

Riding the (Apache) Camel into the cloud!

James Strachan

FuseSource

A Progress Software Company

Your speaker today

■ James Strachan

- james@fusesource.com
- twitter: @jstrachan
- blog: <http://macstrac.blogspot.com/>

■ Software Fellow at FuseSource

- leaders in open source integration & messaging
- we provide training, consulting, support, distributions & tools for open source integration software

■ Open Source hacker

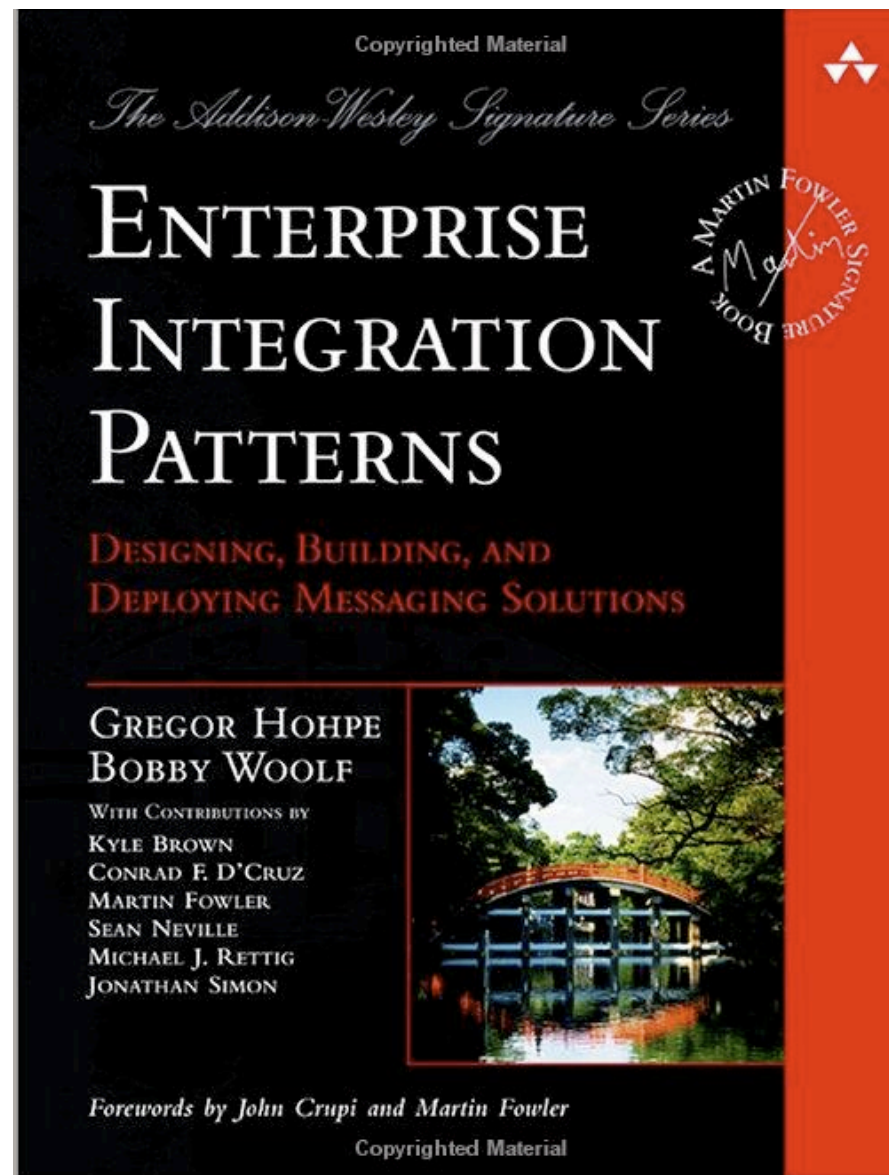
- created the Groovy programming language
- created Apache Camel
- co-founder of
 - Apache ActiveMQ, ServiceMix, Geronimo
 - Scalate
- lets not mention Jelly :)

- what's this Apache Camel thing anyway?
- what did the cloud ever do for us anyway?
- so how'd you ride this Camel in the cloud then?
- Q & A (hopefully...)

What are Enterprise Integration Patterns?


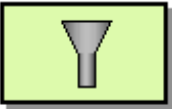
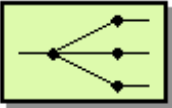
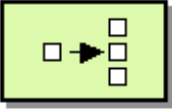

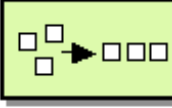


Book by Gregor & Bobby!



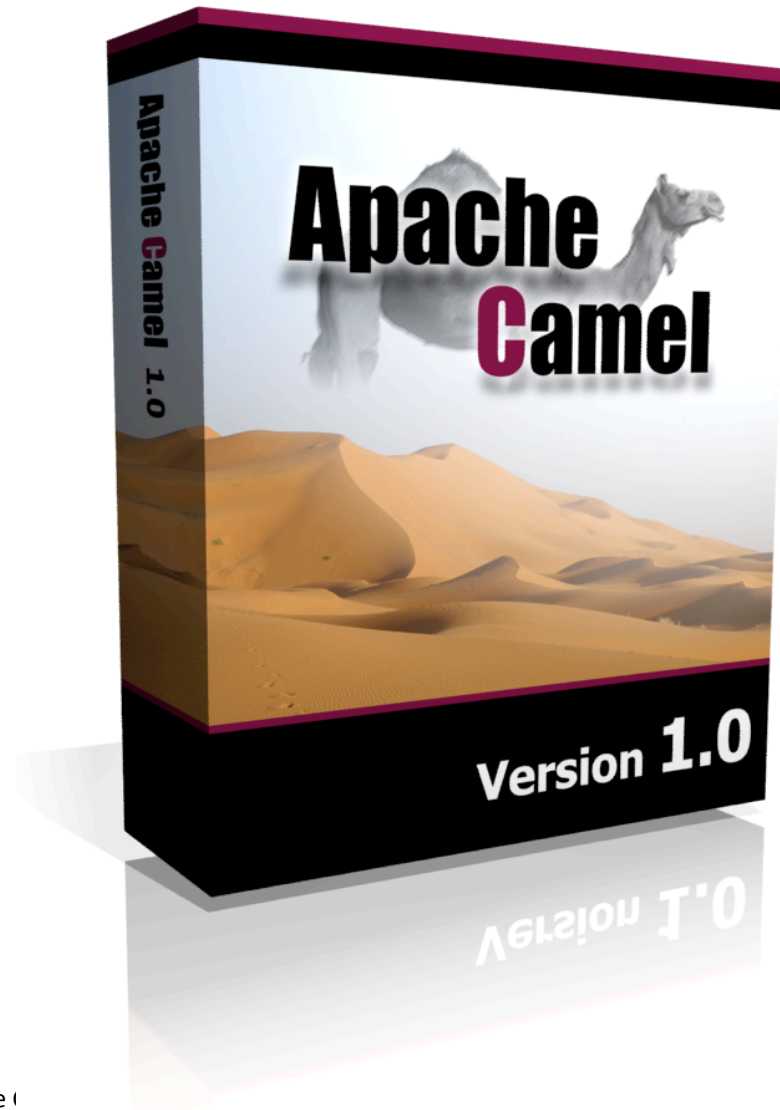
A selection of some of the patterns...

Message Routing

	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?
	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?
	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?

What is Apache Camel?

<http://camel.apache.org/>

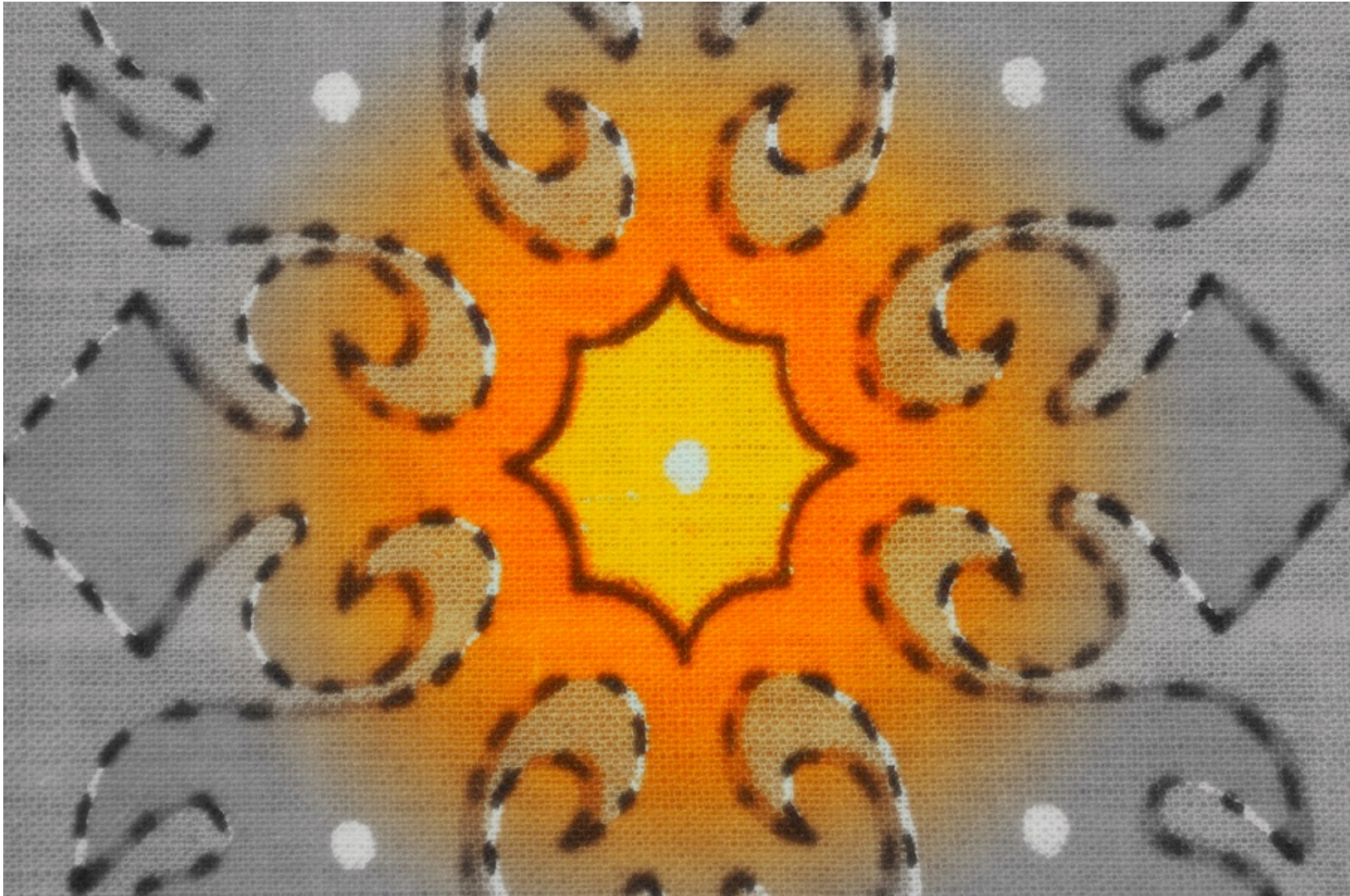


What is Apache Camel?

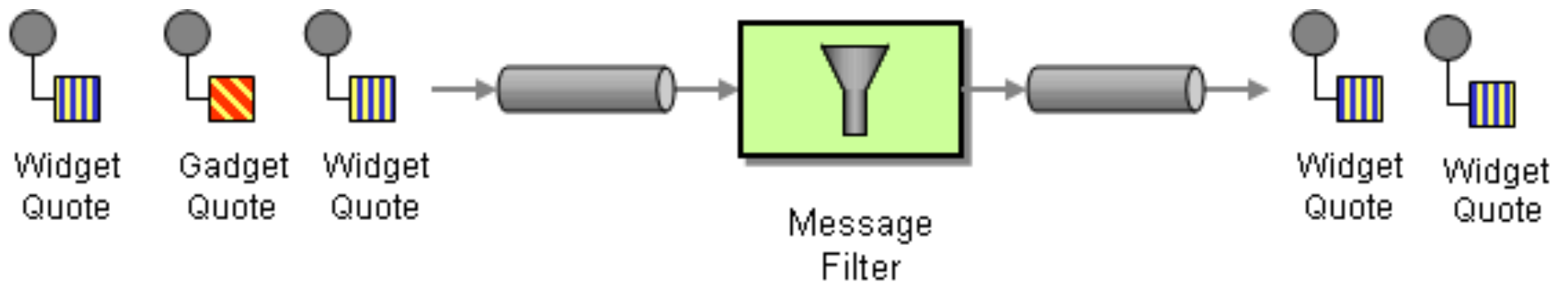
Apache Camel is a Powerful Open Source
Integration Framework
based on known
Enterprise Integration Patterns

<http://camel.apache.org/enterprise-integration-patterns.html>

Lets look at a pattern!



Message Filter



Message Filter : XML

```
<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>
```

Message Filter : Spring XML

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.0.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>
```

Message Filter : XML

```
<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>
```

Expressions & Predicates

15 Expression Languages

BeanShell	Python
EL	Ruby
Groovy	Simple
JavaScript	SpEL
JSR 223	SQL
OGNL	XPath
MVEL	XQuery
PHP	

URIs, Endpoints and Components (120+)

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

activemq	cxfrs	flatpack	jasypt
activemq-journal	cxfrs	freemarker	javaspace
amqp	dataset	ftp/ftps/sftp	jbi
atom	db4o	gae	jcr
bean	direct	hdfs	jdbc
bean validation	ejb	hibernate	jetty
browse	esper	hl7	jms
cache	event	http	jmx
cometd	exec	ibatis	jpa
crypto	file	irc	jt/400

120+ components...

language	properties	seda	stream
ldap	quartz	servlet	string-template
mail/imap/pop3	quickfix	sip	test
mina	ref	smooks	timer
mock	restlet	smpp	validation
msv	rmi	snmp	velocity
nagios	rnc	spring-integration	vm
netty	rng	spring-security	xmpp
nmr	rss	spring-ws	xquery
printer	scalate	sql	xslt

Message Filter : XML

```
<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>
```

Message Filter : Java

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

Message Filter : Java Complete

```
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");

    }
}
```

Message Filter : Scala

```
"direct:a" when(_.in == "<hello/>") {  
    to("mock:a")  
}
```

Create CamelContext in Spring

```
<camelContext xmlns="http://camel.apache.org/schema/spring">  
  <package>com.acme.quotes</package>  
</camelContext>
```

Create CamelContext in Java

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

IDE support

```
public class FilterRoute extends RouteBuilder {  
    public void configure() throws Exception {  
        Endpoint A = endpoint("activemq:queue: quote");  
        Endpoint B = endpoint("mq:quote");  
        Predicate isWidget = xpath("/quote/product = 'widget'");  
        from(A).fi  
    }  
}
```

		filter(Predicate predicate)	FilterDefinition
		filter()	ExpressionClause<FilterDefinition>
		filter(ExpressionDefinition expression)	FilterDefinition
		filter(String language, String expression)	FilterDefinition

Press ^_ to view all accessible classes

Documentation for filter(Predicate)

in: org.apache.camel:camel-core:2.3-SNAPSHOT

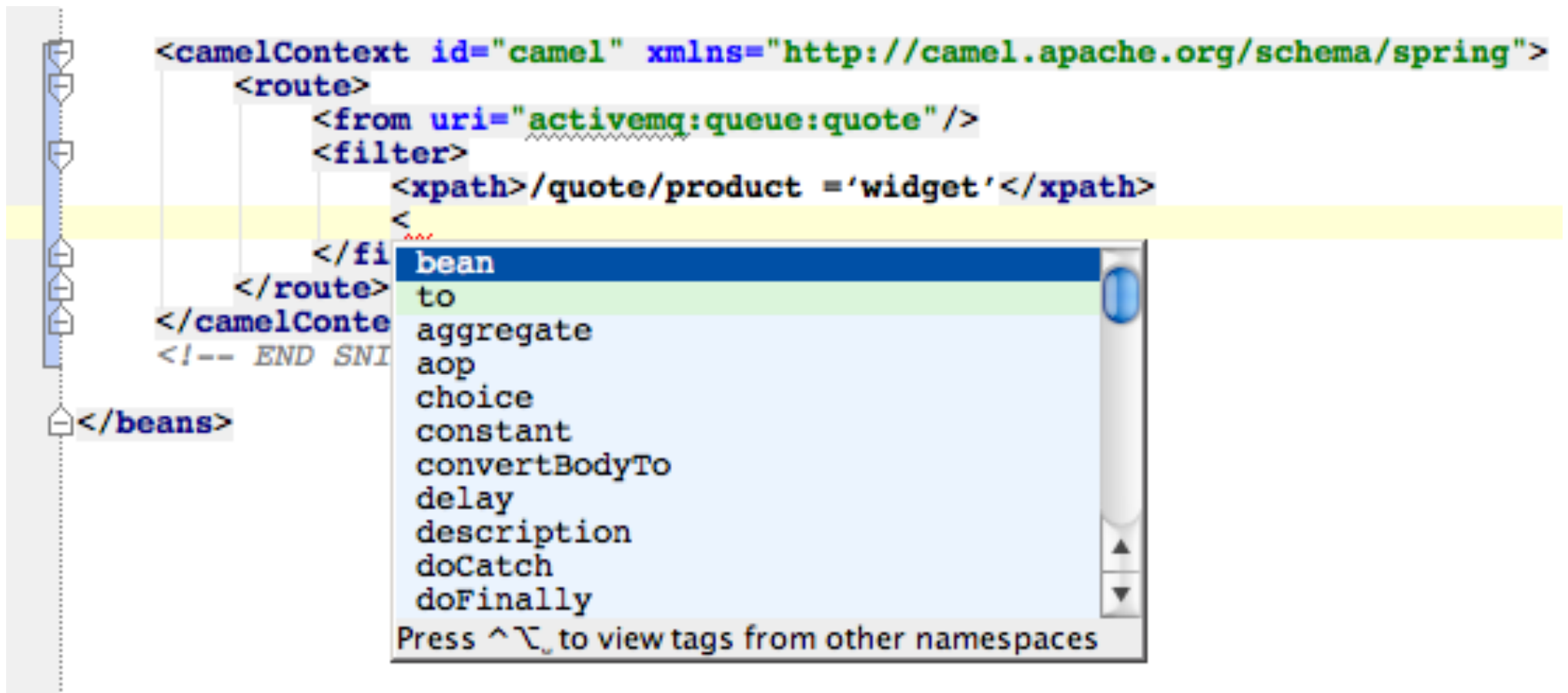
[org.apache.camel.model.ProcessorDefinition](#)

public [org.apache.camel.model.FilterDefinition](#) filter(
[Message Filter EIP](#): Creates a predicate which is applied and only if it is true then the exchange is forwarded to the destination

Parameters:
predicate - predicate to use

Returns:
the builder

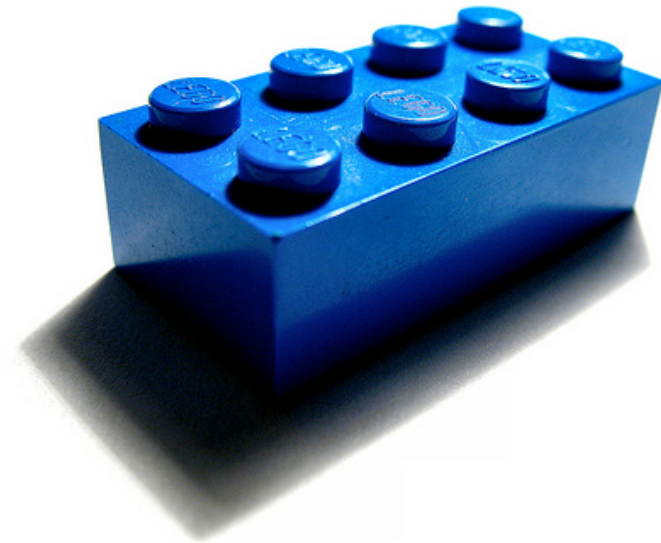
IDE support (XML)



Recap - core concepts of Camel

- Enterprise Integration Patterns
- Routing
- Domain Specific Language (DSL)
- Endpoints & URIs
- Predicates & Expressions
- Components (lots of 'em!)
- Test Kit

and much more ...



Beans



Bean as a Message Translator

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

Bean

```
public class Foo {  
  
    public String someMethod(String name) {  
        return "Hello " + name;  
    }  
}
```

Binding Beans to Camel Endpoints

```
public class Foo {  
  
    @Consume(uri="activemq:cheese")  
    public Object onCheese(String name) {  
        ...  
    }  
}
```

Binding Method Arguments

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

for more annotations see

<http://camel.apache.org/bean-integration.html>

Sending messages to endpoints

```
public class Foo {
    @Produce(uri="activemq:foo.bar")
    ProducerTemplate producer;

    public void doSomething() {
        if (whatever) {
            producer.sendBody("<hello>world!</hello>");
        }
    }
}
```

Sending messages to endpoints

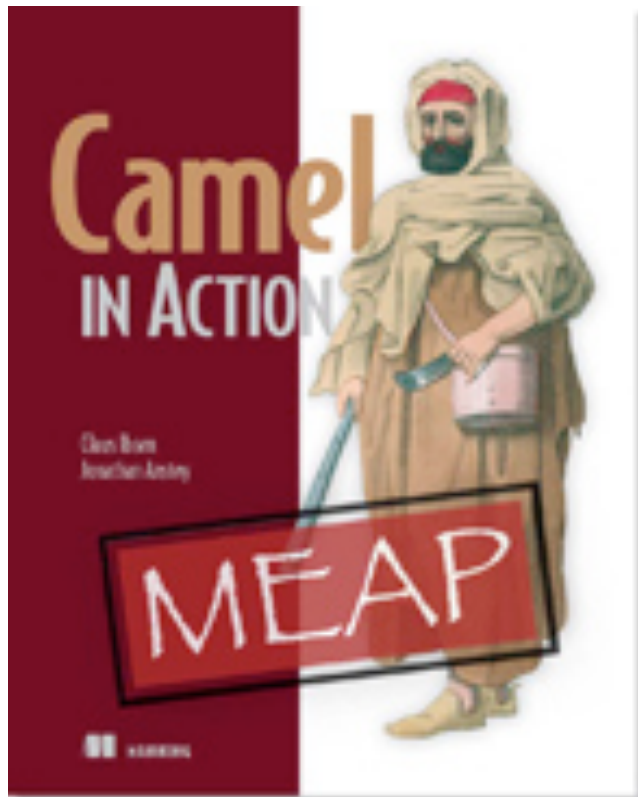
```
public interface MyListener {
    String sayHello(String name);
}

public class MyBean {
    @Produce(uri = "activemq:foo")
    protected MyListener producer;

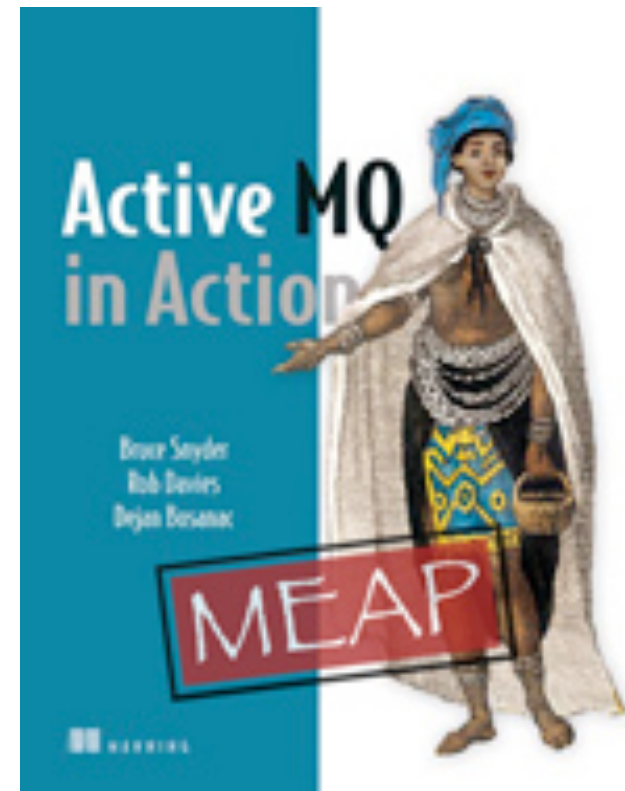
    public void doSomething() {
        // lets send a message
        String response = producer.sayHello("James");
    }
}
```


Fastest way to learn....

- Manning top-15 year to date (2010)



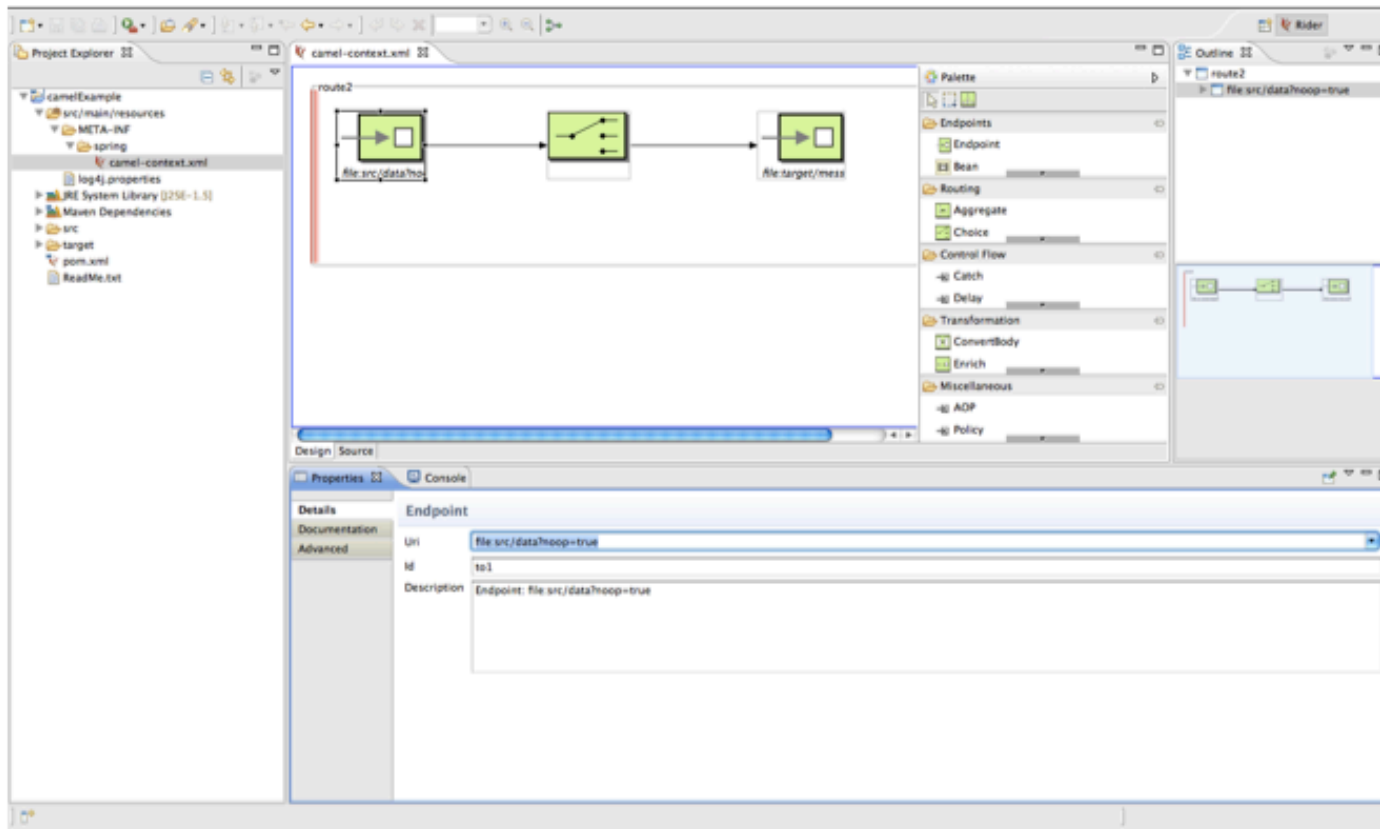
#10



#12

Fuse IDE: the fastest way to get riding!

- free eclipse based tool for Apache Camel & ActiveMQ etc
 - create projects, edit routes, run stuff, visualise, trace & diagnose



<http://fusesource.com/>

Cloud schmoud!



How we got here..

- Operating Systems
- Virtual Processes (JVM / .Net / LLVM)
- Application Server
- Virtualisation of hardware
- Clouds
- Multi-clouds?

Riding the camel into the cloud



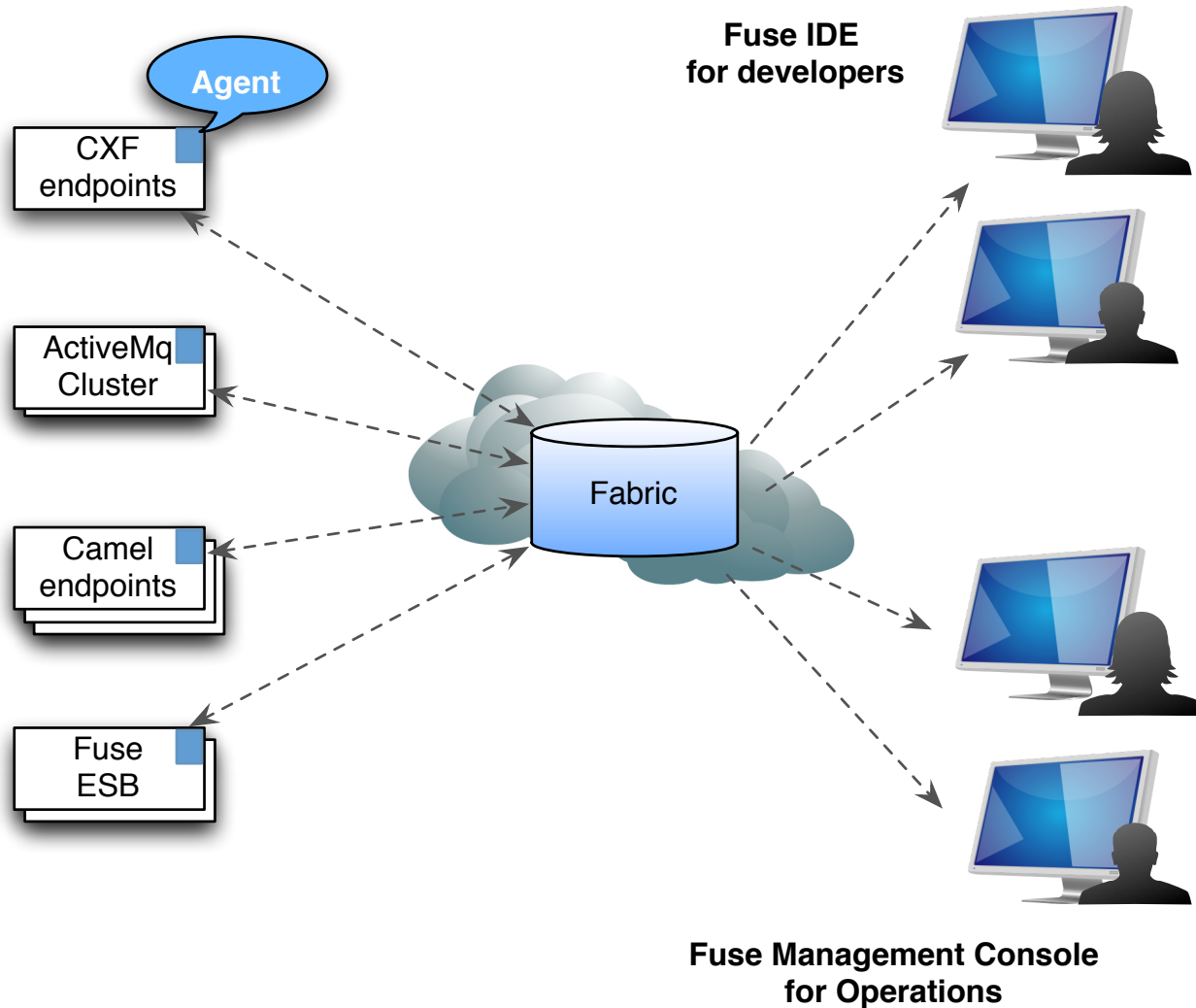
Tips for Camel riding in the cloud

- Loose coupling!
 - messaging is the best loose coupling
 - both time and location!
 - what if you can't?
 - discovery, load balancing & coordination can be hard...

What is Fuse Fabric?

- <http://fuse.fusesource.org/fabric/>
- open source PaaS
 - for running your integration solutions on any kind of cloud
 - Apache License 2.0
 - no catches or hidden bits :)
 - small & lightweight

Fuse Fabric overview



Fuse Fabric Features

- configuration registry
 - rolling upgrades
 - easy integration with source control & releasing
 - real time updates without restarting/reinstalling
- runtime registry
 - discovery
 - coordination
 - federated services
- dynamic provisioning within the cloud
 - containers / processes
 - on existing machines or spin up/down on cloud
 - features in containers

How Fuse Fabric helps Camel Riders

- easy lifecycle, provision & versioning of Camel
 - as bundles/wars/config files
 - rolling upgrade of routes & support multiple version
- centralised environment configuration
 - use profiles to keep things DRY

Camel Fabric discovery & load balancing

- Invoking a remote Camel endpoint

```
from( "activemq:someInput" ).  
    to( "fabric:myName" )
```

- Exposing a Camel endpoint into the fabric

```
from( "fabric:myName:jetty:http://0.0.0.0:0" ).  
    to( "bean:foo" )
```

Camel - HA Routes

- Implements Master / Hot Standby routes
- To ensure only a single JVM runs a particular route at any point in time

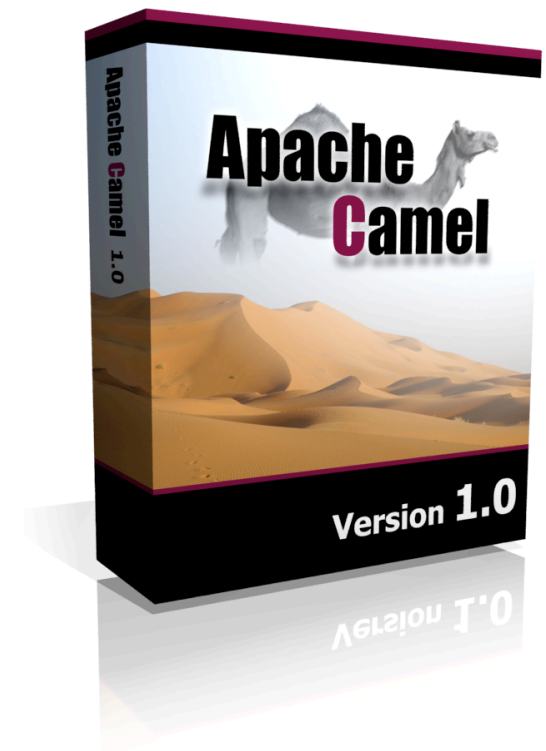
```
from( "master:foo:someUri" ) . . .
```

Riding Camel in the cloud: Summary

- Camel rocks
 - buy the book & try Fuse IDE
- Use messaging
 - easy loose coupling in the cloud
- Fuse Fabric can help
 - fine grained lifecycle, provisioning & configuration
 - HA Camel
 - discovery & load balancing when you can't use messaging

What should I do next?

- Ride that Camel
 - <http://camel.apache.org/>
- Download Fuse IDE
 - <http://fusesource.com>
- Fuse Fabric
 - <http://fuse.fusesource.org/fabric/>
- Don't get the hump!



Any Questions?



twitter: @jstrachan #fusenews

<http://fusesource.com>