

Meet the Camel

EAI made easy

Eric Weikl, BTDD 2013-06-14





perl

The End?

Where it goes from here...

- “We need to run the import more often!”
- “We need to change the input format!”
 - “... of *course* we need to support the old version!”
- “We need better validation!”
 - “... using data from system X”
- “We need an online interface!”

Creeping Integration

a.k.a. Bottom-Up EAI

Meet the Camel

*“Apache Camel is a versatile open-source
integration framework based on known
Enterprise Integration Patterns.”*



Apache Camel

= EIP

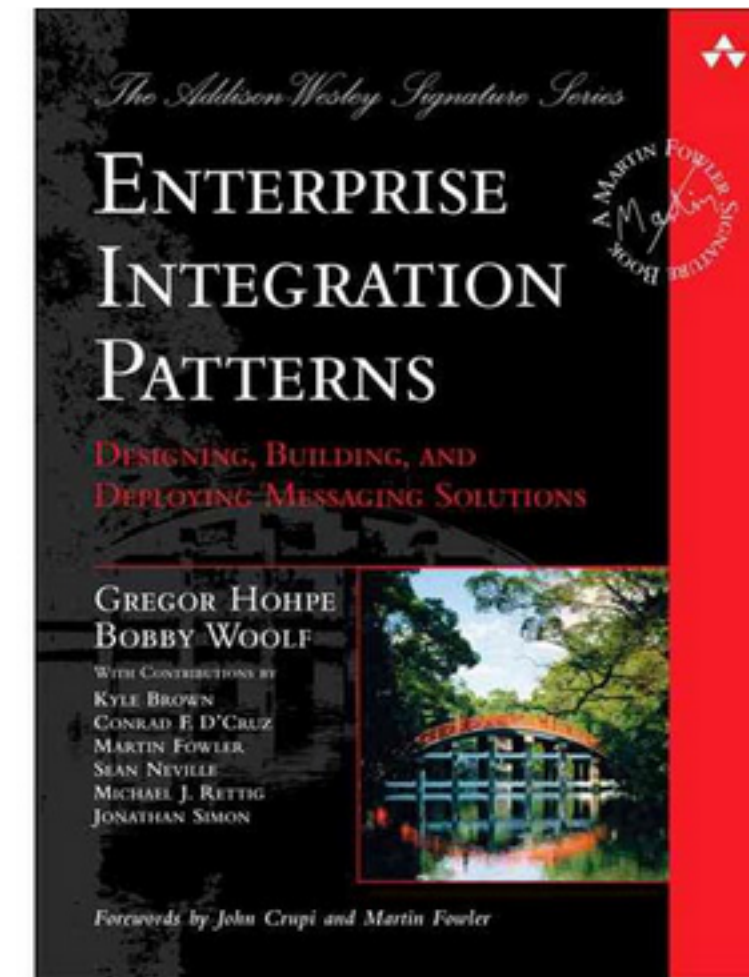
+ API

+ Runtime Environment

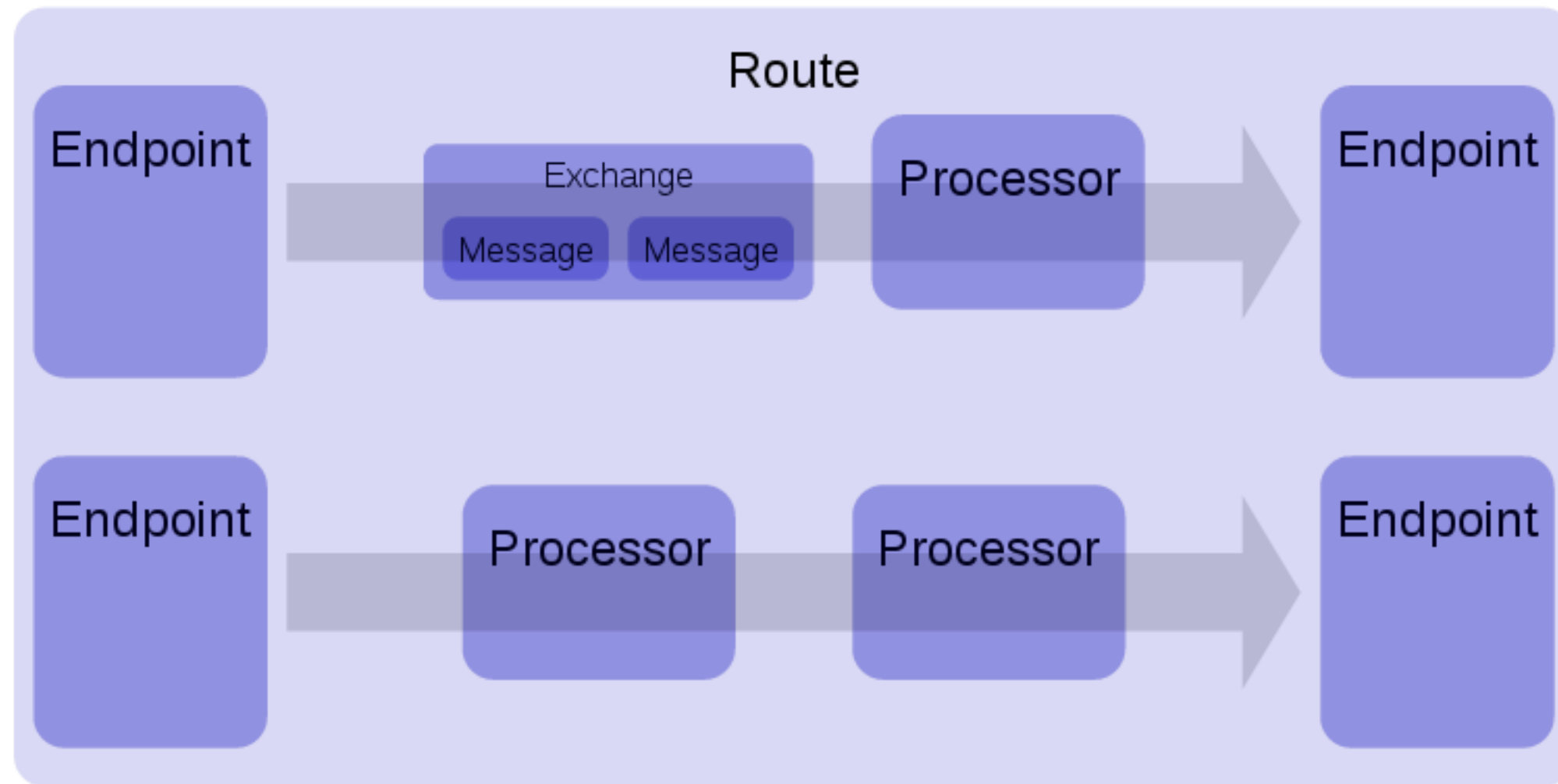
+ Connectors

EIP

- By Gregor Hohpe and Bobby Woolf (2003)
- 65 **patterns** for integrating systems
- Introduced own **vocabulary** and **notation**
- Focus on asynchronous messaging



API



Runtime Environment

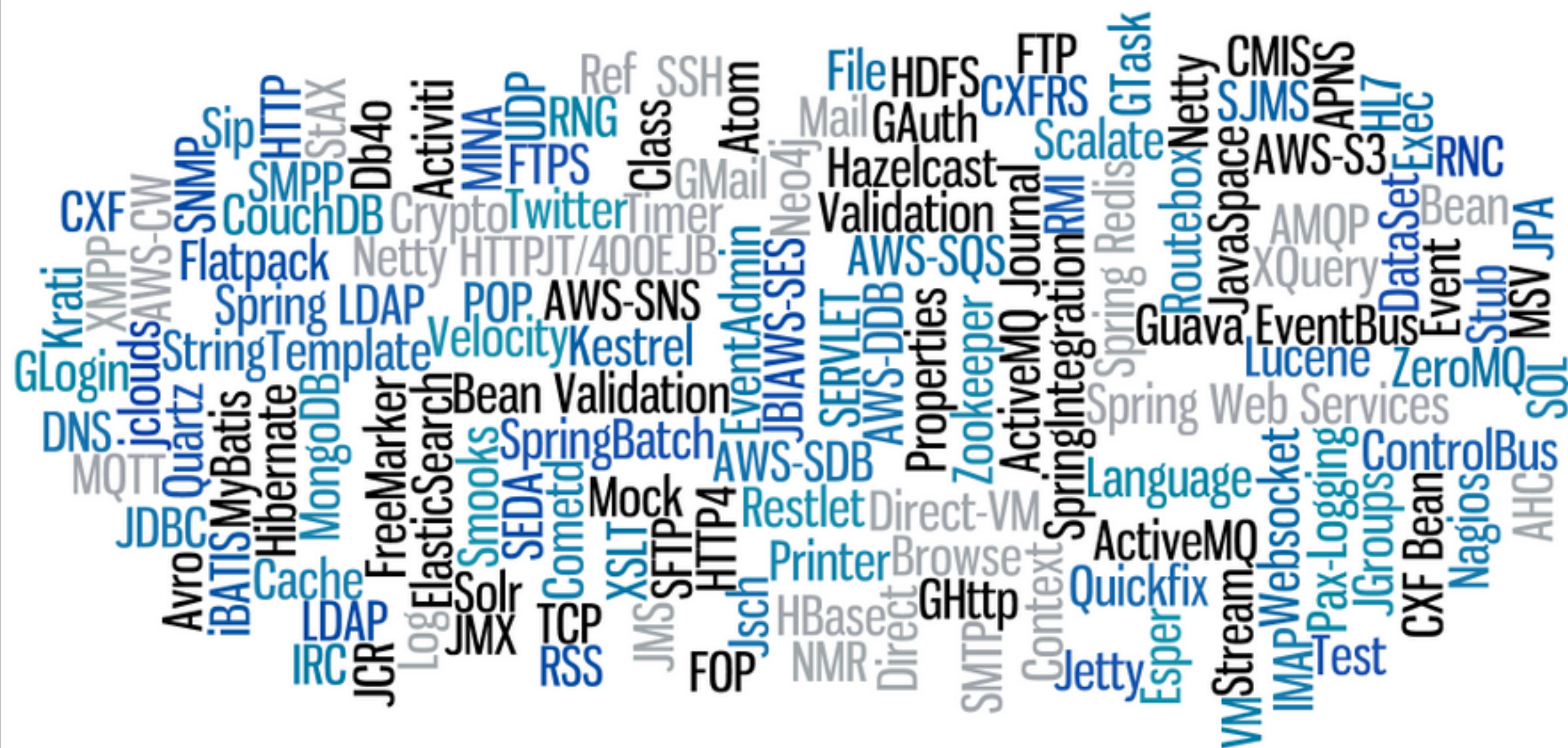
Multi-threading and asynchrony



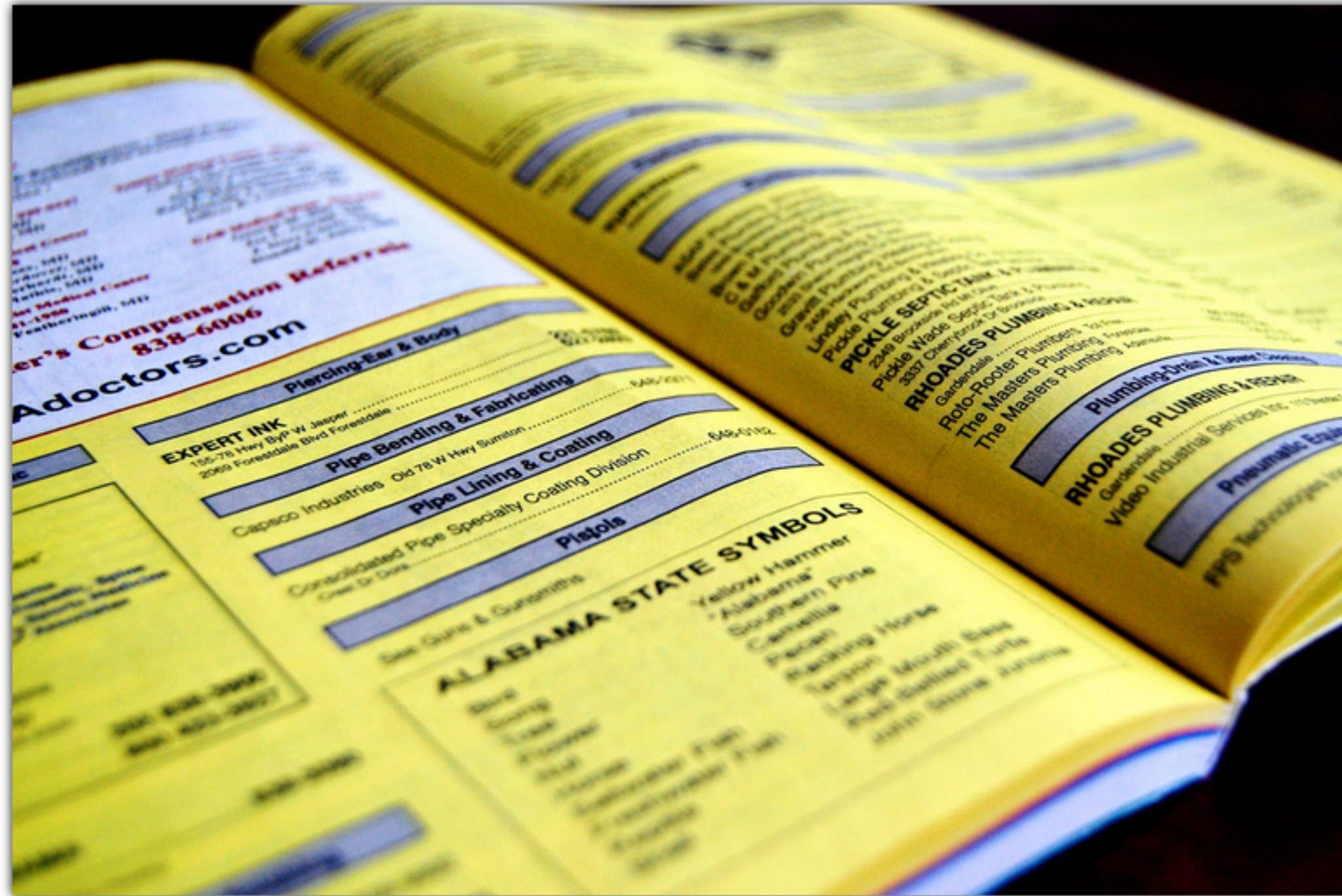
Type Conversion



Connectors



Component Registry



Spring Integration



Coding Camel

Route

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

public class EchoRoute extends RouteBuilder {

    @Override
    public void configure() throws Exception {
        from("stream:in")
            .to("stream:out");
    }
}
```

Combining Routes

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

public class HelloRoute extends RouteBuilder {

    @Override
    public void configure() throws Exception {
        from("stream:in").routeId("userInput")
            .to("direct:output");

        from("direct:output").routeId("consoleOutput")
            .to("stream:out");
    }
}
```


Transform

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

public class HelloRoute extends RouteBuilder {

    @Override
    public void configure() throws Exception {
        from("stream:in")
            .transform().simple("Hello ${body}")
            .to("stream:out");
    }
}
```

Processor

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

public class HelloRoute extends RouteBuilder {

    @Override
    public void configure() throws Exception {
        from("stream:in")
            .process(new HelloTransformer())
            .to("stream:out");
    }
}
```

Processor Interface

```
public class HelloTransformer implements Processor {  
  
    @Override  
    public void process(Exchange exchange) throws Exception {  
        String message = exchange.getIn().getBody(String.class);  
        exchange.getIn().setBody("Hello " + message);  
    }  
  
}
```

Bean

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

public class HelloRoute extends RouteBuilder {

    @Override
    public void configure() throws Exception {
        from("stream:in")
            .bean(new HelloTransformer())
            .to("stream:out");
    }
}
```


Bean Processor

```
public class HelloTransformer {  
    @Handler  
    public String prependHello(@Body String message) throws Exception {  
        return "Hello " + message;  
    }  
}
```

BeanRef

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

public class HelloRoute extends RouteBuilder {

    @Override
    public void configure() throws Exception {
        from("stream:in")
            .beanRef("helloTransformer", "transform")
            .to("stream:out");
    }
}
```

Example

File Batch Import

Step 1: Copy Files

File Endpoint

- scans directory for incoming files
- creates message from file content
- archives processed files
- stores failed files for inspection

File Input

```
public class CopyRoute extends RouteBuilder {

    @Override
    public void configure() throws Exception {
        from(FILE_IN_ENDPOINT)
            .to(LOG_ENDPOINT)
            .to(FILE_OUT_ENDPOINT);
    }

    static final String LOG_ENDPOINT =
        "log:file?showAll=true&multiline=true";

    static final String FILE_IN_ENDPOINT =
        "file:///tmp/ex3/input?"
        + "autoCreate=true"
        + "&preMove=../in-progress"
        + "&move=../done"
        + "&moveFailed=../error";

    static final String FILE_OUT_ENDPOINT =
        "file:///tmp/ex3/output?autoCreate=true";

}
```

Step 2: Parse CSV

CSV Data

ISBN	Author / Editor	Title	Price
978-1-4614-3560-0	de Almeida	Advances in Yersinia Research	\$259.00
978-94-007-3857-7	Le Novère	Computational Systems Neurobiology	\$189.00
978-1-4419-0460-7	Choi	Encyclopedia of Signaling Molecules	\$1,100.00
978-3-642-30337-1	Hermann	Gasotransmitters: Physiology and Pathophysiology	\$189.00
978-1-4614-4498-5	Humphries	Hereditary Retinopathies	\$49.95
978-3-0348-0444-8	Stein-Streilein	Infection, Immune Homeostasis and Immune Privilege	\$189.00
...	...	...	...

CSV Reader

```
public class CsvImportRoute extends RouteBuilder {  
  
    @Override  
    public void configure() throws Exception {  
        from(FILE_IN_ENDPOINT)  
            .setProperty(Exchange.CHARSET_NAME, simple("ISO-8859-1"))  
            .to("direct:unmarshal");  
  
        from("direct:unmarshal")  
            .unmarshal().bindy(BindyType.Csv, BookRecord.class)  
            .split().body()  
            .log("${body}");  
    }  
}
```

CSV Binding

```
@CsvRecord(separator = ",", skipFirstLine = true)
public class BookRecord {
    @DataField(pos = 1, required = true)
    private String isbn;
    @DataField(pos = 2, required = true)
    private String author;
    @DataField(pos = 3, required = true)
    private String title;
    @DataField(pos = 4, required = true)
    private String price;

    // ... getters omitted ...
}
```

“But our files are huge!”

CSV Reader

```
public class HugeCsvImportRoute extends RouteBuilder {  
  
    @Override  
    public void configure() throws Exception {  
        from(FILE_IN_ENDPOINT)  
            .setProperty(Exchange.CHARSET_NAME, simple("ISO-8859-1"))  
            .split(body().tokenize("\n")).streaming()  
            .filter().simple("${property.CamelSplitIndex} > 0")  
            .to("direct:unmarshal");  
  
        from("direct:unmarshal")  
            .unmarshal().bindy(BindyType.Csv, BookRecord.class)  
            .log("${body}");  
    }  
}
```

Step 3: Insert Records

Inserting Data

```
insert into books (  
  isbn, title, author, currency, amount  
) values (  
  ?, ?, ?, ?, ?  
)
```

SQL Insert

```
public class CsvImportRoute extends RouteBuilder {
    @Override
    public void configure() throws Exception {
        from("direct:unmarshal")
            // ...
            .to("direct:insert");

        from("direct:insert")
            .setHeader("isbn", simple("${body.isbn}"))
            .setHeader("title", simple("${body.title}"))
            .setHeader("author", simple("${body.author}"))
            .setHeader("currency", simple("${body.currencyCode}"))
            .setHeader("amount", simple("${body.amount}"))
            .to("sql:insert into books ("
                + "    isbn, title, author, currency, amount"
                + ") values ("
                + "    :#isbn,"
                + "    :#title,"
                + "    :#author,"
                + "    :#currency,"
                + "    :#amount"
                + ")?dataSource=dataSourceFromRegistry");
    }
}
```

DAO Insert

```
public class CsvImportRoute extends RouteBuilder {  
    @Override  
    public void configure() throws Exception {  
        from("direct:unmarshal")  
            // ...  
            .to("direct:insert");  
  
        from("direct:insert")  
            .beanRef("bookDao", "save");  
    }  
}
```

Bookstore 2.0

Supporting multiple file formats

Features

1. new file format with prices in €
2. old clients get their prices recalculated

CSV Format v2

ISBN	Author / Editor	Title	Price	Retail Price (EUR)
978-1-4614-3560-0	de Almeida	Advances in Yersinia Research	\$259.00	197.99
978-94-007-3857-7	Le Novère	Computational Systems Neurobiology	\$189.00	144.99
978-1-4419-0460-7	Choi	Encyclopedia of Signaling Molecules	\$1,100.00	839.99
978-3-642-30337-1	Hermann	Gasotransmitters: Physiology and Pathophysiology	\$189.00	144.99
978-1-4614-4498-5	Humphries	Hereditary Retinopathies	\$49.95	38.99
978-3-0348-0444-8	Stein-Streilein	Infection, Immune Homeostasis and Immune Privilege	\$189.00	144.99
...	...	...	...	...

Route v2

```
public class MultiVersionRoute extends RouteBuilder {
    @Override
    public void configure() throws Exception {
        from(FILE_IN_ENDPOINT)
            .setProperty(Exchange.CHARSET_NAME, simple("ISO-8859-1"))
            .split().body()
            .choice()
                .when(simple("${file:name} contains 'v2'"))
                    .to("direct:unmarshalv2")
                .otherwise()
                    .to("direct:unmarshalv1")
            .endChoice();

        from("direct:unmarshalv2")
            .unmarshal().bindy(BindyType.Csv, BookRecordV2.class)
            .to("direct:insert");

        from("direct:unmarshalv1")
            .unmarshal().bindy(BindyType.Csv, BookRecordV1.class)
            .beanRef("bookTransformer", "convertV1ToV2")
            .to("direct:insert");

        from("direct:insert")
            // ...
    }
}
```

Book Transformer

```
public class BookTransformer {  
    public BookRecordV2 convertV1ToV2(BookRecordV1 book) throws Exception {  
        BookRecordV2 mapped = new BookRecordV2(book);  
        mapped.setRetailPrice(calculateRetailPrice(book));  
        return mapped;  
    }  
  
    private Money calculateRetailPrice(BigDecimal dollarAmount) {  
        // ... call some service ...  
        return Money.of(CurrencyUnit.EUR, amount);  
    }  
}
```

That's Sloooooow

Can we make it faster?

SEDA

Staged Event-Driven Architecture

(in-memory queues with workers)

Parallel Processing

```
public class ParallelMultiVersionRoute extends RouteBuilder {
    @Override
    public void configure() throws Exception {
        from(FILE_IN_ENDPOINT)
            .setProperty(Exchange.CHARSET_NAME, simple("ISO-8859-1"))
            .split().body()
            .choice()
                .when(simple("${file:name} contains 'v2'"))
                    .to("seda:unmarshalv2")
                .otherwise()
                    .to("seda:unmarshalv1")
            .endChoice();

        from("seda:unmarshalv2?concurrentConsumers=2")
            .unmarshal().bindy(BindyType.Csv, BookRecordV2.class)
            .to("direct:insert");

        from("seda:unmarshalv1?concurrentConsumers=5")
            .unmarshal().bindy(BindyType.Csv, BookRecordV1.class)
            .beanRef("bookTransformer", "convertV1ToV2")
            .to("direct:insert");

        from("direct:insert")
            // ...
    }
}
```


Testing

That's a lot of logic

```
public class ParallelStreamingMultiVersionRoute extends RouteBuilder {
    @Override
    public void configure() throws Exception {
        // ... file routes omitted ...

        from("direct:incoming")
            .setProperty(Exchange.CHARSET_NAME, simple("ISO-8859-1"))
            .split(body().tokenize("\n")).streaming()
            .filter().simple("${property.CamelSplitIndex} > 0")
            .choice()
                .when(simple("${file:name} contains 'v2'"))
                    .to("seda:unmarshalv2")
                .otherwise()
                    .to("seda:unmarshalv1")
            .endChoice();

        from("seda:unmarshalv1").id("unmarshalv1")
        // ...
    }
}
```

Let's write a test

```
public class RoutingLogicTest {  
    @Test  
    public void shouldSkipFirstLineInFile() throws Exception {  
        System.out.println("um... help?");  
    }  
}
```

Camel Test Support

JUnit Base Class

Mock Endpoints

Assertions

AOP Framework

Simple Dependency Injection

Using Mock Endpoints

```
public class ParallelStreamingMultiVersionRoute extends RouteBuilder {
    @Override
    public void configure() throws Exception {
        from("direct:incoming").id("csvRouting")
            .setProperty(Exchange.CHARSET_NAME, simple("ISO-8859-1"))
            .split(body().tokenize("\n")).streaming()
            .filter().simple("${property.CamelSplitIndex} > 0")
            .choice()
                .when(simple("${file:name} contains 'v2'"))
                    .to("mock:seda:unmarshalv2")
                .otherwise()
                    .to("mock:seda:unmarshalv1")
            .endChoice();
    }
}
```

Let's try that again

```
public class RoutingLogicTest extends CamelTestSupport {
    @EndpointInject(uri = "direct:incoming")
    ProducerTemplate incoming;

    @Test
    public void shouldSkipFirstLineInFile() throws Exception {
        getMockEndpoint("mock:seda:unmarshalv1").expectedMessageCount(0);
        getMockEndpoint("mock:seda:unmarshalv2").expectedMessageCount(0);

        incoming.sendBody(
            "ISBN,Author / Editor,Title,MediumType\n"
        );

        assertMockEndpointsSatisfied();
    }

    @Override
    protected RouteBuilder createRouteBuilder() throws Exception {
        return new ParallelStreamingMultiVersionRoute();
    }
}
```

Replacing Endpoints With AOP

```
public class RoutingLogicTest extends CamelTestSupport {
    // ... as before ...

    @Before
    public void mockEndpoints() throws Exception {
        context.getRouteDefinition("csvRouting").adviceWith(context,
            new AdviceWithRouteBuilder() {
                @Override
                public void configure() throws Exception {
                    mockEndpointsAndSkip("seda:unmarshalv1", "seda:unmarshalv2");
                }
            }
        );
        context.start();
    }

    public boolean isUseAdviceWith() {
        return true;
    }
}
```


Getting Started

Groovy

```
@Grab(group="org.apache.camel", module="camel-core", version="2.11.0")
@Grab(group="org.apache.camel", module="camel-stream", version="2.11.0")
import org.apache.camel.builder.RouteBuilder
import org.apache.camel.impl.DefaultCamelContext
import org.apache.camel.main.Main

class EchoRoute extends RouteBuilder {
    void configure() {
        from('stream:in')
            .to('stream:out')
    }
}

def main = new Main()
main.enableHangupSupport()
main.addRouteBuilder(new EchoRoute())
main.run()
```


Java & Spring

```
@Configuration
public class MyApplicationContext extends SingleRouteCamelConfiguration {

    @Bean
    public Processor greeter() {
        return new Processor() {
            @Override
            public void process(Exchange exchange) throws Exception {
                String input = exchange.getIn().getBody(String.class);
                exchange.getIn().setBody("Hello " + input);
            }
        };
    }

    @Override
    public RouteBuilder route() {
        return new RouteBuilder() {
            @Override public void configure() throws Exception {
                from("stream:in")
                    .beanRef("greeter")
                    .to("stream:out");
            }
        };
    }
}
```

```
public class MyCmdLineRunner {  
    public static void main(String[] args) throws Exception {  
        Main main = new Main();  
        main.setConfigClassesString(CamelWithSpring.class.getName());  
        main.run(args);  
    }  
}
```

Plain Web Application

web.xml

```
<web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
    http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">

  <context-param>
    <param-name>routeBuilder-MyRoute</param-name>
    <param-value>com.tngtech.camel.MyRoute</param-value>
  </context-param>

  <listener>
    <listener-class>
      o.a.camel.component.servletlistener.JndiCamelServletContextListener
    </listener-class>
  </listener>

  ...

</web-app>
```

OSGi

Enterprise Service Bus

- Apache ServiceMix
- Fuse ESB
- OpenESB

Conclusion

Summary

- Starting with Camel is easy
 - Embeddable in existing application
 - Several deployment options
 - Modular: only use parts you need
 - With relatively few dependencies
 - No extra infrastructure required
- Easier than integrating by hand
- Components for *everything*
- Extensible
- Medium learning curve
- Very good testing support

Thank You!

Questions?

eric.weikl@tngtech.com

[https://www.xing.com/profile/Eric_Weikl/
@ericweikl](https://www.xing.com/profile/Eric_Weikl/@ericweikl)

TNG  **TECHNOLOGY
CONSULTING**

Images

- Cup of Camel - <http://www.flickr.com/photos/yukop/7069769559>
- Pneumatic Central - <http://www.flickr.com/photos/dlanod/235990854/>
- Currency Conversion - <http://www.flickr.com/photos/epsos/8453024068/>
- Yellow Pages - <http://www.flickr.com/photos/jamiesrabbits/4248396588/>
- Spring - <http://www.flickr.com/photos/vietnamfriendly/2408955764>