3.– 6. September 2012
in Nürnberg

Herbstcampus

Wissenstransfer
par excellence

Gemeinsam sind wir stark

BPMN und Camel

Dr. Daniel Lübke & Martin Huber

innoQ Deutschland GmbH

Gemeinsam sind wir stark

BPMN und Apache Camel

Gemeinsam sind wir stark

Daniel Lübke & Martin Huber,
innoQ Deutschland GmbH

http://www.innoq.com

Beratungsunternehmen mit Schwerpunkt auf Softwarearchitektur & effizienter Entwicklung

Gründungsmitglied iSAQB e.V.

~60 Mitarbeiter (Q1 2012)

Beratung, Schulung, Entwicklung

Java/Java EE, Ruby on Rails u.a

Standorte in Düsseldorf, Frankfurt, München, Zürich

Agenda

- Geschäftsprozesse mit BPMN + BPEL
- Probleme
- Architektur mit BPMN und Apache Camel
- Apache Camel auf einen Blick
- Pufferschicht mit Camel & Beispiel

Agenda

- Geschäftsprozesse mit BPMN + BPEL
- Probleme
- Architektur mit BPMN und Apache Camel
- Apache Camel auf einen Blick
- Pufferschicht mit Camel & Beispiel

Agenda

- Geschäftsprozesse mit BPMN + BPEL
- Probleme
- Architektur mit BPMN und Apache Camel
- Apache Camel auf einen Blick
- Pufferschicht mit Camel & Beispiel

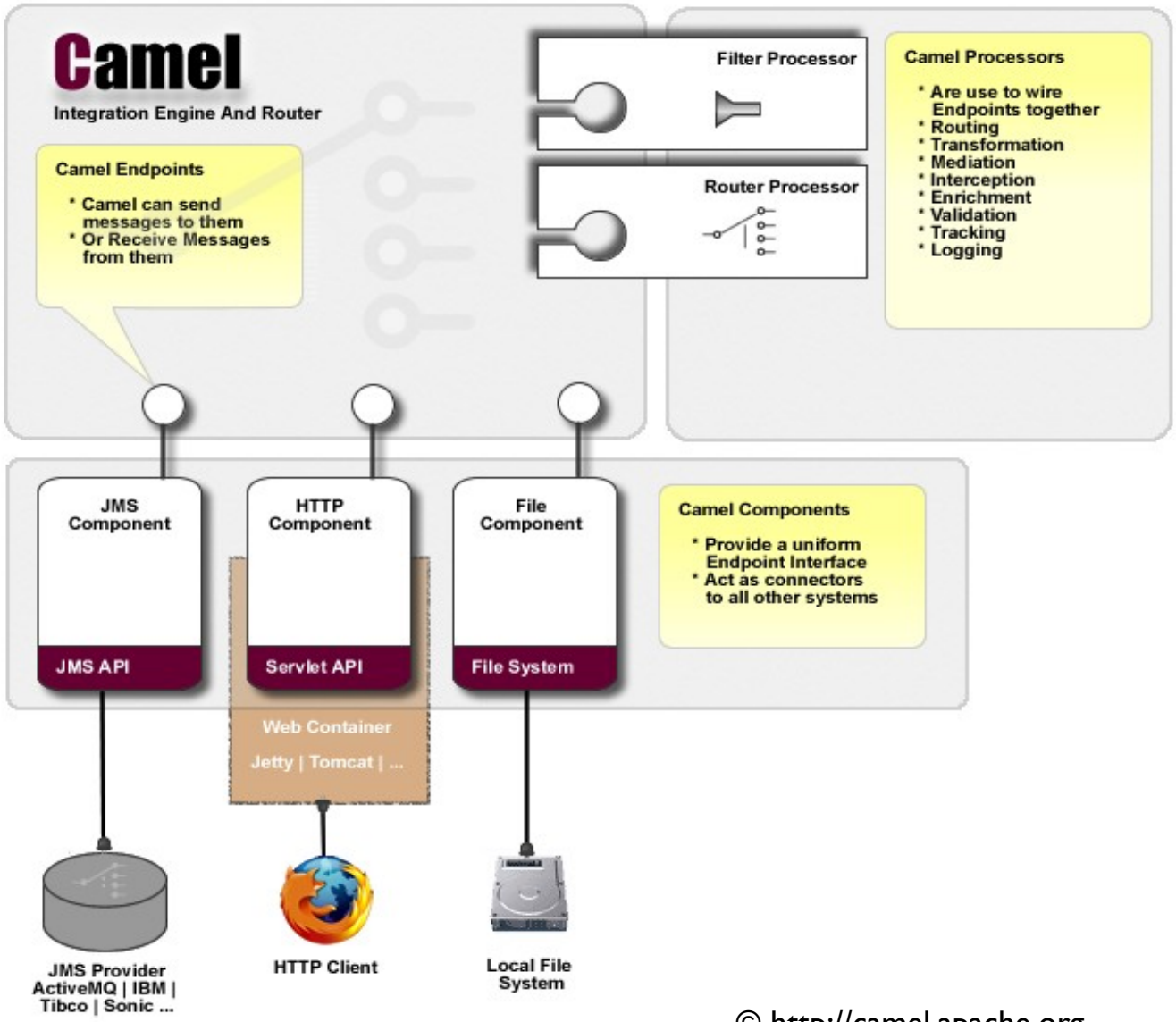
Agenda

- Geschäftsprozesse mit BPMN + BPEL
- Probleme
- Architektur mit BPMN und Apache Camel
- Apache Camel auf einen Blick
- Pufferschicht mit Camel & Beispiel

Agenda

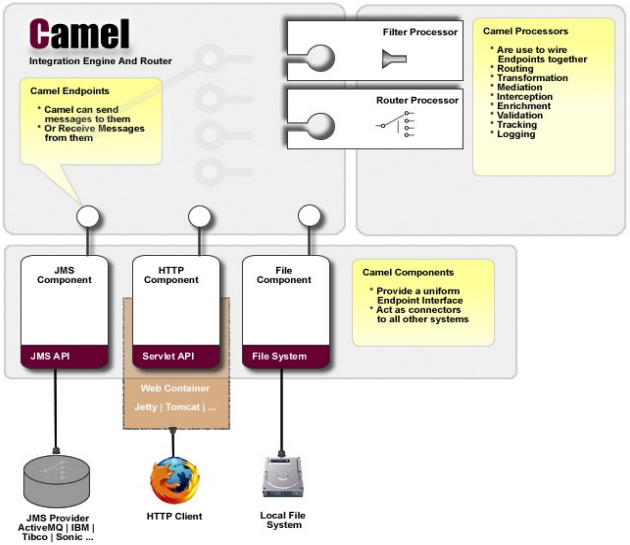
- Geschäftsprozesse mit BPMN + BPEL
- Probleme
- Architektur mit BPMN und Apache Camel
- Apache Camel auf einen Blick
- Pufferschicht mit Camel & Beispiel

Apache Camel auf einen Blick



© http://camel.apache.org

Apache Camel - Konzepte



- Messages und Exchanges
- Processors
- Endpoints als Producers & Consumers
- Routes
- DSL
- TypeConverters
- CamelContext

© http://camel.apache.org

Konzepte: Routing

© http://camel.apache.org

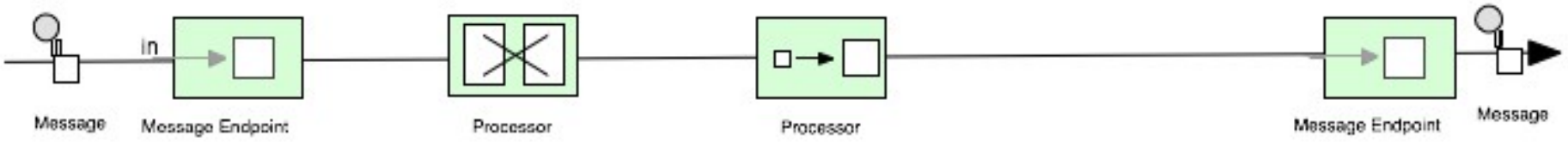
innoQ

Konzepte: Routing

© http://camel.apache.org

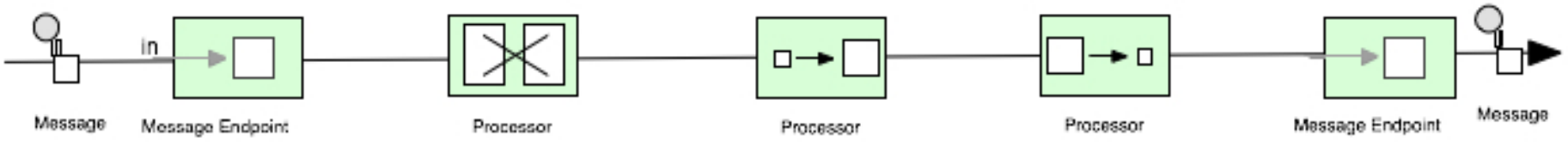
innoQ

Konzepte: Routing



© http://camel.apache.org

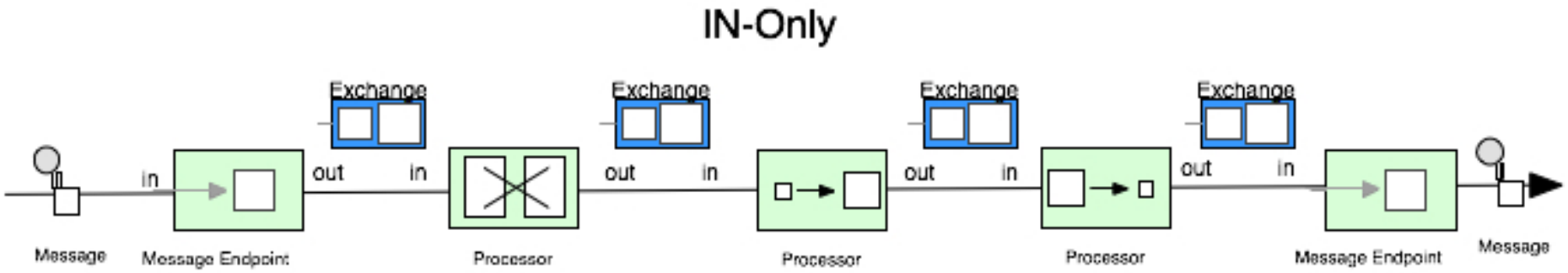
Konzepte: Routing



© http://camel.apache.org

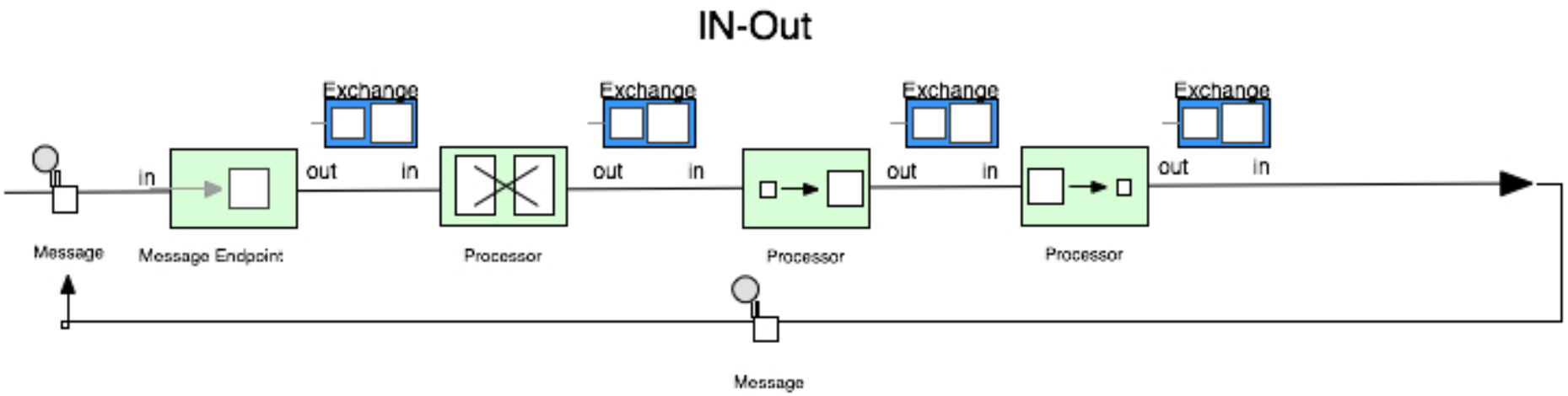
innoQ

Konzepte – Exchanges In-Only



© http://camel.apache.org

Konzepte – Exchanges In-Only



© http://camel.apache.org

Apache Camel - DSL

```
from("activemq:queue:XmlPayBlacks?cacheLevelName=CACHE_NONE")
        .routeId("xmlhandling")
        .unmarshal(xStreamDataFormat)
        .bean(new PayBlackBean(), "getPaidPositions")
        .split().body()
        .marshal(xStreamDataFormat)
        .setHeader("CamelFileName", simple("{date:now:yyyyMMddHHmmssSSS}.xml"))
        .to("file:///Users/martinh/dev/samples/result?fileExist=Append");
```

Apache Camel - DSL

```
from("activemq:queue:XmlPayBlacks?cacheLevelName=CACHE_NONE")
        .routeId("xmlhandling")
        .unmarshal(xStreamDataFormat)
        .bean(new PayBlackBean(), "getPaidPositions")
        .split().body()
        .marshal(xStreamDataFormat)
        .setHeader("CamelFileName", simple("{date:now:yyyyMMddHHmmssSSS}.xml"))
        .to("file:///Users/martinh/dev/samples/result?fileExist=Append");
```

Apache Camel - DSL

```
public class SampleRouter extends RouteBuilder {

    @Override
    public void configure() throws Exception {

        from("activemq:queue:XmlPayBlacks?cacheLevelName=CACHE_NONE")
                .routeId("xmlhandling")
                .unmarshal(xStreamDataFormat)
                .bean(new PayBlackBean(), "getPaidPositions")
                .split().body()
                .marshal(xStreamDataFormat)
                .setHeader("CamelFileName", simple("{date:now:yyyyMMddHHmmssSSS}.xml"))
                .to("file:///Users/martinh/dev/samples/result?fileExist=Append");


        ...
```

Apache Camel–Beans & Prozessoren

```
public class SampleRouter extends RouteBuilder {

    @Override
    public void configure() throws Exception {


        from("activemq:queue:XmlPayBlacks?cacheLevelName=CACHE_NONE")
                .routeId("xmlhandling")
                .unmarshal(xStreamDataFormat)
                .bean(new PayBlackBean(), "getPaidPositions")
                .process(new DoSthProcessor())
                .split().body()
                .marshal(xStreamDataFormat)
                .setHeader("CamelFileName", simple("{date:now:yyyyMMddHHmmssSSS}.xml"))
                .to("file:///Users/martinh/dev/samples/result?fileExist=Append");



        ...
```

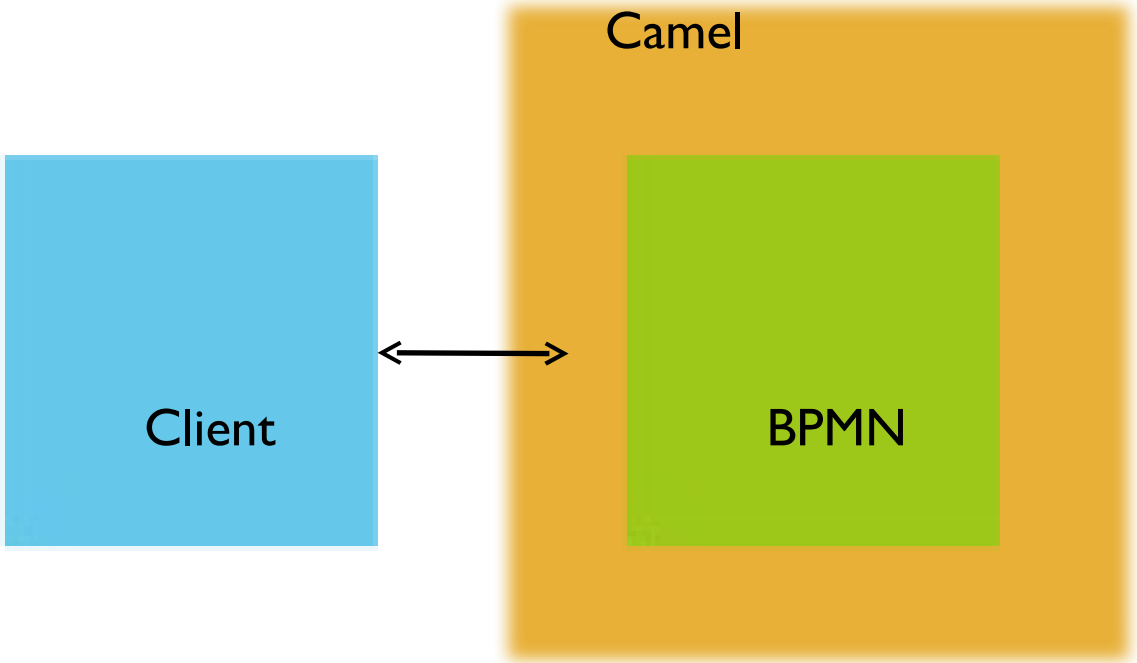
Apache Camel–Beans & Prozessoren

```
public class DoSthProcessor implements Processor {

        @SuppressWarnings("rawtypes")
        @Override
        public void process(Exchange exchange) throws Exception {

                String text =  exchange.getIn().getBody(String.class);
```

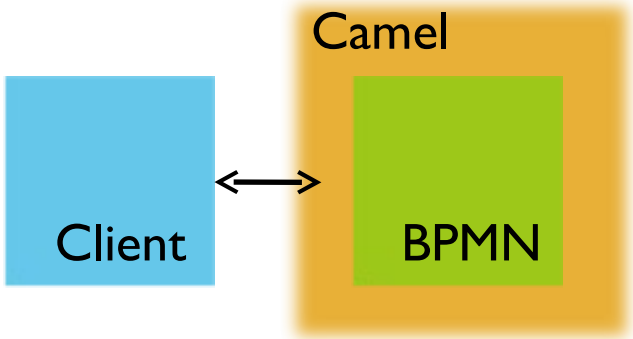
Agenda

- Geschäftsprozesse mit BPMN + BPEL
- Probleme
- Architektur mit BPMN und Apache Camel
- Apache Camel auf einen Blick
- Pufferschicht mit Camel & Beispiel

Pufferschicht



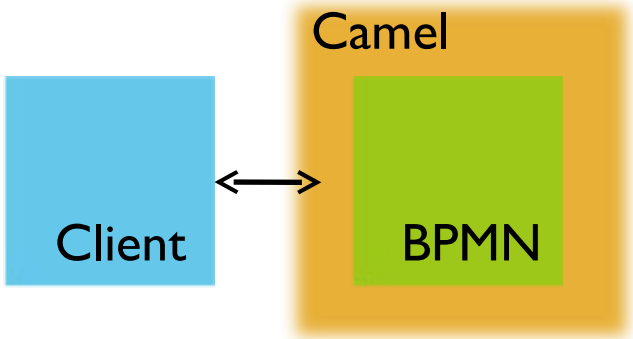
Pufferschicht - Schemakonsistenz



Schema-Transformation, wenn nötig

Routing zu passendem Webservice-Endpoint

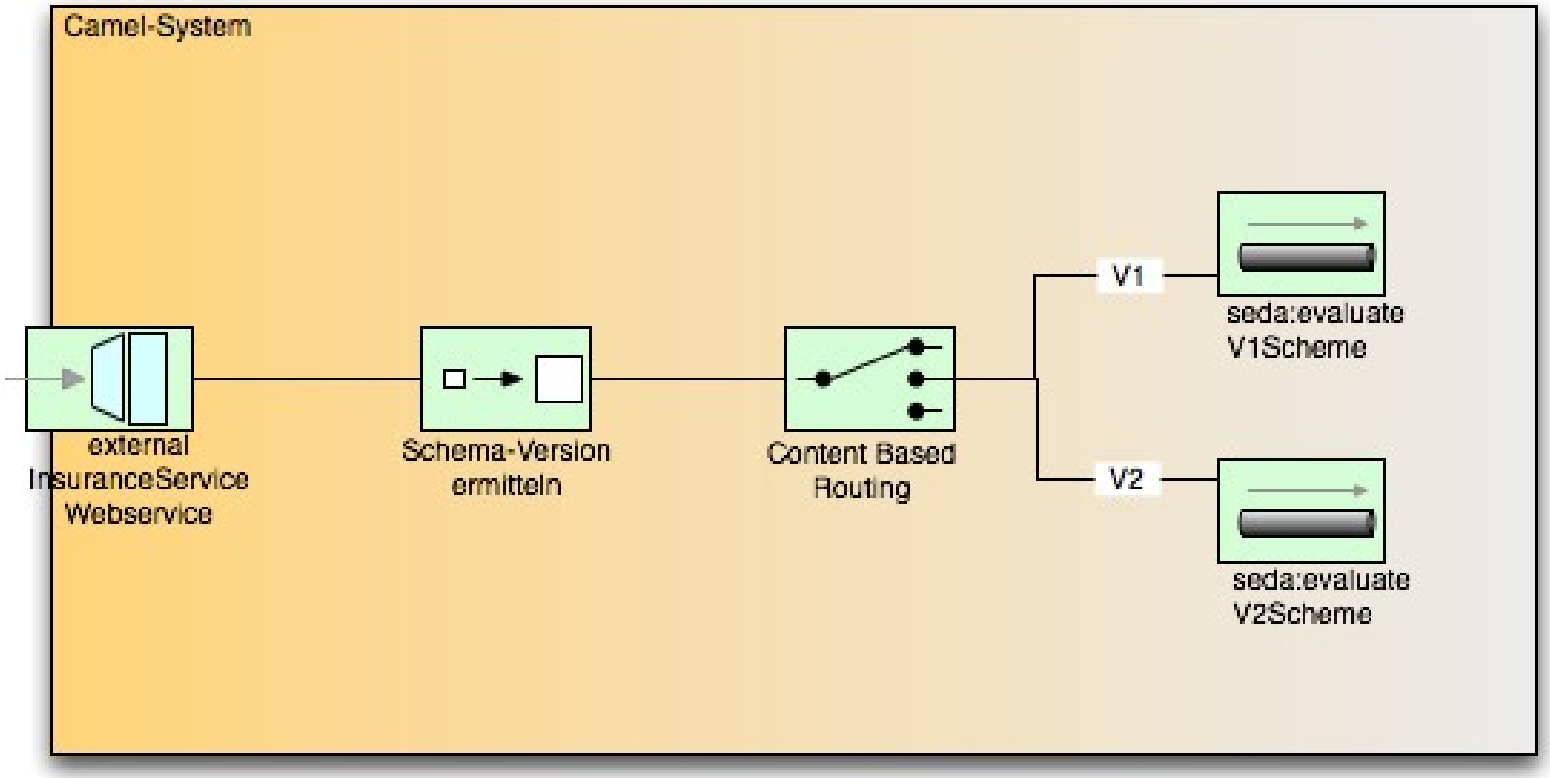
Pufferschicht - Datenspeicher



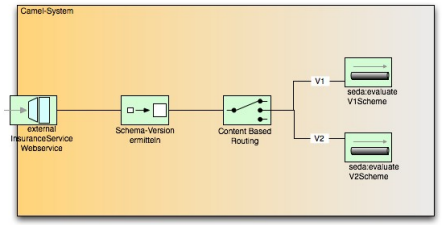
Zwischenspeicherung von Daten, die nicht im BPMN gebraucht werden

Wiederholen der Daten bei Bedarf

Webservice-Schemakonsistenz



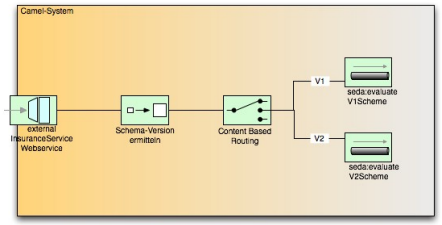
Webservice-Schemakonsistenz




```
public class SchemaEvaluationRouter extends RouteBuilder {

    @Override
     public void configure() throws Exception {

            // ------------ Namespace-Definition ----------------
            Namespaces ns_v1 = new Namespaces("c1", "http://bpel.innoq.com/insurance/v1/types");
            Namespaces ns_v2 = new Namespaces("c2", "http://bpel.innoq.com/insurance/v2/types");

            from("jetty:http://0.0.0.0:9080/insuranceservice?minThreads=5")
                        .process(new ContentPrinter())
                        .process(new PrintHeaderProcessor())
                        .choice()
                            .when().xpath("//c1:CarInsuranceFindProcess", ns_v1)
                                    .setHeader("schemaVersion", constant("1"))
                                    .setHeader("targetVersion").xpath("//c1:targetVersion", String.class, ns_v1)
                                    .to("seda:evaluateV1Scheme")
                            .when().xpath("//c2:CarInsuranceFindProcess", ns_v2)
                                    .setHeader("schemaVersion", constant("2"))
                                    .setHeader("targetVersion").xpath("//c2:targetVersion", String.class, ns_v2)
                                    .to("seda:evaluateV2Scheme")
                            .otherwise()
                                    .to("seda:fault");
```

innoQ

WS-Endpoint-Konfiguration in Spring

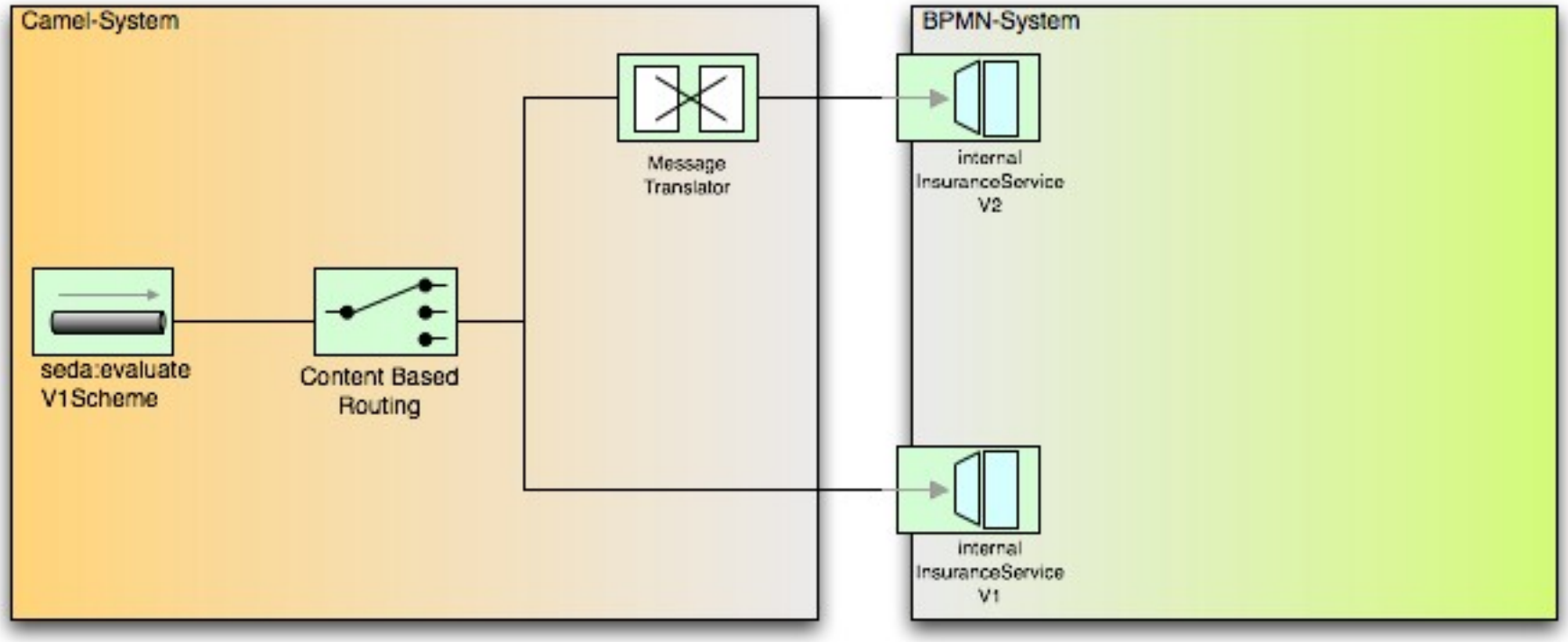



```
<beans xmlns="http://www.springframework.org/schema/beans"
        ...
        http://camel.apache.org/schema/spring http://camel.apache.org/schema/spring/camel-spring.xsd
         http://camel.apache.org/schema/cxf http://camel.apache.org/schema/cxf/camel-cxf.xsd">

        <import resource="classpath:META-INF/cxf/cxf.xml" />
        <import resource="classpath:META-INF/cxf/cxf-extension-soap.xml" />
        <import resource="classpath:META-INF/cxf/cxf-extension-http-jetty.xml" />

        <bean id="jetty" class="org.apache.camel.component.jetty.JettyHttpComponent" />

        <cxf:cxfEndpoint id="insuranceEndpoint-v1" address="http://localhost:9000/process1"
                wsdlURL="/wsdl/insuranceV1.wsdl">
                <cxf:properties>
                        <entry key="dataFormat" value="MESSAGE" />
                        <entry key="publishedEndpointUrl" value="http://localhost:9080/insuranceservice" />
                </cxf:properties>
        </cxf:cxfEndpoint>
        <cxf:cxfEndpoint id="insuranceEndpoint-v2" address="http://localhost:9000/process2"
                wsdlURL="/wsdl/insuranceV2.wsdl">
                <cxf:properties>
                        <entry key="dataFormat" value="MESSAGE" />
                        <entry key="publishedEndpointUrl" value="http://localhost:9080/insuranceservice" />
                </cxf:properties>
        </cxf:cxfEndpoint>
```

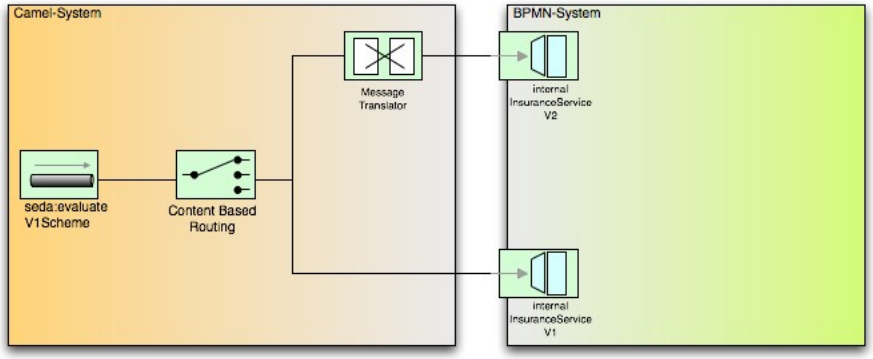
```
from("jetty:http://0.0.0.0:9080/insuranceservice?minThreads=5")
        .process(new ContentPrinter())
```

innoQ

Webservice-Schemakonsistenz



Webservice-Schemakonsistenz




```
from("seda:evaluateV1Scheme")
        .choice()
            .when(header("targetVersion").isEqualTo("2"))
                    .pipeline("direct:transformV1V2", "cxf:bean:insuranceEndpoint-v2")
                    .to("cxf:bean:insuranceEndpoint-v2")
            .otherwise()
                    .to("cxf:bean:insuranceEndpoint-v1");
```

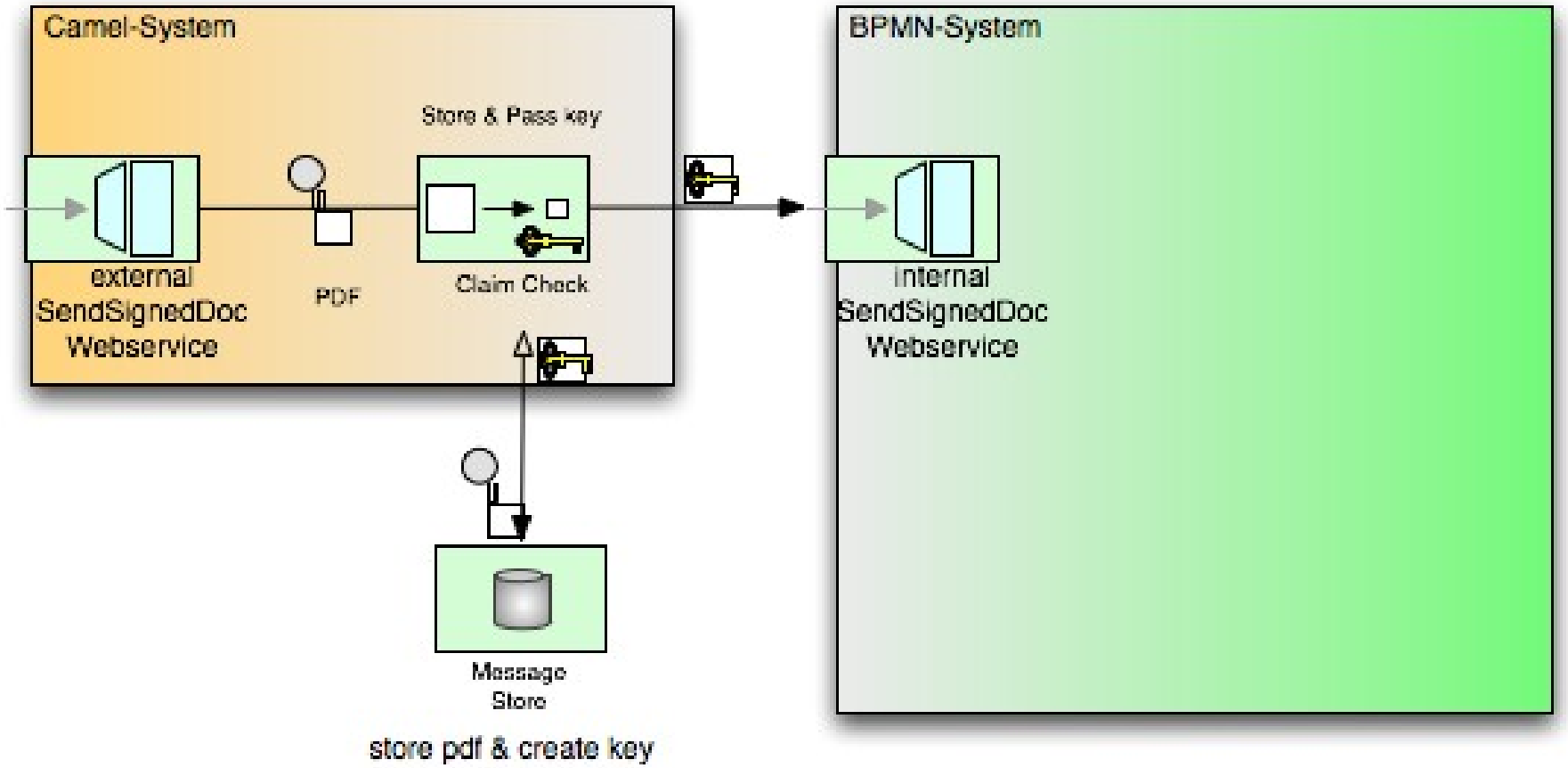
Interne Webservices (Fake)

```
from("cxf:bean:insuranceEndpoint-v1")
        .id("handleProcess_v1")
        .transform().xpath("//c1:CarInsuranceFindProcess", ns_v1)
        .unmarshal(jaxb10)
        .process(new ContentsProcessor())
        .marshal(jaxb10)
        .to("seda:save")
        .process(new ResponseBuilderProcessor(10));

from("cxf:bean:insuranceEndpoint-v2")
        .id("handleProcess_v2")
        .transform().xpath("//c2:CarInsuranceFindProcess", ns_v2)
        .unmarshal(jaxb11)
        .process(new ContentsProcessor())
        .marshal(jaxb11)
        .to("seda:save")
        .process(new ResponseBuilderProcessor(11));
```

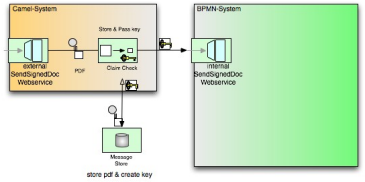
innoQ

Datenspeicher

Datenspeicher - Store

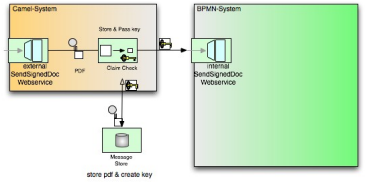


Datenspeicher - Store




```
public class FileEntrePot extends RouteBuilder {

@Override
public void configure() throws Exception {

Namespaces ns_external = new Namespaces("ext", "http://www.example.org/external/");

from("cxf:bean:externalSendSignedDocument")
        .process(new ContentPrinter())
        .setHeader("processid").xpath("//ext:sendSignedDocument/processId/text()",
ns_external)
        .setHeader("CamelFileName", simple("Process_${in.header.processid}.pdf.txt"))
        .multicast()
                .to("seda:handlefile", "seda:forwardmessage")

        .bean(ExternalResponse.class, "createSSDResponse");
```

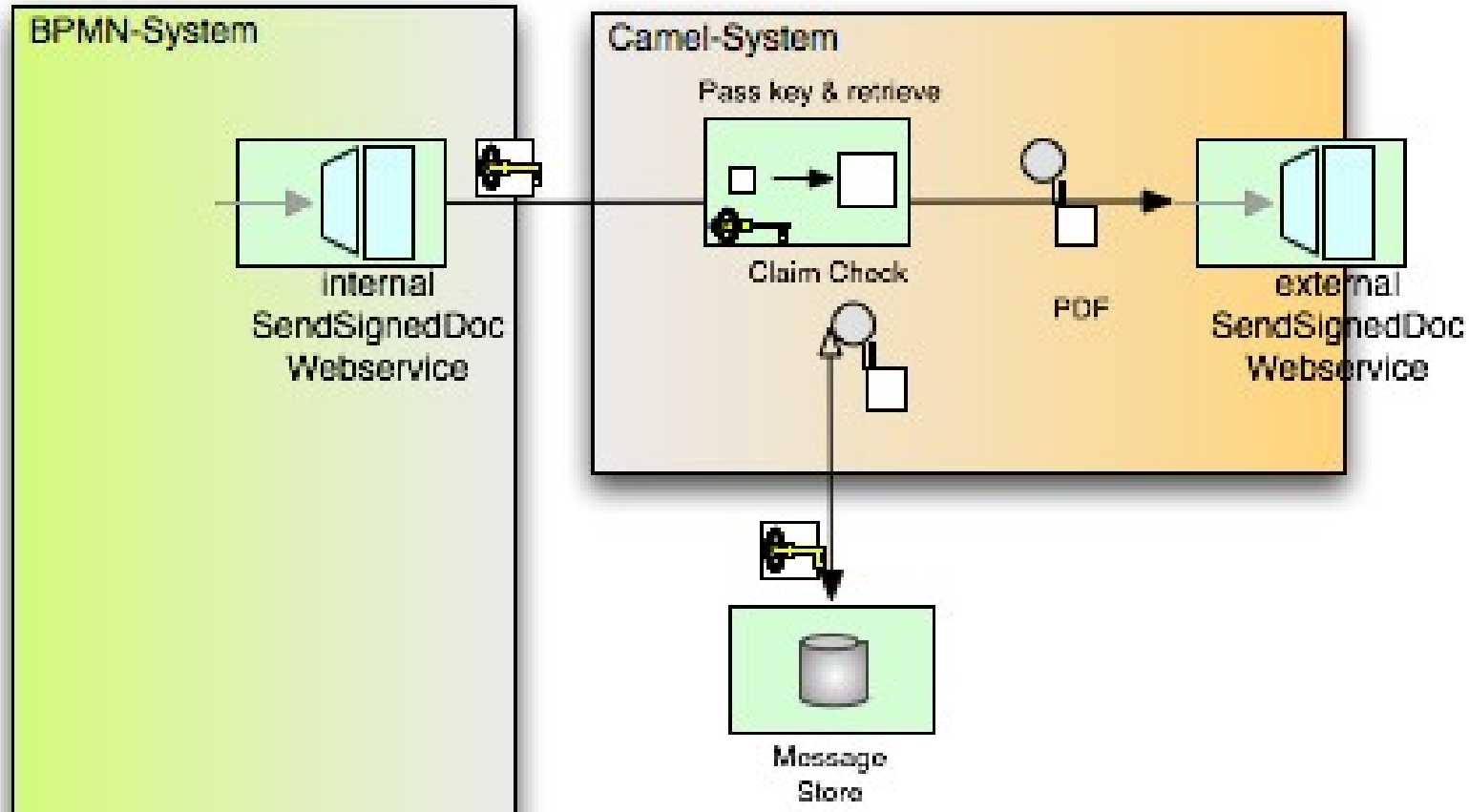
Datenspeicher - Store



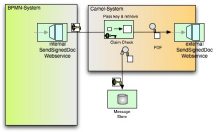

```
.multicast()
        .to("seda:handlefile", "seda:forwardmessage")
```

```
from("seda:handlefile")
        .transform().xpath("//ext:sendSignedDocument/pdf/text()", ns_external)
        .to("file:target/output");

from("seda:forwardmessage")
        .setBody(simple("${in.CamelFileName}"))
        .to("cxf:bean:internalSendSignedDocument");
```

Datenspeicher - Retrieve



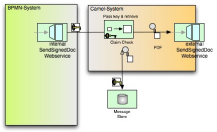
Datenspeicher - Retrieve



```
public class FileReAcquire extends RouteBuilder {

    @Override
    public void configure() throws Exception {

        onException(SoapFault.class)
                        .maximumRedeliveries(0).handled(true)
                        .process(new SoapFaultProcessor());

        from("cxf:bean:internalMakeApplicationheaderFilterStrategy=#dropAllMessageHeadersStrategy")
                        .process(new ReAcquireProcessor())
                        .process(new ContentPrinter())
                        .to("seda:callExternal")
                        .setBody(constant("OK"));

        from("seda:callExternal")
                        .to("cxf:bean:externalMakeApplication");

        from("cxf:bean:externalMakeApplication")
                        .convertBodyTo(String.class)
                        .to("seda:out");
```

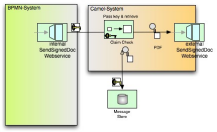
innoQ

Datenspeicher - Retrieve



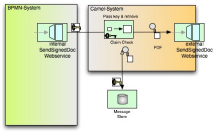
```
public class ReAcquireProcessor implements Processor {

        ConsumerTemplate consumer;
        private static final Namespace ns_ext =
                Namespace.getNamespace("ext“,  "http://www.example.org/external/");
        private static final Namespace ns_int =
                Namespace.getNamespace("ext“, "http://www.example.org/internal/");

        org.jdom.output.XMLOutputter out = new XMLOutputter();

        private static final String EXCEPTION_MESSAGE = "No Document Found!";

        @Override
        @SuppressWarnings({ "rawtypes" })
        public void process(Exchange exch) throws Exception {
                Element message = parse(exch.getIn().getBody(InputStreamCache.class));
                ...
```

Datenspeicher - Store



```
public class ReAcquireProcessor implements Processor {

        ConsumerTemplate consumer;
        private static final Namespace ns_ext =
                Namespace.getNamespace("ext“,  "http://www.example.org/external/");
        private static final Namespace ns_int =
                Namespace.getNamespace("ext“, "http://www.example.org/internal/");

        org.jdom.output.XMLOutputter out = new XMLOutputter();

        private static final String EXCEPTION_MESSAGE = "No Document Found!";

        @Override
        @SuppressWarnings({ "rawtypes" })
        public void process(Exchange exch) throws Exception {
                Element message = parse(exch.getIn().getBody(InputStreamCache.class));
                ...
```

Datenspeicher - Store



```
public class ReAcquireProcessor implements Processor {

        ConsumerTemplate consumer;
        private static final Namespace ns_ext =
                Namespace.getNamespace("ext“,  "http://www.example.org/external/");
        private static final Namespace ns_int =
                Namespace.getNamespace("ext“, "http://www.example.org/internal/");

        org.jdom.output.XMLOutputter out = new XMLOutputter();

        private static final String EXCEPTION_MESSAGE = "No Document Found!";

        @Override
        @SuppressWarnings({ "rawtypes" })
        public void process(Exchange exch) throws Exception {
                Element message = parse(exch.getIn().getBody(InputStreamCache.class));
                ...
```

Vielen Dank!

Daniel Lübke, @
stefan.tilkov@innoq.com
http://www.innoq.com
Phone: +

Martin Huber, @Waterback
martin.huber@innoq.com
http://www.innoq.com
Phone: +49 171 122 66 501

Wir lösen das – persönlich!

innoQ

Vielen Dank!

Dr. Daniel Lübke & Martin Huber

innoQ Deutschland GmbH