Apache Camel Workshop

JBCNConf 2018

Claus Ibsen - Nicola Ferraro

Welcome, who we are

Claus Ibsen

Software Engineer (Red Hat)

10 years Apache Camel

@davsclaus

davsclaus

www.davsclaus.com

Nicola Ferraro

Software Engineer (Red Hat)

Less years Apache Camel :)

@ni_ferraro

nicolaferraro

www.nicolaferraro.me

Agenda

- **Intro to Apache Camel ←**
- Overview of the Workshop
- Coding: Part 1
- Circuit Breakers and Saga
- Coding: Part 2
- Openshift (Optional)
- Summary
- Q/A

What is Apache Camel ?

System Integration

Figure 1.1 Camel is the glue between disparate systems.

Integration Framework

PATTERN BASED INTEGRATION

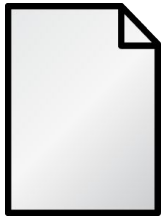
Apache Camel, a powerful pattern-based integration engine with a comprehensive set of connectors and data formats to tackle any integration problem.

ENTERPRISE INTEGRATION PATTERNS

Build integrations using enterprise best practices.

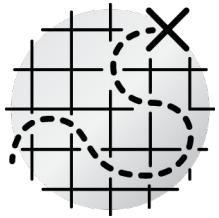
200+ COMPONENTS

Batch, messaging, web services, cloud, APIs, and more ...



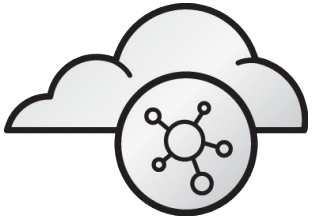
BUILT-IN DATA TRANSFORMATION

JSON, XML, HL7, YAML, SOAP, Java, CSV, and more ...



INTUITIVE ROUTING

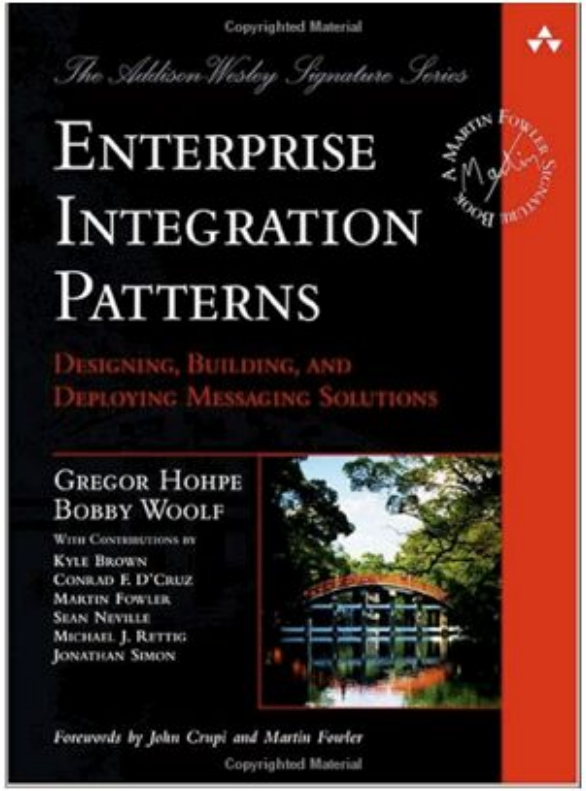
Develop integrations quickly in Java or XML.



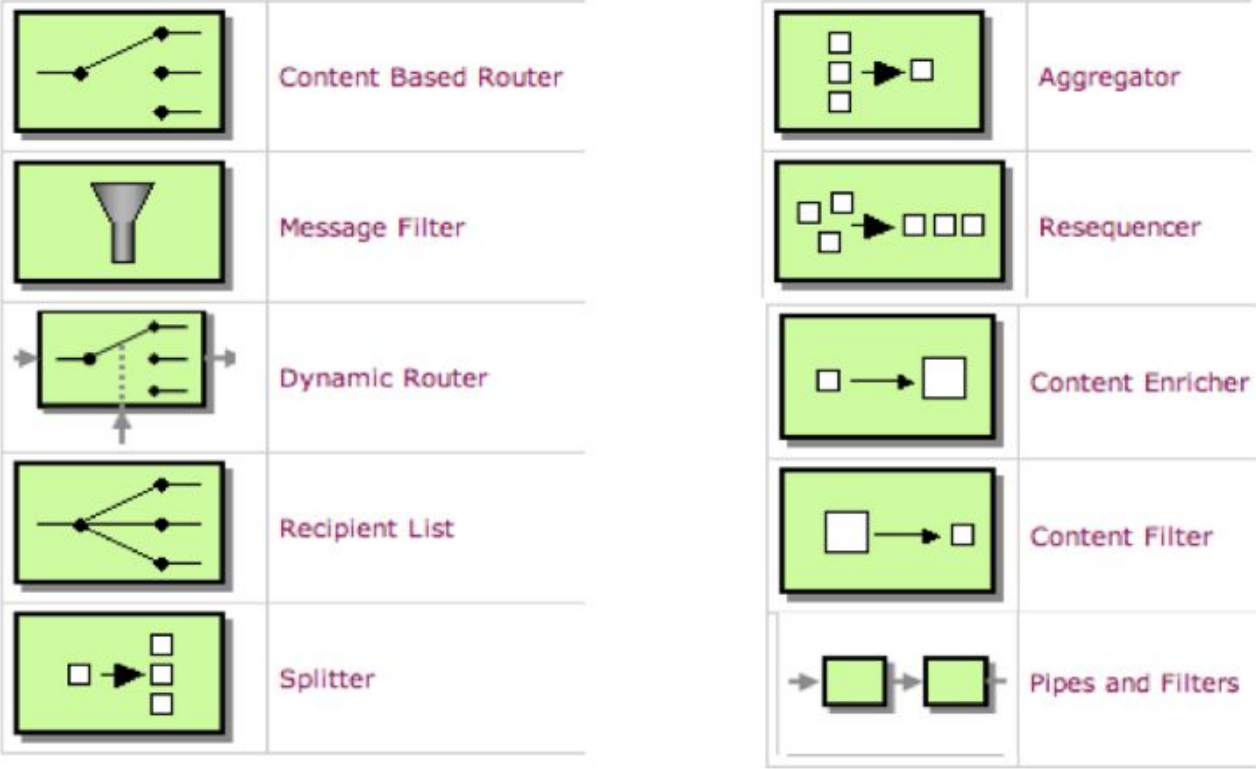
NATIVE REST SUPPORT

Create, connect, and compose APIs with ease.

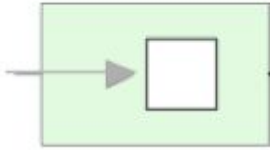
Enterprise Integration Patterns



Enterprise Integration Patterns

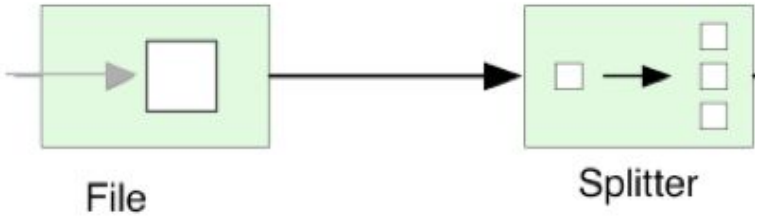


Camel Routes with Splitter

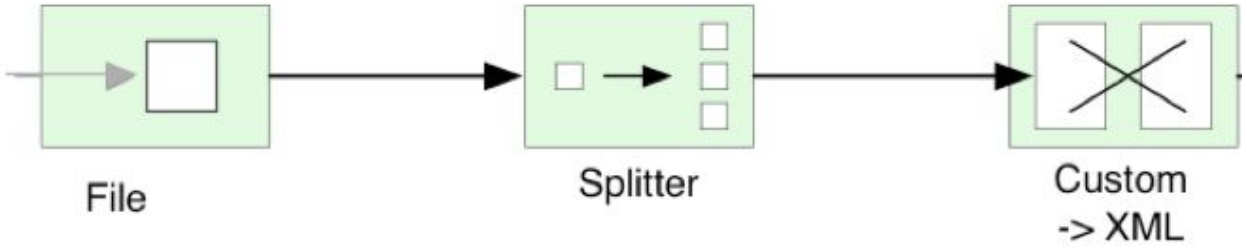


File

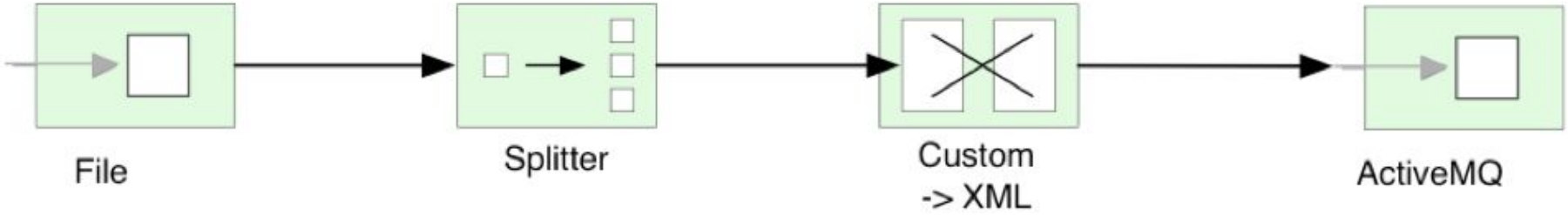
Camel Routes with Splitter



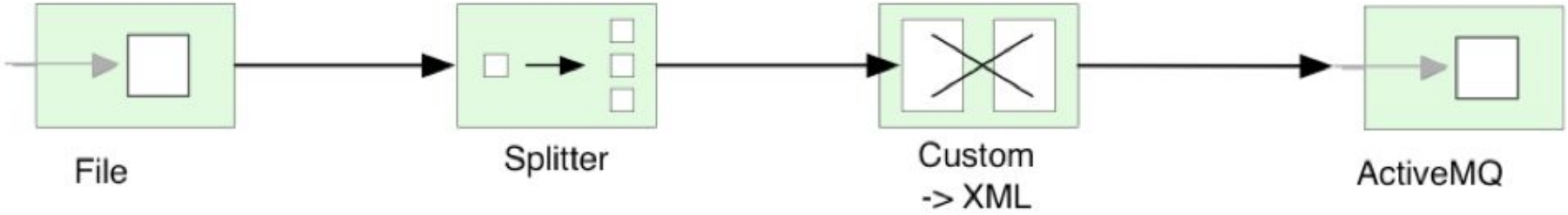
Camel Routes with Splitter



Camel Routes with Splitter

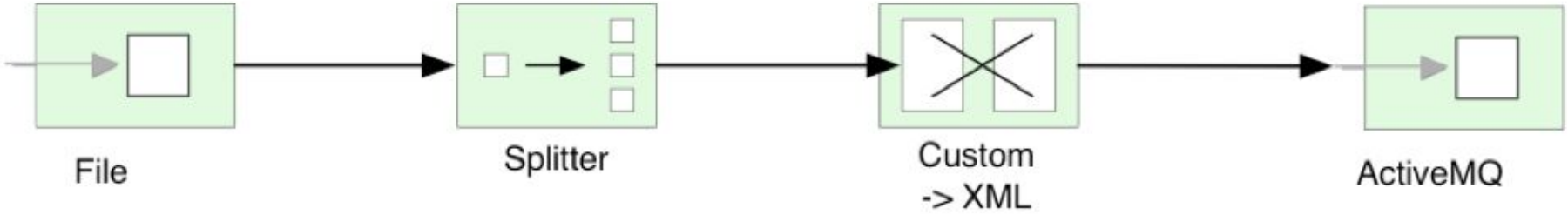


Camel Routes with Splitter



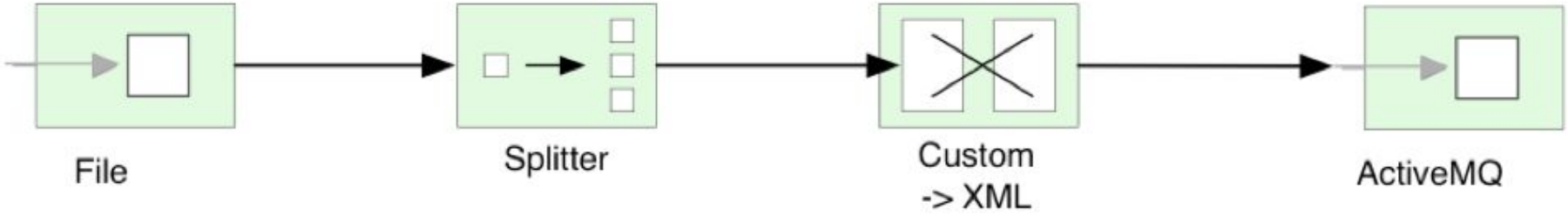

```
from("file:inbox")
```

Camel Routes with Splitter




```
from("file:inbox")
    .split(body().tokenize("\n"))
```

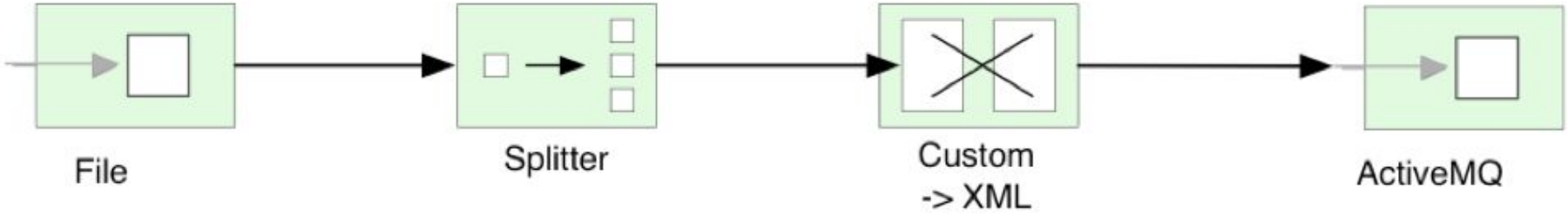
Camel Routes with Splitter




```
from("file:inbox")
    .split(body().tokenize("\n"))
        .marshal(customToXml)
```

Custom data transformation

Camel Routes with Splitter




```
from("file:inbox")
    .split(body().tokenize("\n"))
        .marshal(customToXml)
        .to("activemq:line");
```

Custom data transformation

Camel Routes with Splitter

```
from("file:inbox")
    .split(body().tokenize("\n"))
        .marshal(customToXml)
        .to("activemq:line");
```

Camel Route in Java DSL

```
public void configure() throws Exception {

  from("file:inbox")
    .split(body().tokenize("\n"))
      .marshal(customToXml)
      .to("activemq:line");
}
```

Camel Route in Java DSL

```
public class MyRoute extends RouteBuilder {

  public void configure() throws Exception {

    from("file:inbox")
      .split(body().tokenize("\n"))
        .marshal(customToXml)
        .to("activemq:line");
  }
}
```

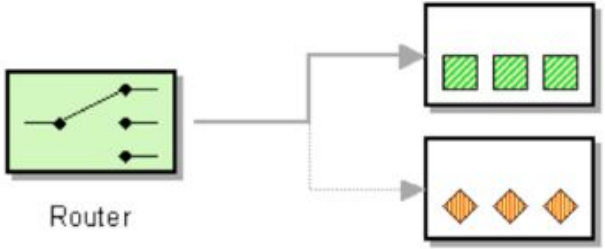
```
import org.apache.camel.builder.RouteBuilder;

public class MyRoute extends RouteBuilder {

  public void configure() throws Exception {

    from("file:inbox")
      .split(body().tokenize("\n"))
        .marshal(customToXml)
        .to("activemq:line");
  }
}
```

Camel Routes



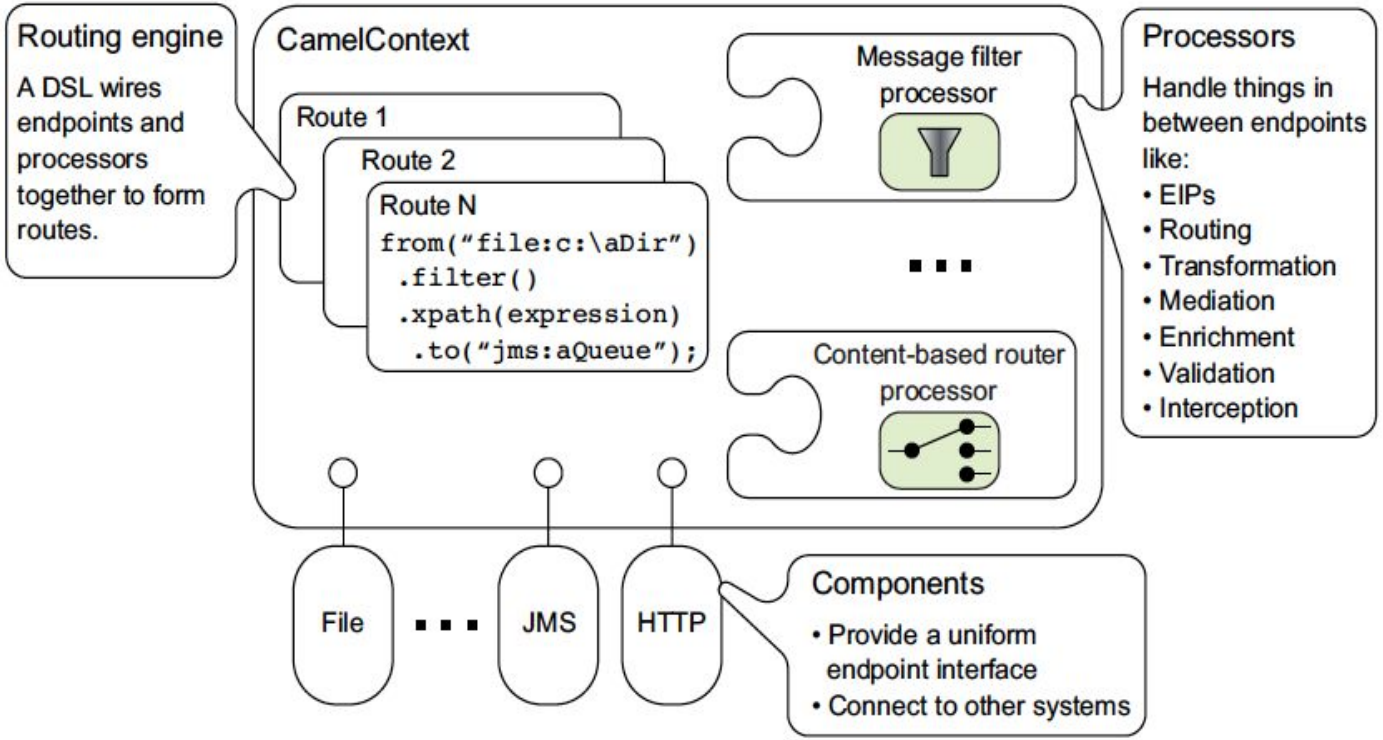

```
from("file:data/inbox")
  .to("jms:queue:order");
```

Java DSL

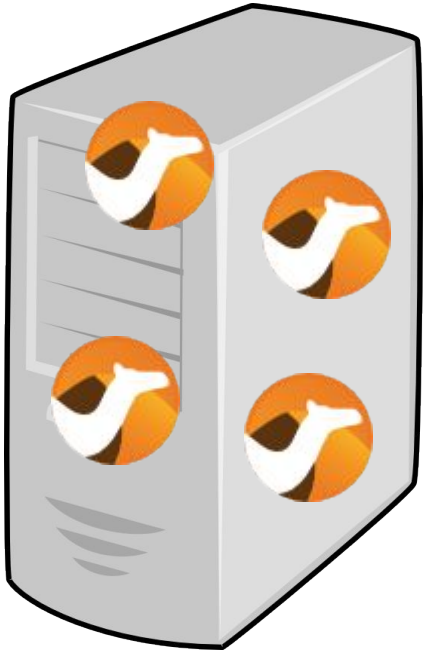
XML DSL

```
<route>
  <from ri="file:data/inbox"/>
  <to uri="jms:queue:order"/>
</route>
```

Camel Architecture



Camel runs everywhere

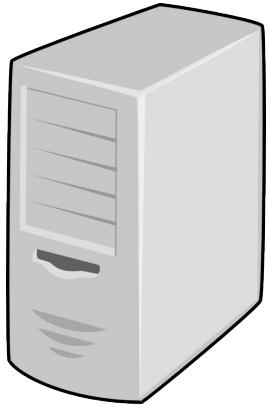


Application Servers

Standalone Runtimes

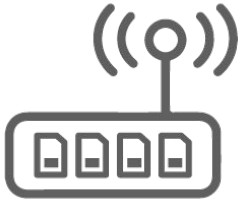
Linux Containers

Camel connects everything



Enterprise Systems

- File
- FTP
- JMS
- AMQP
- JDBC
- SQL
- TCP/UDP
- Mail
- HDFS
- JPA
- MongoDB
- Kafka
- ...



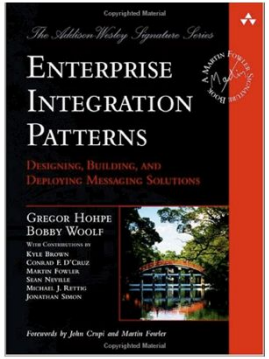
IoT

- CoAP
- MQTT
- PubNub

- AWS
  - S3
  - SQS
  - Kinesis
  - ...
- Google
  - BigQuery
  - PubSub
- Azure
  - Blob
  - Queue

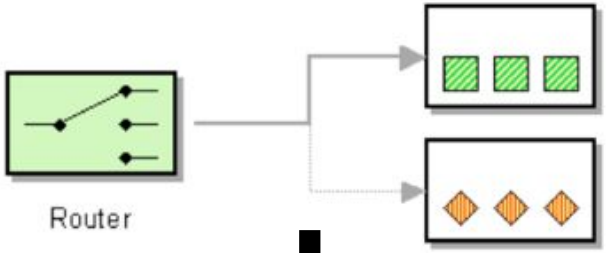
- Box
- Dropbox
- Facebook
- Linkedin
- Salesforce
- SAP
- ServiceNow


TOOL BOX

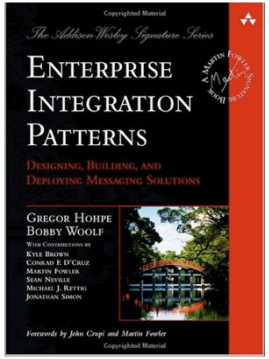

Copyrighted Material
The Addison-Wesley Signature Series
Enterprise Integration Patterns
Designing, Building, and Deploying Messaging Solutions
Gregor Hohpe
Bobby Woolf
With Contributions by
Kyle Brown
Conrad F. D'Cruz
Martin Fowler
Sean Neville
Michael J. Rettig
Jonathan Simon
Forewords by John Crupi and Martin Fowler
Copyrighted Material

+


TOOL BOX

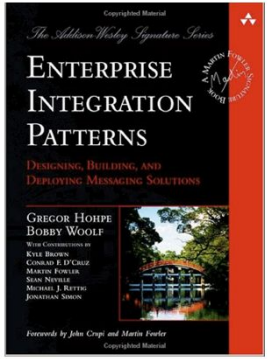

Router

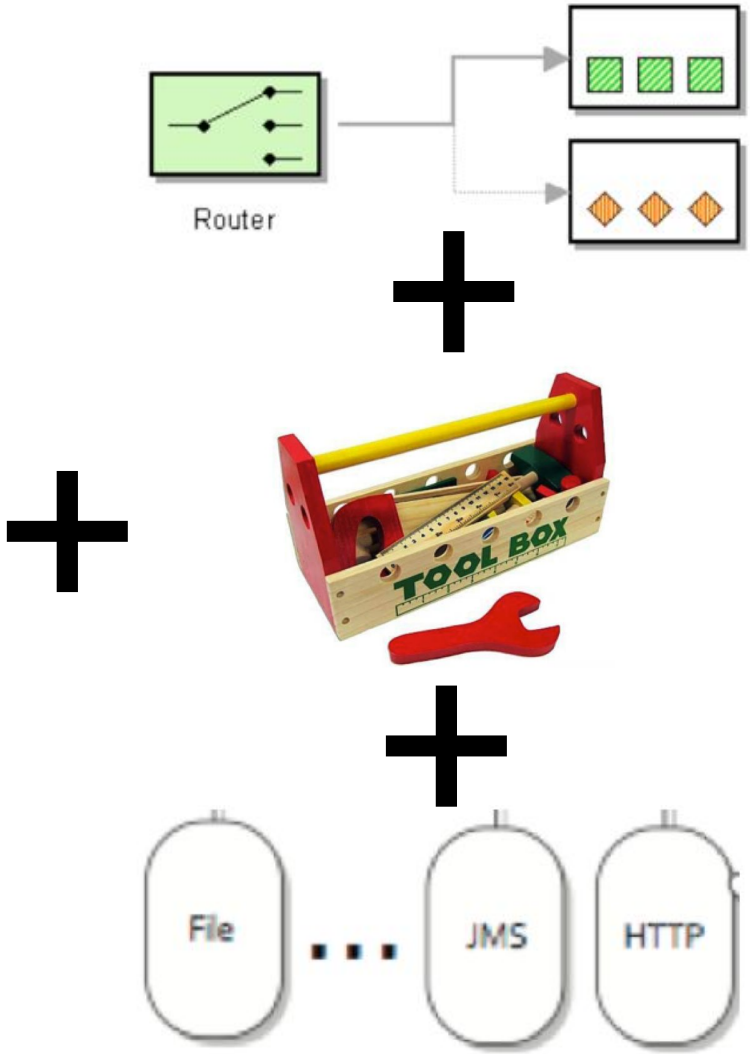
+

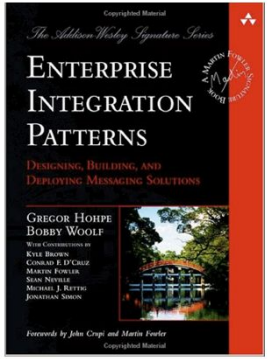

Copyrighted Material
The Addison-Wesley Signature Series
Enterprise Integration Patterns
Designing, Building, and Deploying Messaging Solutions
Gregor Hohpe
Bobby Woolf
With Contributions by
Kyle Brown
Conrad F. D'Cruz
Martin Fowler
Sean Neville
Michael J. Rettig
Jonathan Simon
Forewords by John Crupi and Martin Fowler
Copyrighted Material

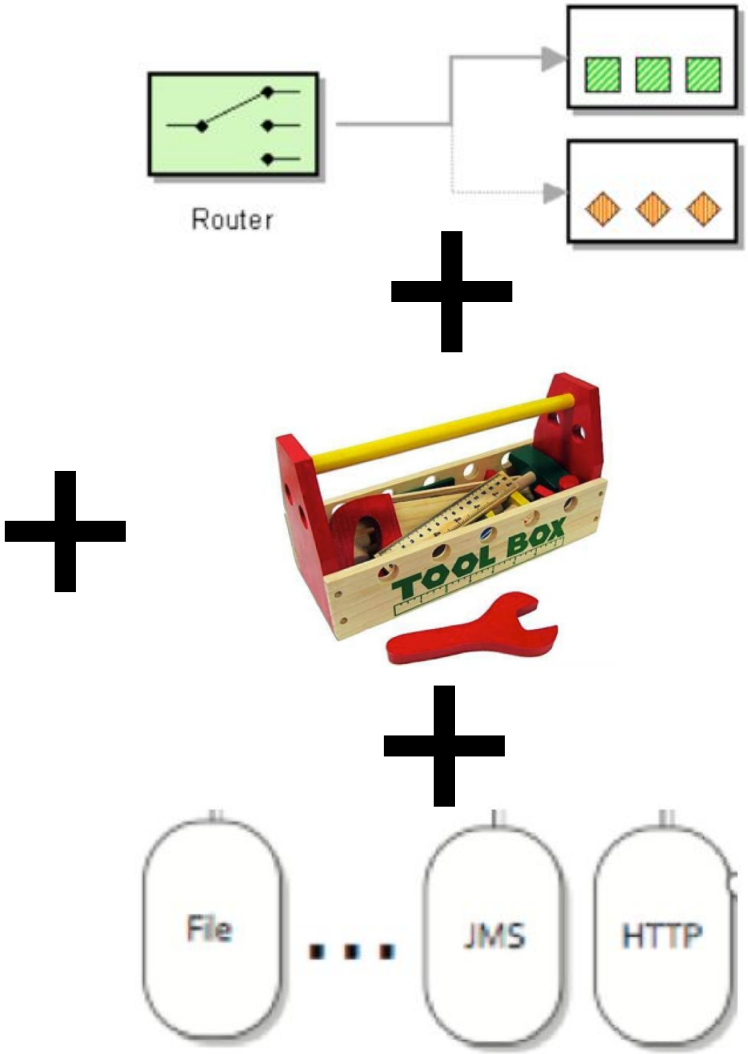
+


TOOL BOX


Copyrighted Material
The Addison-Wesley Signature Series
Enterprise Integration Patterns
Designing, Building, and Deploying Messaging Solutions
Gregor Hohpe
Bobby Woolf
With Contributions by
Kyle Brown
Conrad F. D'Cruz
Martin Fowler
Sean Neville
Michael J. Rettig
Jonathan Simon
Forewords by John Crupi and Martin Fowler
Copyrighted Material


Router
TOOL BOX
File
JMS
HTTP

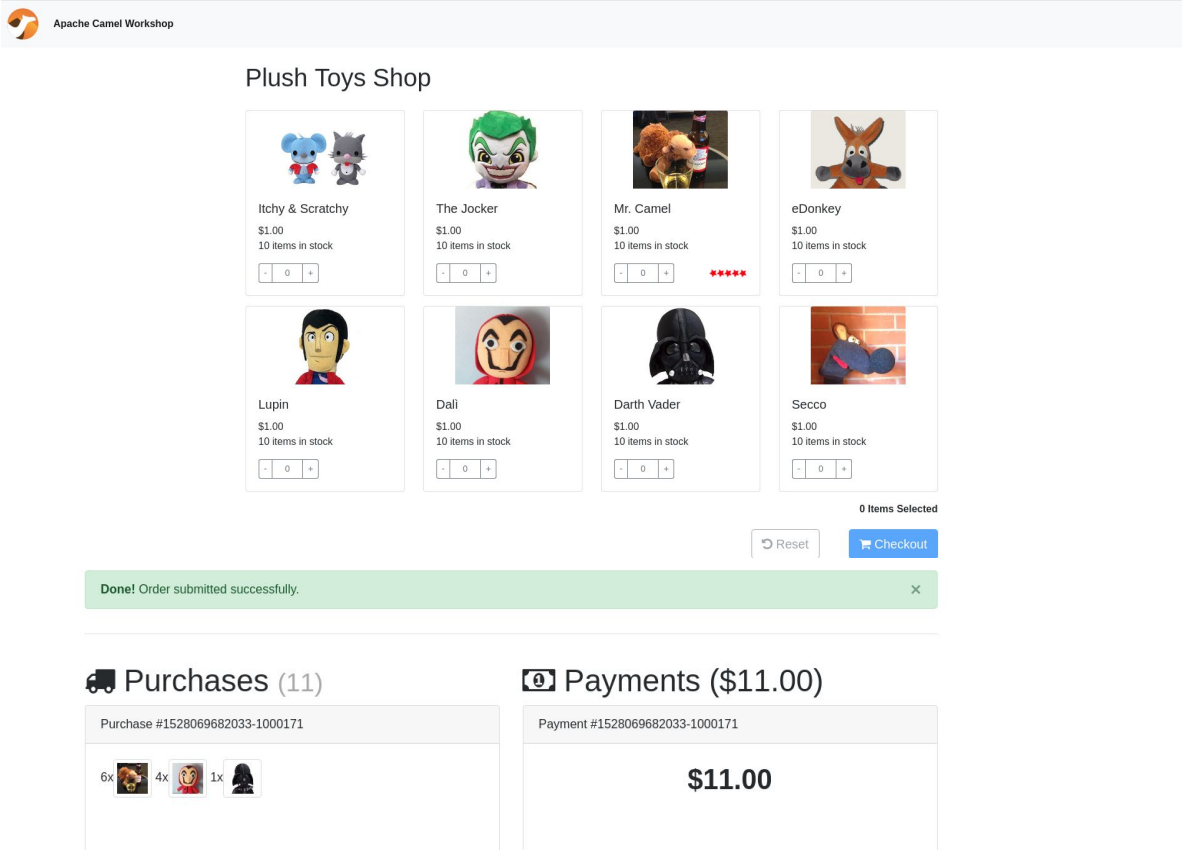

Copyrighted Material
The Addison-Wesley Signature Series
Enterprise Integration Patterns
Designing, Building, and Deploying Messaging Solutions
Gregor Hohpe
Bobby Woolf
With Contributions by
Kyle Brown
Conrad F. D'Cruz
Martin Fowler
Sean Neville
Michael J. Rettig
Jonathan Simon
Forewords by John Crupi and Martin Fowler
Copyrighted Material


Router
TOOL BOX
File
JMS
HTTP

Agenda

- Intro to Apache Camel
- **Overview of the Workshop ←**
- Coding: Part 1
- Circuit Breakers and Saga
- Coding: Part 2
- Openshift (Optional)
- Summary
- Q/A

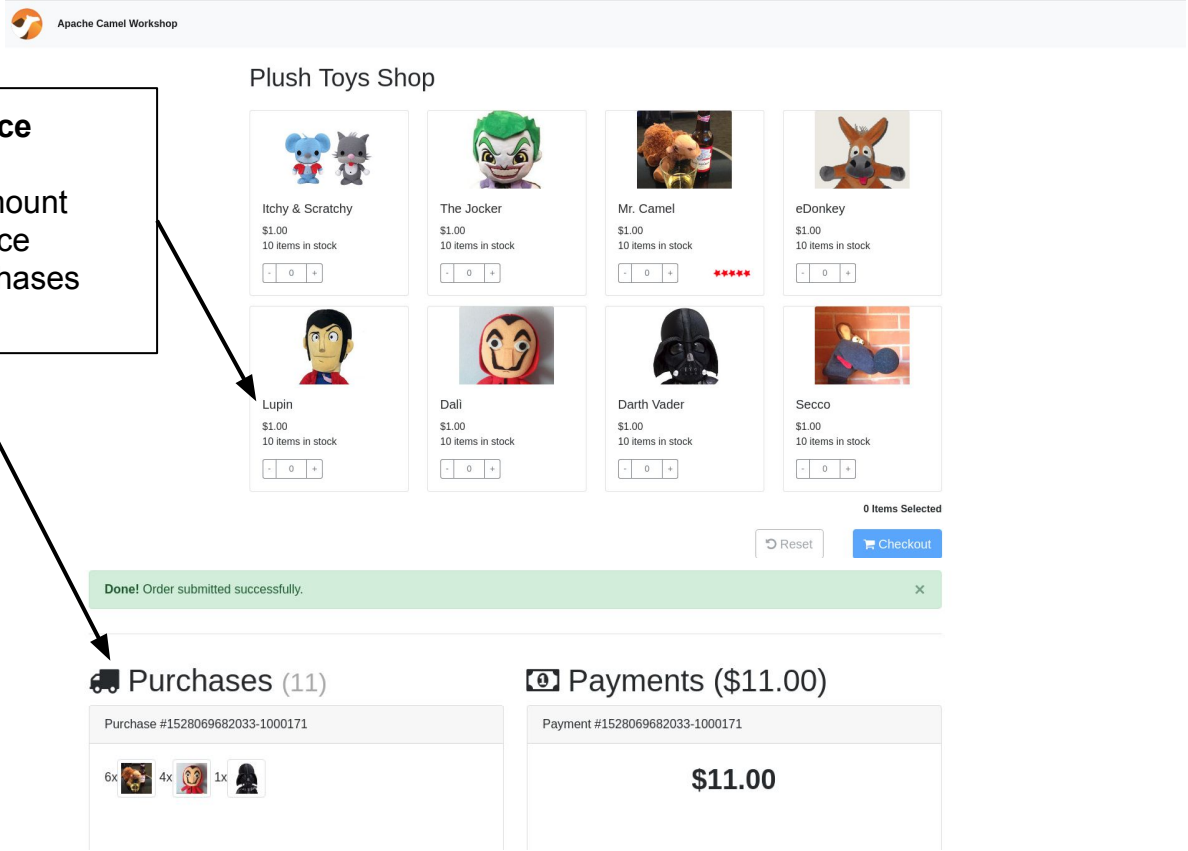
Ovewview of the Workshop



Ovewview of the Workshop

Inventory Service

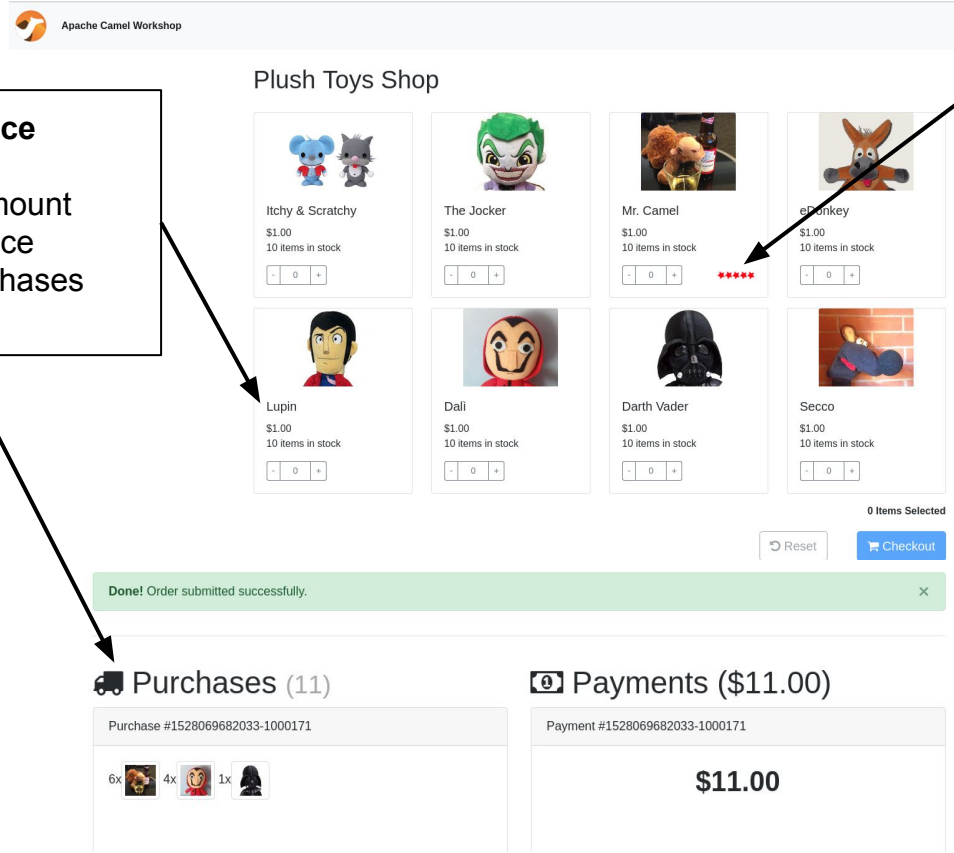
- Items
  - Amount
  - Price
- Item purchases



Ovewview of the Workshop

Recommendation Service
- Featured items

Inventory Service
- Items
  - Amount
  - Price
- Item purchases



Ovewview of the Workshop

Recommendation Service
- Featured items

Inventory Service
- Items
  - Amount
  - Price
- Item purchases

Credit Service
- Payments

Apache Camel Workshop

Plush Toys Shop

Itchy & Scratchy
$1.00
10 items in stock

The Jocker
$1.00
10 items in stock

Mr. Camel
$1.00
10 items in stock

elephkey
$1.00
10 items in stock

Lupin
$1.00
10 items in stock

Dalì
$1.00
10 items in stock

Darth Vader
$1.00
10 items in stock

Secco
$1.00
10 items in stock

0 Items Selected

Reset

Checkout

Done! Order submitted successfully.

Purchases (11)

Purchase #1528069682033-1000171

6x 4x 1x

Payments ($11.00)

Payment #1528069682033-1000171

$11.00

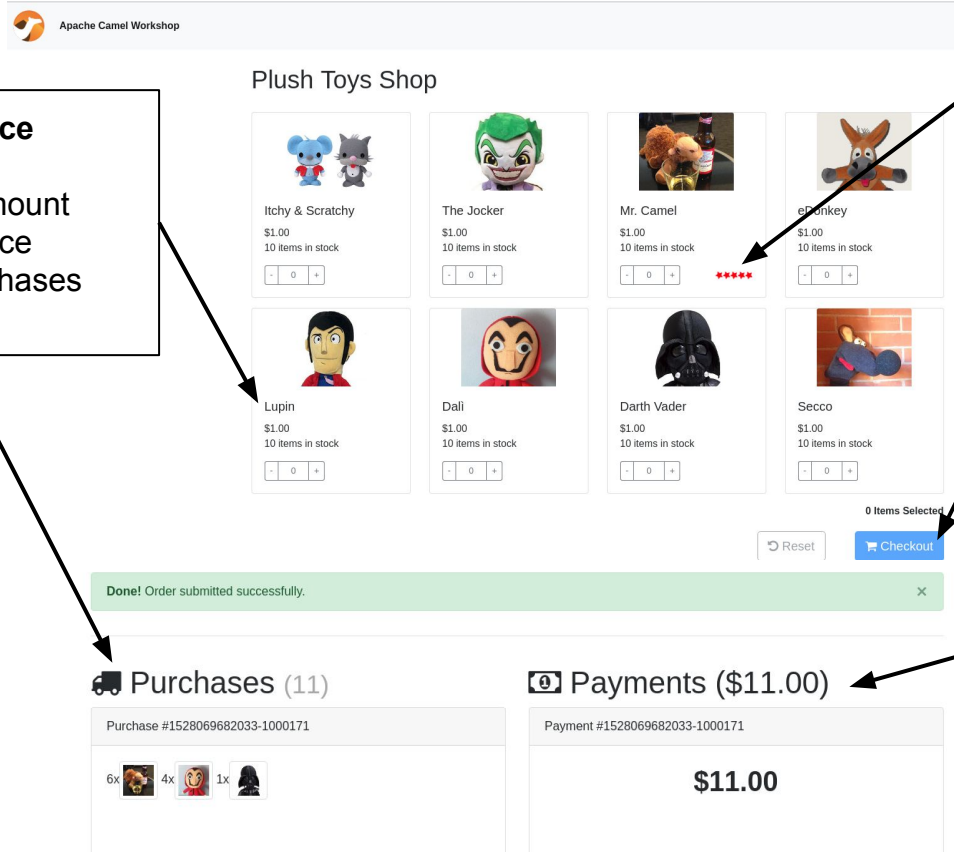
Ovewview of the Workshop

Recommendation Service
- Featured items

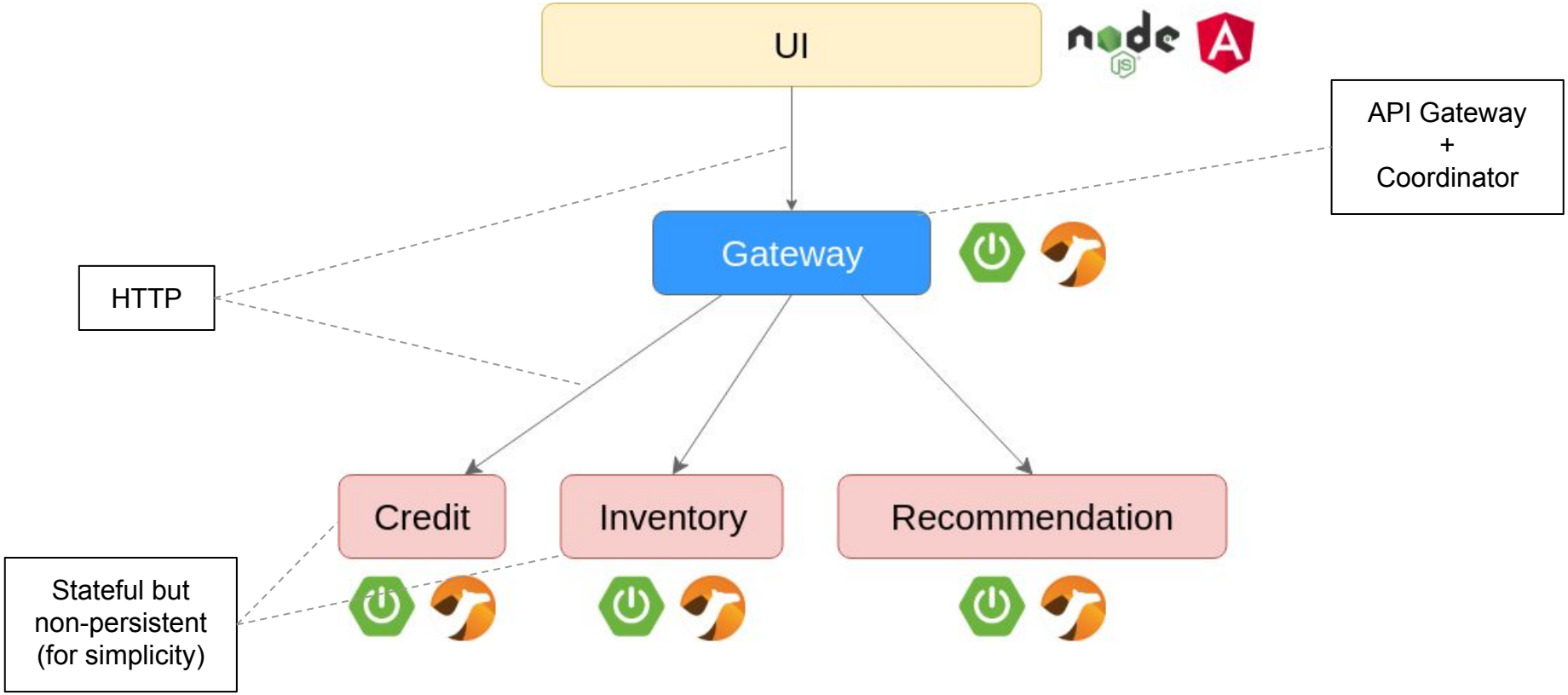
Inventory Service
- Items
  - Amount
  - Price
- Item purchases

Gateway Service
- Order coordinator
- API Normalization

Credit Service
- Payments



Architecture



Agenda

- Intro to Apache Camel
- Overview of the Workshop
- **Coding: Part 1 ←**
- Circuit Breakers and Saga
- Coding: Part 2
- Openshift (Optional)
- Summary
- Q/A

Coding: Part 1

We'll see in this coding session:

- **REST DSL** (camel-servlet to bridge Spring Web)
- **Transformation** (camel-jackson)
- **Validation** (camel-bean-validator)
- **REST Service Documentation** (camel-swagger-java)
- **HTTP Forwarding** (camel-undertow)
- **Enterprise Integration Patterns (EIP)**: service-call, multicast, split, choice, bean call
- **Misc**: property placeholders, simple language

Workshop Instructions

Requirements for Part 1

- Java 8
- Maven 3.5.x
- Your favourite IDE
- HTTPie (to test endpoints, instructions to install in the readme)

Follow the readme:

https://github.com/nicolaferraro/camel-workshop

Start.spring.io

Switch to 1.5.13 before doing anything else

Activities Google Chrome Wed 11:44

Camel Workshop - nicolaferraro/camel- Spring Initializr

Not secure | start.spring.io

SPRING INITIALIZR bootstrap your application now

Generate a Maven Project with Java and Spring Boot 1.5.13

Project Metadata

Artifact coordinates

Group

org.apache.camel.workshop

Artifact

recommendation

Dependencies

Add Spring Boot Starters and dependencies to your application

Search for dependencies

Web, Security, JPA, Actuator, Devtools...

Selected Dependencies

Web DevTools Apache Camel

Generate Project alt + ⏎

Don't know what to look for? Want more options? Switch to the full version.

start.spring.io is powered by Spring Initializr and Pivotal Web Services

Agenda

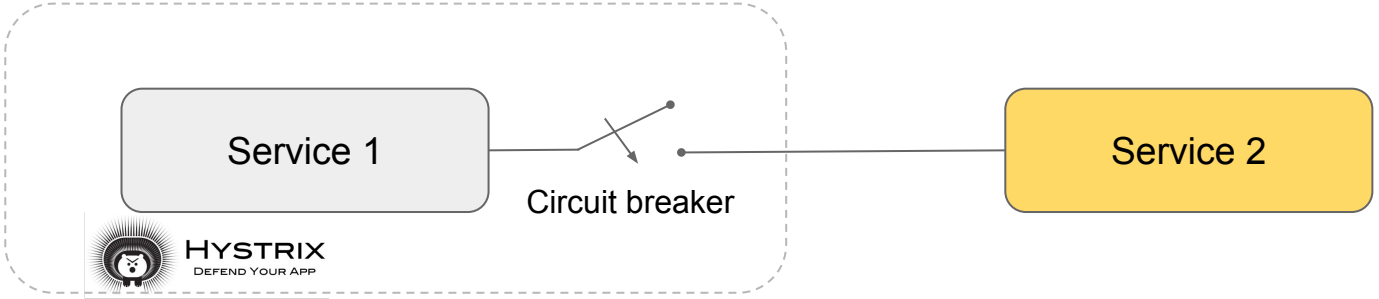
- Intro to Apache Camel
- Overview of the Workshop
- Coding: Part 1
- **Circuit Breakers and Saga ←**
- Coding: Part 2
- Openshift (Optional)
- Summary
- Q/A

Circuit Breakers: open / closed

Circuit breakers are like 2-state light switches:

- Closed: service 1 tries to contact service 2
- Open: service 1 does not even try contact service 2

Why? **To prevent cascading failures, fault tolerance, resilience, fail fast adding a fallback strategy**



The hystrix library is embedded in the calling service

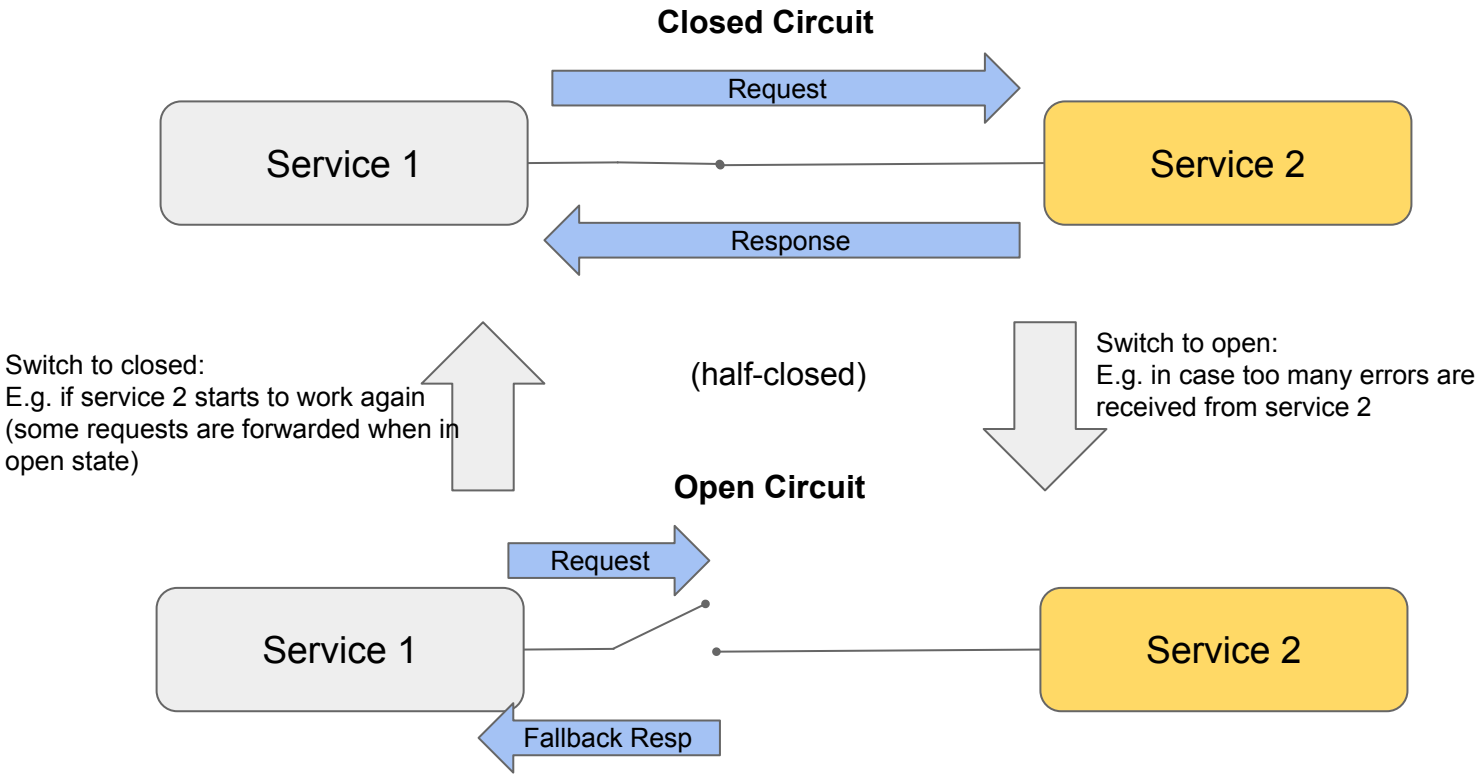
Circuit Breakers

Camel includes a *.hystrix()* EIP for circuit breaking.

```
from("direct:route")
  .hystrix()
    .to("direct:main-action") // ← action(s) we want to make
  .onFallback()
    .setBody(constant("fallback value")) // ← fallback value in case of error
  .end()
```

It's different from a **try/catch**!

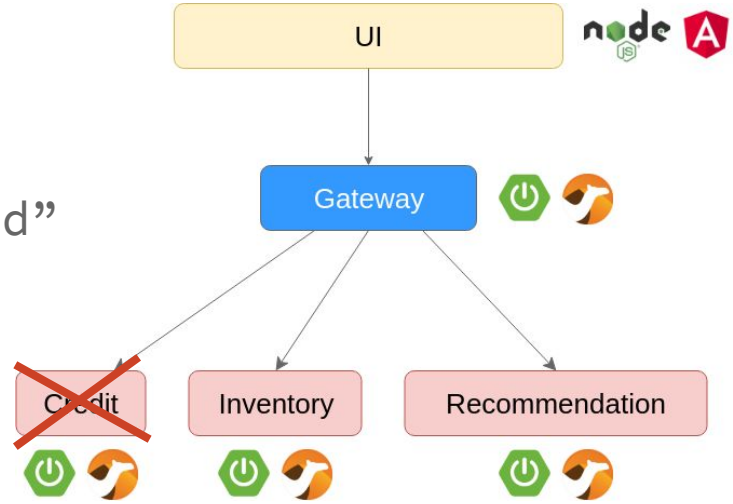
Circuit Breakers: open / closed



Saga

Imagine the following scenario:

- Credit service <u>unavailable</u>
- User makes a order
- Items are marked as “to be shipped” in Inventory
- But credit cannot be decreased!
- **Result = items given for free!**



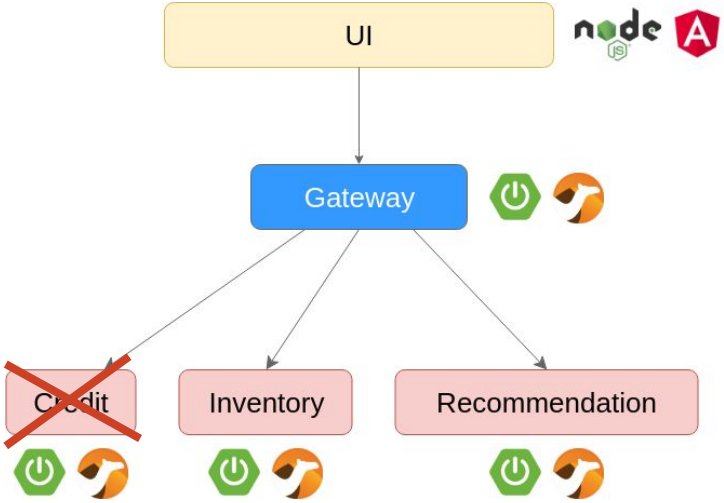
The solution: a saga!

Saga

A pattern for coordinating actions in remote services in order to obtain a consistent outcome.

A different workflow:

- …
- Credit cannot be decreased!
- **Compensation: re-add the items in the inventory!**
- **Result: consistency is reestablished!**



Saga: in Camel

Camel includes a *.saga()* EIP for declaring sagas.

```
from("direct:route")
  .saga().compensation("direct:cancel-action")
    .to("direct:main-action") // ← action(s) we want to make


from("direct:cancel-action")
  .to("...") // ← compensation
```

<u>Rule: if something fails, compensation **must** be done!</u>

Otherwise consistency is not guaranteed.

Saga: in Camel

How consistency is guaranteed:

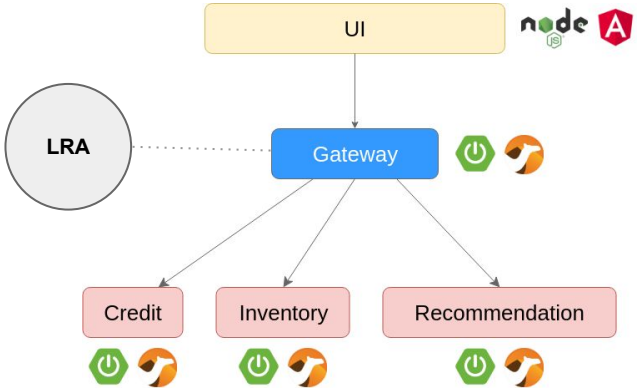
- Compensation is retried until successful
- Compensating action must be idempotent
- Compensating action must be commutative wrt main action
- Status of Sagas can be written to a persistent log

LRA Coordinator for persistence:

https://github.com/eclipse/microprofile-lra

camel-lra supports it already!

Implementation: https://github.com/jbosstm/narayana



Agenda

- Intro to Apache Camel
- Overview of the Workshop
- Coding: Part 1
- Circuit Breakers and Saga
- **Coding: Part 2 ←**
- Openshift (Optional)
- Summary
- Q/A

Coding: Part 2

We'll see in this coding session:

- **Circuit Breaker** (camel-hystrix)
- **Cache** (camel-caffeine)
- **Saga** (camel-core)

Follow **part 2** on:

https://github.com/nicolaferraro/camel-workshop

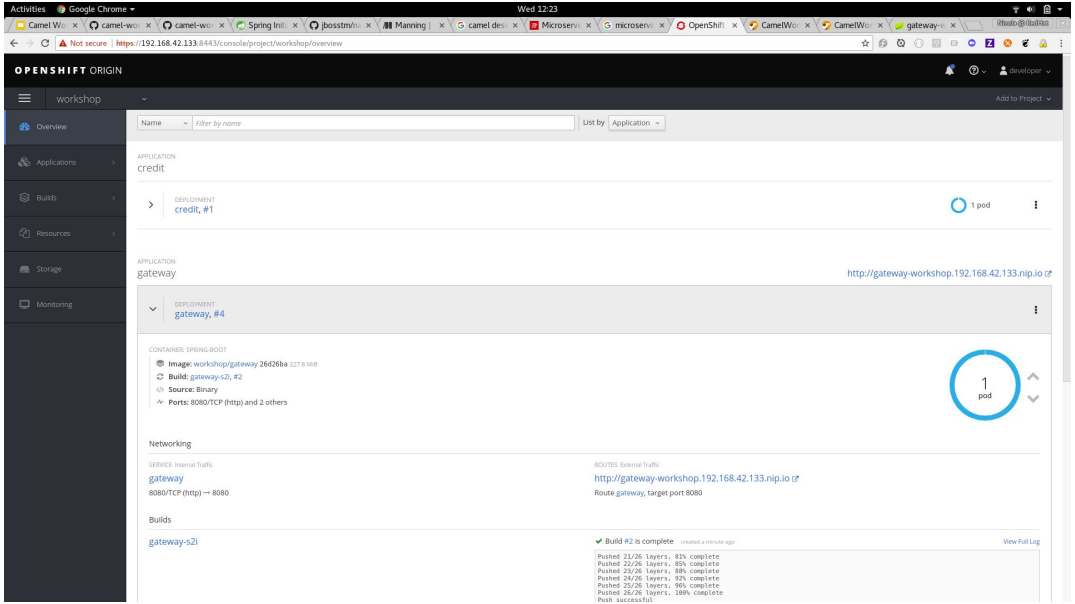
Agenda

- Intro to Apache Camel
- Overview of the Workshop
- Coding: Part 1
- Circuit Breakers and Saga
- Coding: Part 2
- **Openshift (Optional) ←**
- Summary
- Q/A

Openshift (and Kubernetes)

The workshop includes a tutorial to help you deploy the whole system into a Openshift Project.

INSTRUCTOR LEAD DEMO



Agenda

- Intro to Apache Camel
- Overview of the Workshop
- Coding: Part 1
- Circuit Breakers and Saga
- Coding: Part 2
- Openshift (Optional)
- **Summary ←**
- Q/A

Summary

APACHE Camel

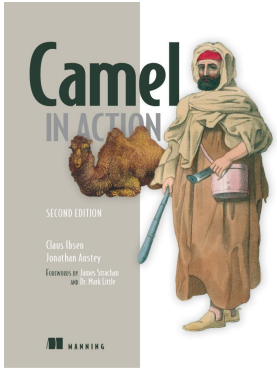
- **Camel is Awesome!**
- Camel loves Spring Boot and Spring Boot loves Camel

  https://spring.io/blog/2018/05/23/spring-tips-apache-camel
- Camel can do microservices
- Camel can integrate with everything

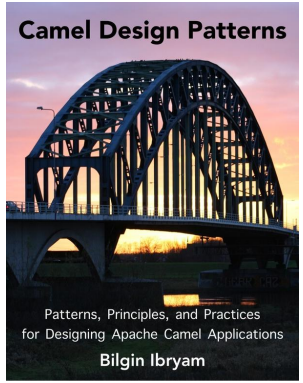
Thank you!

@davsclaus
@ni_ferraro

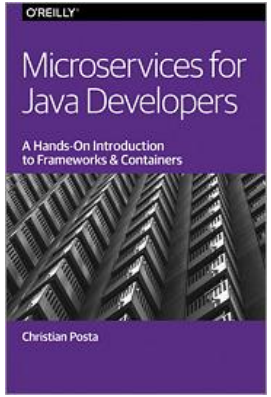
Source code and slides:
https://github.com/nicolaferraro/camel-workshop



https://www.manning.com/books/camel-in-action-second-edition



https://leanpub.com/camel-design-patterns



https://developers.redhat.com/promotions/microservices-for-java-developers/