

LECTURE 5

Introduction to Camel:

Camel Architecture

Introduction to Enterprise Integration Patterns:

Message Routers and Gateways

Basic Consumers

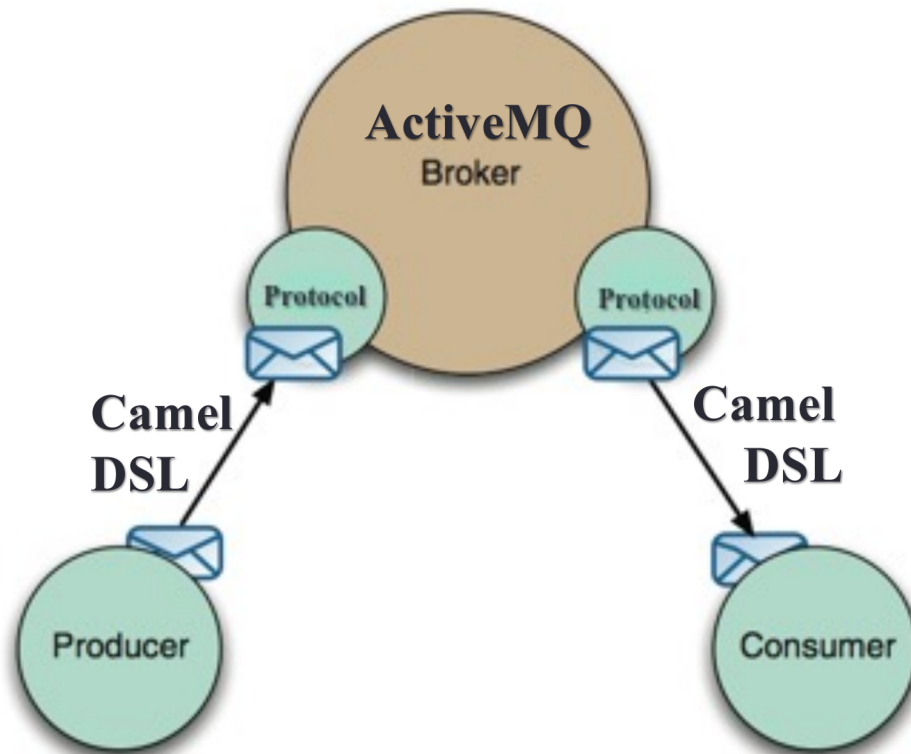
Splitting and Aggregation

INTRODUCTION TO CAMEL ARCHITECTURE

ActiveMQ provides an implementation of the JMS Specification

Camel can send messages to an ActiveMQ instance over the JMS protocol

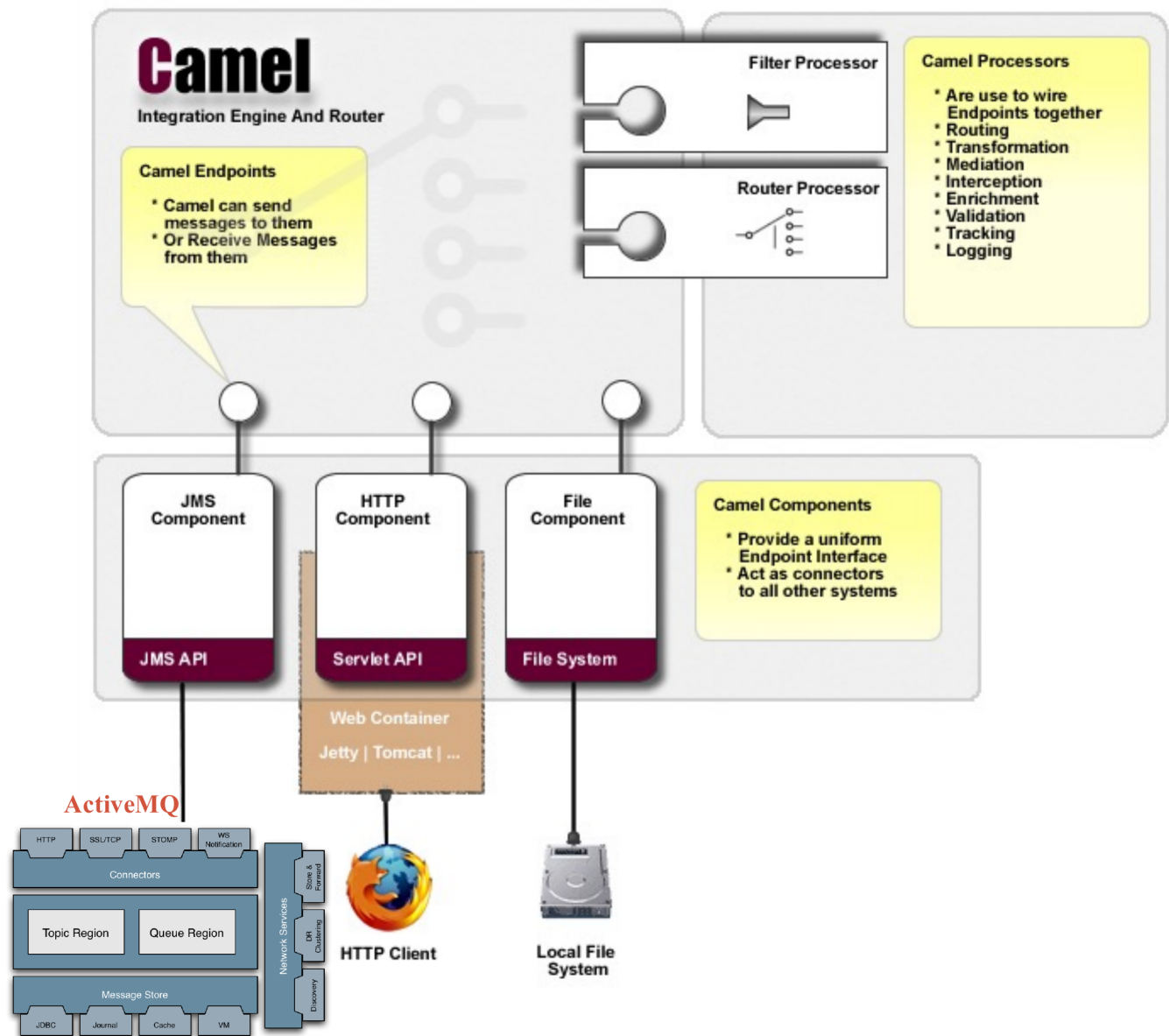
Message Broker Pattern Details



- ActiveMQ is a Message Broker, and Camel is a Java-based implementation of the Enterprise Integration Patterns from Hohpe & Woolf.

Camel Componentry

- At a high level, Camel consists of a CamelContext which contains a collection of Component instances. A Component is essentially a factory for Endpoints.
- Camel Architecture is at <http://camel.apache.org/architecture.html>



Camel Domain-Specific Language (DSL)

- The Camel DSL incorporates components and processors that are actually implementations of Enterprise Integration Patterns (EIPs)
- Enterprise Integration Patterns are reusable models within the domain of data and message integration that facilitate the sending and receiving of messages which contain various types of data
- Examples of EIPs include translators, filters, aggregators, splitters, routers, etc., which are all used in designing and implementing solutions to data and application integration problems

A Diesel What???

- A DSL is a *Domain Specific Language* that is built on top of another language, specifically to “simplify” development of particular types of applications...
- So, Java and C++ are considered general purpose programming languages...i.e., you can do almost anything with them...
- DSLs offer predicates that are familiar to practitioners within a give domain of expertise
 - An E-R Diagram is an example of a domain-specific modeling language, and has predicates such as multiplicity, optionality, mandatory, many-to-one, etc. all database designers will recognize
- DSLs are built for specific purposes, and examples include:
 - HTML (a markup language)
 - SQL (a query language)

First Look at Camel DSL Code

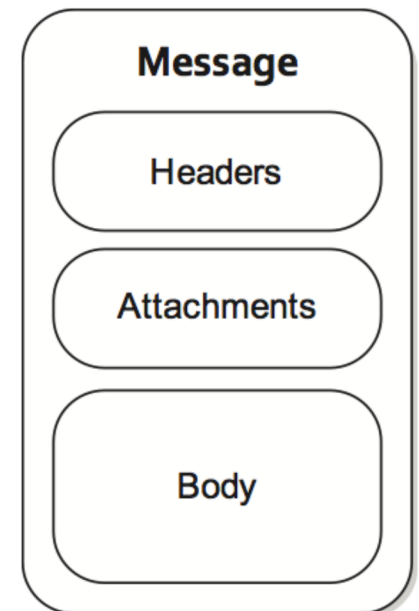
Example: Adding a New Route

```
context.addRoutes(new RouteBuilder() {
    @Override
    public void configure()
    { //for file component see: http://camel.apache.org/file2.html
        from("file:data/inbox?noop=true&fileName=head.101.csv")
            .log("RECEIVED: head.101.csv")
            .unmarshal().csv()
            .split(body())
            .to("jms:queue:MPCS_51050_LAB6");
    }
});
```

- Resources:
 - Camel Code Examples: <http://camel.apache.org/examples.html>
 - Camel Tutorials: <http://camel.apache.org/tutorials.html>
 - Camel Cookbook: <http://camel.apache.org/cookbook.html>
 - Ibsen & Anstey File Component Reference: Chapter 7.2, pp. 192ff.

Camel's Message Model

- In Camel, messages are uniquely identified with an ID of type `java.lang.String`. The identifier is:
 - Created by the message's creator
 - Protocol dependent
 - Doesn't have a guaranteed format
- All that is guaranteed is that a given message's ID is unique within a given context and protocol
- Example Camel Message ID: `ID:cheiron.local-55077-1399678953467-1:1:1:1:1`



Camel's Message Identifiers



[Home](#) | [Queues](#) | [Topics](#) | [Subscribers](#) | [Connections](#) | [Network](#) | [Scheduled](#) | [Send](#)

Browse incomingOrders

Message ID ↑	Correlation ID	Persistence	Priority	Redelivered	Reply To	Timestamp	Type	Operations
ID:cheiron.local-55077-1399678953467-1:1:1:1:1		Persistent	4	false		2014-05-09 18:42:33:631 CDT		Delete
ID:cheiron.local-55077-1399678953467-1:2:1:1:1		Persistent	4	false		2014-05-09 18:42:33:663 CDT		Delete
ID:cheiron.local-64020-1399722911970-1:1:1:1:1		Persistent	4	false		2014-05-10 06:55:12:127 CDT		Delete
ID:cheiron.local-64020-1399722911970-1:2:1:1:1		Persistent	4	false		2014-05-10 06:55:12:153 CDT		Delete
ID:cheiron.local-64308-1399723216664-1:1:1:1:1		Persistent	4	false		2014-05-10 07:00:16:805 CDT		Delete
ID:cheiron.local-64308-1399723216664-1:2:1:1:1		Persistent	4	false		2014-05-10 07:00:16:823 CDT		Delete
ID:cheiron.local-64308-1399723216664-1:3:1:1:1		Persistent	4	false		2014-05-10 07:00:21:956 CDT		Delete
ID:cheiron.local-64308-1399723216664-1:4:1:1:1		Persistent	4	false		2014-05-10 07:00:21:968 CDT		Delete

[View Consumers](#)

Connecting Camel to ActiveMQ

- In order to use Camel's EIP implementations through its DSL, you need to connect to a JMS provider (such as ActiveMQ) by configuring a Camel ConnectionFactory
- To use ActiveMQ, configure an ActiveMQConnectionFactory that points to a running ActiveMQ broker, e.g.:

```
public static void main(String args[]) throws Exception {  
    // create CamelContext  
    CamelContext context = new DefaultCamelContext();  
    // connect to embedded ActiveMQ JMS broker  
    ConnectionFactory connectionFactory =  
        new ActiveMQConnectionFactory("tcp://localhost:61616");  
    context.addComponent("jms",  
        JmsComponent.jmsComponentAutoAcknowledge(connectionFactory));  
    . . .  
}
```