

# Konzeption und Realisierung von Daten- und Funktionsintegrationsszenarien mit Hilfe des EIP-Frameworks Apache Camel

Sven Eisenhauer

Fachbereich Informatik  
Hochschule Darmstadt

Daten- und Systemintegration WS 2009/2010

# Gliederung

- 1 Motivation
  - Unternehmensintegration
  - Enterprise Integration Patterns
  
- 2 Prototyp
  - Szenario
  - Apache Camel
  - Implementierung
  - Live Demo

# Gliederung

- 1 Motivation
  - Unternehmensintegration
  - Enterprise Integration Patterns
- 2 Prototyp
  - Szenario
  - Apache Camel
  - Implementierung
  - Live Demo

# Was macht Integration schwierig?

- Integration findet in heterogenen Umgebungen statt.
- Keine zwei Szenarien sind identisch.
- Jede Integration bildet eigenständige Lösung.
- Vielzahl unterschiedlicher Integrationslösungen

# Was macht Integration schwierig?

- Integration findet in heterogenen Umgebungen statt.
- Keine zwei Szenarien sind identisch.
- Jede Integration bildet eigenständige Lösung.
- Vielzahl unterschiedlicher Integrationslösungen

# Was macht Integration schwierig?

- Integration findet in heterogenen Umgebungen statt.
- Keine zwei Szenarien sind identisch.
- Jede Integration bildet eigenständige Lösung.
- Vielzahl unterschiedlicher Integrationslösungen

# Was macht Integration schwierig?

- Integration findet in heterogenen Umgebungen statt.
- Keine zwei Szenarien sind identisch.
- Jede Integration bildet eigenständige Lösung.
- Vielzahl unterschiedlicher Integrationslösungen

# Gliederung

- 1 Motivation
  - Unternehmensintegration
  - Enterprise Integration Patterns
- 2 Prototyp
  - Szenario
  - Apache Camel
  - Implementierung
  - Live Demo

# Enterprise Integration Patterns helfen

- Integrationszenarien haben Gemeinsamkeiten.
- EIPs bilden diese Gemeinsamkeiten ab.
- Stellen intuitive Beschreibung und Kommunikationsform bereit.
- EIPs unterstützen Entwurf auf logischer Ebene.
- Hersteller- und technologieneutral.
- Lose Kopplung mittels Nachrichtenorientierung

# Enterprise Integration Patterns helfen

- Integrationszenarien haben Gemeinsamkeiten.
- EIPs bilden diese Gemeinsamkeiten ab.
- Stellen intuitive Beschreibung und Kommunikationsform bereit.
- EIPs unterstützen Entwurf auf logischer Ebene.
- Hersteller- und technologieneutral.
- Lose Kopplung mittels Nachrichtenorientierung

# Enterprise Integration Patterns helfen

- Integrationszenarien haben Gemeinsamkeiten.
- EIPs bilden diese Gemeinsamkeiten ab.
- Stellen intuitive Beschreibung und Kommunikationsform bereit.
- EIPs unterstützen Entwurf auf logischer Ebene.
- Hersteller- und technologieneutral.
- Lose Kopplung mittels Nachrichtenorientierung

# Enterprise Integration Patterns helfen

- Integrationszenarien haben Gemeinsamkeiten.
- EIPs bilden diese Gemeinsamkeiten ab.
- Stellen intuitive Beschreibung und Kommunikationsform bereit.
- EIPs unterstützen Entwurf auf logischer Ebene.
- Hersteller- und technologieutral.
- Lose Kopplung mittels Nachrichtenorientierung

# Enterprise Integration Patterns helfen

- Integrationszenarien haben Gemeinsamkeiten.
- EIPs bilden diese Gemeinsamkeiten ab.
- Stellen intuitive Beschreibung und Kommunikationsform bereit.
- EIPs unterstützen Entwurf auf logischer Ebene.
- Hersteller- und technologieutral.
- Lose Kopplung mittels Nachrichtenorientierung

## Enterprise Integration Patterns helfen

- Integrationszenarien haben Gemeinsamkeiten.
- EIPs bilden diese Gemeinsamkeiten ab.
- Stellen intuitive Beschreibung und Kommunikationsform bereit.
- EIPs unterstützen Entwurf auf logischer Ebene.
- Hersteller- und technologieneutral.
- Lose Kopplung mittels Nachrichtenorientierung

# Übersicht EIPs



Aggregator



Channel Adapter



Channel



Channel Purger



Command Message



Competing Consumers



Content Based



Content Filter



Control Bus



Correlation ID



Content Enricher



Datatype Channel



Dead Letter Channel



Detour



Composed Message



Document Message



Durable Subscriber



Envelope Wrapper



Event-Driven Consumer



Event Message



Message Filter



Guaranteed Delivery



Invalid Message



Message Bus



Message Dispatcher



Message Endpoint



Message



Message Branch



Message Router



Selective Consumer



Message Sequence



Message Store



Message Translator



Service Activator



Messaging Bridge



Messaging Gateway



Normalizer



Poling Consumer



Process Manager



Publish-Subscribe Channel



Recipient List



Request Reply

# Gliederung

- 1 Motivation
  - Unternehmensintegration
  - Enterprise Integration Patterns
- 2 Prototyp
  - Szenario
  - Apache Camel
  - Implementierung
  - Live Demo

# Anforderungen

- Bestellungen sollen als CSV-Dateien und über WS an Legacysystem geschickt werden.
- Benachrichtigung per E-Mail über neue Bestellung.
- Jede E-Mail speichern.
- Rückgabe der Bestellnummer über WS.

# Anforderungen

- Bestellungen sollen als CSV-Dateien und über WS an Legacysystem geschickt werden.
- Benachrichtigung per E-Mail über neue Bestellung.
- Jede E-Mail speichern.
- Rückgabe der Bestellnummer über WS.

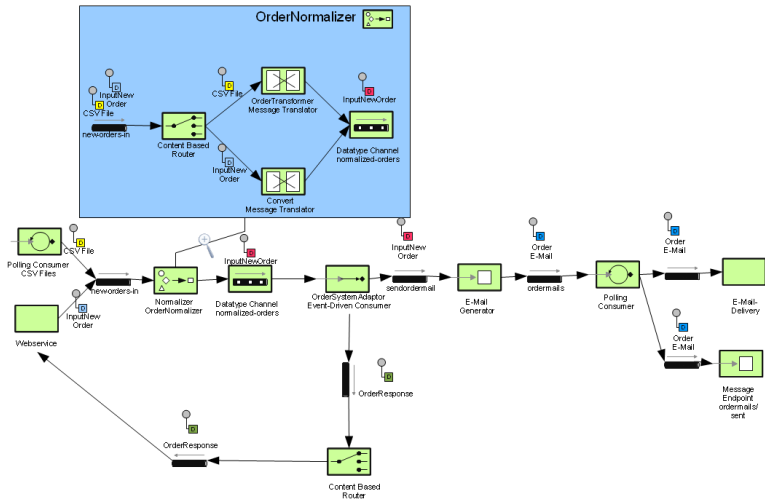
# Anforderungen

- Bestellungen sollen als CSV-Dateien und über WS an Legacysystem geschickt werden.
- Benachrichtigung per E-Mail über neue Bestellung.
- Jede E-Mail speichern.
- Rückgabe der Bestellnummer über WS.

# Anforderungen

- Bestellungen sollen als CSV-Dateien und über WS an Legacysystem geschickt werden.
- Benachrichtigung per E-Mail über neue Bestellung.
- Jede E-Mail speichern.
- Rückgabe der Bestellnummer über WS.

# Beschreibung mittels EIPs



# Gliederung

- 1 Motivation
  - Unternehmensintegration
  - Enterprise Integration Patterns
- 2 Prototyp
  - Szenario
  - **Apache Camel**
  - Implementierung
  - Live Demo

# Allgemeines

- Opensource Framwork für EIPs
- Maven basiert.
- Programmierung in Java DSL oder Konfiguration in XML-Datei
- Plugin-basiert

# Allgemeines

- Opensource Framwork für EIPs
- Maven basiert.
- Programmierung in Java DSL oder Konfiguration in XML-Datei
- Plugin-basiert

# Allgemeines

- Opensource Framwork für EIPs
- Maven basiert.
- Programmierung in Java DSL oder Konfiguration in XML-Datei
- Plugin-basiert

# Allgemeines

- Opensource Framwork für EIPs
- Maven basiert.
- Programmierung in Java DSL oder Konfiguration in XML-Datei
- Plugin-basiert

# Gliederung

- 1 Motivation
  - Unternehmensintegration
  - Enterprise Integration Patterns
- 2 **Prototyp**
  - Szenario
  - Apache Camel
  - **Implementierung**
  - Live Demo

# Routen

- Route als zentrales Element (abstrakte Klasse RouteBuilder)
- Übermittlung von Nachrichten von einem Endpunkt zum nächsten
- Form: from(URI) to (URI)
- CBR mittels choice-Anweisung

# Routen

- Route als zentrales Element (abstrakte Klasse RouteBuilder)
- Übermittlung von Nachrichten von einem Endpunkt zum nächsten
- Form: from(URI) to (URI)
- CBR mittels choice-Anweisung

# Routen

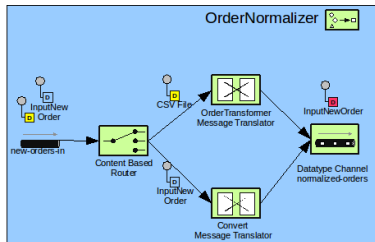
- Route als zentrales Element (abstrakte Klasse RouteBuilder)
- Übermittlung von Nachrichten von einem Endpunkt zum nächsten
- Form: from(URI) to (URI)
- CBR mittels choice-Anweisung

# Routen

- Route als zentrales Element (abstrakte Klasse RouteBuilder)
- Übermittlung von Nachrichten von einem Endpunkt zum nächsten
- Form: from(URI) to (URI)
- CBR mittels choice-Anweisung

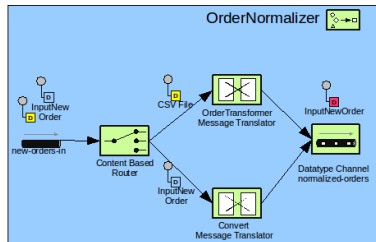
# Datenintegration

- CSV-Consumer
- Transformer Pattern zur Konvertierung von Nutzdaten
- Zusammengefasst als Normalizer
- Einheitliches Datenformat



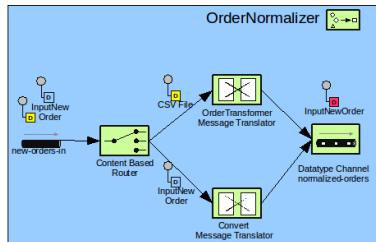
# Datenintegration

- CSV-Consumer
- Transformer Pattern zur Konvertierung von Nutzdaten
- Zusammengefasst als Normalizer
- Einheitliches Datenformat



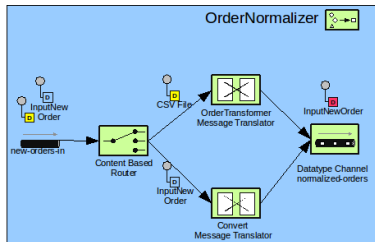
# Datenintegration

- CSV-Consumer
- Transformer Pattern zur Konvertierung von Nutzdaten
- Zusammengefasst als Normalizer
- Einheitliches Datenformat



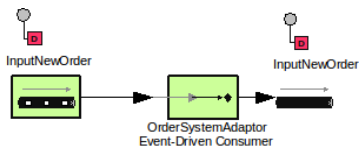
# Datenintegration

- CSV-Consumer
- Transformer Pattern zur Konvertierung von Nutzdaten
- Zusammengefasst als Normalizer
- Einheitliches Datenformat



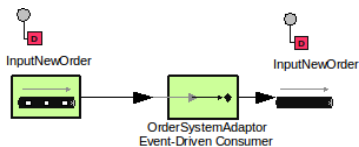
# Funktionsintegration

- Ereignisgesteuerter Konsument
- Interface Processor als Integrationspunkt auf Funktionsebene
- Adapter für Legacy-System



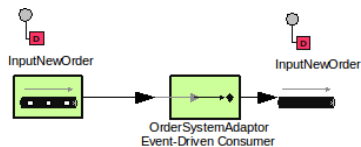
# Funktionsintegration

- Ereignisgesteuerter Konsument
- Interface Processor als Integrationspunkt auf Funktionsebene
- Adapter für Legacy-System



# Funktionsintegration

- Ereignisgesteuerter Konsument
- Interface Processor als Integrationspunkt auf Funktionsebene
- Adapter für Legacy-System



# Gliederung

- 1 Motivation
  - Unternehmensintegration
  - Enterprise Integration Patterns
- 2 Prototyp
  - Szenario
  - Apache Camel
  - Implementierung
  - Live Demo

# Live Demo

- JUnit Test
- Bestellungen via Webservice und CSV-Datei
- Ansicht gespeicherte E-Mails
- Ansicht Bestelldatenbank

# Zusammenfassung

- Apache eignet sich zur Daten- und Funktionsintegration
- Stellt einen leichtgewichtigen ESB dar.
- EIPs unterstützen Design von Integrationslösungen
- Apache Camel unterstützt Implementierung von EIPs
  
- Ausblick
  - Prozessintegration / -management
  - BAM