

# JBug OWL

Red Hat JBoss Fuse

Alle IT Projekte sind Integrationsprojekte

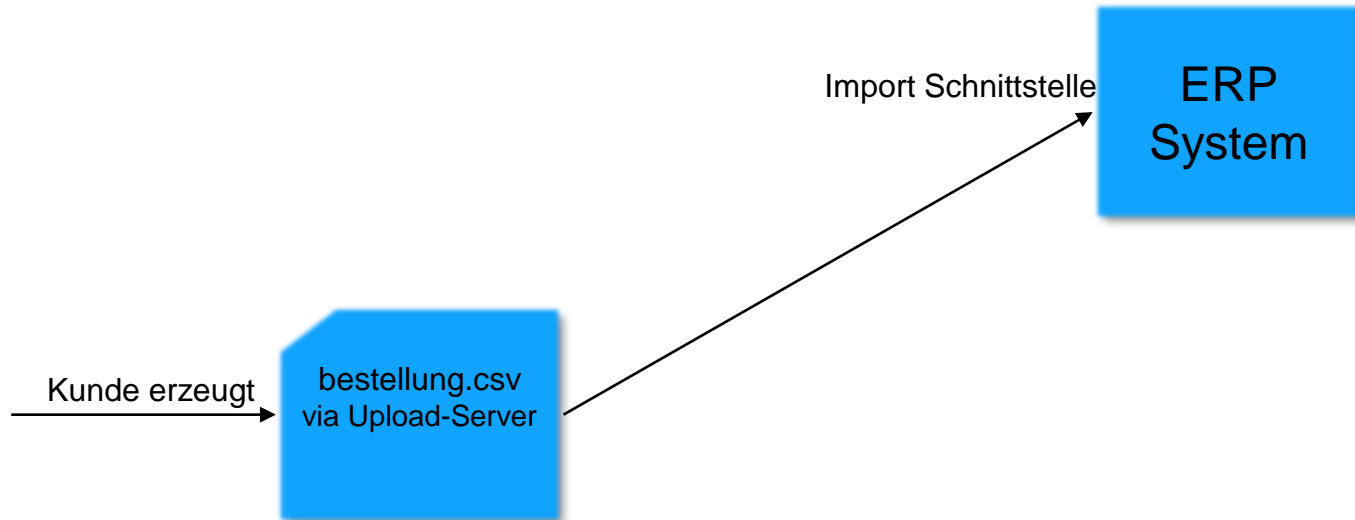
Thorsten Laux & Kamil Swierkot  
S&N AG

# Abstract

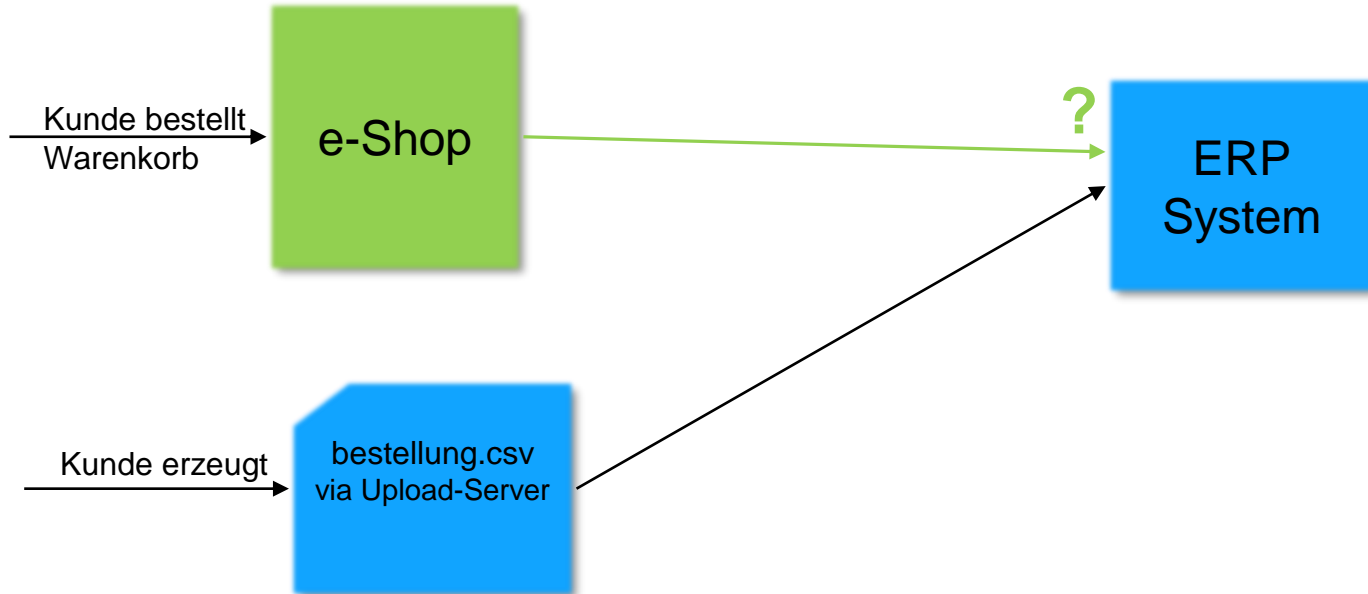
- Integration ist das große Schlagwort in der IT.
- Das IoT treibt diesen Gedanken noch weiter.
- Aber:
  - Immer wieder werden spezifische Connectoren und Adaptern für die Integration vorhandener Alt-/Legacy Systeme geschrieben. Dies erzeugt eine enge Kopplung der Systeme.
  - Jboss Fuse ermöglicht die lose Kopplung von Systemen durch Konfiguration unter Verwendung von EIP\*.

\* EIP: Enterprise Integration Pattern nach Hohpe, et. al

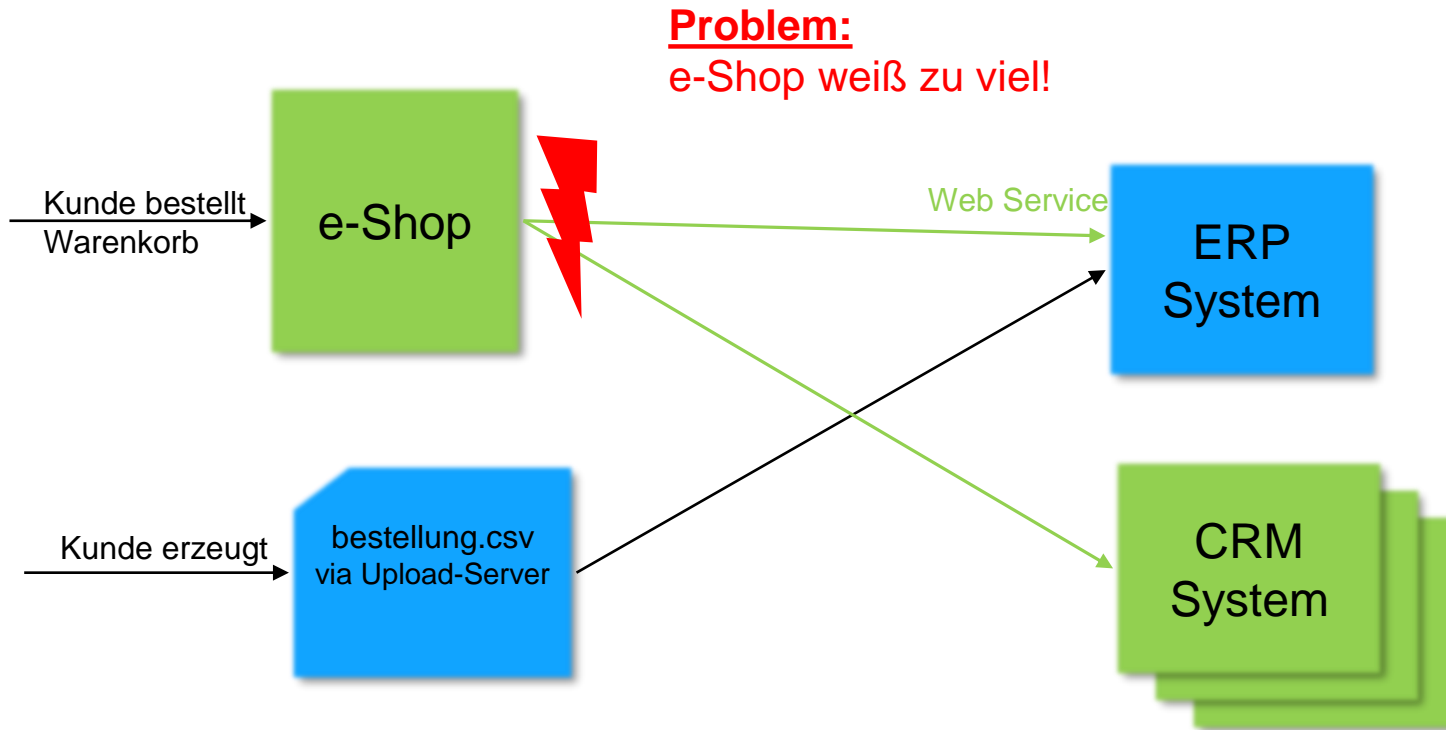
# Integration: B2B Szenario



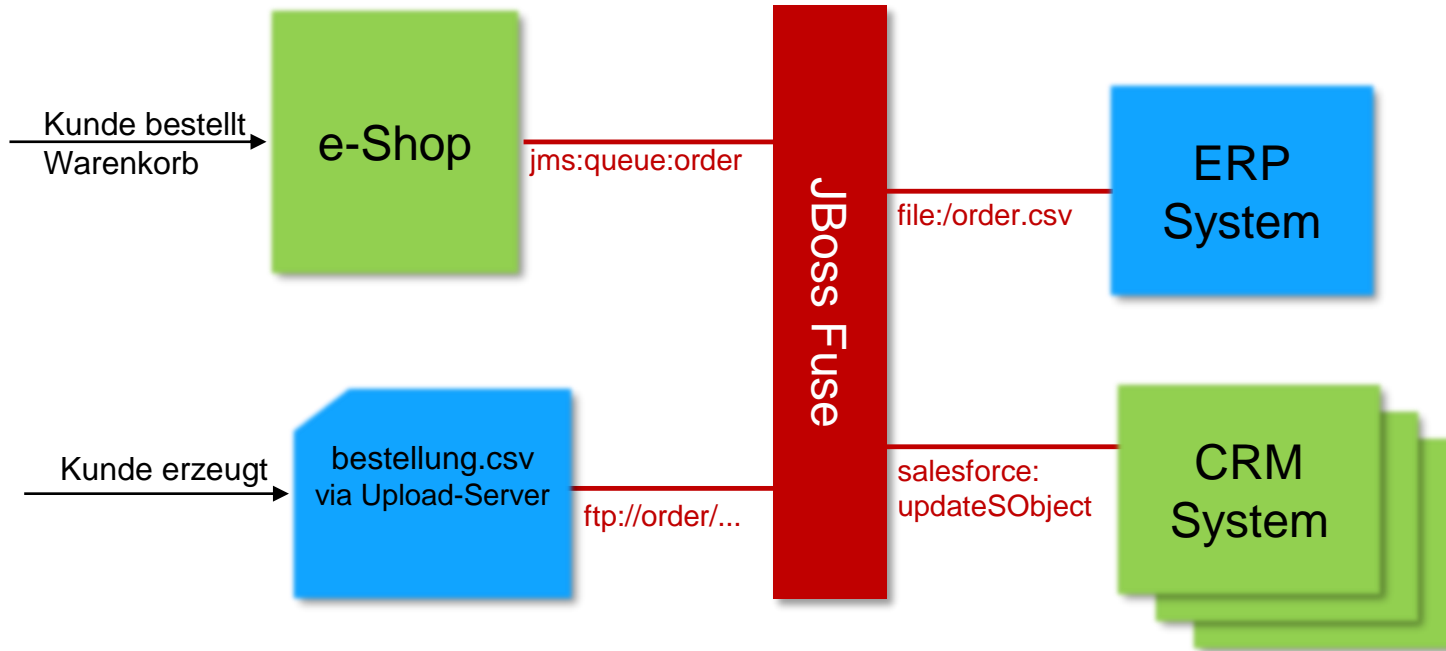
# Integration: B2B Szenario



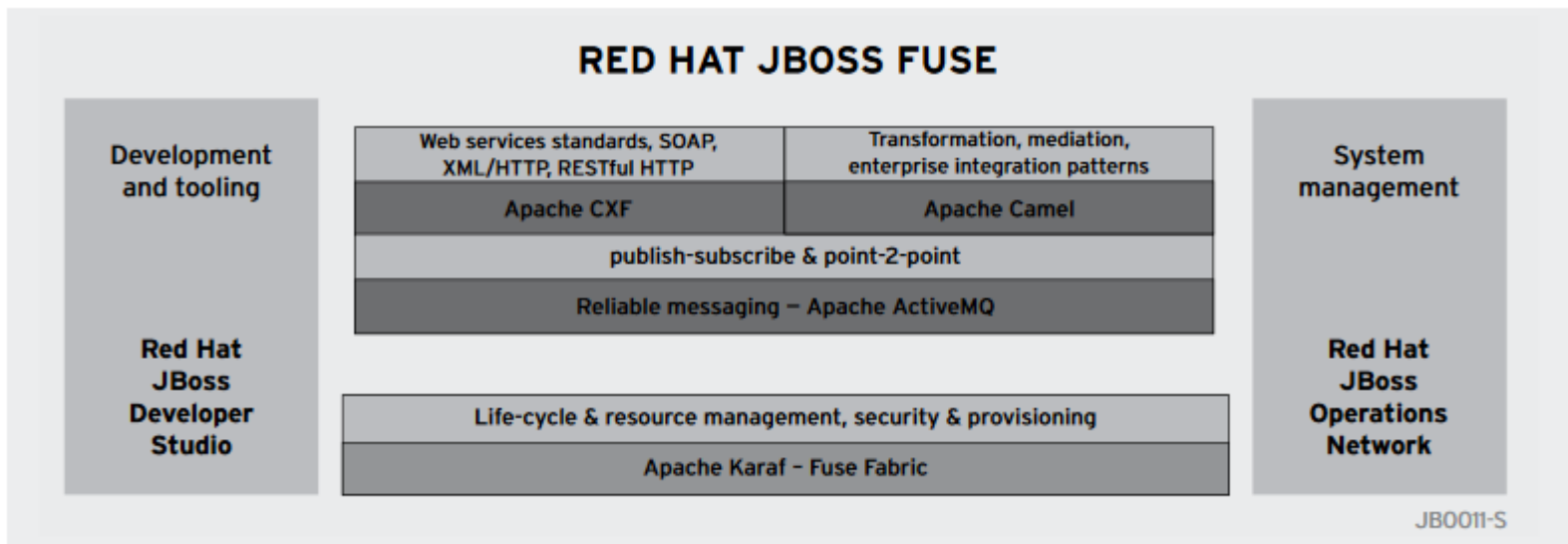
# Integration: B2B Szenario



# Integration: B2B Szenario - Beispiellösung



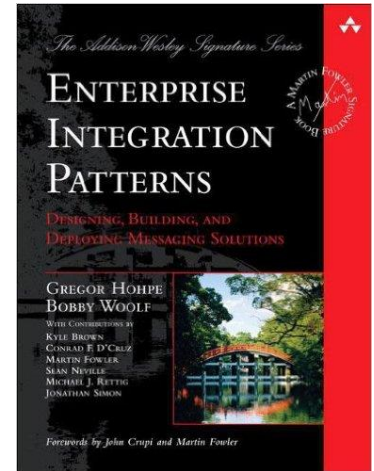
# Übersicht – JBoss Fuse



<https://www.redhat.com/en/files/resources/mi-connected-enterprise-fuse-datasheet-task0133880-inc0311312mm-201511-v2.pdf>

# Integration – Enterprise Integration Pattern (EIP)

- ❖ Buch von Gregor Hohpe und Bobby Woolf
- ❖ Pattern und Rezepte für typische Integrationsprobleme
- ❖ Gemeinsames Vokabular für Entwickler
- ❖ Nachrichtenzentriert
- ❖ Basis für alle größeren Integrationsprodukte

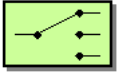

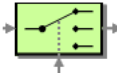
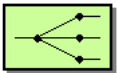
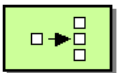
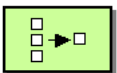


<http://www.eaipatterns.com/>



# Integration – EIP Auswahl

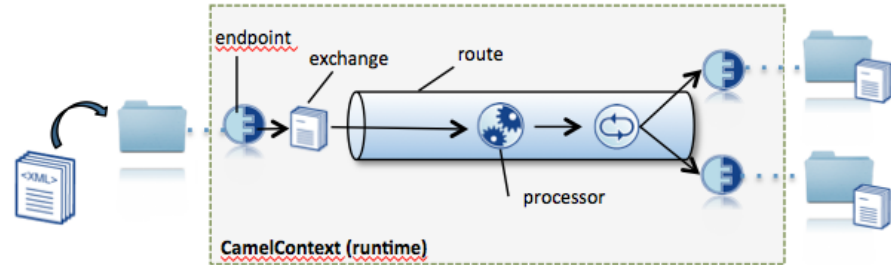
## Message Routing

	Content Based Router	How do we handle a situation where the implementation of a single logical function (e.g., inventory check) is spread across multiple physical systems?
	Message Filter	How can a component avoid receiving uninteresting messages?
	Dynamic Router	How can you avoid the dependency of the router on all possible destinations while maintaining its efficiency?
	Recipient List	How do we route a message to a list of (static or dynamically) specified recipients?
	Splitter	How can we process a message if it contains multiple elements, each of which may have to be processed in a different way?
	Aggregator	How do we combine the results of individual, but related messages so that they can be processed as a whole?

<http://camel.apache.org/enterprise-integration-patterns.html>

# Integration - Camel

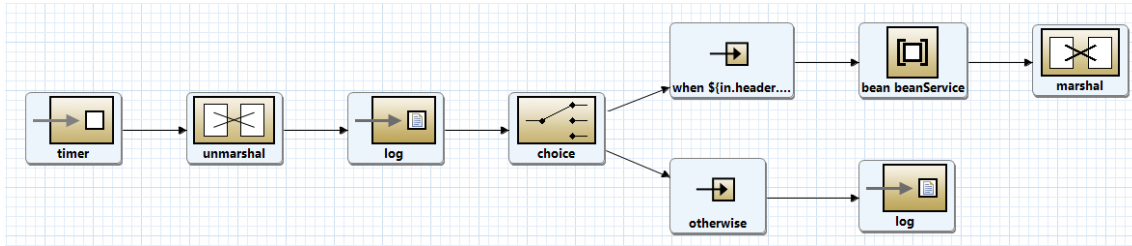
- Integration mit Apache Camel
- Implementiert EIP
  - Stellt >140 Endpoints bereit
  - Processors  $\approx$  Transportweg zwischen Endpoints
    - Transformationen
    - Validierung
    - Routing
  - Datenaustausch zwischen Komponenten über Exchange
- Große und aktive Community



<http://blog.octo.com/wp-content/uploads/2011/03/camel-usecase.png>

# Integration - Camel

## Route werden über DSL oder XML definiert



```
from("file:/data/in?idempotent=true&idempotentKey=${file:name}-${file:modified}&include=*.csv").routeId
  .idempotentConsumer(header("CamelFileName").append(header("CamelFileLastModified")),
    FileIdempotentRepository.fileIdempotentRepository(new File(config.getIdempotentStore()).getSto
  .log("Found file: ${simple{in.header.CamelFileName}}")
  .choice()
    .when(header("CamelFileName").regex(".{8}.*"))
      .convertBodyTo(String.class)
      .bean(new EtnMatcherBean(config.getEtnLookup()))
    .choice()
      .when(header("etnCount").isEqualTo(1))
        .bean(new MqttMessageProducer(config.getWorkplaceId()))
        .log("Sending To MQTT Broker:${in.body}")
        .bean(new MqttSender(config.getWorkplaceId(), config.getMqttConfig()))
```

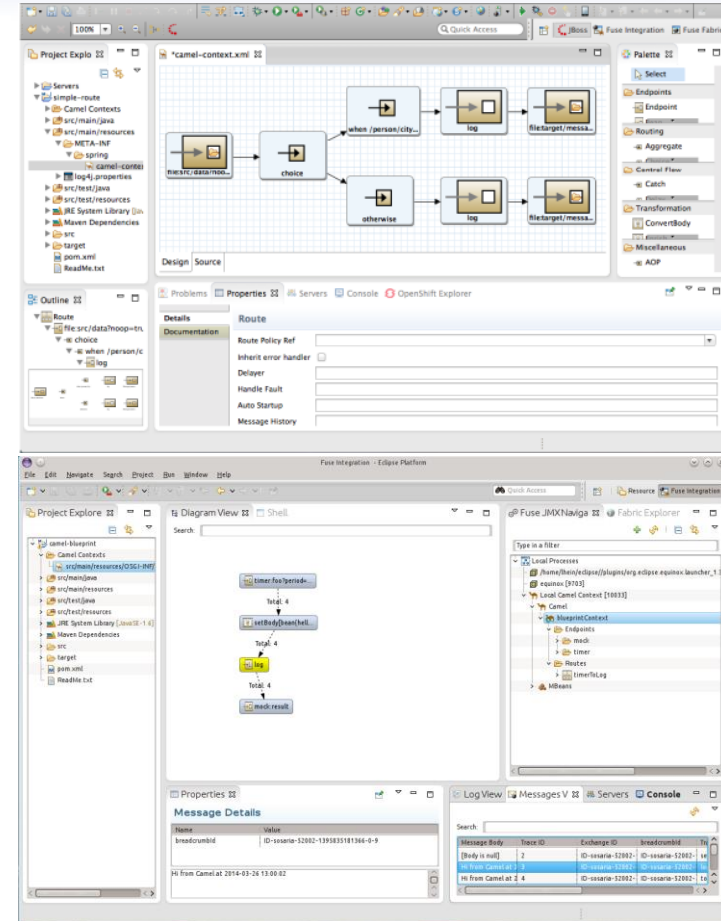
```
<camelContext trace="false" xmlns="http://camel.apache.org/schema/spring">
  <route>
    <from uri="cxf:bean:WS?dataFormat=MESSAGE" id="timer">
      <description/>
    </from>
    <unmarshal>
      <soapjxb contextPath="com.redhat.gpe.training" version="2.0" />
    </unmarshal>
    <log message="Received Request!" />
    <choice>
      <when>
        <simple>${in.header.SOAPAction} contains "saveCustomer"</simple>
        <bean ref="beanService" method="saveCustomer"/>
        <marshal>
          <soapjxb contextPath="com.redhat.gpe.training" version="2.0" />
        </marshal>
      </when>
      <otherwise>
        <log message="NotFound! ${in.header.SOAPAction}" />
      </otherwise>
    </choice>
  </route>
</camelContext>
```

# Live Demo

📦 Im JBoss Developer ...

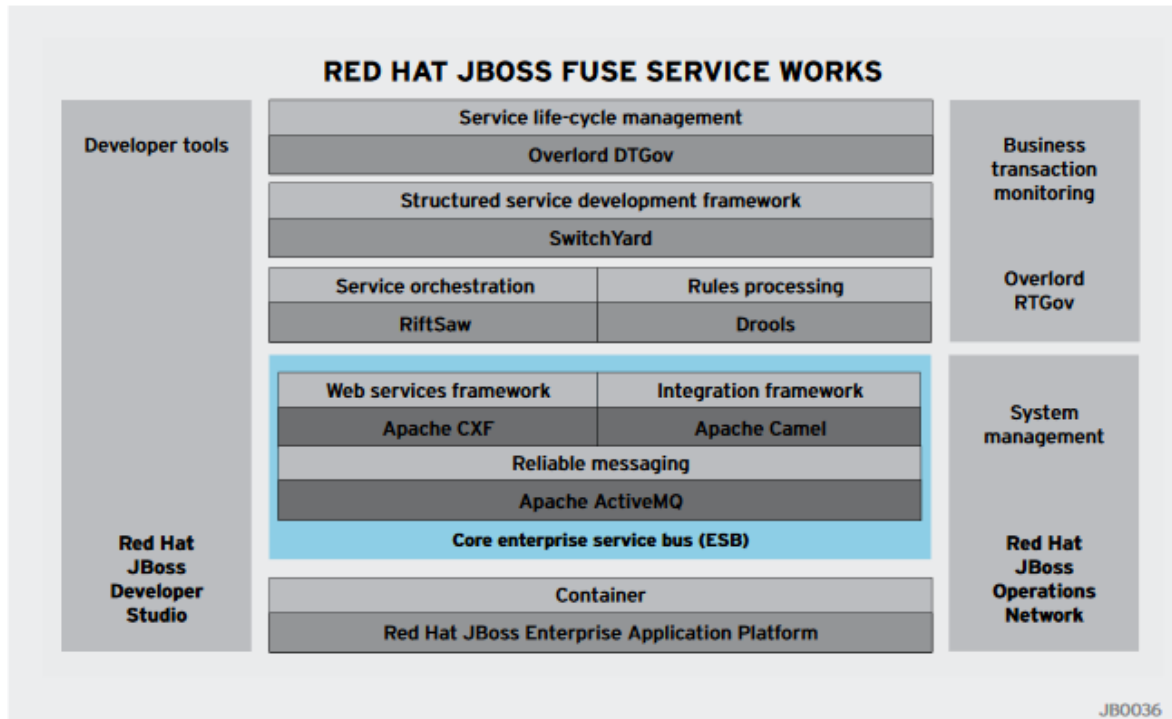
# Tooling

- Integration in JBoss Developer Studio
- Visueller Editor für Camel Routes
- Visueller Trace während Test
- Deployment und Steuerung in Fabric8 und Apache Karaf



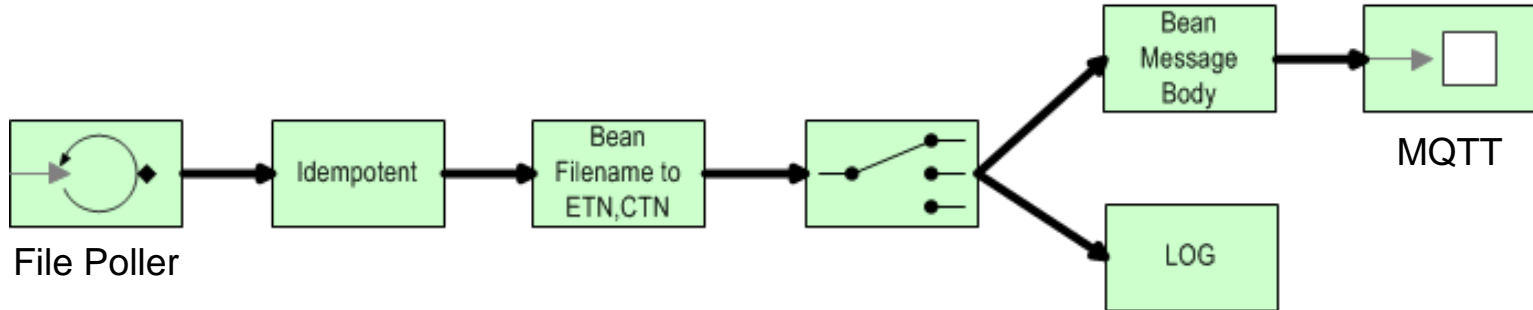
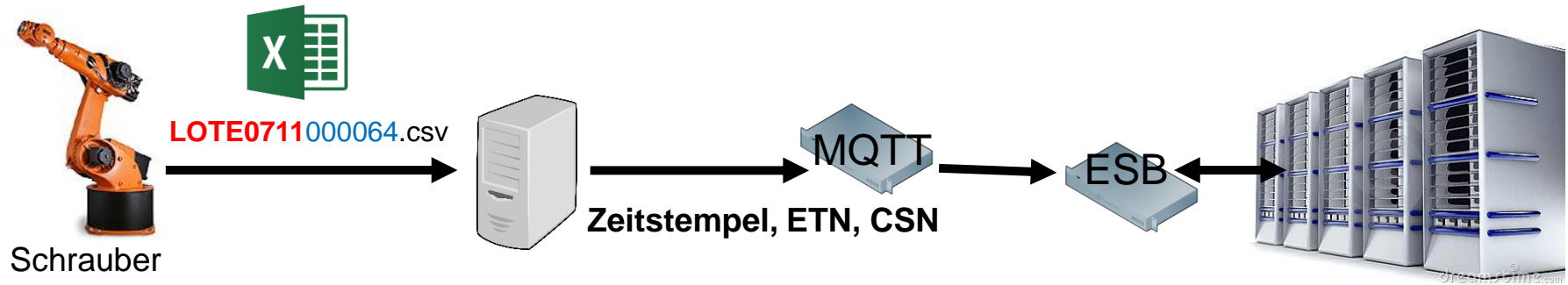
<http://tools.jboss.org/features/apachecamel.html>

# Übersicht – JBoss Fuse Service Works

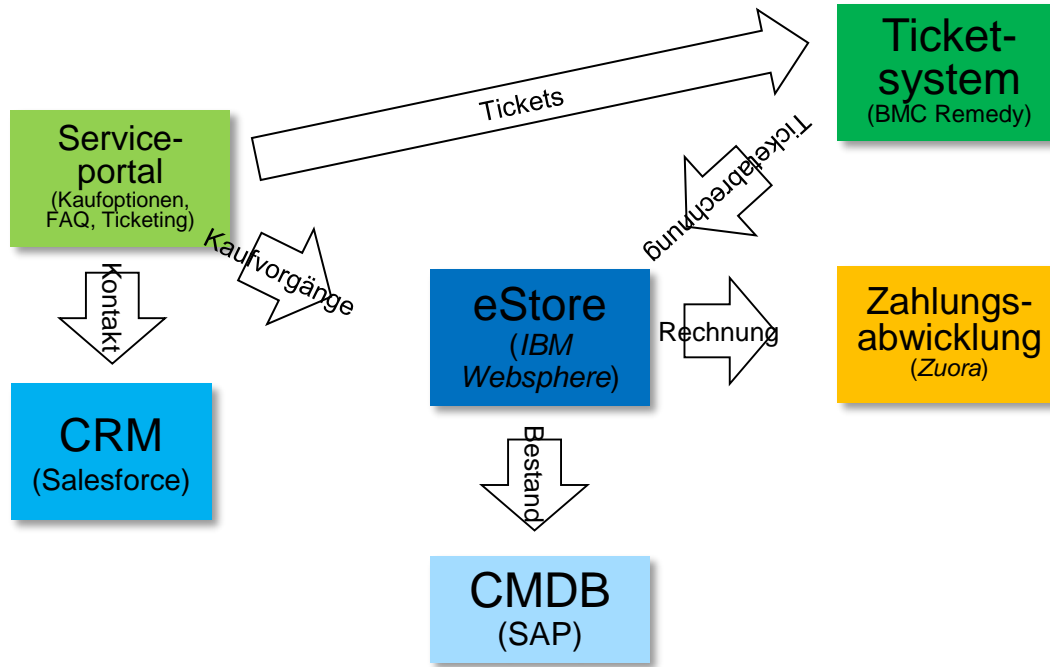


<https://www.redhat.com/de/files/resources/en-rhjb-fuse-service-works-11609347.pdf>

# Beispiele aus der Praxis

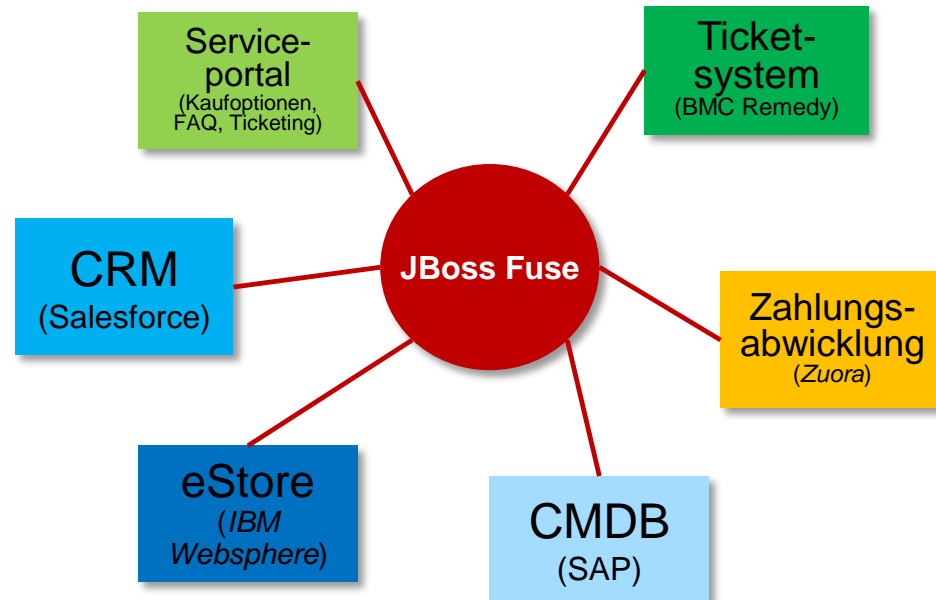


# Beispiele aus der Praxis: SaaS Cloud





# Beispiele aus der Praxis: SaaS Cloud



**Vielen Dank !**

Thorsten Laux & Kamil Swierkot  
{tlaux,kswierkot}@s-und-n.de

# Messaging

- ◆ Messaging über Apache ActiveMQ
- ◆ ActiveMQ unterstützt
  - ◆ Point-To-Point und Publish/Subscribe Verbindungen
  - ◆ Persistent und nicht Persistent Kommunikation
  - ◆ Verschiedene Protokolle: JMS, AMQP, MQTT, ...
  - ◆ Kann horizontal und vertikal skalieren
  - ◆ Verschieden Sprachen
- ◆ Kann auch in einem Java Applikation eingebettet werden

# Webservices

- ◆ Realisiert durch Apache CXF
  - ◆ Leichtgewichtiges Framework für WebServices
  - ◆ Erlaubt das Bereitstellen & Konsumieren von Services
  - ◆ Unterstützt
    - ◆ SOAP (JAX-WS)
      - ◆ Via WSDL
      - ◆ Via Annotations
    - ◆ REST (JAX-RS)

# Dependency Injection

- ◆ Fuse unterstützt zwei DI Frameworks
  - ◆ Spring
  - ◆ Blueprint
    - ◆ Kann Dependencies zur Laufzeit nachladen
    - ◆ Speziell für OSGi definiert und zugeschnitten
    - ◆ XML basiert
- ◆ Realisieren XML DSL für Apache Camel
- ◆ ActiveMQ kann über DI eingebettet werden

# Lifecycle

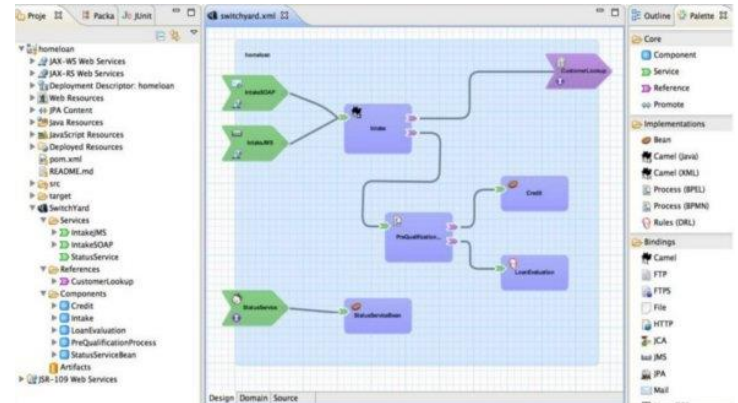
- ◆ Apache Karaf
  - ◆ Leichtgewichtige OSGi Laufzeitumgebung (4.3)
  - ◆ Konsole für Applikationsmanagement und -administration
- ◆ Jboss EAP
- ◆ Fabric8
  - ◆ Container basierte Deployments via Docker und Kubernetes
  - ◆ Tools für zentralisierte Konfiguration
- ◆ Apache Camel kann auch Java Applikation eingebettet werden

# Overlord

- ◆ Project Lifecycle Management/Governance
  - ◆ S-RAMP:
    - ◆ Artifakt, Metadata und Informations Repository
    - ◆ Discovery
  - ◆ Design Time Governance
    - ◆ Workflows für S-RAMP Artefakte
  - ◆ Runtime Governance
    - ◆ Monitor Services Activity via Events
    - ◆ Events auswerten und Informationen darstellen

# Switch Yard

- Service Component Architecture
- Strukturiert Service ...
  - Design
  - Implementierung
  - Deployment
- Integriert Camel, Java und BPEL (RiftSaw)



<http://switchyard.jboss.org/>



# BRMS

- ◆ Business Rule Management System
  - ◆ Process Definition (BPMN 2.0)
  - ◆ Execution
  - ◆ Management
- ◆ Business Rules
  - ◆ Grafische Oberfläche
  - ◆ Decision Tables
- ◆ Complex Event Processing (CEP)