



# JBoss ESB Introduction

Martin Večeřa  
mvecera@redhat.com  
October 2011

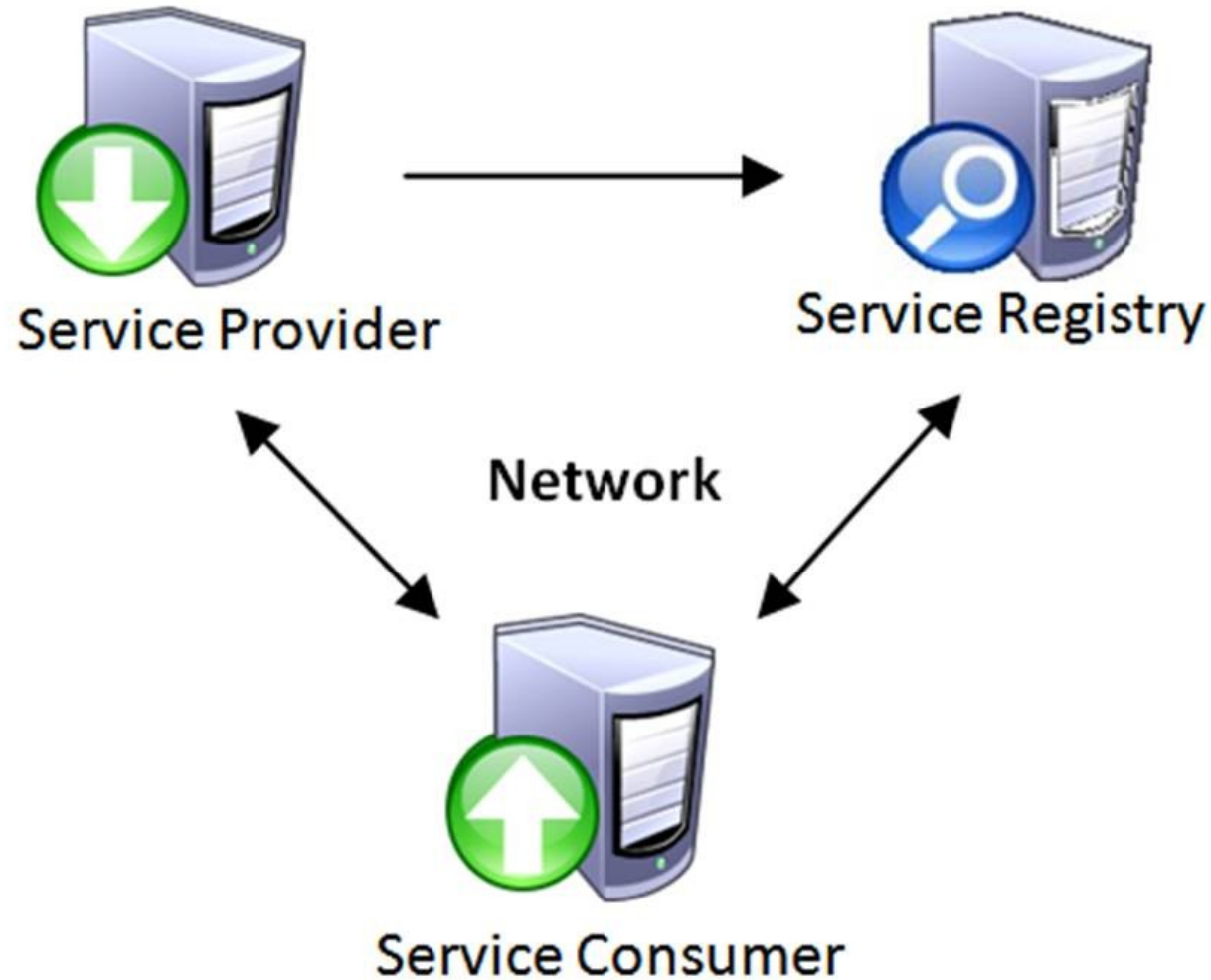
# Agenda

- Service Oriented Architecture
  - Principles
  - SOA Manifesto
- JBoss ESB
  - Architecture
  - Service, Action, Message
  - Built-in Actions

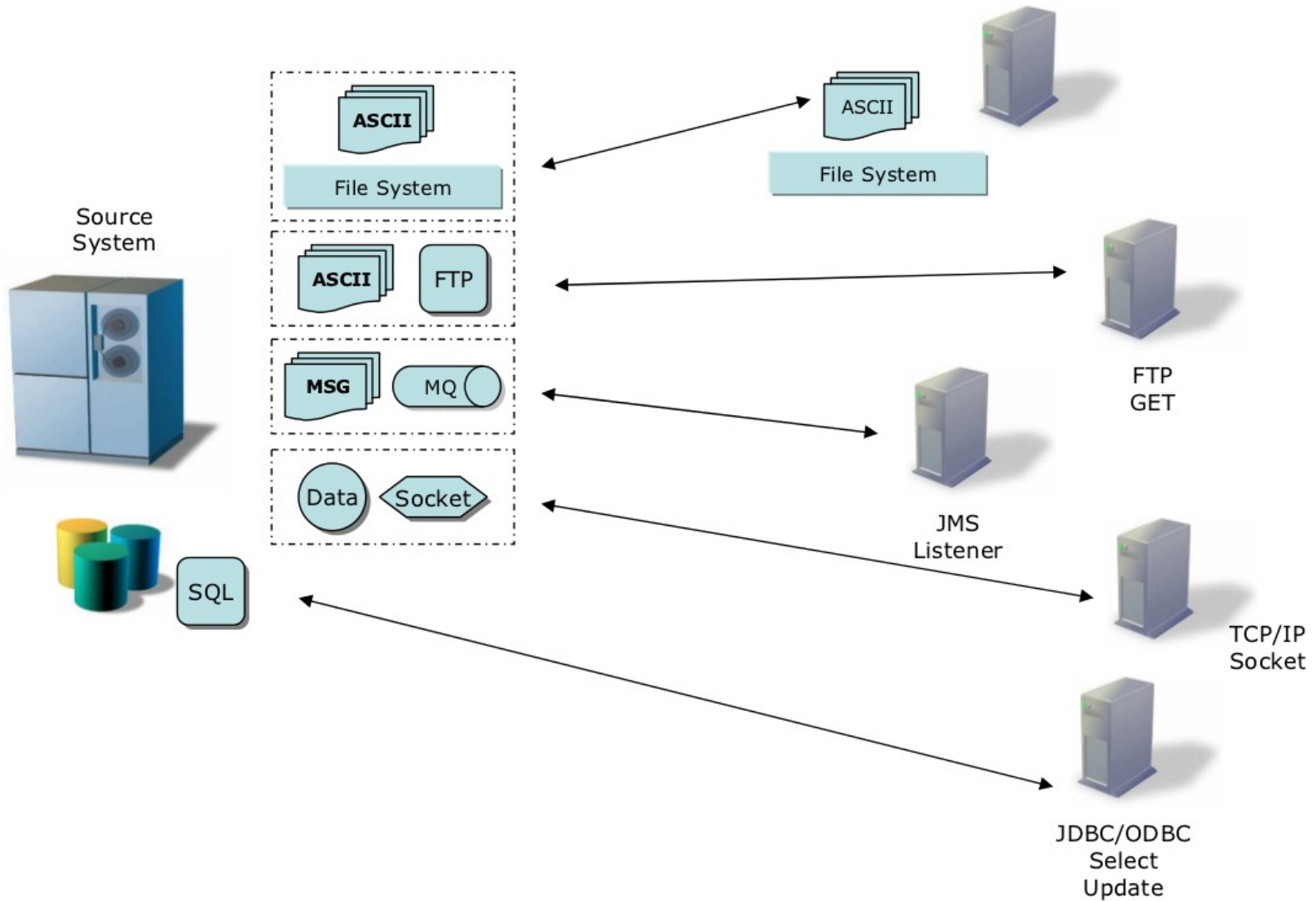
# Service Oriented Architecture

- Services providing functions
- Types of services
  - Task service, Entity service, Utility service
- Basic requirements
  - Reusability
  - Loose coupling
  - Well defined contract (WSDL, XML Schema, WS Policy)
  - Statelessness, Discoverability, Composability, Autonomy...
- How to access services
  - Service registry
- <http://www.soa-manifesto.org/>

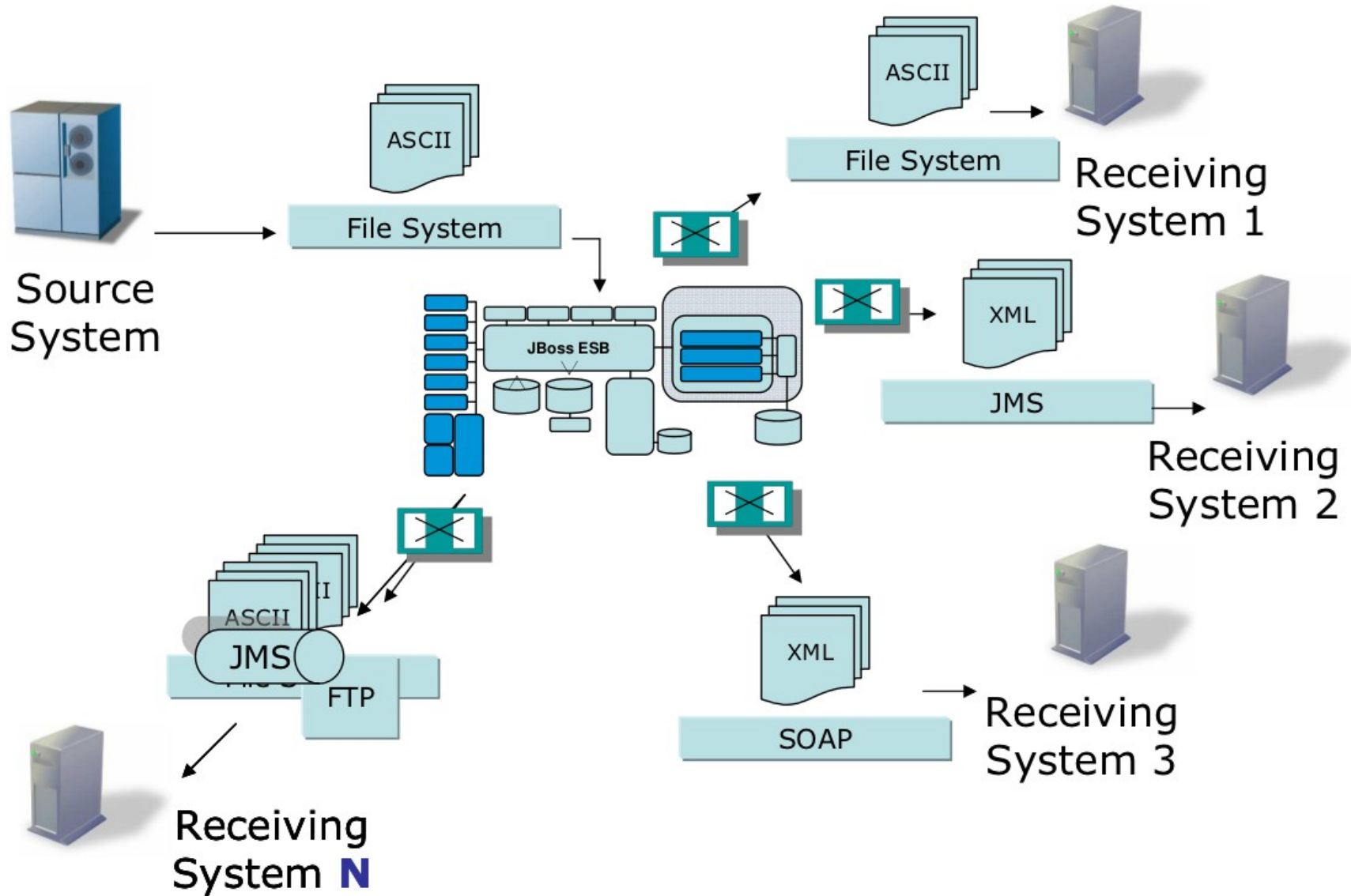
# Service Registry



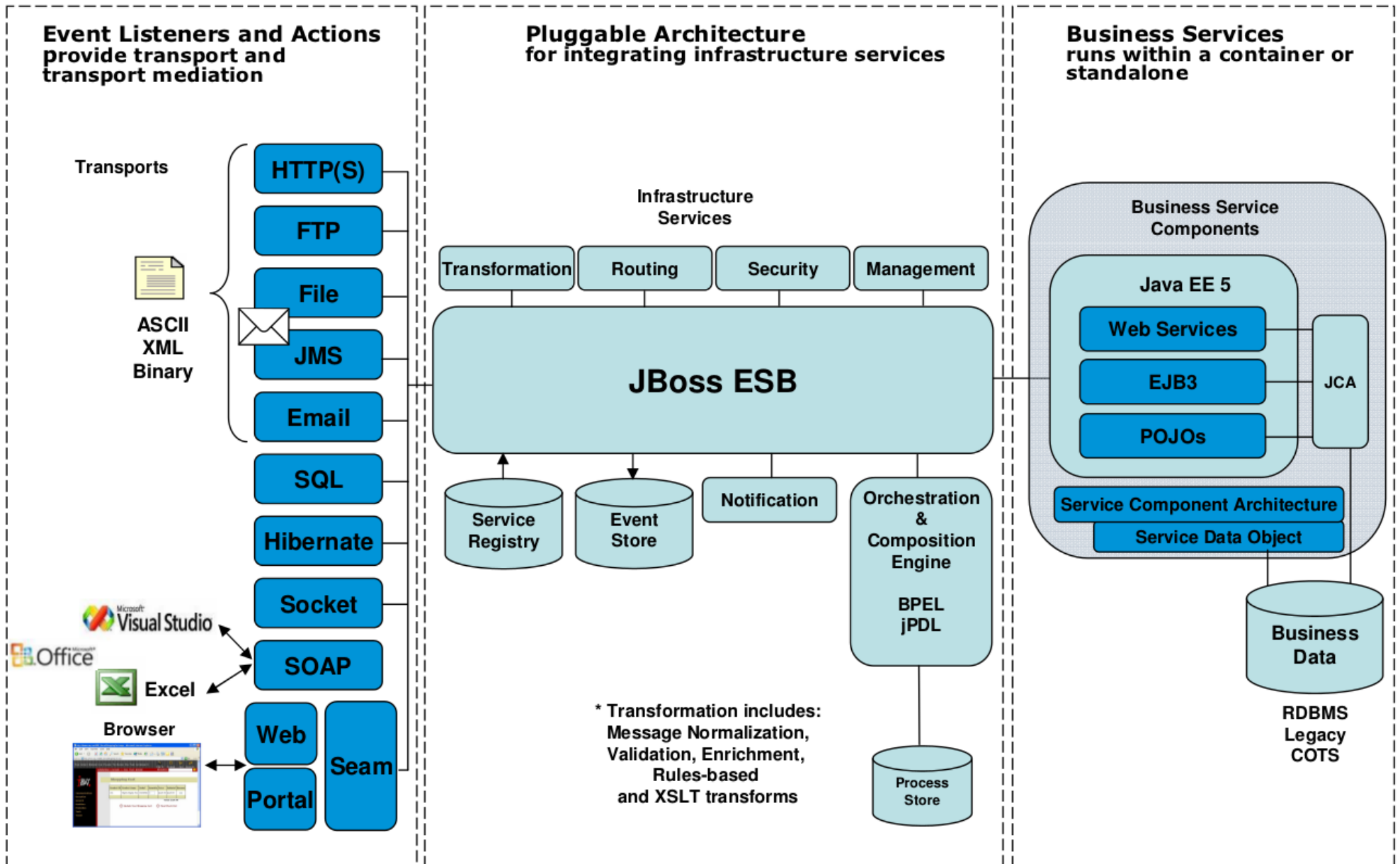
# World Without SOA



# ESB as a SOA Implementation



# ESB Architecture



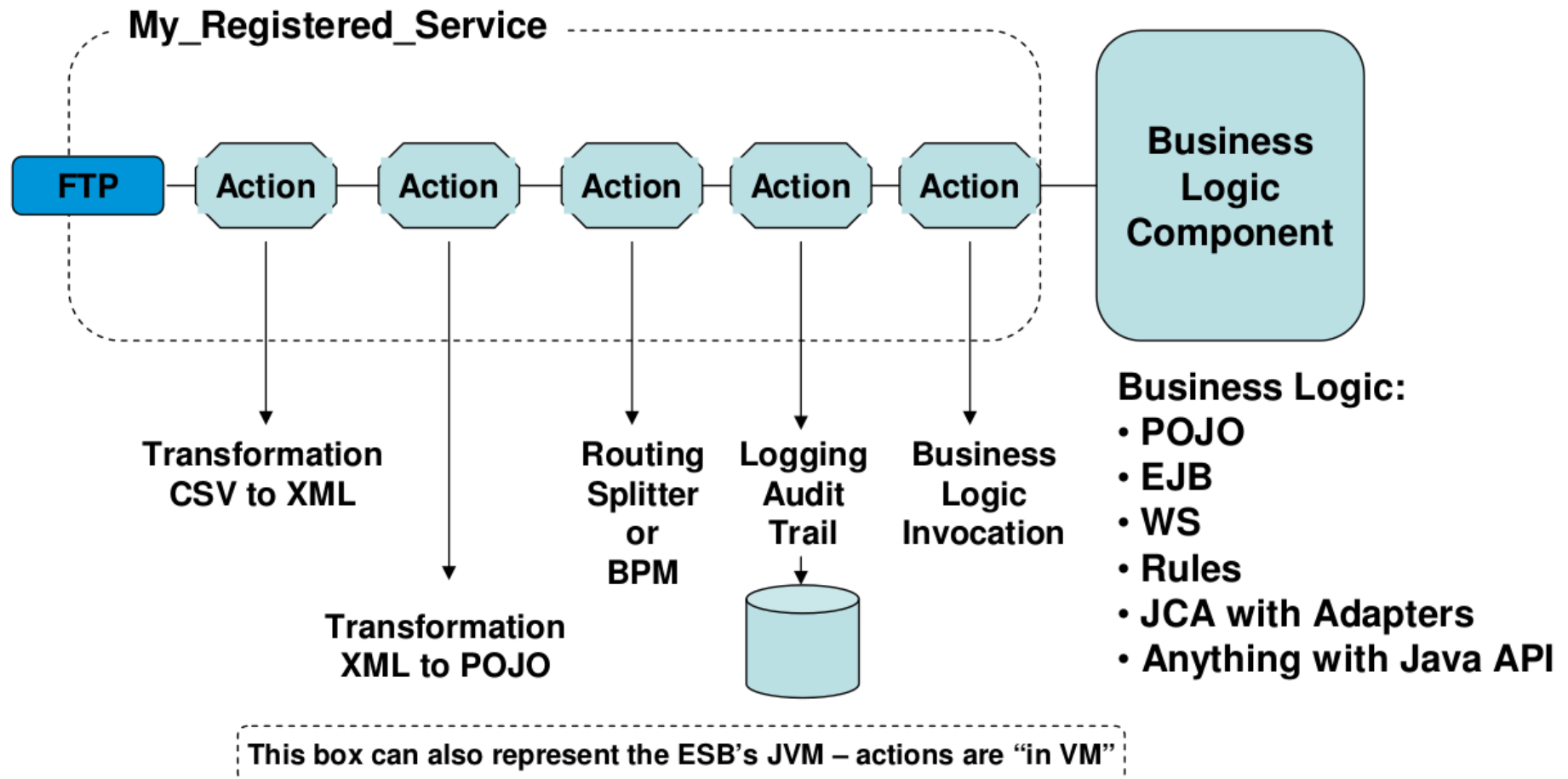
# JBoss ESB Provides

- Message Delivery
  - JMS (JBossMQ, JBoss Messaging, ActiveMQ, MQSeries, Oracle AQ)
  - (S)FTP, SOAP, HTTP, email, database, shared file system
- Action chaining/pipeline
  - For custom service enablement & mediation
- Transformation
  - Smooks, XSLT, Java, Groovy, StringTemplate
- Content-based routing
  - JBoss Rules, XPath DSL
  - Recipient List/Static Routing is also supported
- Orchestration
  - BPM, Intelligent routing, EIPs and hot deployment
- Registry
  - JAX-R based
  - UDDI support out-of-the-box (Scout and jUDDI)



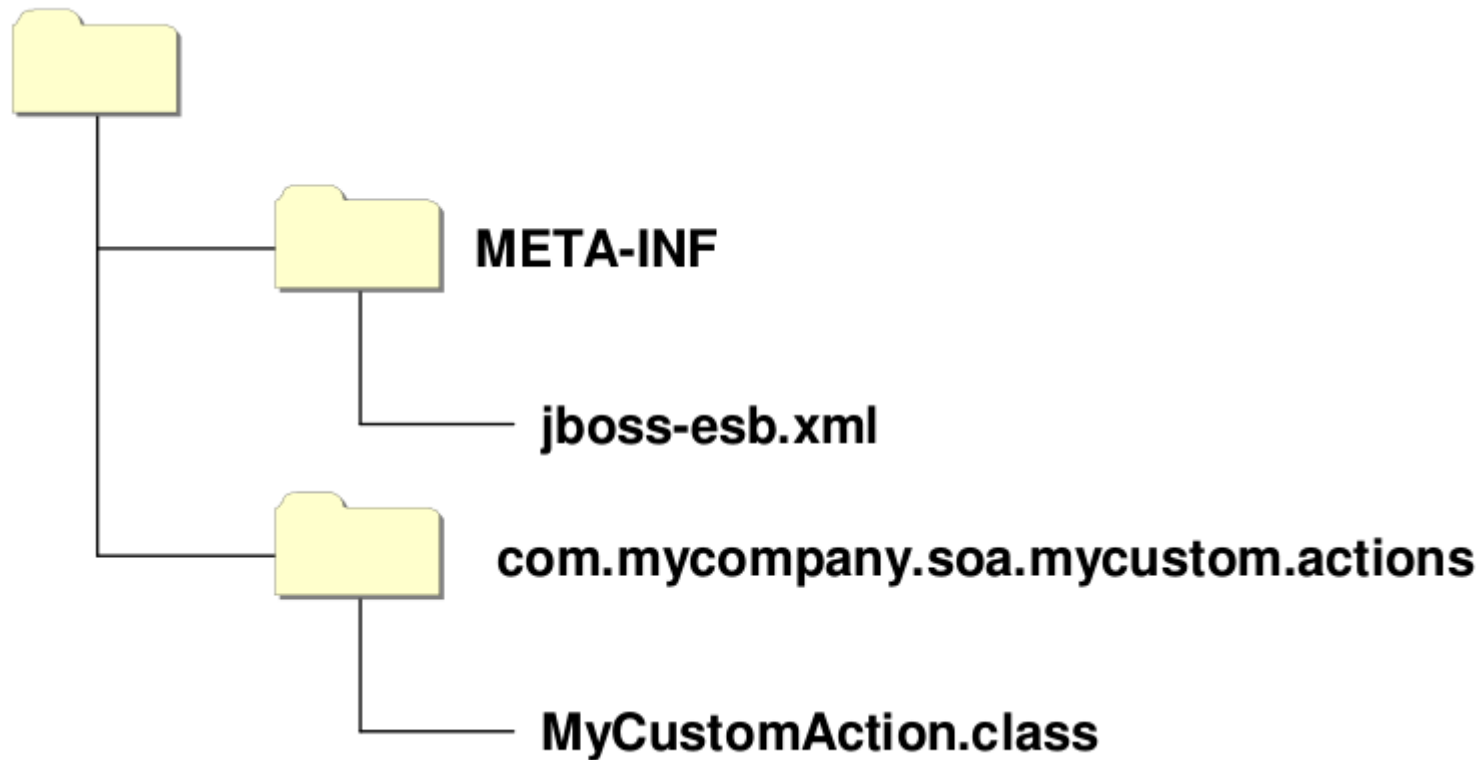
# ESB Service

Actions are reusable mediation components that can be chained together to form the capabilities of a registered service. Actions can be dynamically added/removed at runtime.



# ESB Deployment

- Configuration for one or more services which show up in the registry



# Hello World – jboss-esb.xml

```
<jbossesb>
  <providers>
    <jms-provider name="JBossMessaging"
      connection-factory="ConnectionFactory">

      <jms-bus busid="quickstartGwChannel">
        <jms-message-filter dest-type="QUEUE"
          dest-name="queue/quickstart_helloworld_Request_GW"/>
      </jms-bus>

      <jms-bus busid="quickstartEsbChannel">
        <jms-message-filter dest-type="QUEUE"
          dest-name=" queue/quickstart_helloworld_Request_ESB"/>
      </jms-bus>
    </jms-provider>
  </providers>
```

...

# Hello World – jboss-esb.xml

...

```
<services>
  <service category="FirstServiceESB" name="SimpleListener"
    description="Hello World">
    <listeners>
      <jms-listener name="JMS-Gateway" busidref="quickstartGwChannel"
        maxThreads="1" is-gateway="true"/>
      <jms-listener name="helloWorld" busidref="quickstartEsbChannel"
        maxThreads="1"/>
    </listeners>
    <actions>
      <action name="action1" class="quickstart.helloworld.MyListenerAction"
        process="displayMessage"/>
    </actions>
  </service>
</services>
</jbossesb>
```

# ESB (Un)Aware Message & Configuration

- Gateway listener
- ESB aware listener
  - `org.jboss.soa.esb.message.Message`

- In-VM

```
<service category="HelloWorld" name="Service1"
  description="Service 1" invmScope="GLOBAL">
```

- Maximum number of threads

```
<jms-listener name="helloWorld" busidref="qsEsbChannel1"
  maxThreads="1"/>
```

# Message Exchange Pattern

```
<actions mep="OneWay">
```

- Default Behavior
- OneWay
- RequestResponse

# Message

```
public interface Message {  
    public Header getHeader ();  
    public Context getContext ();  
    public Body getBody ();  
    public Fault getFault ();  
    public Attachment getAttachment ();  
    public URI getType ();  
    public Properties getProperties ();  
    public Message copy() throws Exception;  
}
```

# Hello World Action

```
public class MyAction {  
  
    @ConfigProperty(use = Use.OPTIONAL)  
    private String myConfigProperty;  
  
    @Process  
    public Message process(Message message) {  
        System.out.println("Body: " +  
            new String(message.getBody().getContents()));  
        // modify the contents, do anything Java can do...  
        return message;  
    }  
}
```



# Transformation

- ByteArrayToString, LongToDateConverter, ObjectInvoke, ObjectToCSVString, ObjectToXStream, XStreamToObject
- XsltAction
- Smooks

```
<?xml version='1.0' encoding='UTF-8'?>
<smooks-resource-list xmlns="http://www.milyn.org/xsd/smooks-1.1.xsd"
                      xmlns:jb="http://www.milyn.org/xsd/smooks/javabean-1.2.xsd"
                      default-target-profile="from:dvdstore">

  <!-- Populate the Customer -->
  <jb:bean beanId="customer" class="org.jboss.soa.esb.samples.Customer" createOnElement="customer">
    <jb:value property="userName" data="customer/@userName" />
    <jb:value property="firstName" data="customer/@firstName" />
    <jb:value property="lastName" data="customer/@lastName" />
    <jb:value property="state" data="customer/@state" />
  </jb:bean>

</smooks-resource-list>
```

# Routing

- Aggregator, EchoRouter, HttpRouter, JMSRouter, EmailRouter
- ContentBasedRouter
  - XPath, RegEx, Drools

```
rule "Routing Rule using XPATH Greater"
  when
    xpathGreaterThan "/Order/@totalAmount", "50.0"
  then
    Log : "Really It's EXPRESS Shipping";
    Destination : "ExpressShipping:ExpressShippingService";
end
```

```
rule "Routing Rule using XPATH Less"
  when
    xpathLessThan "/Order/@totalAmount", "50.0"
  then
    Log : "Really It's NORMAL Shipping";
    Destination : "NormalShipping:NormalShippingService";
end
```

# WS Orchestration

- Possibility to consume WS
- Possibility to expose ESB service as a WS

```
<action name="proxy" class="org.jboss.soa.esb.actions.soap.proxy.SOAPProxy">  
  <property name="wsdl" value="https://host/foo/HelloWorldWS?wsdl"/>  
  <property name="endpointUrl" value="https://host/foo/HelloWorldWS"/>  
</action>
```

# Q&A

- Answers can be found here:

[http://docs.jboss.org/jbossesb/docs/4.10/manuals/html/Programmers\\_Guide](http://docs.jboss.org/jbossesb/docs/4.10/manuals/html/Programmers_Guide)