





Leverage Enterprise Integration Patterns with Apache Camel and Twitter

Bruno Borges

Oracle Product Manager for Latin America
Java EE, GlassFish, WebLogic, Coherence

An abstract graphic on the right side of the slide consists of overlapping, semi-transparent geometric shapes in shades of blue and gold. The shapes are interconnected by thin white lines, creating a complex, crystalline structure that resembles a network or a modern architectural design.

MAKE THE
FUTURE
JAVA

ORACLE®

Who am I?

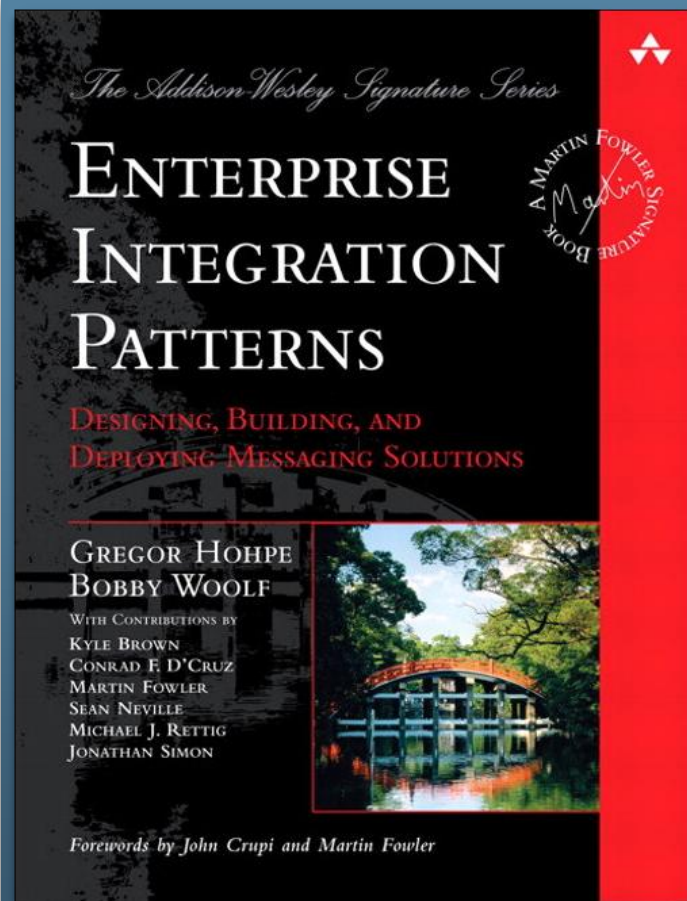
Bruno Borges

- Java developer since 2000
- Speaker at Conferences
 - JustJava, JavaOne Brazil, The Developers' Conference, ApacheCon
- Evangelized Apache Camel and Apache Wicket in Brazil
- Joined Oracle on July 2012
 - Product Manager for Java EE, GlassFish and WebLogic – Latin America
- Married, lives in Sao Paulo, has a Golden Retriever, hiker and gamer

Agenda

- Enterprise Integration Patterns
- Introduction to Apache Camel
- Social Media and Social Data
- Camel Twitter
- Demonstration

Enterprise Integration Patterns



“If you are involved with the operation or development of an enterprise application, there will doubtless come a time when you will need to integrate your application with another using the emerging preferred approach of messaging.”

Randy Stafford

Oracle

Enterprise Integration Patterns

Integration, integration, integration...

- Why do we need patterns for integration?
- Why is it so hard?
- Asynchronous messages
- Where and when to use them?
- Cloud Computing depends on it

Apache Camel

<http://camel.apache.org>



What is Apache Camel?

A Java framework that enables the developer to:

- Use implementations of Enterprise Integration Patterns
- Design routes for Enterprise Integration Patterns
- Use out-of-the-box or develop components and endpoints
- Process asynchronous and synchronous messages
- Connect distinct and independent systems
- Receive, transform and deliver data

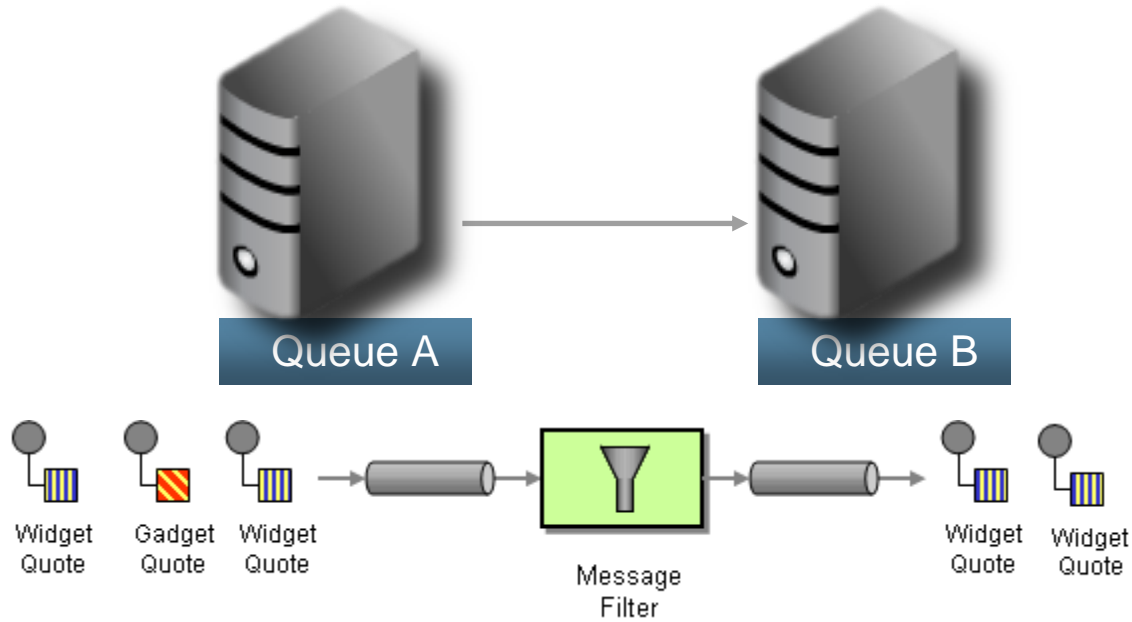
Quick overview of Apache Camel

The core of Camel is about

- Components
- Endpoints
- Routes
- Exchanges and Messages
- Consumers
- Producers
- Processors

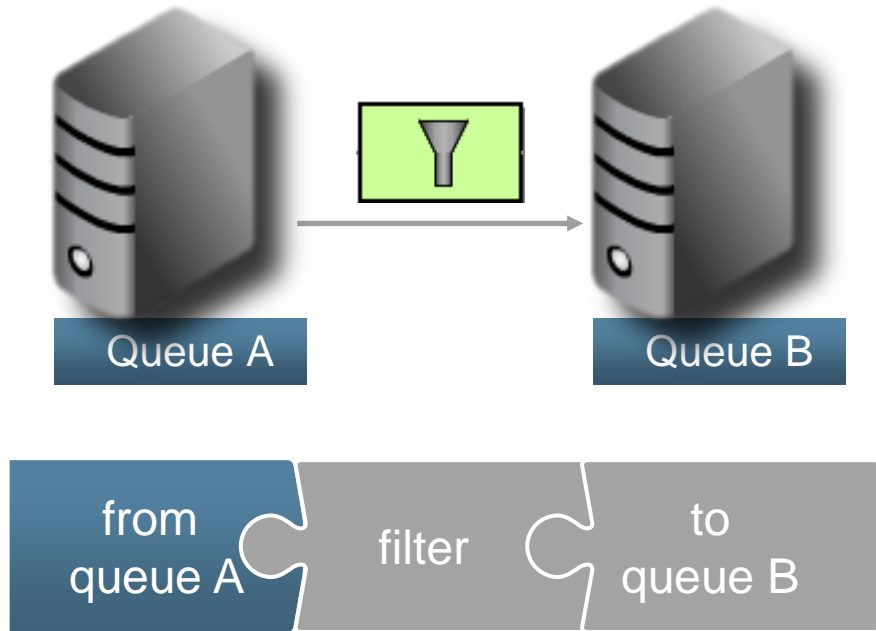
What can you do with Apache Camel?

Example of exchanging and filtering messages between queues



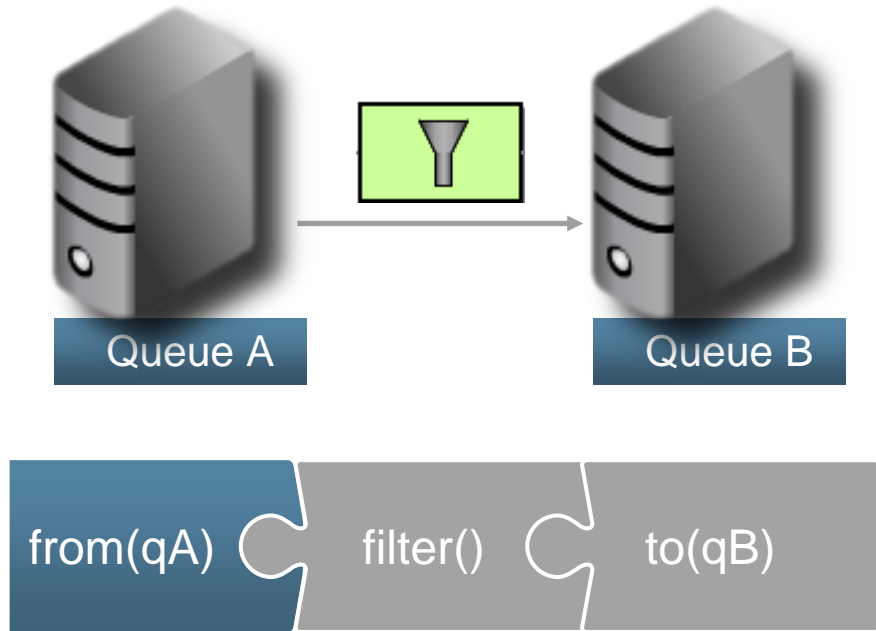
What can you do with Apache Camel?

Example of exchanging a message between queues



What can you do with Apache Camel?

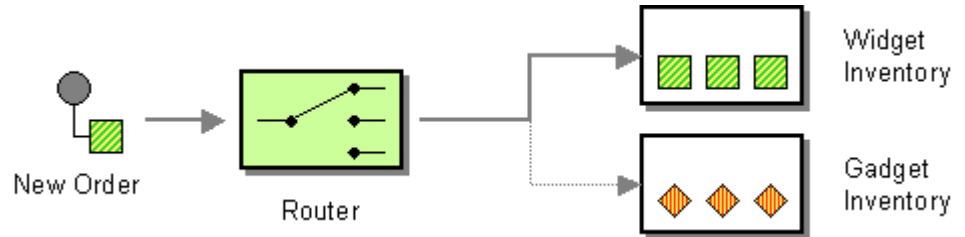
Example of exchanging a message between queues



Example using Spring

Because you love XML

```
<camelContext xmlns="http://camel.apache.org/schema/spring">
  <route>
    <from uri="file:///var/usr/inbox/">
      <choice>
        <when> <xpath>$foo = 'widget'</xpath>
          <to uri="seda:widget"/> </when>
        <when> <xpath>$foo = 'gadget'</xpath>
          <to uri="seda:gadget"/> </when>
        <otherwise> <to uri="seda:lixo"/> </otherwise>
      </choice>
    </route>
  </camelContext>
```

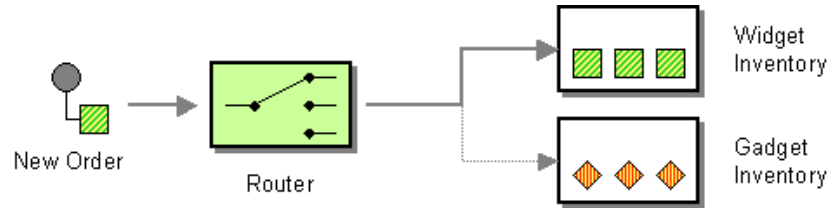
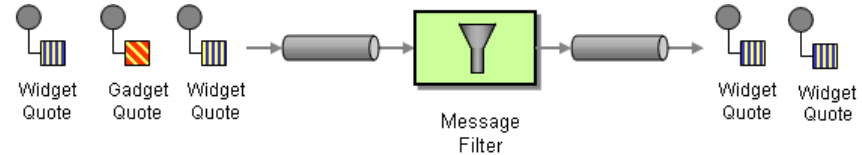


Examples using Scala

With the Scala DSL

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

```
"direct:b" ==> {  
  when(_.in == "<hallo/>") {  
    to ("mock:c")  
  } otherwise {  
    to ("mock:e")  
  }  
  to ("mock:d")  
}
```



Apache Camel

Implementation of Enterprise Integration Patterns

- DSLs for Java, Spring XML and Scala
- Standard URIs for Endpoints
- Message routing based on Predicates and Expressions
- Lots of components
 - JMS, HTTP, MINA, JDBC, FTP, WebServices, EJB, JPA, Hibernate, IRC, JCR, AS/400, LDAP, Mail, Nagios, POP, Printers, Quartz, Restlet, RMI, RSS, Scalate, XMPP...
- Integrates with CDI
 - CamelContext can be started with EJB3's @Startup/@Singleton

Social Media and Social Data



Social Media

The Rise of Social Media as a Communications Channels

- 29% of consumers post negative comments on social networks
- 49% of 16-24 y'o have posted negative comment following bad CX
- 71% that had complaints on social networks were ignored

- Of 29% who did get responses, 51% had positive reaction after
- 17% of those who had a response, wrote a positive comment
- 13% deleted their negative post

Source: 2011 Customer Experience Index Report “The Era of Impatience”
<http://www.oracle.com/us/products/applications/uk-cei-report-1641675.pdf>

Social Data

Reading and processing all that data

- Twitter for instance
 - Twitter, on average, receives 3,9k tweets per second
 - Peaks during main events can reach 20k tweets per second
 - This can more than 300 million tweets per day
 - One tweet may have 140 characters
 - 560 bytes (Twitter uses UTF-8)
 - Per day, more than 150 GB of raw text tweets
 - Not counting indexing and storage overhead
- Let's not forget about Facebook, Google+, LinkedIn and many others

Social Data

“It is not information overload.
It’s filter failure”

Clay Shirky

Independent Consultant, Teacher and Writer

Filtering and Processing Social Data

It looks like a lot with Enterprise Integration Patterns, isn't?

- Read status updates
- Filter based on general keywords
- Identify the issue based on their content
- Route to who will best process that tweet
- Reply if possible
- Store for future conversations with original sender
 - Store retweets that mention users who thought the same thing
- Customize as needed

Camel Twitter

Component for Apache Camel



Camel Twitter

The initial idea: March 2009

<http://blog.brunoborges.com.br/2009/03/leverage-eip-with-apache-camel-and.html>

B3

Blog do Bruno Borges

Início

Projetos

26 março 2009

Leverage EIP with Apache Camel and Twitter



I had no clue **Enterprise Integration Patterns** could actually work together with **Twitter**, for a business need. The thing is that, everybody is looking for ways to integrate everything within everything. And that's what this article tries to explain.

The idea to integrate **Apache Camel** and **Twitter** came when a customer was looking for a way to advertise on Twitter into different accounts, each one for different segments as no one actually wants to follow 100+ departments in the same account. That said, the enterprise will have 100+ Twitter accounts and its customers will now be able to follow only the departments they actually want to.

Camel Twitter

The initial idea: March 2009

<http://blog.brunoborges.com.br/2009/03/leverage-eip-with-apache-camel-and.html>

B3

Blog do Bruno Borges

Início

Projetos

30 março 2009

Leverage EIP with Apache Camel and Twitter: Part 2



On my first post I introduced you about a problem a customer brought to me to solve. And I thought using Apache Camel to advertise on Twitter could actually fit into his business needs (and mine as well). The first thing I had to think about was on how to receive ads from different customers in different formats. This way I could let them continue to use any format they are already using to publish ads online.



Camel Twitter

Proposed as a Camel Component to the ASF in 2009

- CAMEL-1520
 - <https://issues.apache.org/jira/browse/CAMEL-1520>
- Presented at ApacheCon NA'09 during the BarCamp
- Discussions with Camel committers at ApacheCon led me to build a broader component, that could support other social networks
- Challenge: same URIs and features for different Social Networks (“social data providers”). Starting with Facebook, Twitter, LinkedIn, Foursquare, and (the already dead) Google Buzz

Camel Social

How would it look like?

- Started in 2010
- Goal
 - The Camel Social component objective was to be able to poll social data from several networks in a uniform way to be processed through a route
- URI format
 - `social://[social provider] [/social path]?parameters`
- Features
 - Post, Read, Search

Camel Social

Examples

Sending a tweet

```
"direct:foo" to "social://twitter/status"
```

Reading a Facebook Timeline

```
"social://facebook/timeline/brunocborges" ==> {  
    to("log:facebookWall")  
}
```

Searching for events on Facebook

```
"direct:doSearch" to "social://facebook/events"  
                  to "log:events"
```

Camel Social

An unsuccessful attempt

<http://code.google.com/p/camel-social>

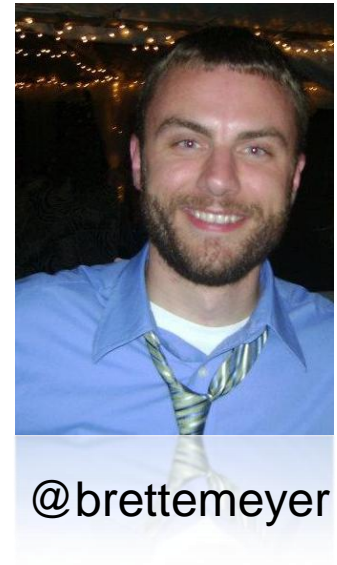
- Problem
 - It was very hard to design an standard API for different Social Nets
- Died in 2010
- What about Spring Social?
 - Didn't exist back then
 - Now they have an abstract API for Social Providers and for OAuth
 - One extra API per Social Network
 - It “could” bring the component back to life

Camel Twitter

A rebirth, thanks to GitHub

- May 2011 – project rebooted on GitHub
 - Focus only on Twitter
- <http://github.com/brunoborges/camel-twitter>
- Kudos to Brett Meyer, who finished the job
- Bilgin Ibryam added support for the Twitter Streaming API

- Thank you guys! I owe you two a beer :-)



Camel Twitter

Final version

- Available since Apache Camel 2.10
- Features
 - Send a tweet (update status)
 - Send and read Direct Messages (DMs)
 - Search using REST (polling) or Streaming
 - Sample streaming from public tweets
 - Timeline reads for home, mentions, retweets, specific user
- Uses the well-know Twitter4J library (Apache licensed)

Examples

Using the Scala DSL

Sending a tweet

```
"jms:queue:tweetingQueue1" to "twitter://timeline/user"
```

Reading a Timeline

```
"twitter://timeline/home?type=polling&delay=5" ==> {  
  to("log:homeTweets")  
}
```

Searching for keywords

```
"direct:doSearch" to "twitter://search?keywords=JavaOne"  
                  to "log:homeTweets"
```

Camel Twitter Demo

Searching for pictures about Batman



Camel Twitter Demo

How it works

- Connects to the Search Streaming channel based on keywords
- Filter for tweets with media only
- Check Coherence Cache and filter duplicates
- Put image/tweet URL as key/value on Coherence Cache
- Transform tweets into small POJOs
- Transform POJOs into JSON Strings
- Sends to users connected to the WebSocket channel

* based on the Camelympics project developed by Bilgin Ibryam

<https://github.com/bibryam/camelympics>

About Camel Coherence Component

The #1 Distributed Caching technology

- A fork of the Camel Cache component
 - Instead of EHCache, now uses Oracle Coherence
- Same features as the core component
 - **Operations:** add, check, get, remove, removeAll
 - **URI:** coherence://cacheName
- Coherence is LRU by default ('local-scheme')
 - Great for the demo! Stores the recent tweets without blowing the Heap
 - If more storage is needed, deploy new Coherence Cache instance

QUESTIONS?



THANK YOU!

@brunoborges

blogs.oracle.com/brunoborges



The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

