Integration vs. Mule ESB vs.

MaibornWolff et al

Consulting
Developing
Speaking
Coaching
Writing

Main Tasks

Evaluation of Technologies and Products

Requirements Engineering

Enterprise Architecture Management

Business Process Management

Architecture and Development of Applications

Planning and Introduction of SOA

Integration of Legacy Applications

Cloud Computing

Contact

Email: kai.waehner@mwea.de

Blog: www.kai-waehner.de/blog

Twitter: @KaiWaehner

Social Networks: Xing, LinkedIn

What is the Key Message?

MaibornWolff et al

?

Do not reinvent the „integration wheel“!

Do not reinvent the „integration wheel“!

There are some good alternatives for Integration!

MaibornWolff et al

Do not reinvent the „integration wheel“!

There are some good alternatives for Integration!

Often an ESB is the wrong Choice!

MaibornWolff et al

1) Systems Integration

2) Integration Frameworks

3) Spring Integration

4) Mule ESB

5) Apache Camel

6) And the Winner is ...

MaibornWolff et al

1) **Systems Integration**

2) Integration Frameworks

3) Spring Integration

4) Mule ESB

5) Apache Camel

6) And the Winner is ...

MaibornWolff et al

Growth

- Applications
- Interfaces
- Technologies
- Products

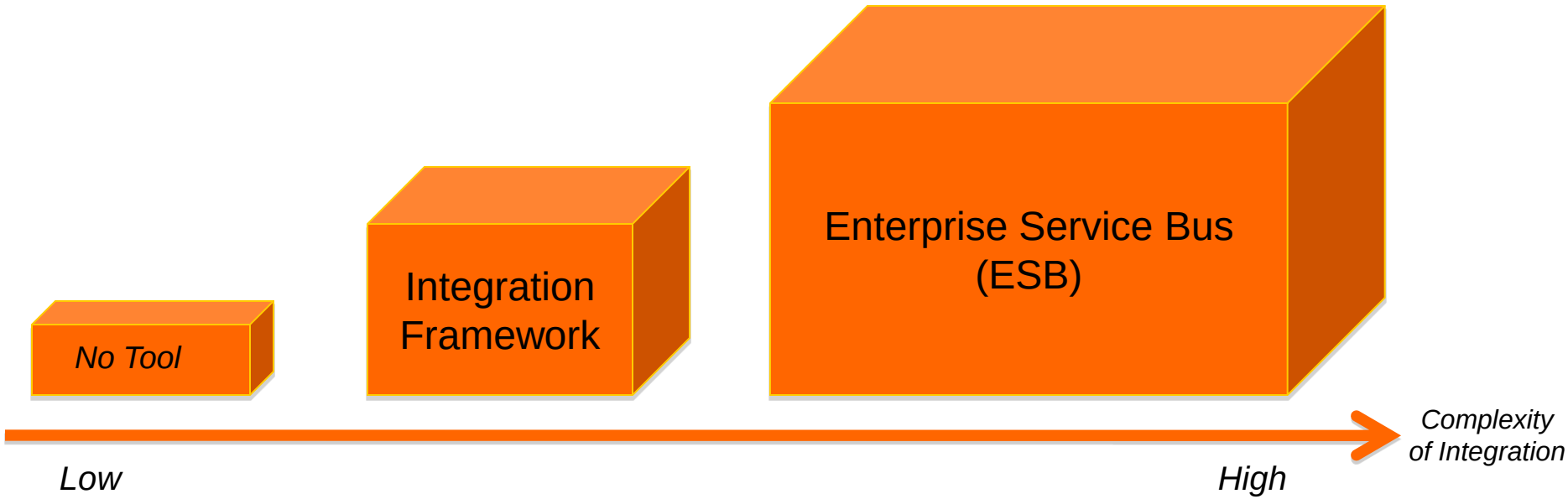

NEW
PARADIGM
AHEAD

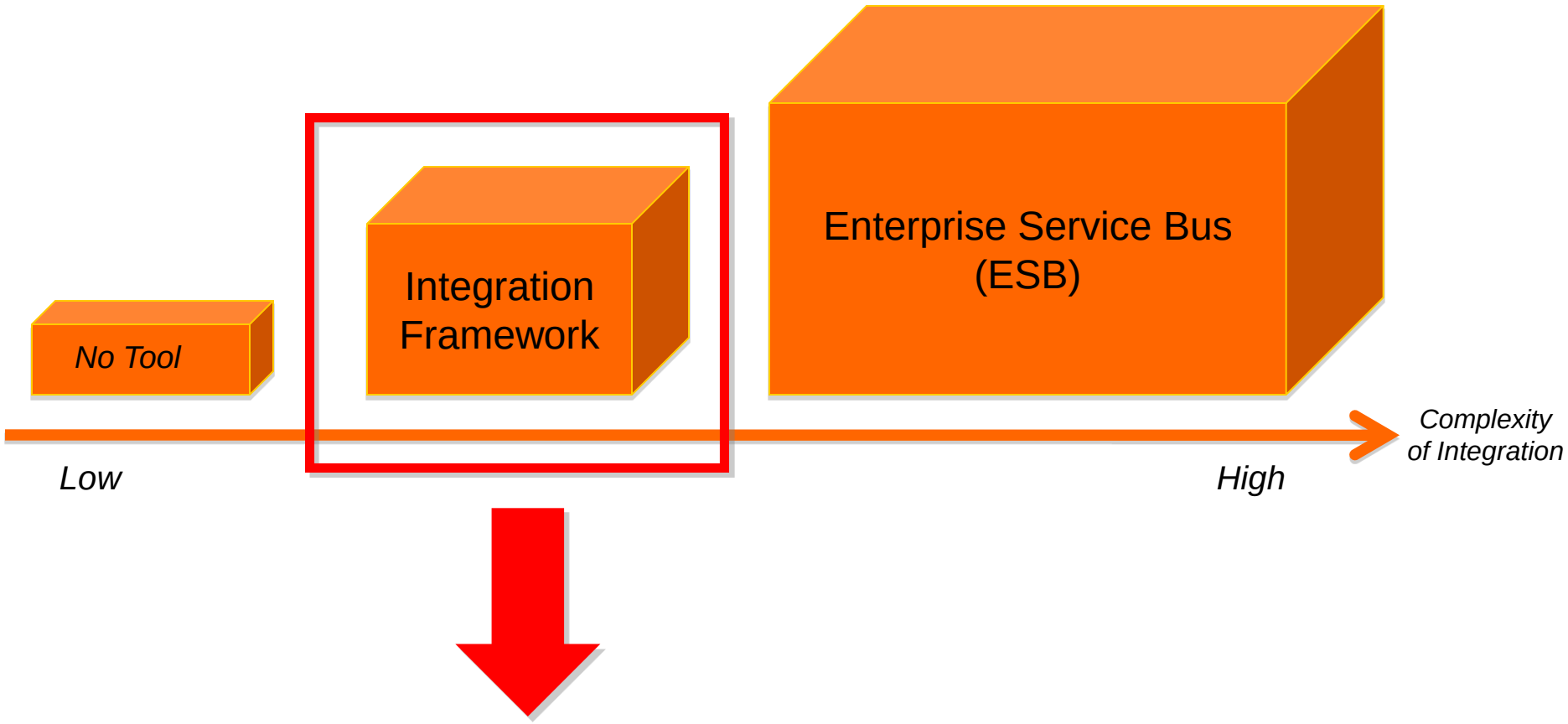
Everybody communicates to everybody

All Roads lead to Rome ...

Alternatives for Systems Integration

MaibornWolff et al





Spring Integration vs. Mule ESB vs. Apache Camel

MaibornWolff et al

1) Systems Integration

2) Integration Frameworks

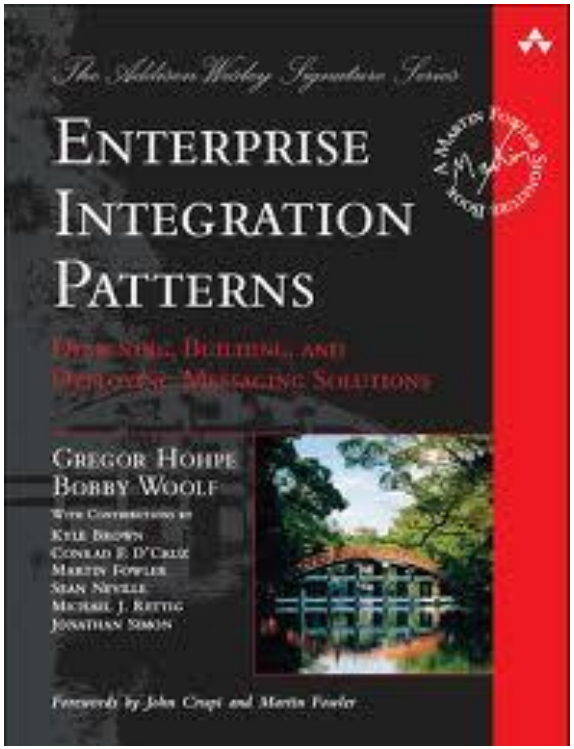
3) Spring Integration

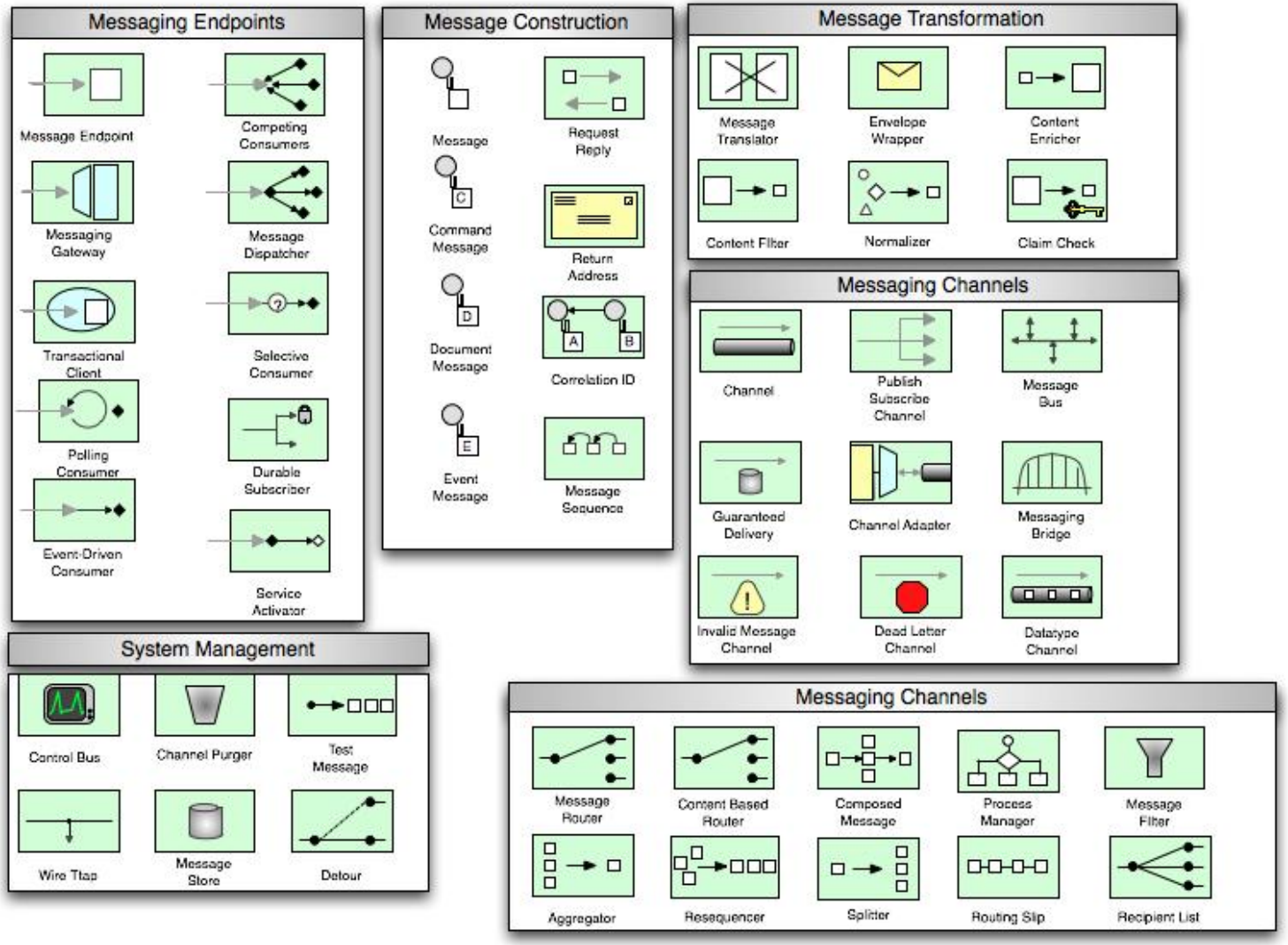
4) Mule ESB

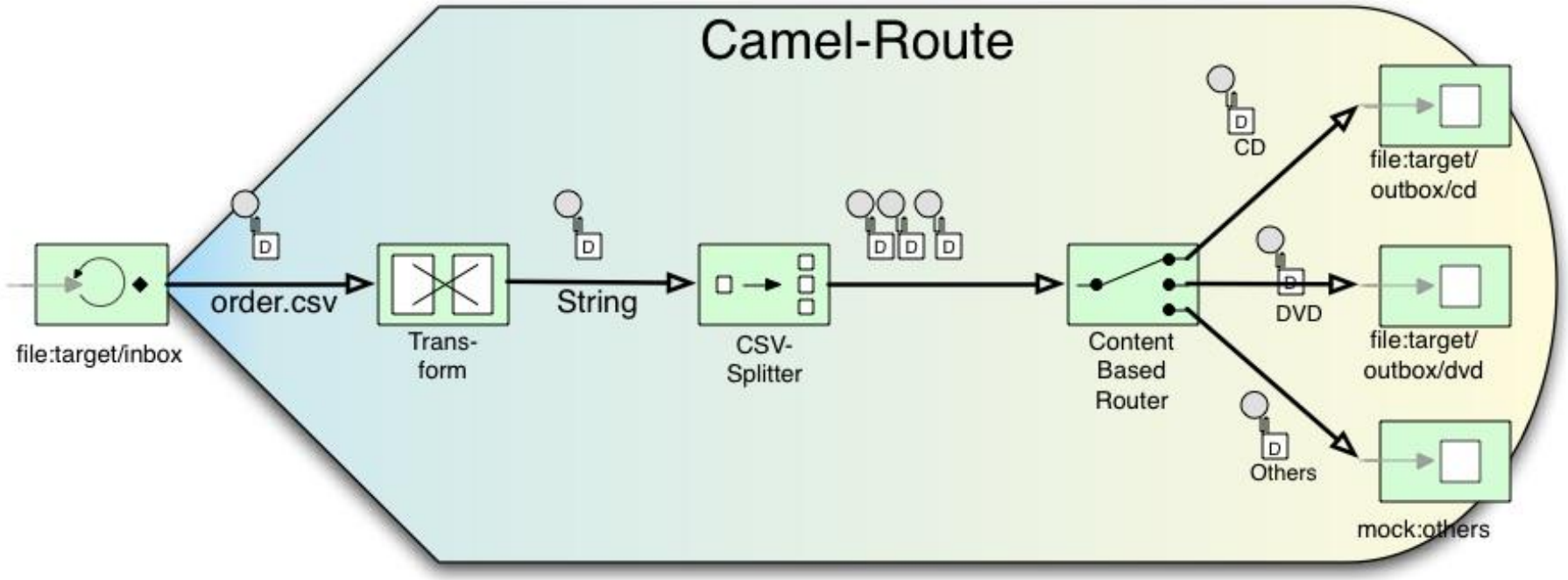
5) Apache Camel

6) And the Winner is ...

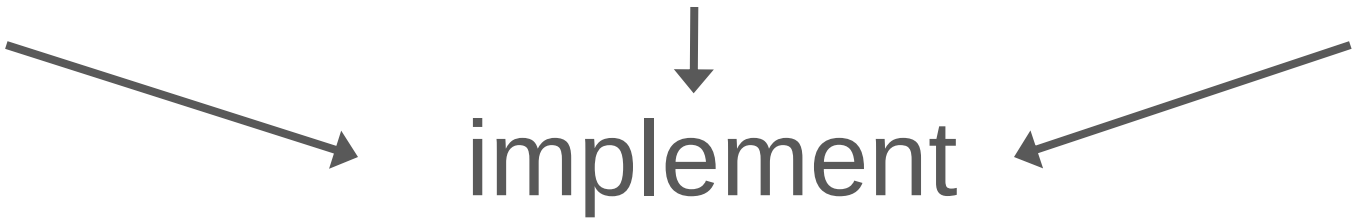
- Standardized Modeling
- Efficient Realization
- Automatic Testing


ENTERPRISE
INTEGRATION
PATTERNS
GREGOR HOHPE
BOBBY WOOLF

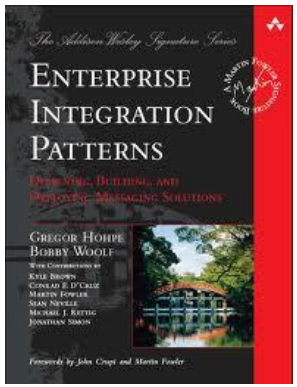

Messaging Endpoints
Message Endpoint
Competing Consumers
Messaging Gateway
Message Dispatcher
Transactional Client
Selective Consumer
Polling Consumer
Durable Subscriber
Event-Driven Consumer
Service Activator
Message Construction
Message
Request Reply
Command Message
Return Address
Document Message
Correlation ID
Event Message
Message Sequence
Message Transformation
Message Translator
Envelope Wrapper
Content Enricher
Content Filter
Normalizer
Claim Check
Messaging Channels
Channel
Publish Subscribe Channel
Message Bus
Guaranteed Delivery
Channel Adapter
Messaging Bridge
Invalid Message Channel
Dead Letter Channel
Datatype Channel
System Management
Control Bus
Channel Purger
Test Message
Wire Tap
Message Store
Detour
Messaging Channels
Message Router
Content Based Router
Composed Message
Process Manager
Message Filter
Aggregator
Resequencer
Splitter
Routing Slip
Recipient List


Camel-Route
file:target/inbox
order.csv
Trans-
form
String
CSV-
Splitter
Content
Based
Router
CD
DVD
Others
file:target/
outbox/cd
file:target/
outbox/dvd
mock:others

Integration vs. **Mule ESB** vs.



implement



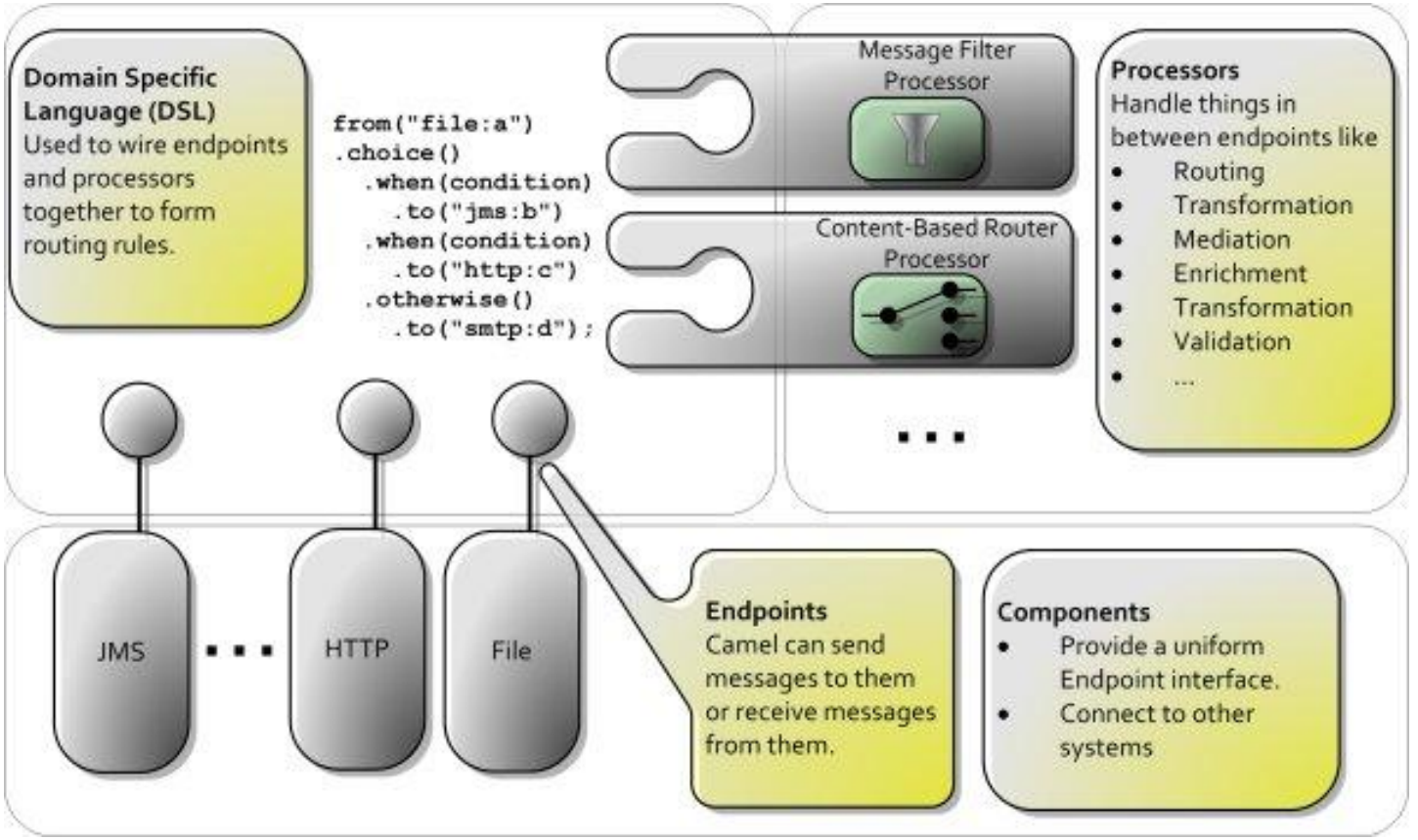
- Open source
- Basic concepts / architecture
- Testability
- Commercial support
- Error handling
- Monitoring
- Enterprise readiness
- Developer-centric vs. designer-centric
- Expandability
- Deployment
- Popularity
- Tool support
- Connectivity
- Domain specific language (DSL)

Similarities

Integration of different Technologies

*„A **domain-specific language (DSL)** is a programming language or specification language dedicated to a particular problem domain, a particular problem representation technique, and / or a particular solution technique.“*

Wikipedia



http://java.dzone.com/articles/apache-camel-integration

(Exemplarily: Apache Camel => Concepts are all the same, only different names)

Standalone

Application Server

Web Container

Spring Container

OSGi

Cloud

- Maturity
- Transactions
- Concurrency
- Error handling
- Monitoring
- Testability

Integration vs. **Mule ESB** vs.

Apache License

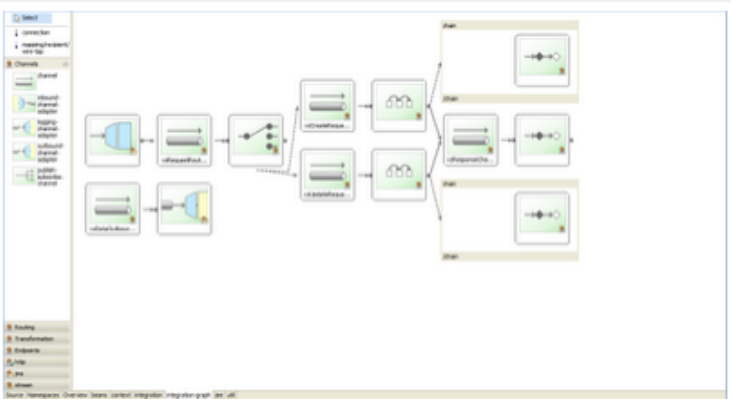
Common Public Atttribution License (CPAL)

Apache License

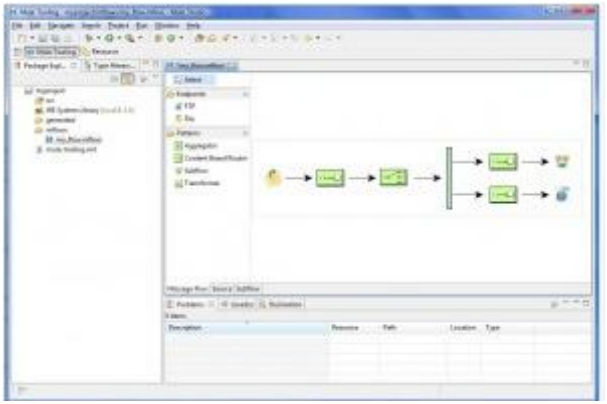
Integration vs. **Mule ESB** vs.

MaibornWolff et al

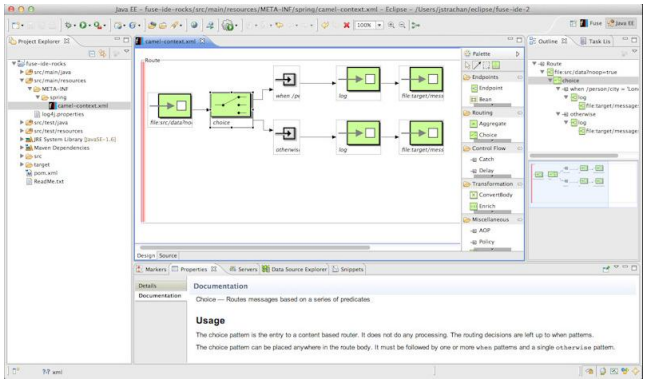
Integration Graph for Spring Integration



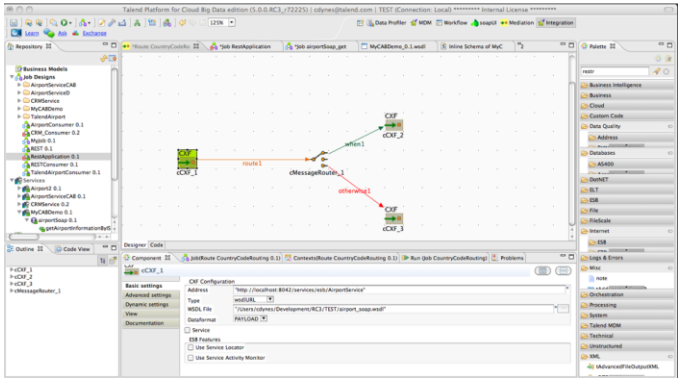
Mule Studio for Mule ESB



Fuse IDE for Apache Camel



Talend Studio for Apache Camel



MaibornWolff et al

- Concepts of each Framework
- Code Example
- Live Demo

MaibornWolff et al

- Open source
- Basic concepts / architecture
- Testability
- Commercial support
- Error handling
- Monitoring
- Enterprise readiness
- Developer-centric vs. designer-centric
- Expandability
- Deployment
- Popularity
- Tool support
- **Connectivity**
- **Domain specific language (DSL)**

Focus

MaibornWolff et al

1) Systems Integration

2) Integration Frameworks

3) Spring Integration

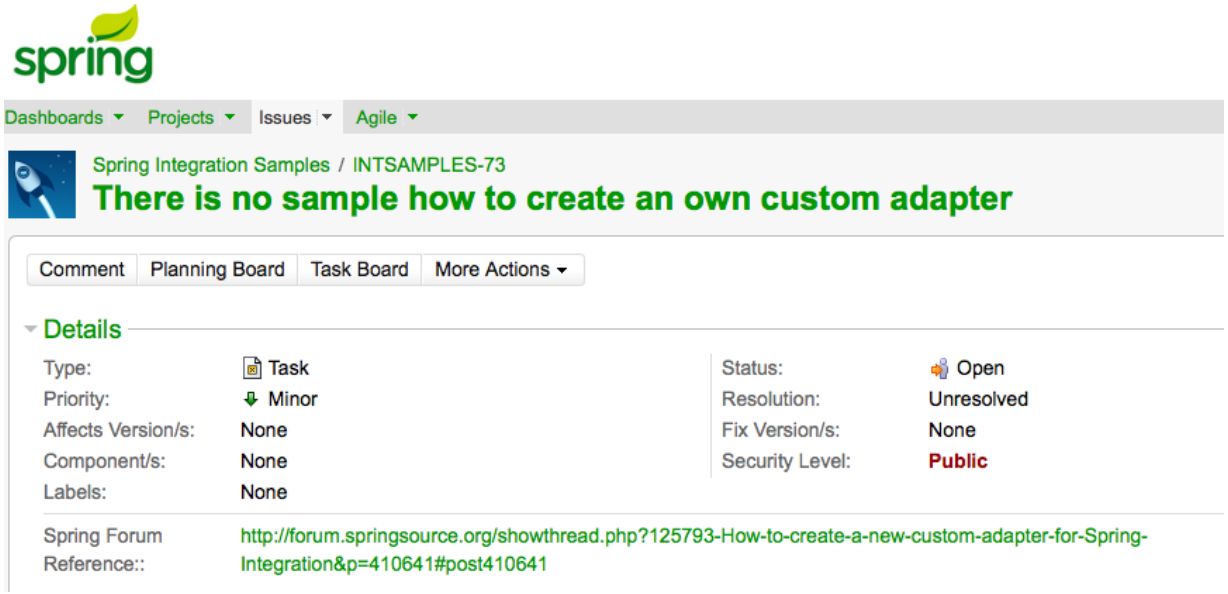
4) Mule ESB

5) Apache Camel

6) And the Winner is ...

AMQP
Feed
File
FTP(S)
GemFire
HTTP
TCP
UDP
JDBC
JMS
Mail
MongoDB
Redis
RMI
SFTP
Stream
Twitter
Web Service
XML
XMPP

How to create own custom adapter?



XML

(Not production-ready yet)

```
<file:inbound-channel-adapter
        id="incomingOrders"
        directory="file:incomingOrders"/>

<payload-type-router input-channel="incomingOrders">
        <mapping type="com.kw.DvdOrder"
                channel="dvdOrders" />
        <mapping type="com.kw.VideogameOrder"
                channel="videogameOrders" />
        <mapping type="com.kw.OtherOrder"
                channel="otherOrders" />
</payload-ty router>

<file:outbound-channel-adapter
        id="dvdOrders"
        directory="dvdOrders"/>

<jms:outbound-channel-adapter
        id="videogamesOrders"
        destination="videogameOrdersQueue"
        channel="videogamesOrders"/>

<logging-channel-adapter id="otherOrders" level="INFO"/>
```

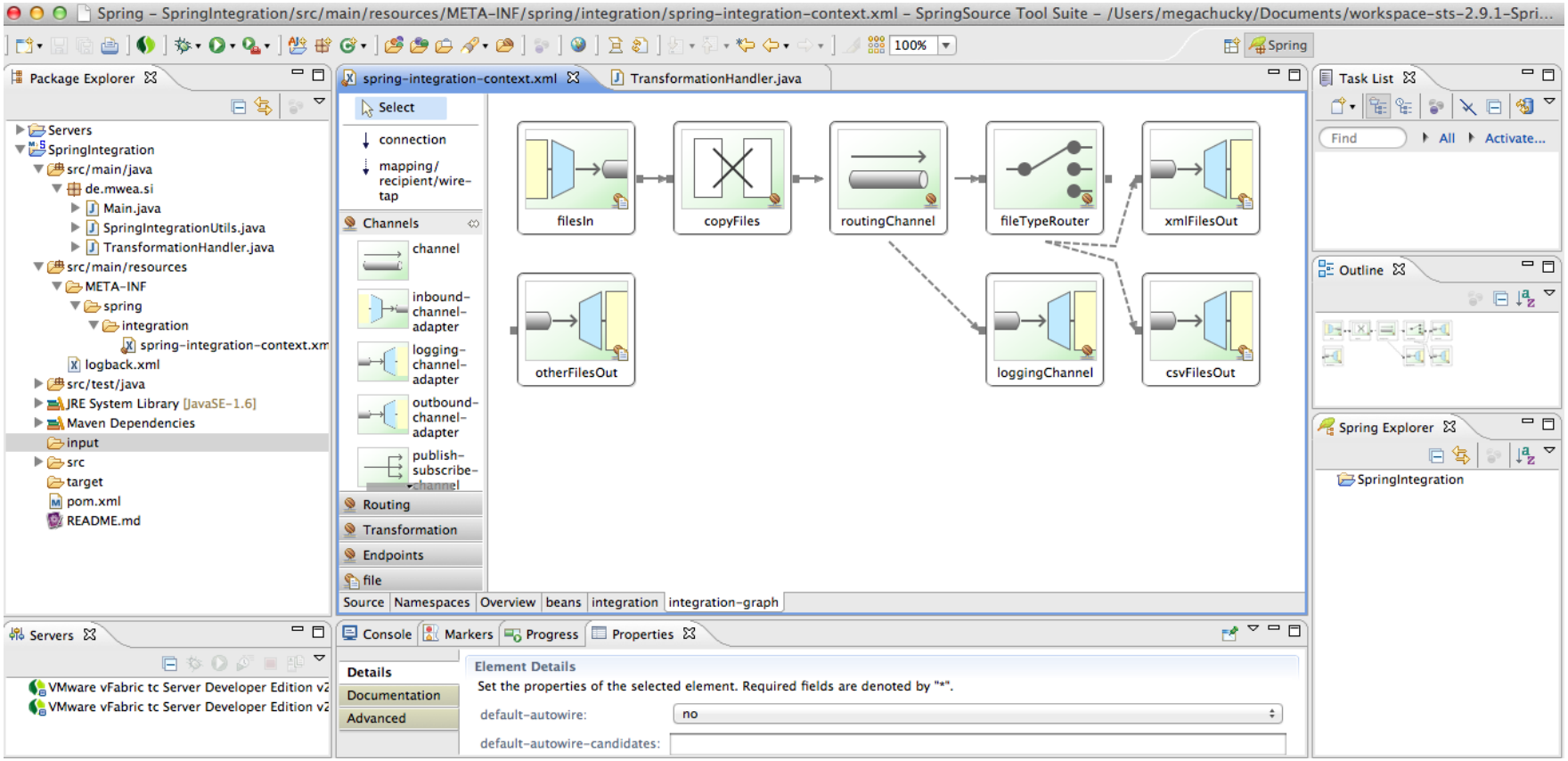
```
val messageFlow =
     filter.using{payload: String => payload == "World"} -->
     transform.using { payload: String => "Hello " + payload} -->
     handle.using { payload: String => println(payload) }

messageFlow.send("World")
```

http://blog.springsource.org/2012/03/05/introducing-spring-integration-scala-dsl/

Spring Integration in Action

Integration Graph

MaibornWolff et al



Pro

- Visual Designer for Spring Integration flows
- Vice versa Editing (code generation vs. coding by hand)
- Open Source (part of Spring IDE at github)
- Simple Eclipse plugin – „Just Spring Integration“
- Export of flow diagrams

Contra

- Non-intuitive usability
- Unclear diagrams
- Immature (e.g. missing documentation, problems with code examples)

MaibornWolff et al

1) Systems Integration

2) Integration Frameworks

3) Spring Integration

4) Mule ESB

5) Apache Camel

6) And the Winner is ...

Embedding Mule in a Java Application or Webapp

This page describes how to start and stop Mule from a Java application or to embed it in a Webapp (such as a JSP or servlet), and how to interact with Mule from your code in both scenarios.

Starting Mule from a Java Application

To start Mule from any Java application, you can call one of its configuration builders. To use Mule XML configuration:

```
DefaultMuleContextFactory muleContextFactory = new DefaultMuleContextFactory();
SpringXmlConfigurationBuilder configBuilder = new SpringXmlConfigurationBuilder("mule-c
muleContext = muleContextFactory.createMuleContext(configBuilder);
```

Make sure you store a reference to the MuleContext, as you will need it to stop Mule.

If you have multiple configuration files, you can provide a comma-separated list or an array of configuration files:

```
SpringXmlConfigurationBuilder configBuilder =
    new SpringXmlConfigurationBuilder(new String[] { "mule-config.xml", "another-conf
```

You then call the `start` method to start the server:

```
muleContext.start();
```

Stopping Mule from a Java Application

To stop Mule, you stop its context like this:

```
muleContext.stop();
muleContext.dispose();
```

http://www.mulesoft.org/documentation/display/MULE3USER/Embedding+Mule+in+a+Java+Application+or+Webapp

MaibornWolff et al

AS400 Data Queue
Abdera
Amazon SQS
jBPM
CICS CTG
CXF
Email
FTP
Hibernate
HTTP/S
Legs4Mule
IMAP/S

Servlet
SFTP
SMTP/S
SOAP
STDIO
TCP
UDP
VM
XMPP
WebSphere MQ
WSDL

Atom
Base64 encoded
Byte arrays
CSV
Encrypted

GZIP
Hex Strings
HTML/ XHTML
Java Objects

JSON
EDI
COBOL Copybook
XML

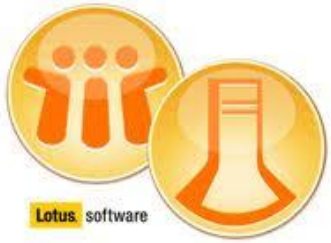
Amazon S3
Authorize.net
Apple Push
Bit.ly
CMIS

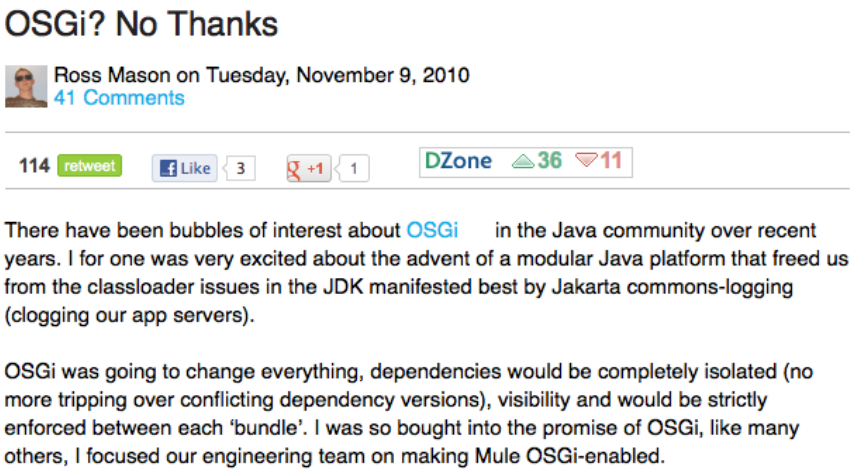
CyberSource
Facebook
Flickr
HBase
Magento

JCR
JDBC
Jersey
Jetty
JMS
LDAP
Multicast
POP3/S
Quartz
Restlet
RMI
SAP

Many more Connectors +
easy to create own Connectors

Several Proprietary Connectors available, for instance:





OSGi? No Thanks

Ross Mason on Tuesday, November 9, 2010
41 Comments

114 retweet Like 3 +1 1 DZone 36 11

There have been bubbles of interest about OSGi in the Java community over recent years. I for one was very excited about the advent of a modular Java platform that freed us from the classloader issues in the JDK manifested best by Jakarta commons-logging (clogging our app servers).

OSGi was going to change everything, dependencies would be completely isolated (no more tripping over conflicting dependency versions), visibility and would be strictly enforced between each 'bundle'. I was so bought into the promise of OSGi, like many others, I focused our engineering team on making Mule OSGi-enabled.

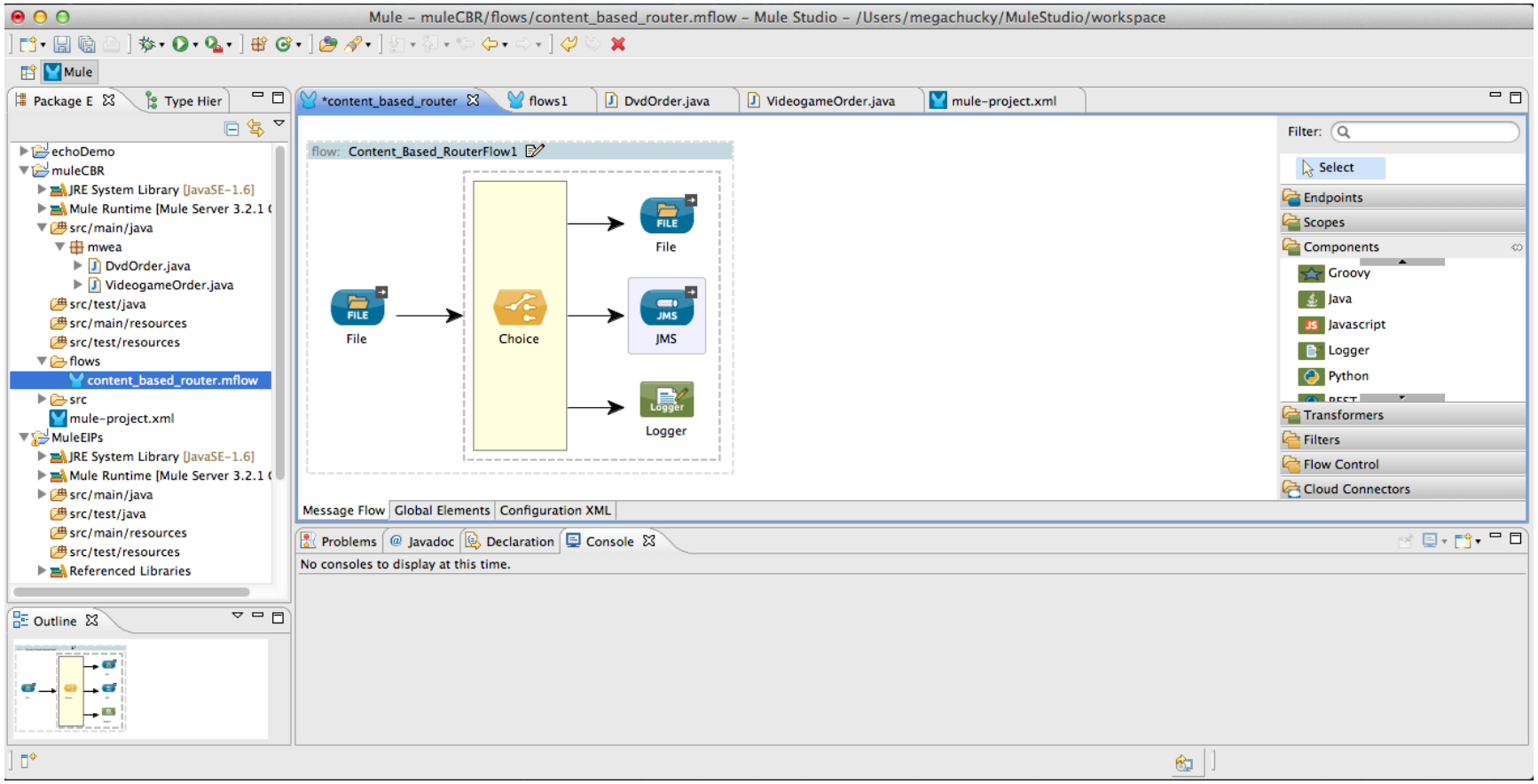
„OSGi adds another complexity to building applications. [...] OSGi is a great specification for middleware vendors, but a terrible specification for the end user.“

Ross Mason, MuleSoft, November 2010

XML

```
<flow name="muleFlow">
    <file:inbound-endpoint path="incomingOrders"/>
    <choice>
      <when expression="payload instanceof com.kw.DvdOrder"
            evaluator="groovy">
            <file:outbound-endpoint path="incoming/dvdOrders"/>
      </when>
      <when expression="payload instanceof com.kw.DvdOrder"
            evaluator="groovy">
            <jms:outbound-endpoint queue="videogameOrdersQueue"/>
      </when>
      <otherwise>
            <logger level="INFO"/>
      </otherwise>
    </choice>
</flow>
```

Mule in Action


Mule – muleCBR/flows/content_based_router.mflow – Mule Studio – /Users/megachucky/MuleStudio/workspace
Mule
Package E
Type Hier
echoDemo
muleCBR
JRE System Library [JavaSE-1.6]
Mule Runtime [Mule Server 3.2.1
src/main/java
mwea
DvdOrder.java
VideogameOrder.java
src/test/java
src/main/resources
src/test/resources
flows
content_based_router.mflow
src
mule-project.xml
MuleEIPs
JRE System Library [JavaSE-1.6]
Mule Runtime [Mule Server 3.2.1
src/main/java
src/test/java
src/main/resources
src/test/resources
Referenced Libraries
Outline
*content_based_router
flows1
DvdOrder.java
VideogameOrder.java
mule-project.xml
flow: Content_Based_RouterFlow1
File
Choice
File
JMS
JMS
Logger
Logger
Filter:
Select
Endpoints
Scopes
Components
Groovy
Java
Javascript
Logger
Python
Transformers
Filters
Flow Control
Cloud Connectors
Message Flow
Global Elements
Configuration XML
Problems
Javadoc
Declaration
Console
No consoles to display at this time.

Pro

- Visual Designer for Mule Flows
- Visual „Live Monitoring“
- Vice versa Editing (Code generation vs. coding by hand)
- Simple Eclipse plugin – „Just Mule“
- Intuitive GUI

Contra

- Proprietary
- Subscription required for many enterprise features (such as monitoring)

MaibornWolff et al

1) Systems Integration

2) Integration Frameworks

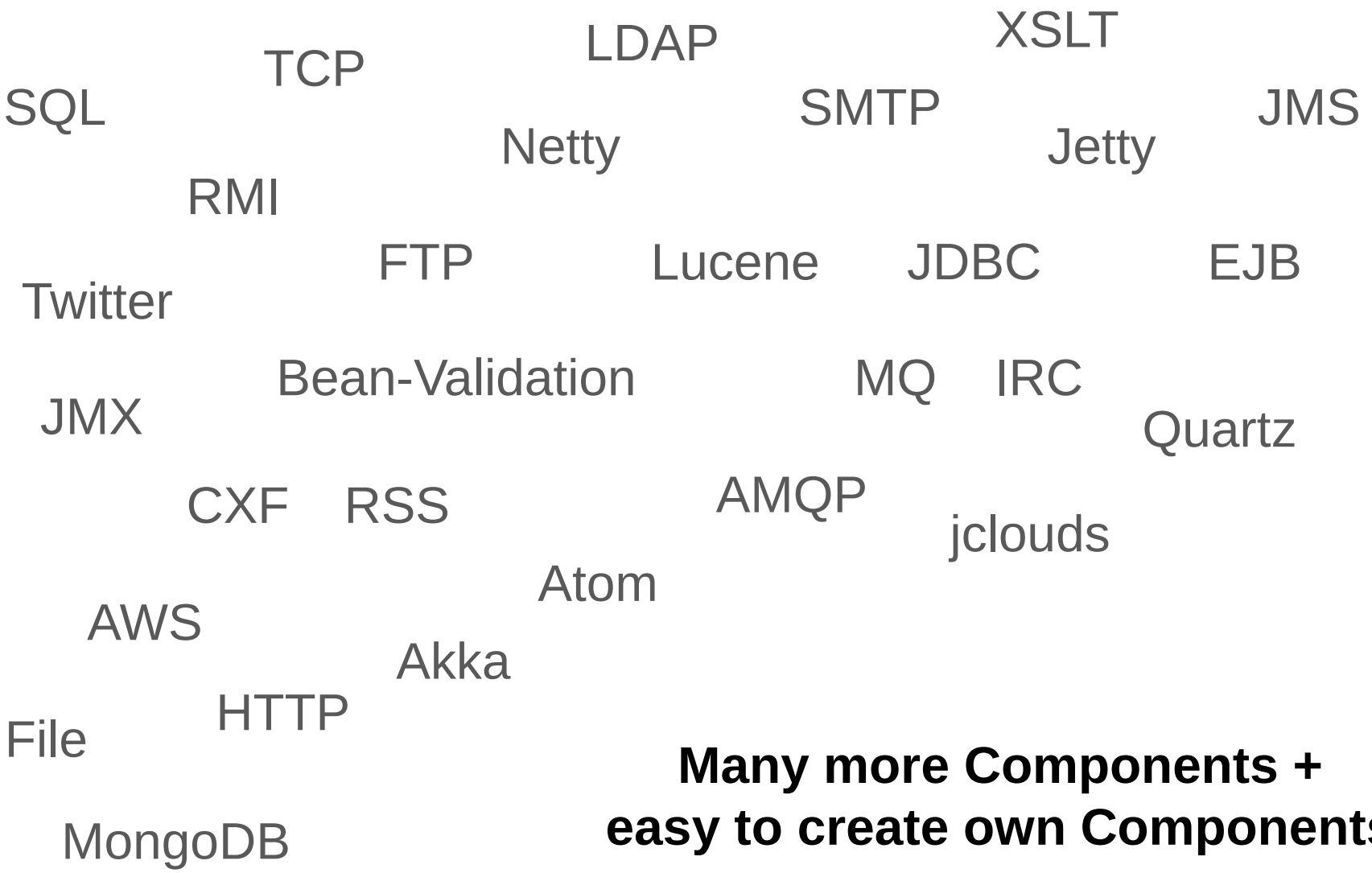
3) Spring Integration

4) Mule ESB

5) Apache Camel

6) And the Winner is ...

XML



Many more Components +
easy to create own Components

```
<route>
    <from uri="file:incomingOrders"/>
    <choice>
      <when>
        <simple>${in.header.type} is 'com.kw.DvdOrder'</simple>
            <to uri="file:incoming/dvdOrders"/>
      </when>
      <when>
        <simple>${in.header.type} is 'com.kw.VideogameOrder'</simple>
            <to uri="jms:videogameOrdersQueue"/>
      </when>
      <otherwise>
        <to uri="log:OtherOrders"/>
      </otherwise>
    </choice>
  </route>
```

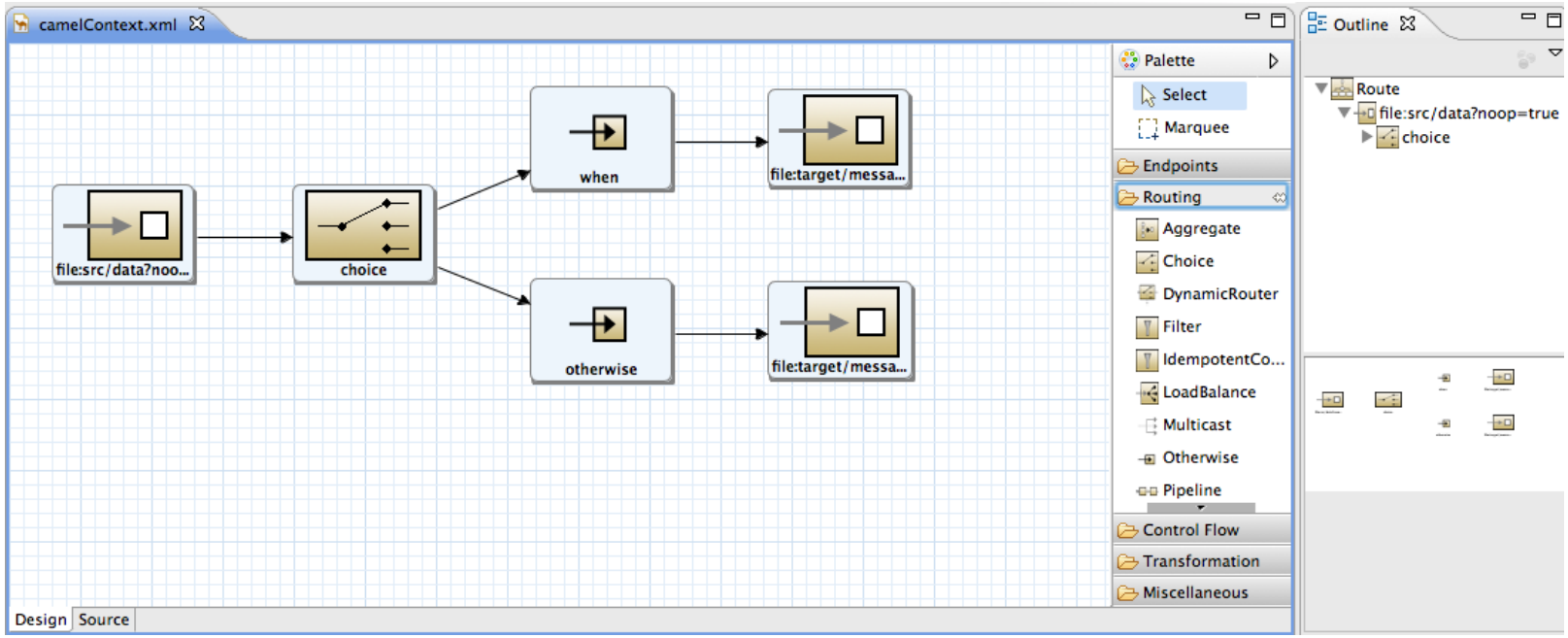
```
from("file:incomingOrders ")
    .choice()
        .when(body().isInstanceOf(com.kw.DvdOrder.class))
            .to("file:incoming/dvdOrders")
        .when(body().isInstanceOf(com.kw.VideogameOrder.class))
            .to("jms:videogameOrdersQueue ")
        .otherwise()
            .to("mock:OtherOrders ");
```

Apache Camel in Action

Fuse IDE

vs.

Talend Open Studio for ESB

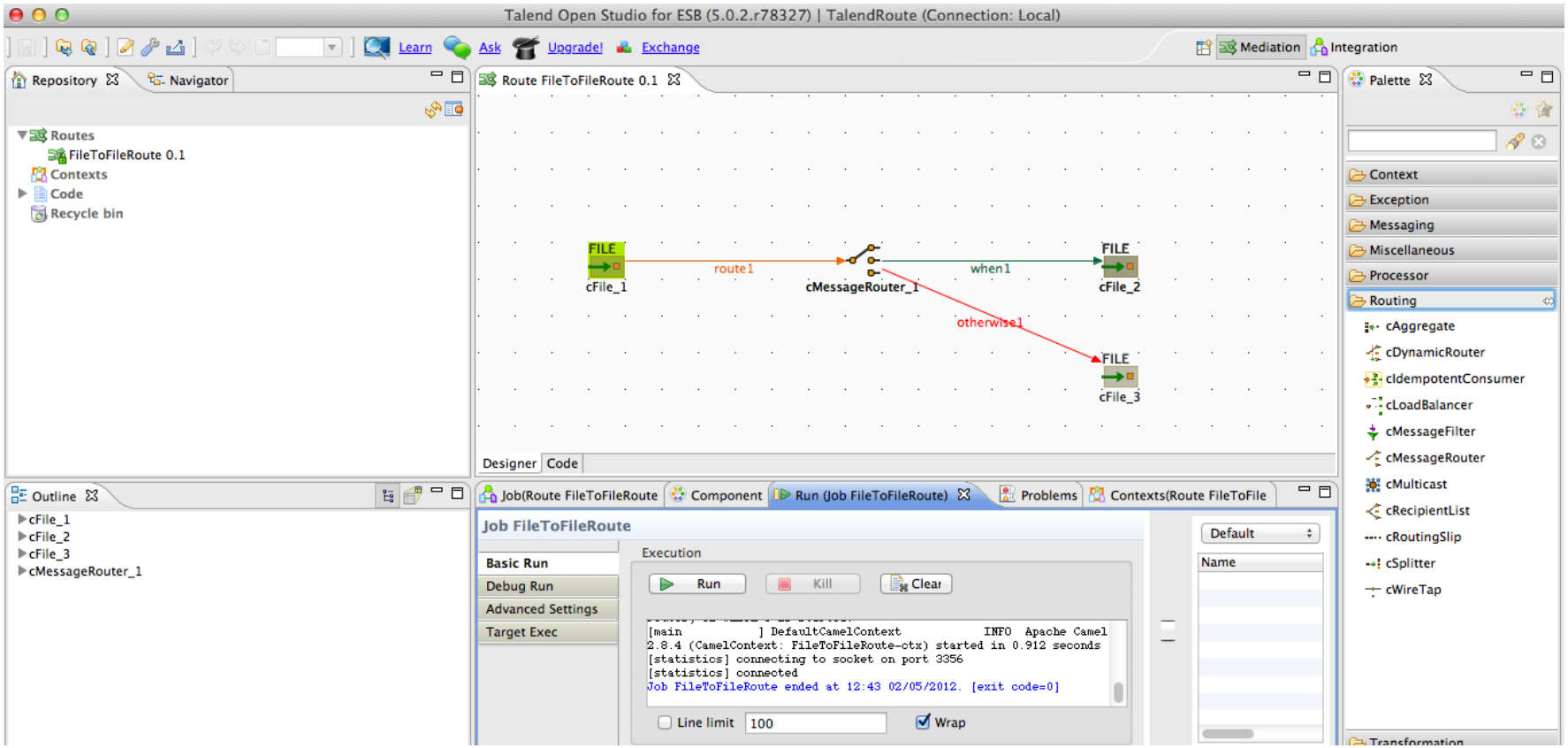

camelContext.xml
file:src/data?noo...
choice
when
otherwise
file:target/messa...
file:target/messa...
Design
Source
Palette
Select
Marquee
Endpoints
Routing
Aggregate
Choice
DynamicRouter
Filter
IdempotentCo...
LoadBalance
Multicast
Otherwise
Pipeline
Control Flow
Transformation
Miscellaneous
Outline
Route
file:src/data?noop=true
choice

Pro

- Visual Designer for Camel routes
- Visual „Live Monitoring“ (for debugging, browsing, tracing)
- Vice versa Editing (code generation vs. coding by-hand)
- Just a simple Eclipse-Plugin – „Just Camel“
- Intuitive GUI
- Export of route diagrams
- JUnit Test Wizard (generates scaffolding for tests)

Contra

- Proprietary
- Subscription required
- Only XML DSL (this is not really a contra, because only Camel offers other DSLs)


Talend Open Studio for ESB (5.0.2.r78327) | TalendRoute (Connection: Local)
Learn
Ask
Upgrade!
Exchange
Mediation
Integration
Repository
Navigator
Routes
FileToFileRoute 0.1
Contexts
Code
Recycle bin
Route FileToFileRoute 0.1
FILE
cFile_1
route1
cMessageRouter_1
when1
FILE
cFile_2
otherwise1
FILE
cFile_3
Designer
Code
Palette
Context
Exception
Messaging
Miscellaneous
Processor
Routing
cAggregate
cDynamicRouter
cIdempotentConsumer
cLoadBalancer
cMessageFilter
cMessageRouter
cMulticast
cRecipientList
cRoutingSlip
cSplitter
cWireTap
Outline
cFile_1
cFile_2
cFile_3
cMessageRouter_1
Job(Route FileToFileRoute
Component
Run (Job FileToFileRoute)
Problems
Contexts(Route FileToFile
Job FileToFileRoute
Basic Run
Debug Run
Advanced Settings
Target Exec
Execution
Run
Kill
Clear
[main] DefaultCamelContext INFO Apache Camel
2.8.4 (CamelContext: FileToFileRoute-ctx) started in 0.912 seconds
[statistics] connecting to socket on port 3356
[statistics] connected
Job FileToFileRoute ended at 12:43 02/05/2012. [exit code=0]
Line limit
100
Wrap
Default
Name

Pro

- Visual Designer for Camel routes
- Visual „Live Monitoring“
- Open Source (at github)
- Community Edition (not all features)
- Zero Coding

Contra

- Zero Coding
- Only Java DSL (plus a lot of boilerplate code)
- No vice versa code editing (only code generation)
- No intuitive user interface

=> no simple Eclipse plugin

=> based on Eclipse, but it is more an own product (1GB)

=> full ESB only

MaibornWolff et al

1) Systems Integration

2) Integration Frameworks

3) Spring Integration

4) Mule ESB

5) Apache Camel

6) And the Winner is ...

spring Integration

Mule ESB

Apache Camel

spring Integration

Mule ESB

Apache Camel

- Spring Project
- „Typical“ JVM Technologies
- No additional Framework

spring Integration

Mule ESB

Apache Camel

Integration

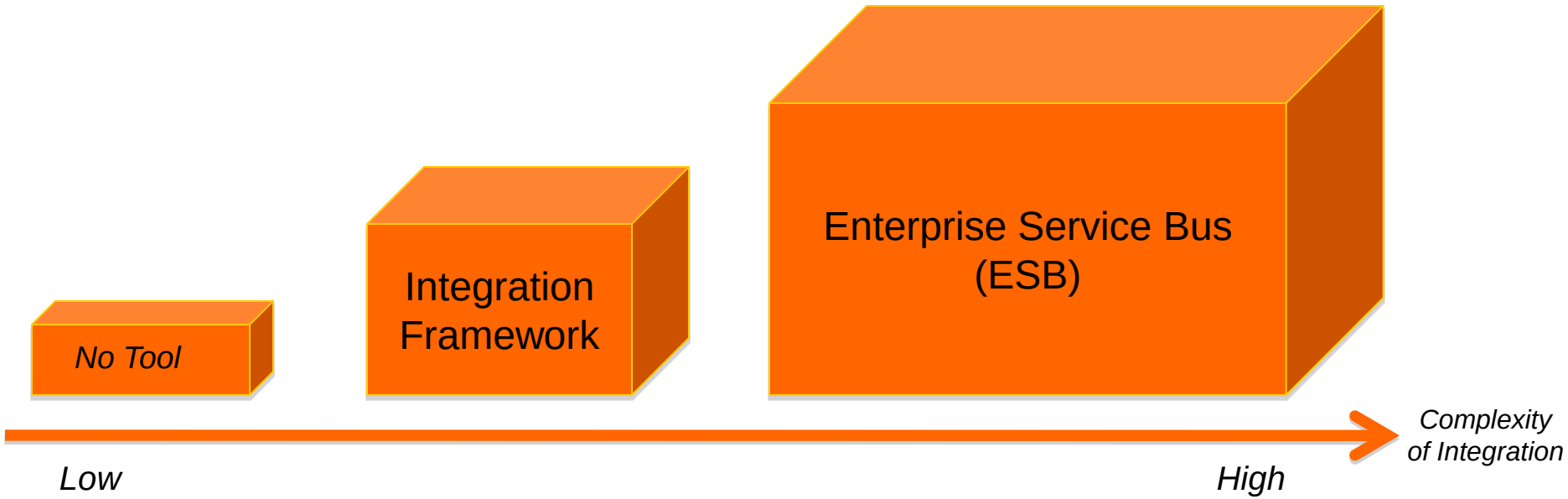
- One of the available proprietary Connectors is required

Mule ESB

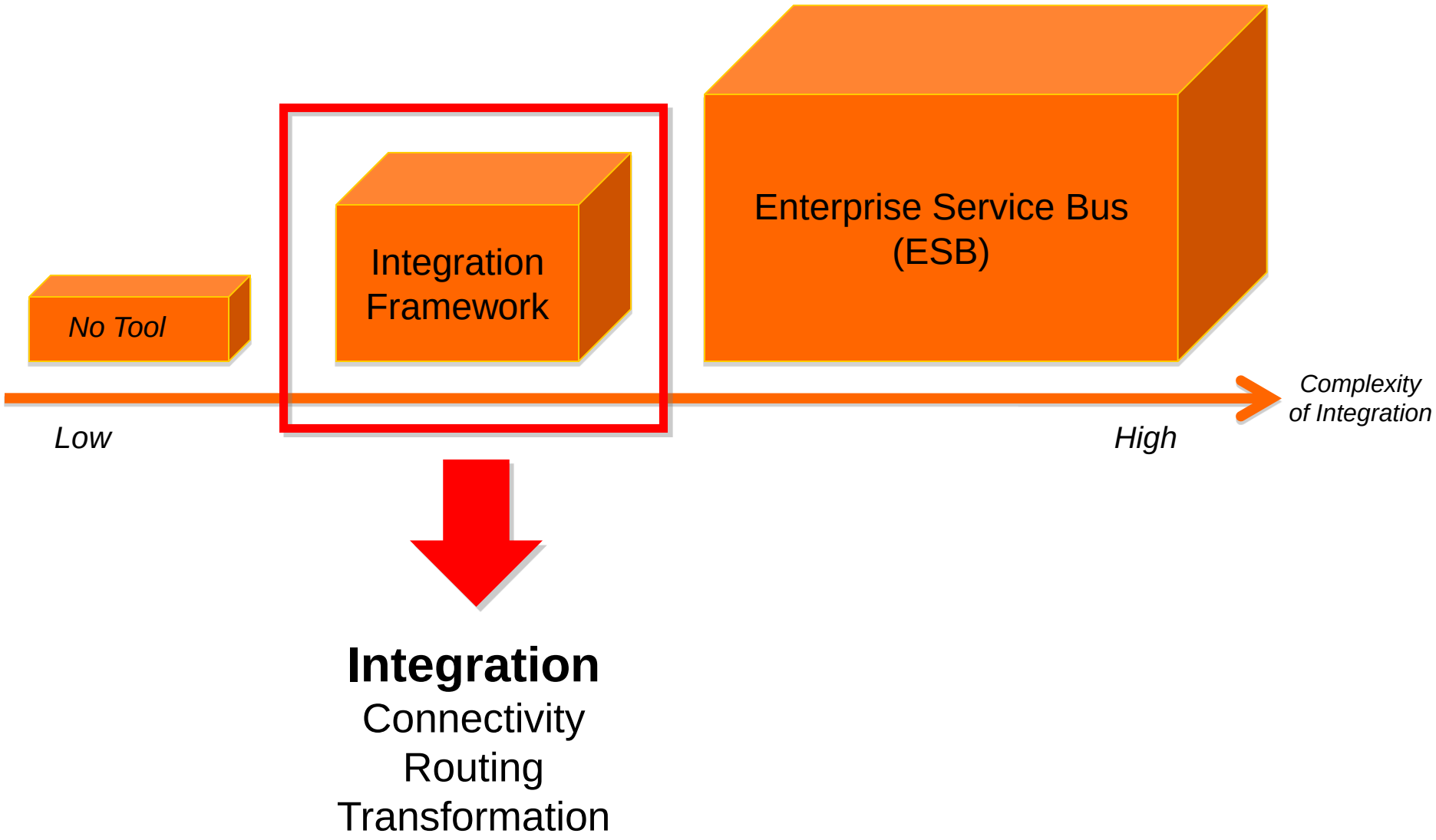
Integration

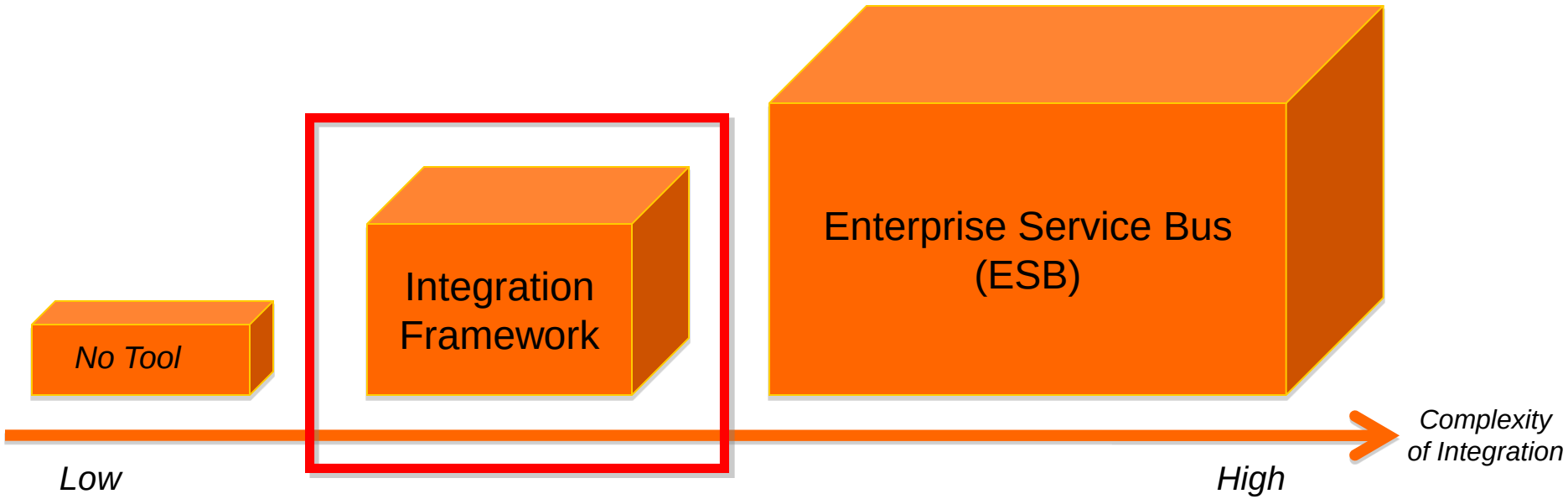
Mule ESB

- In all other Cases


No Tool
Integration Framework
Enterprise Service Bus (ESB)
Low
High
Complexity of Integration

MaibornWolff et al



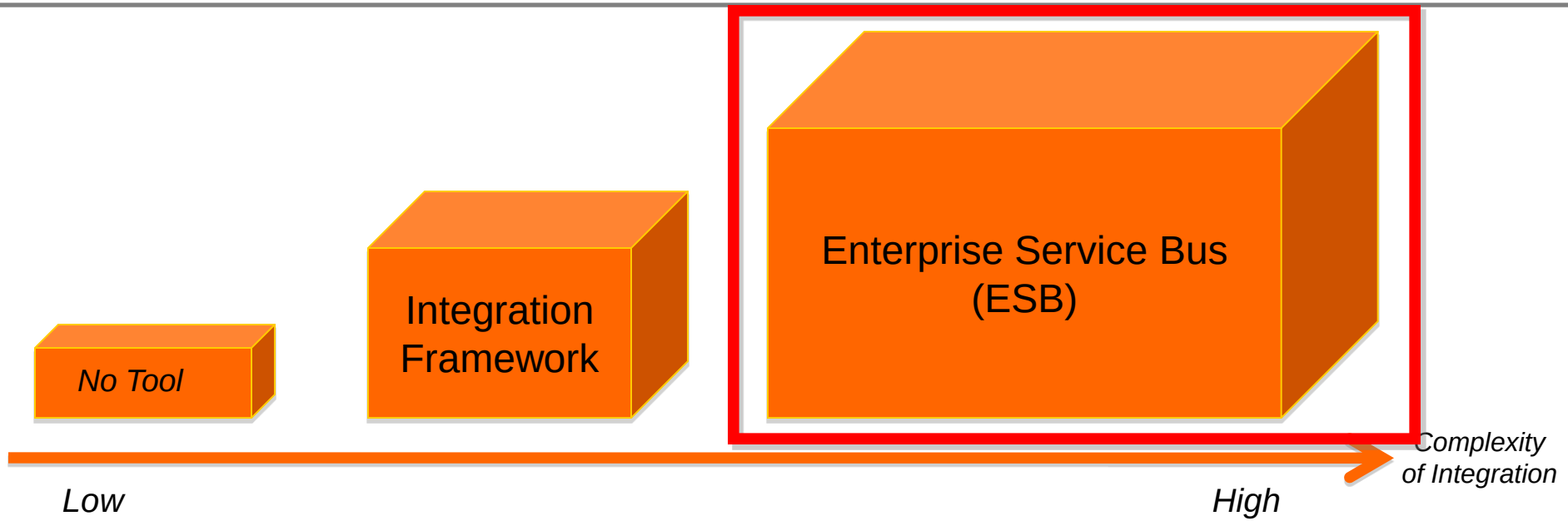


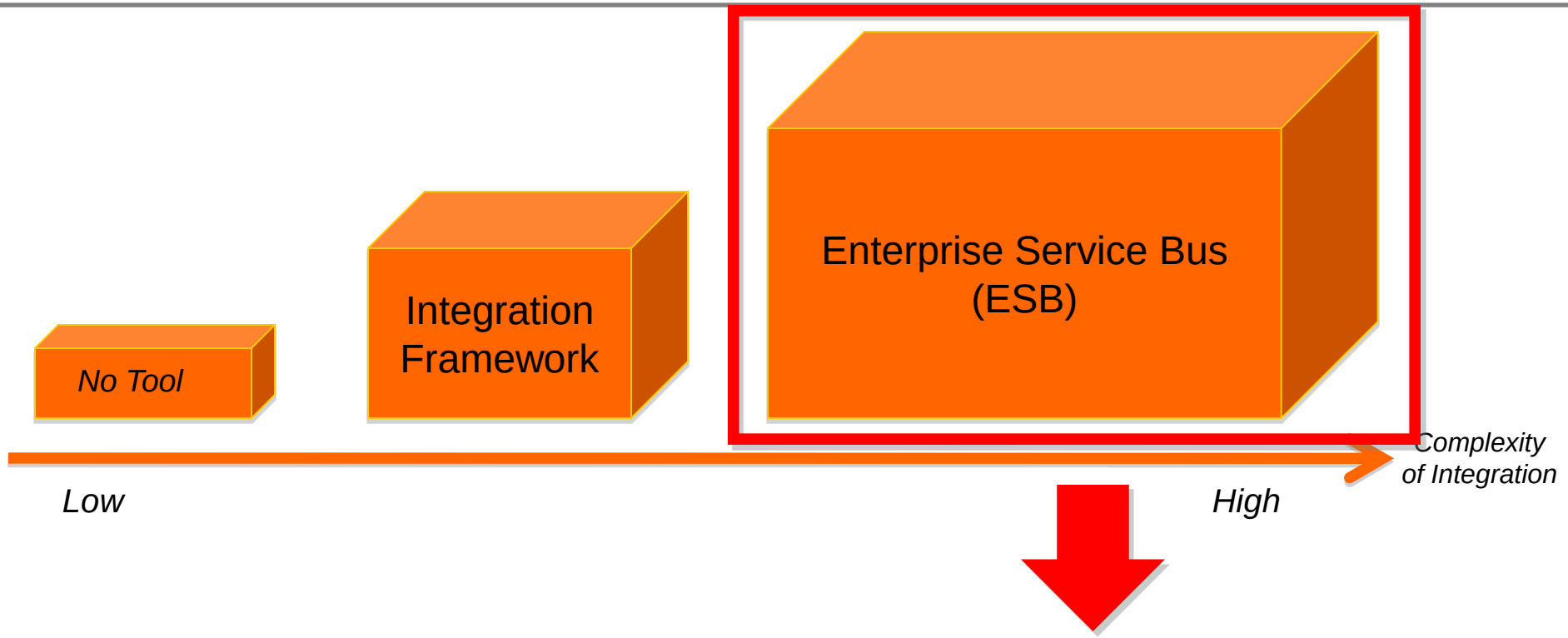
Integration
Connectivity
Routing
Transformation

Spring Integration
vs.
Mule ESB
vs.
Apache Camel

Alternatives for Systems Integration

MaibornWolff et al



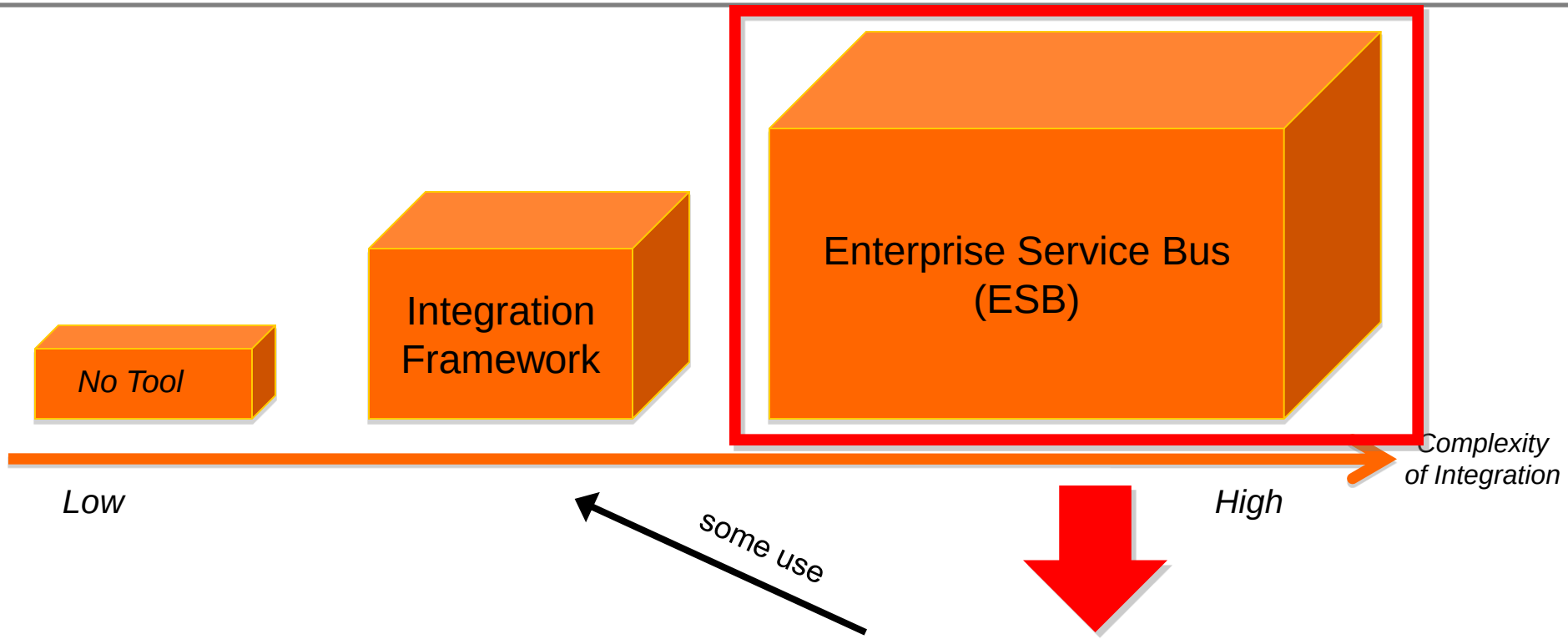


Integration

Connectivity
Routing
Transformation

Middleware Product

Integration
Business Process Management
Business Activity Monitoring
Registry / Repository
Rules Engine

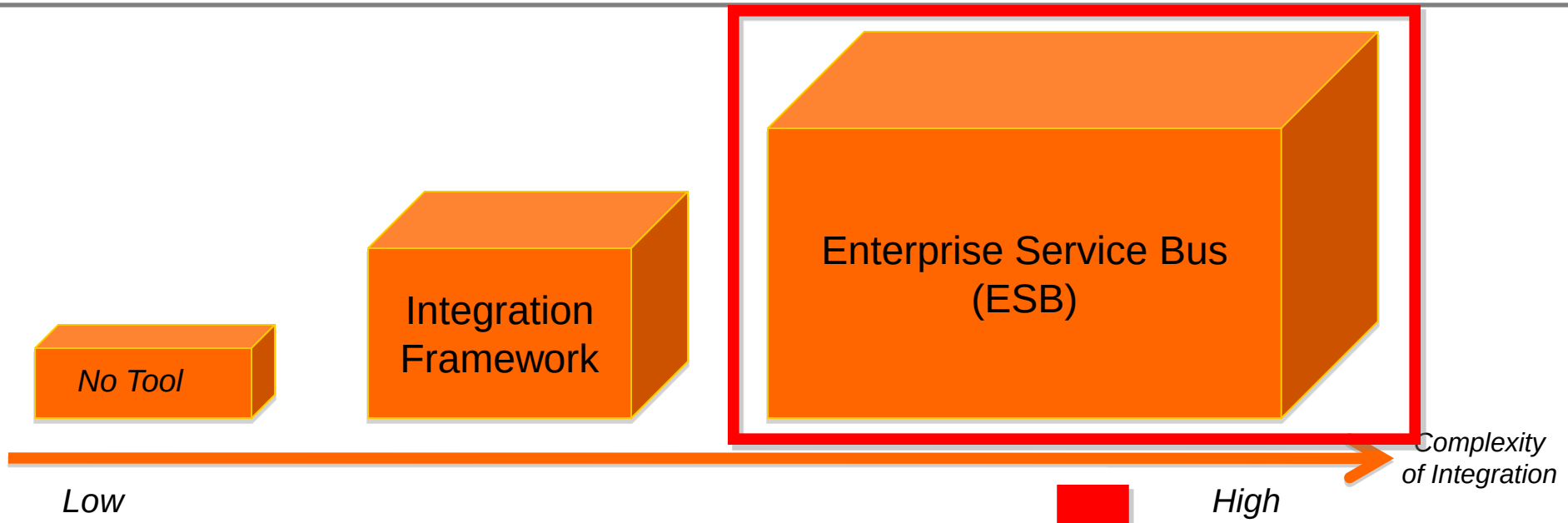


Integration

Connectivity
Routing
Transformation

Middleware Product

Integration
Business Process Management
Business Activity Monitoring
Registry / Repository
Rules Engine

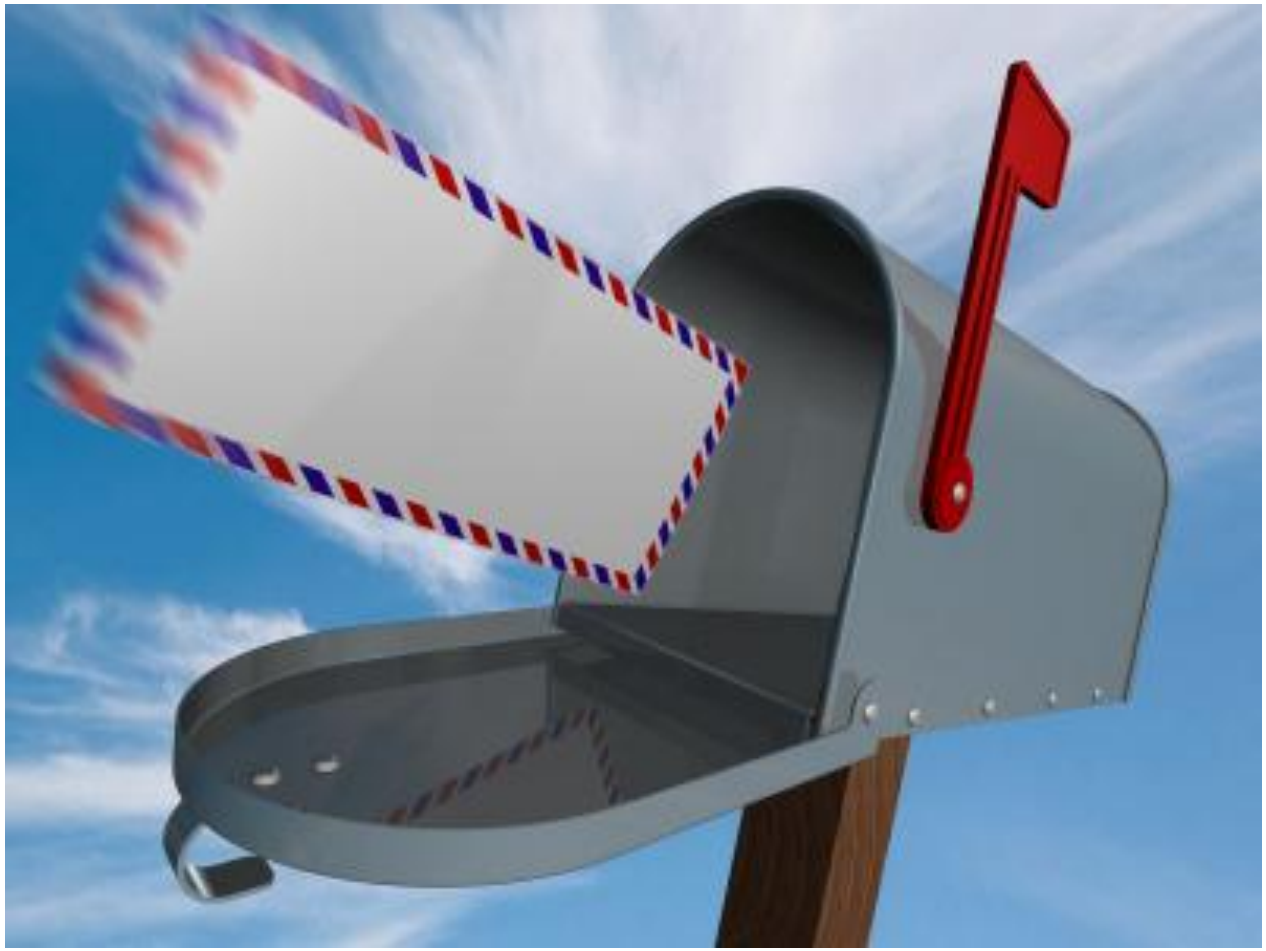


Apache ServiceMix,
Mule ESB, WSO2 ESB
vs.
IBM Message Broker,
IBM ESB, Oracle ESB,
webMethods ESB

Middleware Product

Integration
Business Process Management
Business Activity Monitoring
Registry / Repository
Rules Engine

Did you get the Key Messages?



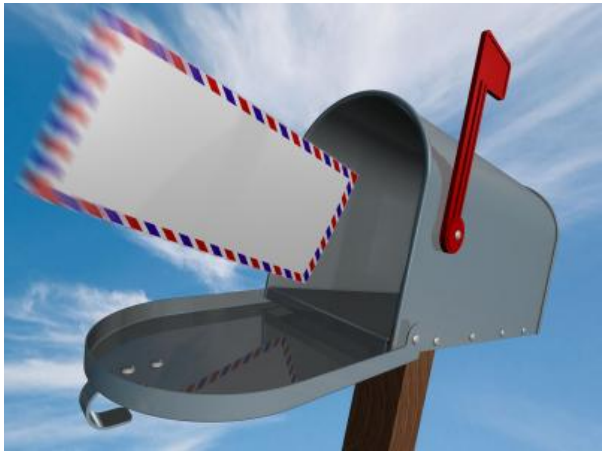
MaibornWolff et al

Do not reinvent the „integration wheel“

There are some good alternatives for Integratic

Often an ESB is the wrong Choic

Did you get the Key Messages?



Become a Part of the Open Source Community

MaibornWolff et al

or **Mule ESB** or 

= **Smart EAI**

Kai Wähner

MaibornWolff et al: www.mwea.de

Email: kai.waehner@mwea.de

Twitter: @KaiWaehner

Blog: www.kai-waehner.de/blog