



APACHE®

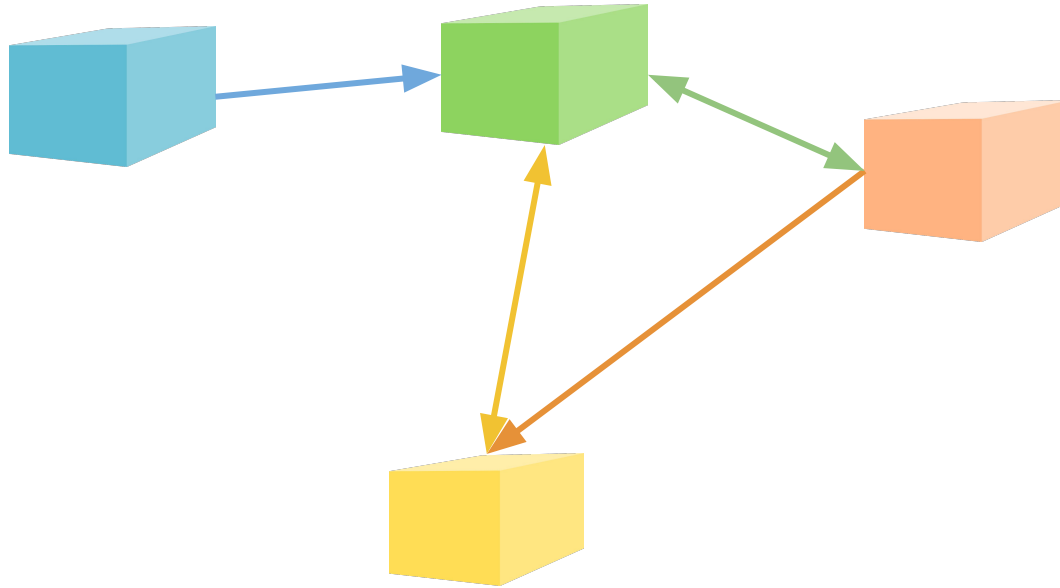
Camel

Integrate everything

Zoran Regvart

 @zregvart

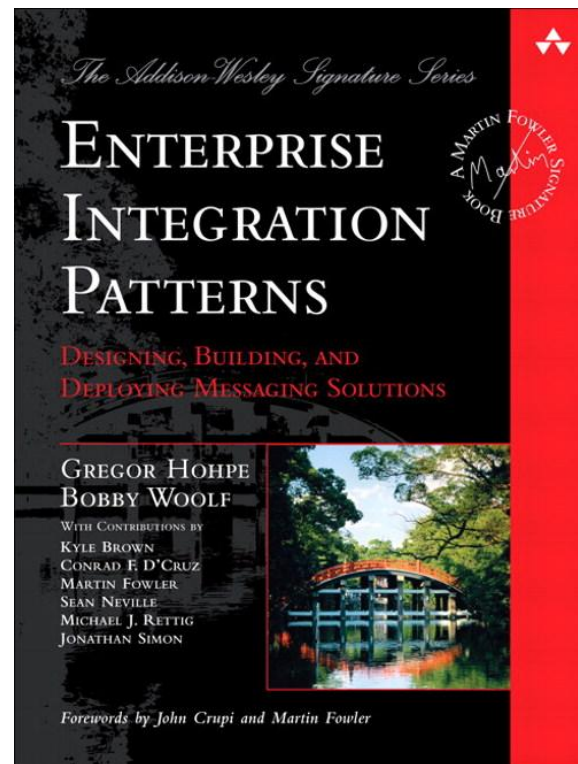
System integration



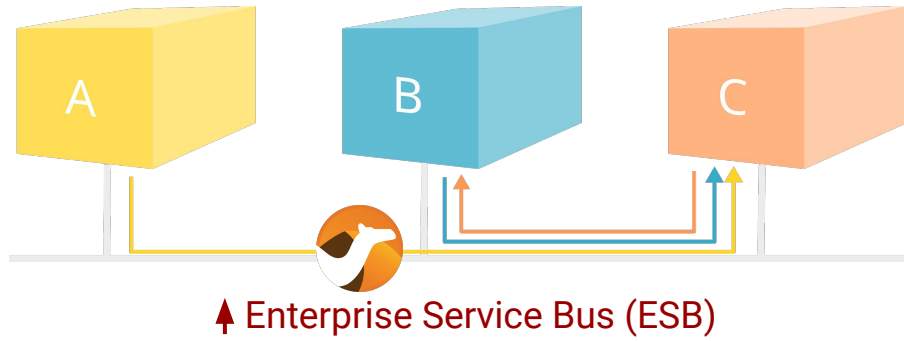
Integration patterns

Book by Gregor Hohpe and Bobby Woolf

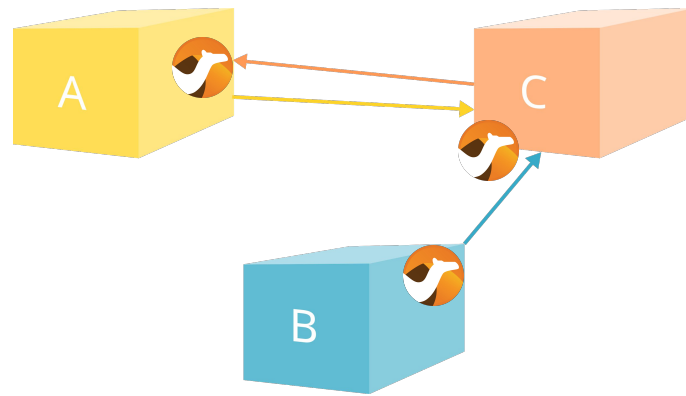
Common patterns and best practices



Not invented here



↓ Microservices



Apache Camel

Integrates with ~300 different systems using best practice patterns

Developed in **Java**, needs **Java Virtual Machine** to run

Actually an acronym: **C**oncise **A**pplication **M**essage **E**xchange **L**anguage

11 year anniversary coming this July



CC Image courtesy of [University of Liverpool Faculty of Health & Life Sciences](#)



CC Image courtesy of [Keven Law](#) from Los Angeles, USA

Concepts

Patterns how to reuse integration best practices

Components how to integrate with systems

Endpoints how to integrate with specific systems

Data transformation how to adapt data

Route how to define the flow of data

Patterns

Aggregate

Circuit breaker

Delay

Dynamic routing

Enrich

Failover

Filter

Idempotency

Load balancing

Marshall/Unmarshall

Multicast

Resequence

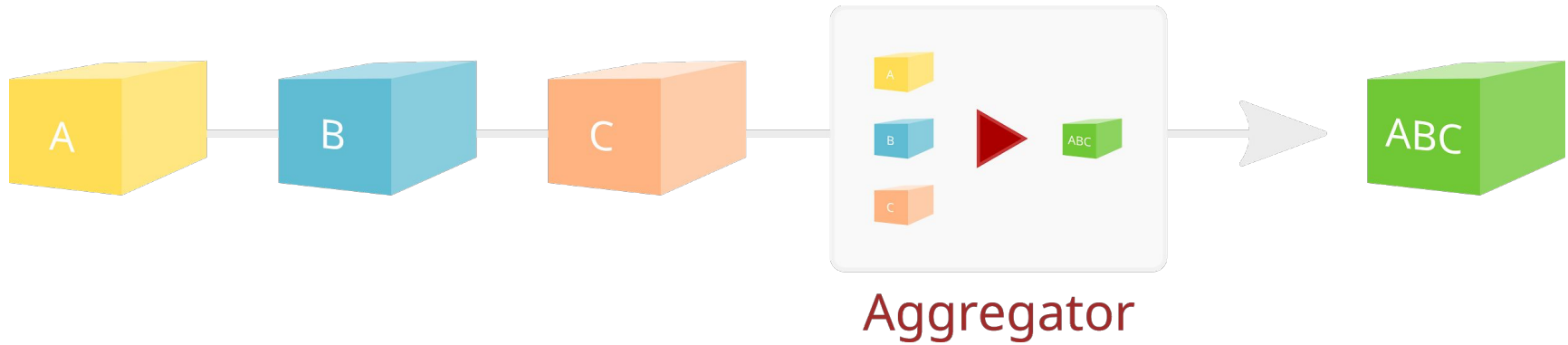
✧ Saga ✧

Service call

Split

Throttle

Aggregate pattern



Components

~300 systems/technologies you can integrate with

HTTP, message systems, APNS, Amazon *, AS2, Box, Braintree, Cassandra, CMIS, Consul, CouchDB, DNS, Docker, ElasticSearch, Facebook, File system, FTP *, GIT, GitHub, Google *, Hadoop, Hipchat, HL7, Kafka, Kubernetes, LDAP, LinkedIn, Mail, MongoDB, Nagios, OData, Openshift, Printer, REST, RSS, Salesforce, Scheduling, ServiceNow, SIP, Slack, SMS, SOAP, Solr, Spark, Splunk, SSH, SQL, Telegram, Twitter, Weather, Websocket, XMPP, Yammer, Zookeeper

Endpoints

One **component** can have many **endpoints**

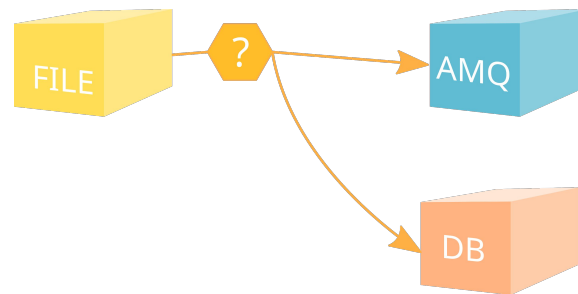
Endpoints can receive (*consume*) or send (*produce*)

```
"file:/data/orders?recursive=true"
```

schema path query

Endpoint URI

Routes



Route defines the flow of data

```
from("file:/data/orders")
  .unmarshal().json()
  .choice()
    .when()
      .jsonpath("$. [?(@.backorder==true)]")
      .to("sjms:queue:back_orders")
    .otherwise()
      .to("sql:INSERT INTO ORDERS...")
```



```
<route>
  <from uri="file:/data/order" />
  <unmarshal>
    <json />
  </unmarshal>
  <choice>
    <when>
      <jsonpath>$...</jsonpath>
      <to uri="sjms:queue:..." />
    </when>
    <otherwise>
      <to uri="sql:INSERT..." />
    </otherwise>
  </choice>
</route>
```

```
from("file:/data/orders")
  .unmarshal()
  .json()
  .choice()
  .when()
    .jsonpath("$...")
    .to("sjms:queue:...")
  .otherwise()
    .to("sql:INSERT...")
```

Demo



"25,67"



"25,67"



Convert
to JSON



Runtime

Include in your own application: use as a **library**

Within standalone runtimes: **Camel Main, Spring Boot, Vert.x**

Application server: **JEE - Wildfly** or **OSGI - Apache Karaf**

Camel and IoT

Data coming from various **devices**, in various **formats** over various **protocols**

IoT is an integration problem

On device

Constrained

Low power

Specialised

At the edge

Connectivity

Messaging

Computing

At scale

Distribution

Scheduling

Insights



kubernetes



OPENSIFT

IoT components in Camel

CoAP

Eclipse Kura

Eclipse Milo (OPC UA): client, server

IEC 60870: client, server

MQTT (Eclipse Paho, FuseSource MQTT client, *JMS and Apache ActiveMQ)

PubNub

SNMP

JBoss Camel Developer Tools

The screenshot displays the JBoss Camel Developer Tools IDE interface. The main workspace shows a Camel blueprint editor with three routes:

- Route generate-order:** Starts with a timer component (timer:order?period=500...), followed by a Bean orderGenerator, a SetHeader updateFileNa... component, and a file:workdir component.
- Route file-to-jms-route:** Starts with a file:workdir component, followed by a Log logReceivingOrder component, and an activeMQ:incomingOrders component.
- Route jms-cbr-route:** Starts with an activeMQ:incomingOrders component, followed by a Choice router. The Choice router has three branches:
 - When forder:order:order:rc...:** Contains a Log logLIK component and a file:workdir/outlik component.
 - When forder:order:order:rc...:** Contains a Log logJUS component and a file:workdir/outbus component.
 - Otherw...:** Contains a Log component and a files component.After the Choice router, there is a Log logEndProcessing component.

The bottom panel shows the configuration for the CamelContext amq-example-context. The configuration table is as follows:

Property	Value
Allow Use Original Message	
Auto Startup	true
Delayer	
Depends-on	
Error Handler Ref	
Handle Fault	
Id	amq-example-context
Lazy Load Type Converters (deprecated)	<input type="checkbox"/>
Management Name Pattern	#name#
Message History	true
Runtime Endpoint Registry Enabled	

Hawt.io

The screenshot displays the Hawtio dashboard for a Camel route named 'bytesToAMQ'. The route configuration on the left includes a timer, a 'Hello World' message, a delayed message, a header, and a message body.

The 'Routes' table shows the following data:

State	Context	Route	Completed #	Failed #	Failed Handled #	Total #	Inflight #	Mean Time	Min Time	Max Time	Total Time	Delta Time
⊙	camel-1	bytesToAMQ	80	0	0	80	0	1005	1003	1018	80463	1
⊙	camel-1	foo	3	0	0	3	0	32	4	89	97	0
⊙	camel-1	route1	3	0	0	3	0	48	13	115	145	4
⊙	camel-1	timerToAMQ	80	0	0	80	1	1506	1504	1525	120520	0

The 'Route metrics' section for 'bytesToAMQ' includes a frequency chart and a histogram. The frequency chart shows calls/second (80 total) for 1 min, 5 min, 15 min, and Mean. The histogram shows percentiles from 75% to 99.9%.

The 'Logs' section shows the following entries:

- 2014-09-08 09:23:51.970 WARN org.infinispan.configuration.parsing.Parser\$1 ISPN000176: The 'wakeUpInterval' attribute of the 'eviction' configuration XML element is deprecated. Setting the 'wakeUpInterval' attribute of the '...
- 2014-09-08 09:23:53.905 WARN io.hawt.sample.Main Don't run with scissors!
- 2014-09-08 09:23:53.906 ERROR io.hawt.sample.Main Someone somewhere is not using Fuse! :)
- 2014-09-08 09:23:53.909 ERROR io.hawt.sample.Main Expected exception for testing: java.lang.NullPointerException

More tooling

Camel [IDEA plugin](#)

Camel [LSP server](#) (Eclipse IDE, Visual Studio Code, Eclipse Che)

Route [test coverage](#)

Camel community

~400 contributors

63 comitters (~15 companies)

31 PMC members

~120 releases

~1000 subscribers on users@ mailing list

Resources

[Camel website](#) and the new documentation on [GitHub](#)

Books: [Camel in Action 2nd edition](#) (☆free IoT chapter☆), [Apache Camel Developer's Cookbook](#), [Camel Design Patterns](#), [Apache Camel Essentials](#)

IRC [#apache-camel](#) on Freenode

Gitter <https://gitter.im/apache/apache-camel>

Mailing list users@camel.apache.org

A close-up photograph of a brown camel resting its head on the ground. The camel's head is the central focus, with its eyes closed and its large, textured nose pointing towards the viewer. The camel's fur is a rich, warm brown color. The background is a light-colored, sandy or dusty ground with some small, dry leaves scattered around. The overall lighting is bright and natural, suggesting an outdoor setting.

Thank you!

Questions?