Apache Camel

Willem Jiang
Senior Engineer Progress Software
2010-08

BUSINESS MAKING PROGRESS™

Who is Willem Jiang ?

BUSINESS MAKING PROGRESS

- Senior Software Engineer at Progress Software

  Working on Apache project for about 5 years

  Full time Apache Camel and CXF committer and PMC member

- Contact

ningjiang@apache.org

http://willem.blogspot.com

http://jnn.javaeye.com

http://twitter.com/willemjiang

PROGRESS SOFTWARE

Agenda

BUSINESS MAKING PROGRESS

- What is Apache Camel
- EIP examples
- Some cool features of Camel
- Riding tips of camel
- Q and A

PROGRESS SOFTWARE

What is Camel?

BUSINESS MAKING PROGRESS

PROGRESS SOFTWARE

A Little History of Camel

BUSINESS MAKING PROGRESS

- Camel is found as a sub project of ActiveMQ
- The first code is committed
  - r519901 | jstrachan | 2007-03-19 11:54:57 +0100 (Mon, 19 Mar 2007) | 1 line
  - Initial checkin of Camel routing library
- Some codes were came from ServiceMix
  - ServiceMix EIP component
  - some other ServiceMix components
- Apache Camel 1.0 released June 2007
- Apache Camel 2.0 released August 2009
- Apache Camel 2.4 released July 2010
- Apache Camel is more than 3 years old

PROGRESS SOFTWARE

ServiceMix, ActiveMQ, CXF and Camel

BUSINESS MAKING PROGRESS

PROGRESS SOFTWARE

ServiceMix, ActiveMQ, CXF and Camel

BUSINESS MAKING PROGRESS

- ServiceMix is an open source ESB.

PROGRESS SOFTWARE

ServiceMix, ActiveMQ, CXF and Camel

BUSINESS MAKING PROGRESS

- ServiceMix is an open source ESB.
- ActiveMQ is an open source messaging provider.

PROGRESS SOFTWARE

ServiceMix, ActiveMQ, CXF and Camel

BUSINESS MAKING PROGRESS.

- ServiceMix is an open source ESB.
- ActiveMQ is an open source messaging provider.
- CXF is an open source service framework.

PROGRESS SOFTWARE

ServiceMix, ActiveMQ, CXF and Camel

BUSINESS MAKING PROGRESS.

- ServiceMix is an open source ESB.
- ActiveMQ is an open source messaging provider.
- CXF is an open source service framework.

Camel is an open source integration framework based on known Enterprise Integration Patterns.

PROGRESS SOFTWARE

Integration Framework

BUSINESS MAKING PROGRESS.

PROGRESS SOFTWARE

Integration Framework

BUSINESS MAKING PROGRESS

- Why we need to do the integration?

PROGRESS SOFTWARE

Integration Framework

BUSINESS MAKING PROGRESS

- Why we need to do the integration?
  - Application is built on different tech stacks

PROGRESS SOFTWARE

Integration Framework

BUSINESS MAKING PROGRESS.

- Why we need to do the integration?
  - Application is built on different tech stacks
  - We want to make apps work together

PROGRESS SOFTWARE

Integration Framework

BUSINESS MAKING PROGRESS.

- Why we need to do the integration?
  - Application is built on different tech stacks
  - We want to make apps work together
- How can we do it ?

PROGRESS SOFTWARE

Integration Framework

BUSINESS MAKING PROGRESS

- Why we need to do the integration?
  - Application is built on different tech stacks
  - We want to make apps work together
- How can we do it ?
  - Big Database (Integration the data, it's not a good one)

PROGRESS SOFTWARE

Integration Framework

BUSINESS MAKING PROGRESS

- Why we need to do the integration?
  - Application is built on different tech stacks
  - We want to make apps work together
- How can we do it ?
  - Big Database (Integration the data, it's not a good one)
  - Message follow (Let the Applications talk to each other)

PROGRESS SOFTWARE

Integration Framework

BUSINESS MAKING PROGRESS

- Why we need to do the integration?
  - Application is built on different tech stacks
  - We want to make apps work together
- How can we do it ?
  - Big Database (Integration the data, it's not a good one)
  - Message follow (Let the Applications talk to each other)
  - Event Driven (High performance with a good scalability)

PROGRESS SOFTWARE

Integration Framework

BUSINESS MAKING PROGRESS

- Why we need to do the integration?
  - Application is built on different tech stacks
  - We want to make apps work together
- How can we do it ?
  - Big Database (Integration the data, it's not a good one)
  - Message follow (Let the Applications talk to each other)
  - Event Driven (High performance with a good scalability)
- How about an Integration Framework ?

PROGRESS SOFTWARE

Integration Framework

BUSINESS MAKING PROGRESS

- Why we need to do the integration?
  - Application is built on different tech stacks
  - We want to make apps work together
- How can we do it ?
  - Big Database (Integration the data, it's not a good one)
  - Message follow (Let the Applications talk to each other)
  - Event Driven (High performance with a good scalability)
- How about an Integration Framework ?
  - Your application code will not touch with any middleware API

PROGRESS SOFTWARE

Integration Framework

BUSINESS MAKING PROGRESS.

- Why we need to do the integration?
  - Application is built on different tech stacks
  - We want to make apps work together
- How can we do it ?
  - Big Database (Integration the data, it's not a good one)
  - Message follow (Let the Applications talk to each other)
  - Event Driven (High performance with a good scalability)
- How about an Integration Framework ?
  - Your application code will not touch with any middleware API
  - You can focus on your business

PROGRESS SOFTWARE

Integration Framework

BUSINESS MAKING PROGRESS

- Why we need to do the integration?
  - Application is built on different tech stacks
  - We want to make apps work together
- How can we do it ?
  - Big Database (Integration the data, it's not a good one)
  - Message follow (Let the Applications talk to each other)
  - Event Driven (High performance with a good scalability)
- How about an Integration Framework ?
  - Your application code will not touch with any middleware API
  - You can focus on your business
  - Let the framework do the heave lift work for you

PROGRESS SOFTWARE

Integration Framework

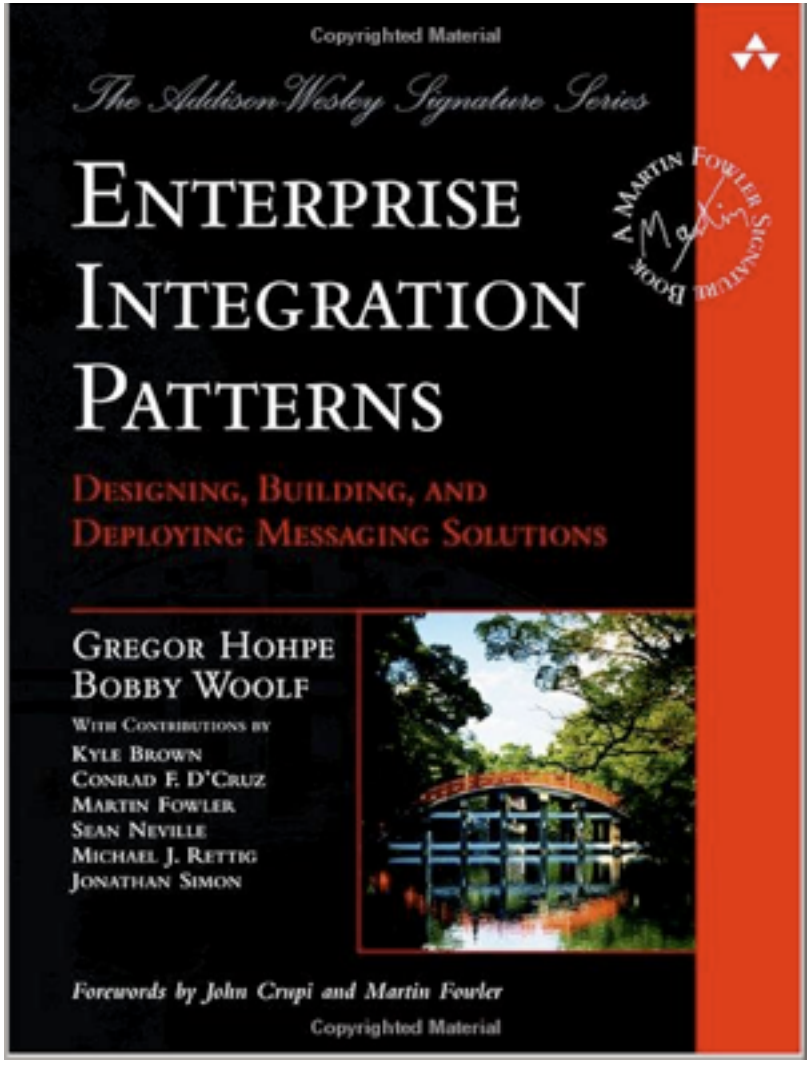
BUSINESS MAKING PROGRESS.

- Why we need to do the integration?
  - Application is built on different tech stacks
  - We want to make apps work together
- How can we do it ?
  - Big Database (Integration the data, it's not a good one)
  - Message follow (Let the Applications talk to each other)
  - Event Driven (High performance with a good scalability)
- How about an Integration Framework ?
  - Your application code will not touch with any middleware API
  - You can focus on your business
  - Let the framework do the heave lift work for you

A Camel can carry 4 times as much load as other beasts of burden!

PROGRESS SOFTWARE

Book by Gregor & Bobby!



Agenda

BUSINESS MAKING PROGRESS

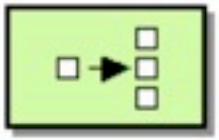
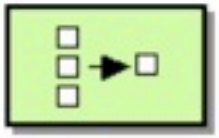
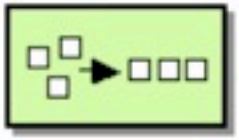
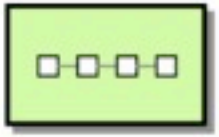
- What is Apache Camel
- EIP examples
- Some cool features of Camel
- Riding tips of camel
- Q and A

PROGRESS SOFTWARE

Message Routing

Message Routing in EIP

BUSINESS MAKING PROGRESS.

	Content Based Router	How do we handle a situation where the implementation of a single logical function (e.g., inventory check) is spread across multiple physical systems?
	Message Filter	How can a component avoid receiving uninteresting messages?
	Dynamic Router	How can you avoid the dependency of the router on all possible destinations while maintaining its efficiency?
	Recipient List	How do we route a message to a list of dynamically specified recipients?
	Splitter	How can we process a message if it contains multiple elements, each of which may have to be processed in a different way?
	Aggregator	How do we combine the results of individual, but related messages so that they can be processed as a whole?
	Resequencer	How can we get a stream of related but out-of-sequence messages back into the correct order?
	Routing Slip	How do we route a message consecutively through a series of processing steps when the sequence of steps is not known at design-time and may vary for each message?
	Throttler	How can I throttle messages to ensure that a specific endpoint does not get overloaded, or we don't exceed an agreed SLA with some external service?
	Delayer	How can I delay the sending of a message?
	Load Balancer	How can I balance load across a number of endpoints?
	Multicast	How can I route a message to a number of endpoints at the same time?
	Loop	How can I repeat processing a message in a loop?

PROGRESS SOFTWARE

Simple Routing

Simple Routing


```
from("file:src/data?noop=true").
    to("jms:queue:myqueue");
```

More Simple Routing

More Simple Routing


```
from("rmi://localhost:1099/patch/to/service").
    to("netty:tcp://remotehost:1234");
```

Camel Components

BUSINESS MAKING PROGRESS.

ActiveMQ	File	JBI	MINA	RMI	TCP
ActiveMQ Journal	FIX	JCR	Mock	RNC	Test
AMQP	Flatpack	JDBC	MSMQ	RNG	Timer
Atom	FTP	Jetty	MSV	SEDA	UDP
Bean	Hibernate	JMS	Multicast	SFTP	Validation
CXF	HTTP	JPA	POJO	SMTP	Velocity
DataSet	iBATIS	JT/400	POP	Spring Integration	VM
Direct	IMAP	List	Quartz	SQL	XMPP
Esper	IRC	Log	Queue	Stream	XQuery
Event	JavaSpace	Mail	Ref	String Template	XSLT

http://camel.apache.org/components.html

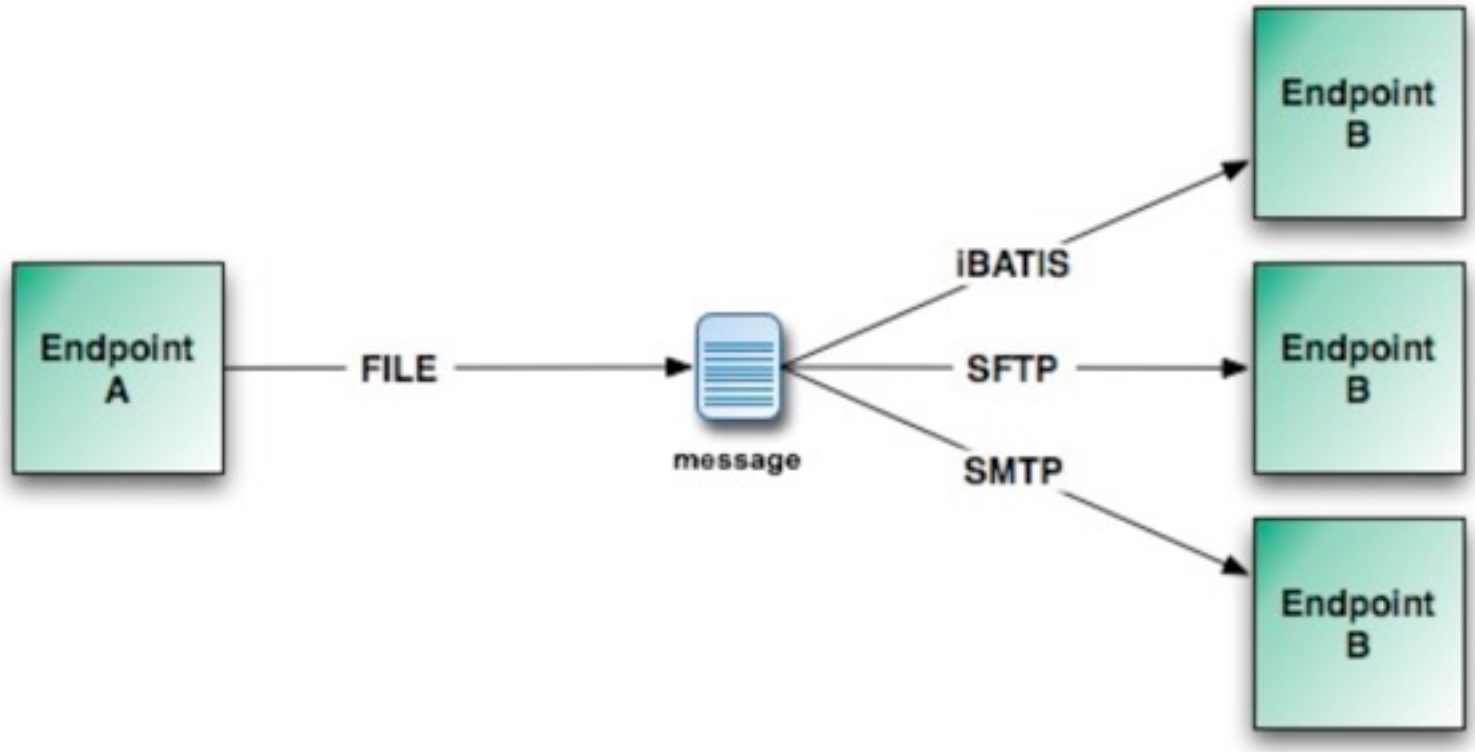
PROGRESS SOFTWARE

Pipeline

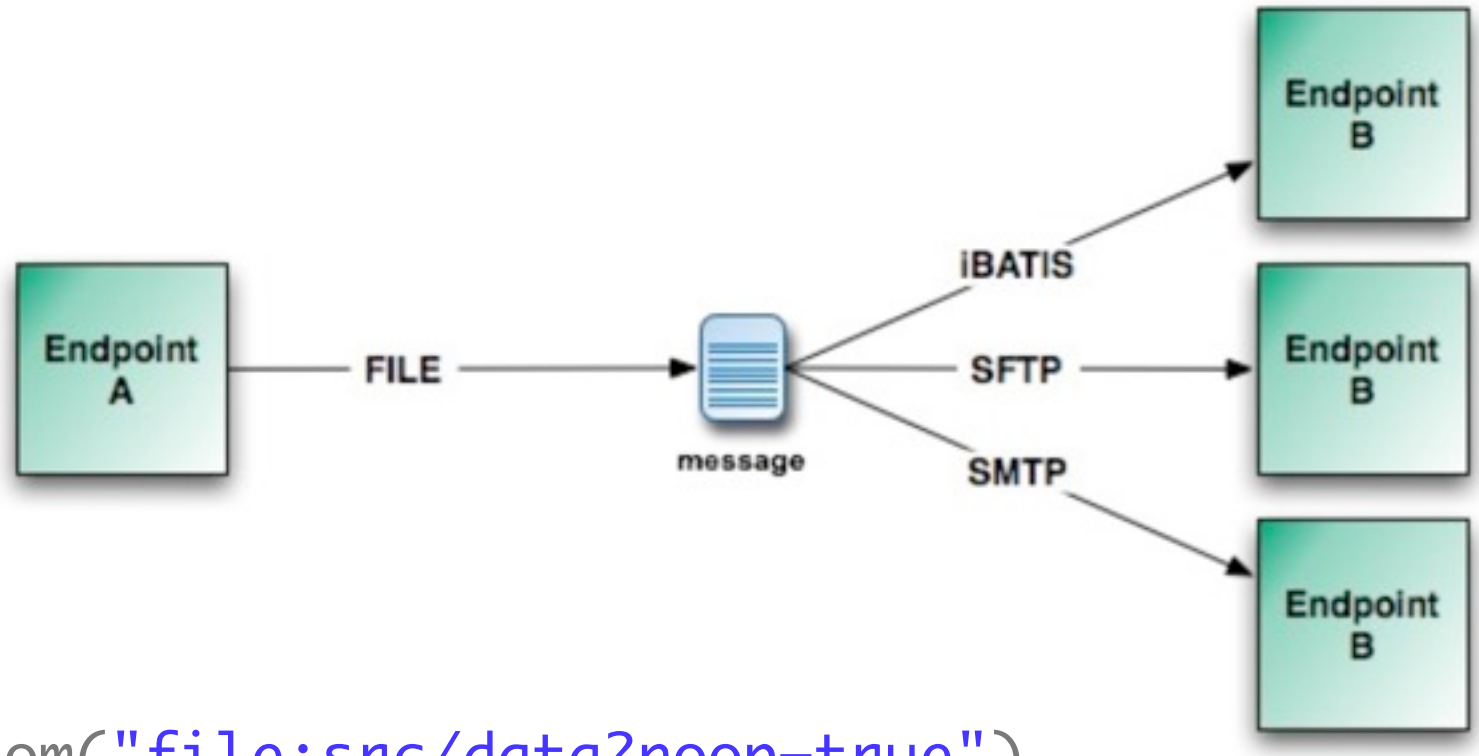
Pipeline


```
from("EndpointA").
    pipleline("EndpointB", "EndpointQ",
"EndpointR");
```

Multicast Routing



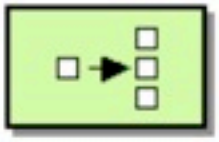
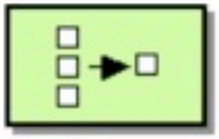
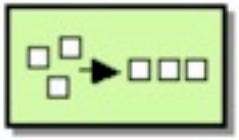
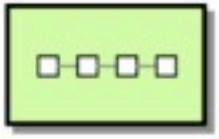
Multicast Routing




```
from("file:src/data?noop=true").
   multicast("ibatis://xxx", "sftp://xxx",
"smtp://xxx");
```

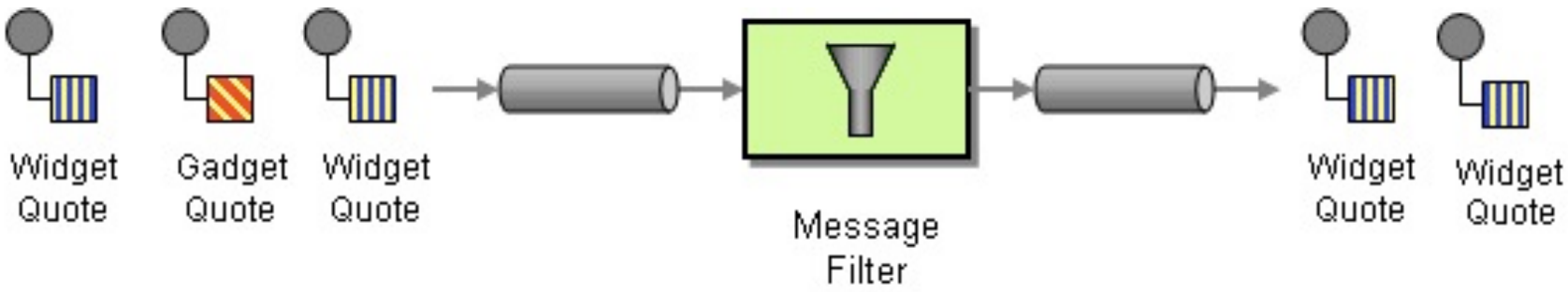
Some examples of the EIP implementation

BUSINESS MAKING PROGRESS.

	Content Based Router	How do we handle a situation where the implementation of a single logical function (e.g., inventory check) is spread across multiple physical systems?
	Message Filter	How can a component avoid receiving uninteresting messages?
	Dynamic Router	How can you avoid the dependency of the router on all possible destinations while maintaining its efficiency?
	Recipient List	How do we route a message to a list of dynamically specified recipients?
	Splitter	How can we process a message if it contains multiple elements, each of which may have to be processed in a different way?
	Aggregator	How do we combine the results of individual, but related messages so that they can be processed as a whole?
	Resequencer	How can we get a stream of related but out-of-sequence messages back into the correct order?
	Routing Slip	How do we route a message consecutively through a series of processing steps when the sequence of steps is not known at design-time and may vary for each message?
	Throttler	How can I throttle messages to ensure that a specific endpoint does not get overloaded, or we don't exceed an agreed SLA with some external service?
	Delayer	How can I delay the sending of a message?
	Load Balancer	How can I balance load across a number of endpoints?
	Multicast	How can I route a message to a number of endpoints at the same time?
	Loop	How can I repeat processing a message in a loop?

PROGRESS SOFTWARE

Message Filter



Message Filter

BUSINESS MAKING PROGRESS

```
<camelContext xmlns="http://camle.apache.org/schema/spring">
  <route>
    <from uri="activemq:topic:Quotes"/>
    <filter>
      <xpath>/quote/product = ‘widget’</xpath>
      <to uri="mqseries:WidgetQuotes"/>
    </filter>
  </route>
</camelContext>
```

```
from("activemq:topic:Quotes).
  filter().xpath("/quote/product = ‘widget’").
    to("mqseries:WidgetQuotes");
```


PROGRESS SOFTWARE

Language Support For Message Processing

BUSINESS MAKING PROGRESS

- BeanShell
- Javascript
- Groovy
- Python
- PHP
- Ruby
- JSP EL
- OGNL
- SQL
- Xpath
- XQuery
- Simple

PROGRESS SOFTWARE

Message Filter : Spring XML

BUSINESS MAKING PROGRESS

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:schemaLocation="http://www.springframework.org/schema/beans
       http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
       http://camel.apache.org/schema/spring
       http://camel.apache.org/schema/spring/camel-spring.xsd">

 <camelContext xmlns="http://camel.apache.org/schema/spring">
    <route>
      <from uri="activemq:topic:Quotes"/>
      <filter>
        <xpath>/quote/product = 'widget'</xpath>
        <to uri="mqseries:WidgetQuotes"/>
      </filter>
    </route>
  </camelContext>

</beans>
```


PROGRESS SOFTWARE

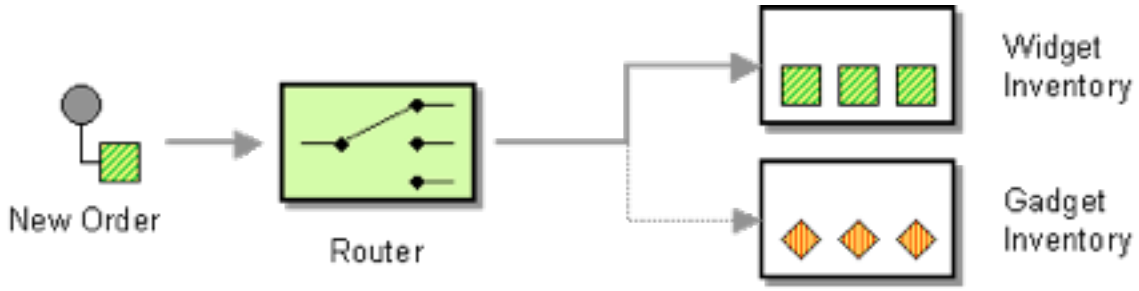
Message Filter : Java Complete

BUSINESS MAKING PROGRESS

```
package com.acme.quotes;

import org.apache.camel.builder.RouteBuilder;

public class MyRouteBuilder extends RouteBuilder {

    public void configure() {

        // forward widget quotes to MQSeries
        from("activemq:topic:Quotes).
                filter().xpath("/quote/product = ‘widget’").
                to("mqseries:WidgetQuotes");
 }
}
```


PROGRESS SOFTWARE

Content Base Router




```
from("activemq:NewOrders”).
  choice().when("/quote/product = ‘widget’").
    to("activemq:Orders.Widgets").
  choice().when("/quote/product = ‘gadget’").
    to("activemq:Orders.Gadgets").
  otherwise().to("activemq:Orders.Bad");
```

Content Based Router

```
<camelContext xmlns="http://camel.apache.org/schema/spring">
  <route>
    <from uri="activemq:NewOrders"/>
    <choice>
      <when>
        <xpath>/order/product = 'widget'</xpath>
        <to uri="activemq:Orders.Widgets"/>
      </when>
      <when>
        <xpath>/order/product = 'gadget'</xpath>
        <to uri="activemq:Orders.Gadgets"/>
      </when>
      <otherwise>
        <to uri="activemq:Orders.Bad"/>
      </otherwise>
    </choice>
  </route>
</camelContext>
```

Agenda

BUSINESS MAKING PROGRESS

- What is Apache Camel
- EIP examples
- Some cool features of Camel
- Riding tips of camel
- Q and A

PROGRESS SOFTWARE

Some Cool features of Camel

BUSINESS MAKING PROGRESS

- Routing and mediation engine
- Domain-specific language
- Extensive component library
- Beans
- Type Conversion
- Data Format
- Test tools
- Maven tools
- ...

PROGRESS SOFTWARE

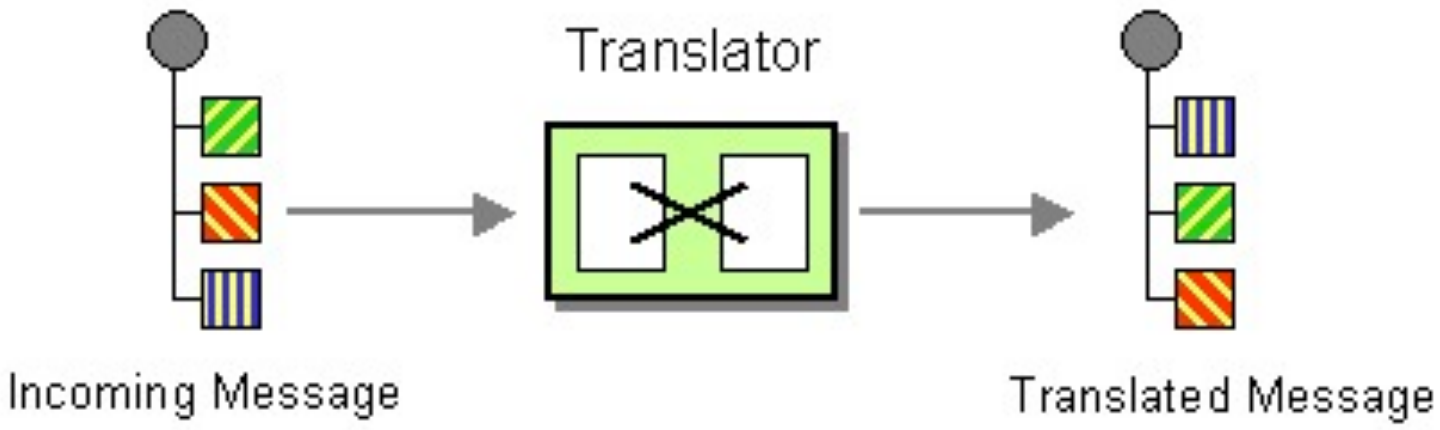
Beans

BUSINESS MAKING PROGRESS

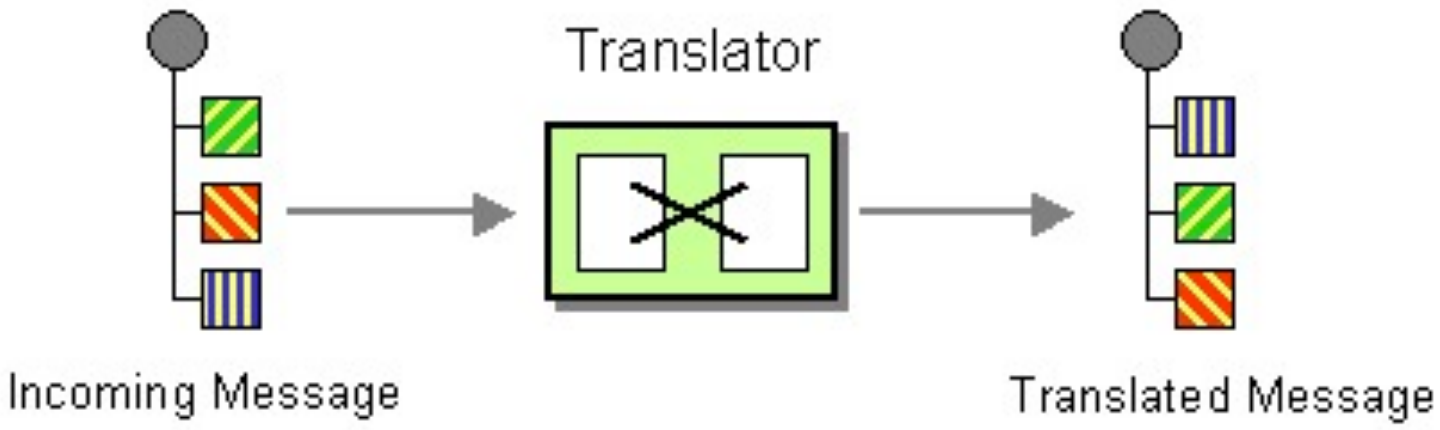
- Powerful bean integration
  - Adapt to you beans
  - EIP as @annotations
    - @Produce
    - @Consumer
    - @RecipientList
    - @RoutingSlip

PROGRESS SOFTWARE

Bean as a Message Translator

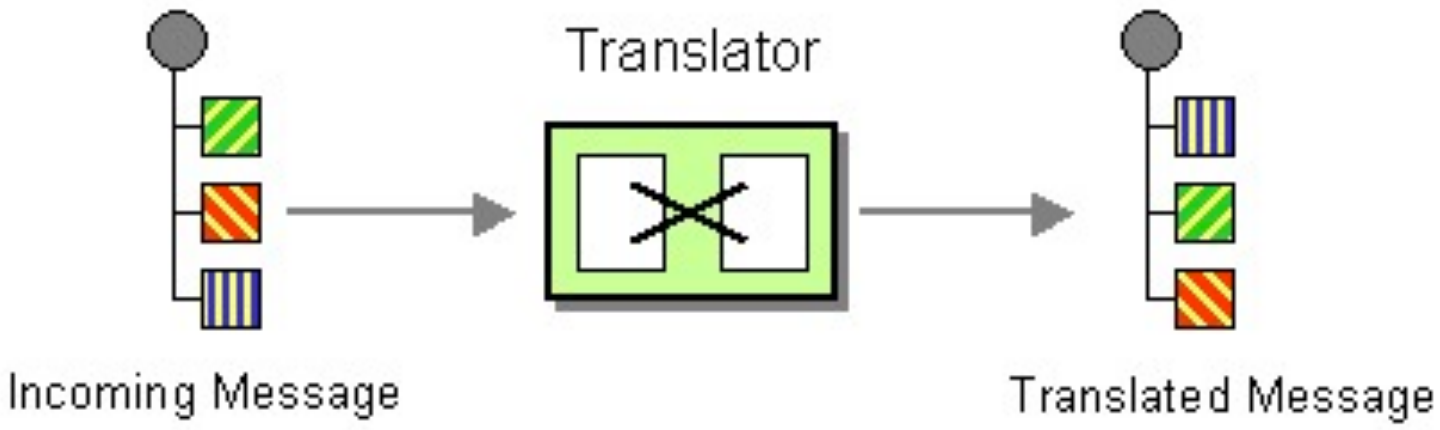


Bean as a Message Translator



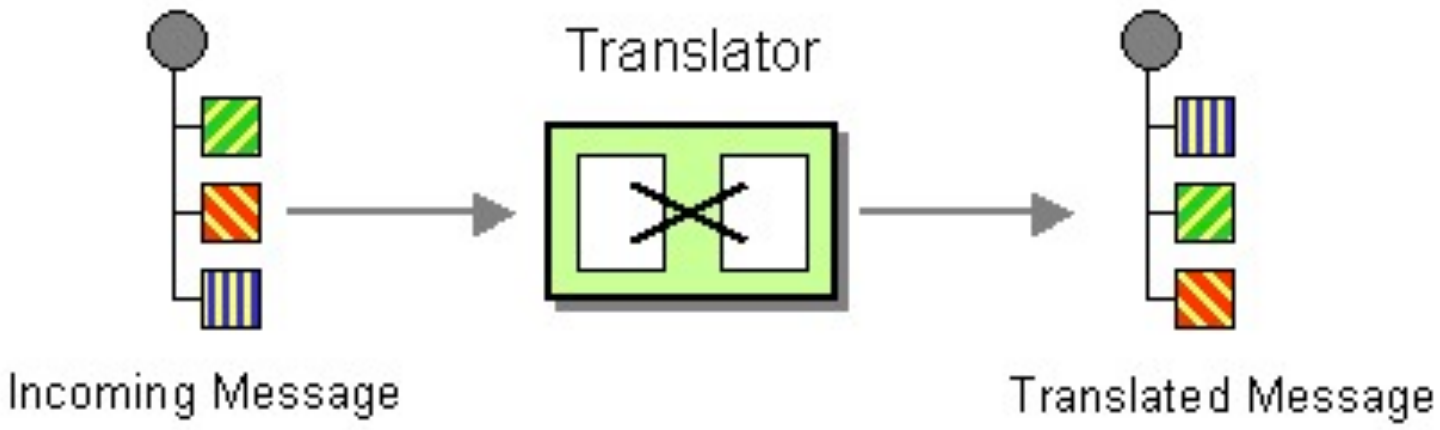

```
from("activemq:Incoming”).
  beanRef("myBeanName").
    to("activemq:Outgoing");
```

Bean as a Message Translator




```
from("activemq:Incoming”).
  beanRef("myBeanName").
    to("activemq:Outgoing");

public class Foo {

  public void someMethod(String name) {
    ...
  }
}
```

Bean as a Message Translator




```
from("activemq:Incoming”).
  beanRef("myBeanName").
    to("activemq:Outgoing");

public class Foo {

  public void someMethod(String name) {
    ...
  }
}
```

Binding Method Arguments

BUSINESS MAKING PROGRESS

```
public class Foo {

  public void onCheese(
    @XPath("/foo/bar") String name,
    @Header("JMSCorrelationID") String id) {
    ...
  }
}
```


PROGRESS SOFTWARE

Sending the message

```
public class MyCoolBean {

    @Produce(uri = "log:foo")
    protected ProducerTemplate producer;

    public void sendMsg() {
        producer.sendBody("Hello World");
    }

}
```

Consume the message and EIP annotation

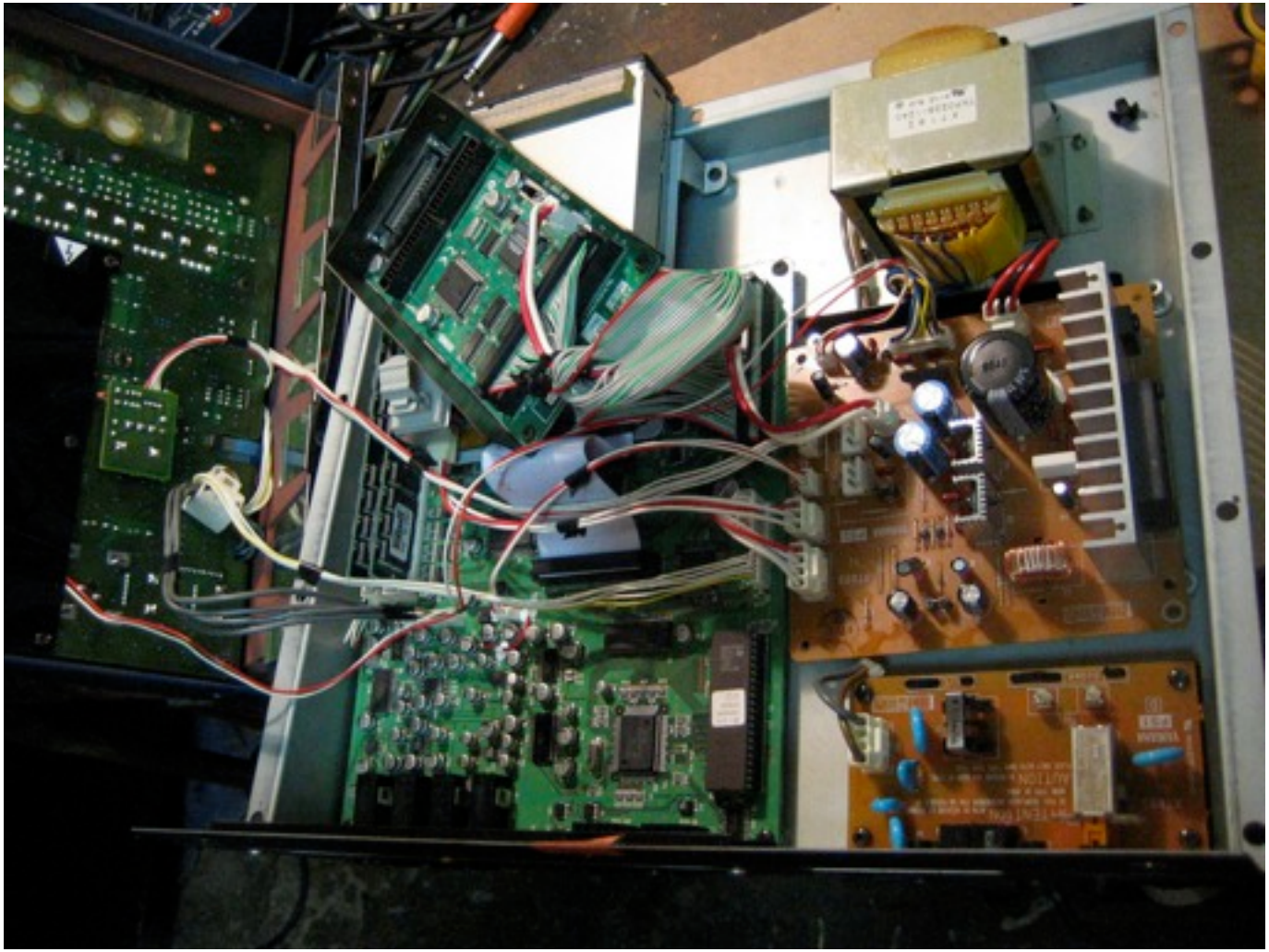
BUSINESS MAKING PROGRESS

```
public class RouterBean {

    @Consume(uri = "direct:start")
    @RecipientList
    public String[] route(String body) {
        return new String[]{"mock:a",
"mock:b"};
    }
}
```


PROGRESS SOFTWARE

Type Conversion

BUSINESS MAKING PROGRESS



PROGRESS SOFTWARE

Type Conversion

BUSINESS MAKING PROGRESS

```
package com.acme.foo.converters;

import org.apache.camel.Converter;
import java.io.*;

@Converter
public class IOConverter {

    @Converter
    public static InputStream toInputStream(File file)
throws FileNotFoundException {
        return new BufferedInputStream(new FileInputStream
(file));
    }
}
```

```
# META-INF/services/org/apache/camel/TypeConverter

com.acme.foo.converters
```


PROGRESS SOFTWARE

Type Conversion

BUSINESS MAKING PROGRESS

```
from("direct:start").convertBodyTo
(InputStream.class).to("mock:result");
```

```
from("direct:start").process(new Processor(){
 public void process(Exchange exchange) {
  Message in = exchange.getIn();
  in.setBody(in.getBody(InputStream.class));
 }
}).to("mock:result");
```


PROGRESS SOFTWARE

DataFormat

BUSINESS MAKING PROGRESS

- Camel has 18 Data Formats

bindy	protobuf
castor	serialization
csv	soap
crypto	tidy markup
flatpack	xml beans
gzip	xml security
hl7	xstream
jaxb	zip
json	dozer

PROGRESS SOFTWARE

Data Format

BUSINESS MAKING PROGRESS

```
from("activemq:QueueWithJavaObjects).
  marshal().jaxb().
    to("mqseries:QueueWithXmlMessages");

from("activemq:QueueWithXmlMessages).
  unmarshal().jaxb().
    to("mqseries:QueueWithJavaObjects");
```


PROGRESS SOFTWARE

Test kit

BUSINESS MAKING PROGRESS

- Mock Endpoint
  - You can use it check if the message is processed rightly
  - jMock like API, declarative expectations
  - The expectations can be asserted in a test case

- Test
  - JUnit based (3.x, 4.x)
  - Supports Spring
  - Easy to test
  - Quick prototyping

PROGRESS SOFTWARE

Test support

BUSINESS MAKING PROGRESS

```
@ContextConfiguration
public class ProduceTemplateTest extends AbstractJUnit38SpringContextTests {
    @Autowired
    protected ProducerTemplate producer;

    @EndpointInject(uri = "mock:result")
    protected MockEndpoint result;

    public void testProducerTemplate() throws Exception {
        result.expectedBodiesReceived("hello");
        // lets send a message
        producer.sendBody("direct:start", "hello");
        result.assertIsSatisfied();
    }
}
```


PROGRESS SOFTWARE

Test support

BUSINESS MAKING PROGRESS

```
@ContextConfiguration
public class ProduceTemplateTest extends AbstractJUnit38SpringContextTests {
    @Autowired
    protected ProducerTemplate producer;

    @EndpointInject(uri = "mock:result")
    protected MockEndpoint result;

    public void testProducerTemplate() throws Exception {
        result.expectedBodiesReceived("hello");
        // lets send a message
        producer.sendBody("direct:start", "hello");
        result.assertIsSatisfied();
    }
}
```

```
<camelContext id="camelContext" xmlns="http://camel.apache.org/schema/spring">
    <template id="camelTemplate" />
    <route>
       <from uri="direct:start"/>
       <to uri="mock:result"/>
    </route>
 </camelContext>
```


PROGRESS SOFTWARE

Maven tooling

BUSINESS MAKING PROGRESS

- Maven Archetypes to create new projects

```
mvn archetype:generate \
  -DarchetypeGroupId=org.apache.camel.archetypes \
  -DarchetypeArtifactId=camel-archetype-java \
  -DarchetypeVersion=2.4.0
```

- Maven goals to run the project
  - mvn camel:run

PROGRESS SOFTWARE

Agenda

BUSINESS MAKING PROGRESS

- What is Apache Camel
- EIP examples
- Some cool features of Camel
- Riding tips of camel
- Q and A

PROGRESS SOFTWARE

Riding tips of camel

Running Camel

BUSINESS MAKING PROGRESS

```
CamelContext context = new DefaultCamelContext();
context.addRoutes(new MyRouteBuilder());
context.start();
```


PROGRESS SOFTWARE

Running Camel

BUSINESS MAKING PROGRESS

```
CamelContext context = new DefaultCamelContext();
context.addRoutes(new MyRouteBuilder());
context.start();
```

```
<camelContext id="camel" xmlns="http://camel.apache.org/
schema/spring">
    <route>
      <from uri="direct:start"/>
      <from to="mock:result"/>
    </route>
  </camelContext>
```


PROGRESS SOFTWARE

Running Camel

BUSINESS MAKING PROGRESS

```
CamelContext context = new DefaultCamelContext();
context.addRoutes(new MyRouteBuilder());
context.start();
```

```
<camelContext id="camel" xmlns="http://camel.apache.org/
schema/spring">
    <route>
      <from uri="direct:start"/>
      <from to="mock:result"/>
    </route>
  </camelContext>
```

```
<camelContext id="camel" xmlns="http://camel.apache.org/
schema/spring">
    <package>com.exaple.your.package</package>
  </camelContext>
```


PROGRESS SOFTWARE

Camel Riding from Java

BUSINESS MAKING PROGRESS

- /META-INF/spring/camelContext.xml
- set the CLASSPATH
- java org.apache.camel.spring.Main

PROGRESS SOFTWARE

Maven Tooling

BUSINESS MAKING PROGRESS

```
<project>
...
   <build>
   <plugins>
      <plugin>
        <groupId>org.apache.camel</groupId>
        <artifactId>camel-maven-plugin</artifactId>
      </plugin>
    </plugins>
  </build>
  </project>
```

mvn camel:run

PROGRESS SOFTWARE

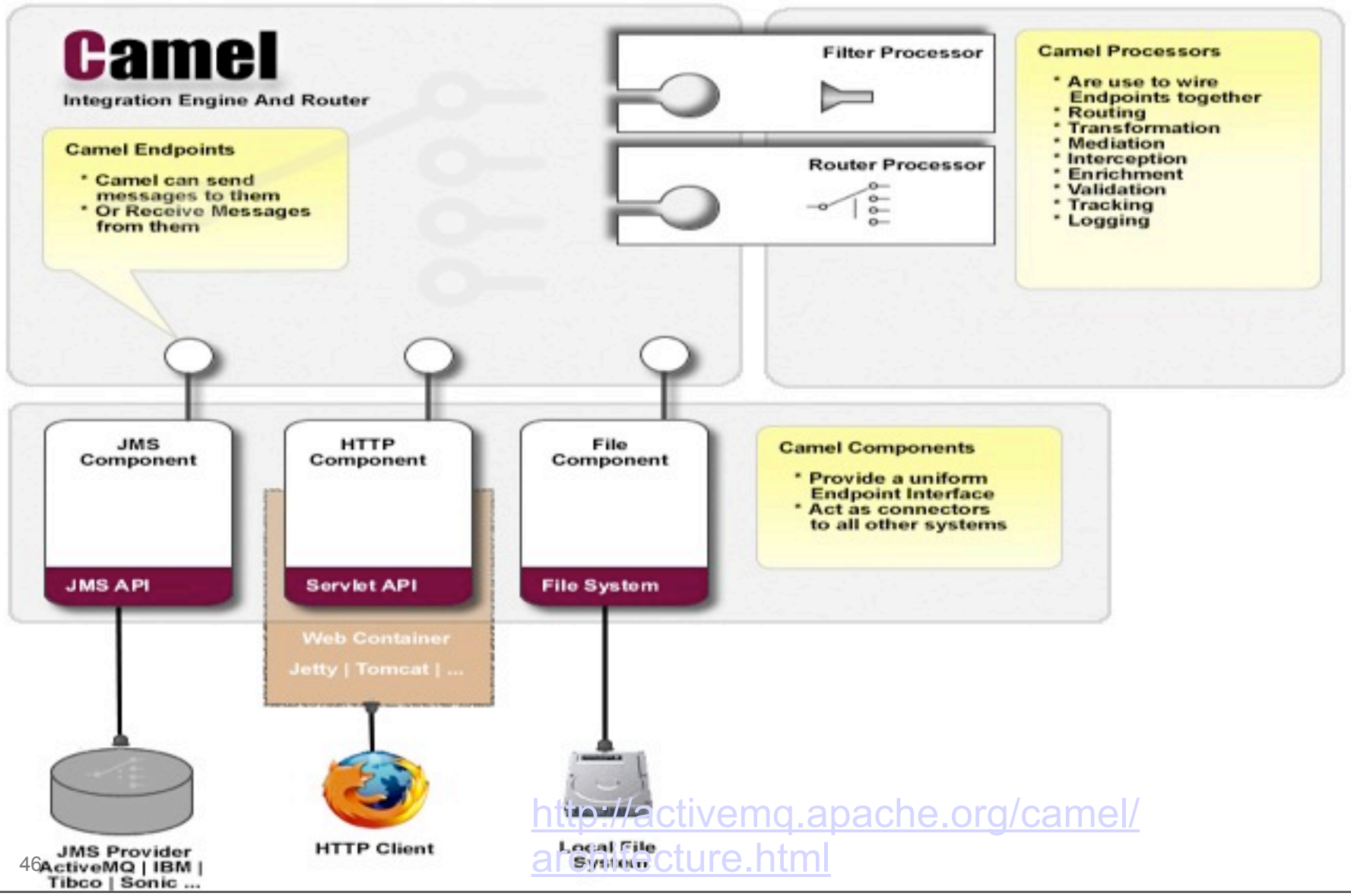
Running Camel

BUSINESS MAKING PROGRESS

- Lightweight
  - You can use it as Java library
- Embeddable
  - You can deploy it in most of containers
- Know Deployment Option
  - Standalone Java Application
  - Web Application
  - J2EE Application
  - JBI Container
  - OSGi
  - Google App Engine
  - Java Web Start
  - Spring Application

PROGRESS SOFTWARE

How camel do this routing work ?



http://activemq.apache.org/camel/architecture.html

How to write your own component?

BUSINESS MAKING PROGRESS

- Using the Maven Archetypes to create the project
  - camel-archetype-component
- Component
  - The Endpoint Factory
- Endpoint
  - ScheduledPollEndpoint
  - DefaultEndpoint
  - ResourceEndpoint
- Consumer
- Producer

PROGRESS SOFTWARE

How to write your own routing rule in Camel

BUSINESS MAKING PROGRESS

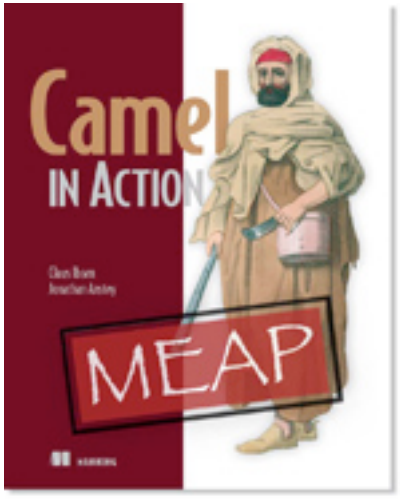
- What's the magic of from, to, choice .....
  - /camel-core/src/main/java/org/apache/camel/model
- Implementing it in DSL way
  - Defining the type class that you want
- Implementing it in a Spring configuration way
  - Adding the annotation for JAXB consuming

PROGRESS SOFTWARE

Q and A

BUSINESS MAKING PROGRESS

- Where do I get more information?
  - Camel website: http://camel.apache.org
  - Fuse website: http://fusesource.com
  - Camel in Action book: http://manning.com/ibsen



PROGRESS SOFTWARE

Thank You

BUSINESS MAKING PROGRESS.

PROGRESS SOFTWARE