    </transform>
  </route>
```


Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.

Take advantage of OSGi Config Admin

- You can update properties in the command shell or by modifying a properties file.
- Updating the HTTP endpoint at runtime is simple:

```
smx@root> config:edit org.fusesource.camel-config
smx@root> config:propset httpEndpoint jetty:http://0.0.0.0:7777/placeorder
smx@root> config:update
smx@root> osgi:restart 111
```


Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.

Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential

Reference existing services

- You should reuse existing services rather than rolling your own
- Reference ActiveMQ ConnectionFactory for JMS messaging

```
<reference id="connectionFactory" interface="javax.jms.ConnectionFactory"
  filter="(name=localhost)" />

<bean id="jms" class="org.apache.activemq.camel.component.ActiveMQComponent">
  <property name="connectionFactory" ref="connectionFactory" />
</bean>
```

- Reference Aries TransactionManager for transactions

```
<reference id="transactionManager" interface="javax.transaction.TransactionManager" />
```


Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved. FuseSource Confidential

Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.

FuseSource Confidential

FuseSource
A Progress Software Company

Deploying only what you need

- You should reduce the boot features to only what you need.
  - featuresBoot property in etc/org.apache.karaf.features.cfg
- Vanilla install of ServiceMix 4.4 loads **249 bundles**
- For our example we were able to reduce that to **88 bundles**

Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.

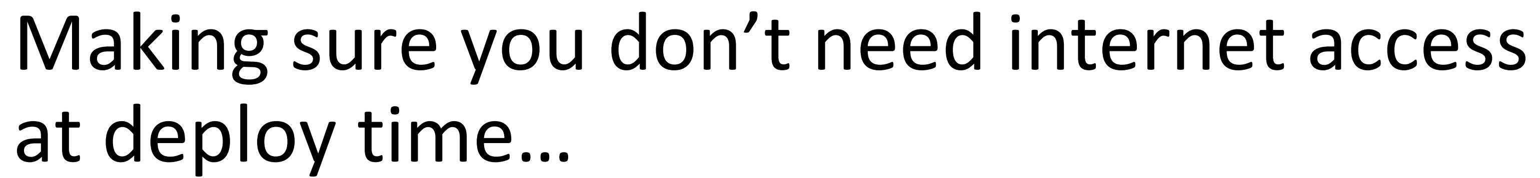
FuseSource Confidential

FuseSource
A Progress Software Company

Deploying only what you need

- Features we kept:
  - config – Config Admin shell
  - camel – core Camel support
  - activemq-broker – Embedded ActiveMQ broker
  - camel-activemq – ActiveMQ support in Camel
- Features we removed:
  - camel-nmr – didn't initially need NMR support
  - camel-cxf – we are not using CXF
  - jbi-cluster, servicemix-* – not using JBI at all
  - war – we didn't deploy any WAR

Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved. FuseSource Confidential



Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential

Making sure you don't need internet access at deploy time

- Maven is great for development time as you never have to go out and download a library yourself – it just downloads from repositories on the Internet.
- In a production environment however, you should make sure all libraries are already available locally to the ESB.
  - You may not have Internet access in your environment
  - Having all libraries locally available reduces the risk of failure at deploy time

Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential
FuseSource
A Progress Software Company

Making sure you don’t need internet access at deploy time

- Use the features-maven-plugin to package up all 3rd party dependencies of your application.

```
<plugin>
  <groupId>org.apache.karaf.tooling</groupId>
  <artifactId>features-maven-plugin</artifactId>
  <executions>
    <execution>
      <id>add-features-to-repo</id>
      <phase>generate-resources</phase>
      <goals>
        <goal>add-features-to-repo</goal>
      </goals>
      <configuration>
        <descriptors>
          <descriptor>mvn:org.apache.camel.karaf/apache-camel/${camel-version}/xml/features</descriptor>
          <descriptor>mvn:org.apache.servicemix/apache-servicemix/${servicemix-version}/xml/features</descriptor>
          <descriptor>mvn:org.apache.activemq/activemq-karaf/${activemq-version}/xml/features</descriptor>
          <descriptor>file:${basedir}/target/classes/features.xml</descriptor>
        </descriptors>
        <features>
          <feature>rider-auto-osgi</feature>
        </features>
        <repository>target/repo</repository>
      </configuration>
    </execution>
  </executions>
</plugin>
```


Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential
FuseSource
A Progress Software Company

Making sure you don't need internet access at deploy time

- These dependencies should then be made available to ServiceMix by adding its URL to the org.ops4j.pax.url.mvn.repositories property in etc/org.ops4j.pax.url.mvn.cfg
  - Could be a local file system directory or a repository manager that you import the archive into.
- Future versions of Karaf/ServiceMix will have this process automated by using new Maven plugins and KAR files
  - KAR - think feature descriptor + dependencies in a ZIP

Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.

Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved. FuseSource Confidential
FuseSource
A Progress Software Company

Cross bundle routing with the NMR

- The NMR stands for Normalized Message Router.
- The name is artifact of the JBI origins of ServiceMix – payloads don't need to be normalized if you are not communicating with JBI components.
  - Send whatever payload you like
- Useful as a fast in-memory communication link between routes that exist in the same or separate bundles.

Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.

FuseSource Confidential

FuseSource
A Progress Software Company

Cross bundle routing with the NMR

- For example we could switch to using NMR in between the routes in our example.

```
<route id="HTTPtoJMS">
    <from uri="{{httpEndpoint}}"/>
    <inOnly uri="nmr:incomingOrders"/>
    <transform>
        <constant>OK</constant>
    </transform>
</route>

<route id="FileToJMS">
    <from uri="{{fileEndpoint}}"/>
    <to uri="nmr:incomingOrders"/>
</route>

<route id="NormalizeMessageData">
    <from uri="nmr:incomingOrders"/>
    <convertBodyTo type="java.lang.String"/>
    <choice>
```


Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential

Recap

- Apache ServiceMix/Fuse ESB is a great container for Camel.
- Try to let the tooling generate your OSGi MANIFEST unless you need to override options.
- Keep Spring-DM for now but consider Blueprint for new projects.
- Use features to group your bundles.
- Use the ConfigAdmin OSGi service to configure your routes.

Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential

Recap

- Reduce boot features in ServiceMix to only what your application requires.
- Reference existing services.
- Make your feature's dependencies available locally to the ESB rather than relying on Maven downloads from the Internet.
- Use the NMR for communication between routes in the container.

Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
FuseSource Confidential

Useful references

- FuseSource - http://fusesource.com
  - http://fusesource.com/products/enterprise-servicemix/#documentation
- http://java.dzone.com/articles/open-source-integration-apache
- Camel in Action - http://manning.com/ibsen/
- OSGi in Action - http://www.manning.com/hall/

Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.

Any Questions?

Copyright © 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.