|  |
| --- |
| EAI  Extract, Transform and Load (ETL) with EAI  20.02.2015  4BHIT  Melanie Göbel, Tobias Perny |

Inhaltsverzeichnis

[Aufgabenstellung 2](#_Toc412718024)

[Resources 2](#_Toc412718025)

[Aufwand und Zeit 3](#_Toc412718026)

[Schätzung und Realität 3](#_Toc412718027)

[Zeitaufzeichnung Göbel 3](#_Toc412718028)

[Zeitaufzeichnung Perny 3](#_Toc412718029)

[Installation und Inbetriebnahme 4](#_Toc412718030)

[Ausführen 4](#_Toc412718031)

[Technologiebeschreibung 5](#_Toc412718032)

[EAI 5](#_Toc412718033)

[EIP 5](#_Toc412718035)

[Insbesondere ETL 5](#_Toc412718036)

[Verwendete Patterns 6](#_Toc412718038)

[File Transfer 6](#_Toc412718039)

[Message Translator 6](#_Toc412718040)

[Message Endpoint 6](#_Toc412718041)

[Quellen 7](#_Toc412718042)

# Aufgabenstellung

"The ETL (Extract, Transform, Load) is a mechanism for loading data into systems or databases using some kind of Data Format from a variety of sources; often files then using Pipes and Filters, Message Translator and possible other Enterprise Integration Patterns.  
So you could query data from various Camel Components such as File, HTTP or JPA, perform multiple patterns such as Splitter or Message Translator then send the messages to some other Component.  
To show how this all fits together, try the ETL Example." [1]  
  
ETL ist ein wichtiger Prozess bei einem Datawarehouse. Zeigen Sie wie Enterprise Integration Patterns [2] dabei eingesetzt werden können (8 Punkte, nur jene, die in dem Beispiel vorkommen). Verwenden Sie dazu das ETL Example [3]. Dokumentieren Sie die Implementierung sowie alle notwendigen Schritte ausführlich in einem Protokoll (8 Punkte). Fügen Sie den verwendeten Code nach den Metaregeln an und geben Sie alles als ZIP-Archiv (Gesamtes Framework mit Anleitung, wie das System gestartet werden kann) ab.

### Resources

[1] Extract Transform Load (ETL); Apache Camel; Online: <http://camel.apache.org/etl.html>; abgerufen 13.02.2015  
[2] Enterprise Integration Patterns; G.Hohpe, B.Woolf; 2003; Online: <http://www.enterpriseintegrationpatterns.com/toc.html>; abgerufen 13.02.2015  
[3] Extract Transform Load (ETL) Example; Apache Camel; Online: <http://camel.apache.org/etl-example.html>; abgerufen 13.02.2015

# Aufwand und Zeit

## Schätzung und Realität

|  |  |  |  |
| --- | --- | --- | --- |
| Arbeit | Geschätzte Zeit (in min) | Zuständig | Benötigte Zeit (in min) |
| Ausführen von EAI | 30 | Göbel, Perny | 30 |
| Technologiebeschreibung | 60 | Göbel | 40 |
| Sourcecodedokumenation | 60 | Perny | 70 |
| Verwendete Patterns | 120 | Göbel | 150 |
| Dokumentation (Zeit, Deckblatt, Formatierung) | 30 | Perny, Göbel | 20 |
| Insgesamt | 300  = 6 Stunden |  | 310 = 6 std. 10min. |

## Zeitaufzeichnung Göbel

|  |  |  |
| --- | --- | --- |
| Arbeit | Datum | Dauer (in min) |
| Ausführen von EAI, Dokumentation | 15.02.2015 | 30 |
| Technologiebeschreibung | 22.02.2015 | 40 |
| Verwendete Patterns | 26.02.2015 | 150 |

## Zeitaufzeichnung Perny

|  |  |  |
| --- | --- | --- |
| Arbeit | Datum | Dauer (in min) |
| Sourcecodedokumentation | 26.02.2015 | 70 |

# Installation und Inbetriebnahme

## Ausführen

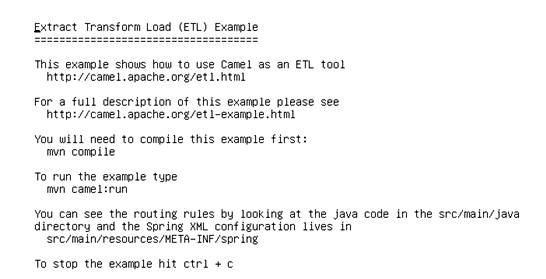
Runterladen von Apache-Camel [1]

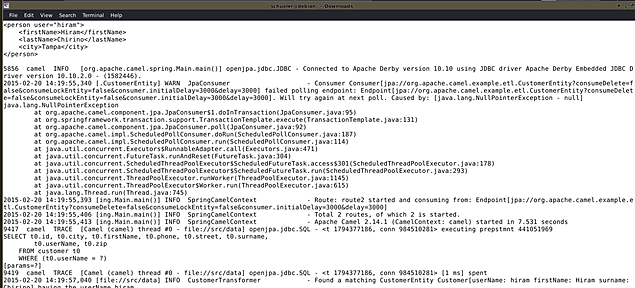
Entpacken mit *tar –xvf*

*Cd examples/camel\_example-etl*

Falls noch nicht vorhanden Maven installieren. *Apt-get install maven*

Lesen des README und ausführen von mvn compile und mvn camel:run



Fehler: Zwischen den Datensätzen eine NullPointerException: 

# Technologiebeschreibung

## EAI

Enterprise Application Integration, kurz EAI, ist eine IT-Infrastruktur zur Kopplung von IT-Systemen z.B: ERP, CRM, SCM und andere betriebswirtschaftliche Systeme. [2]

Das Hauptprinzip von EAI ist eine loose Kopplung und beschreibt nicht wie das Softwareprodukt selbst auszusehen hat. [3]

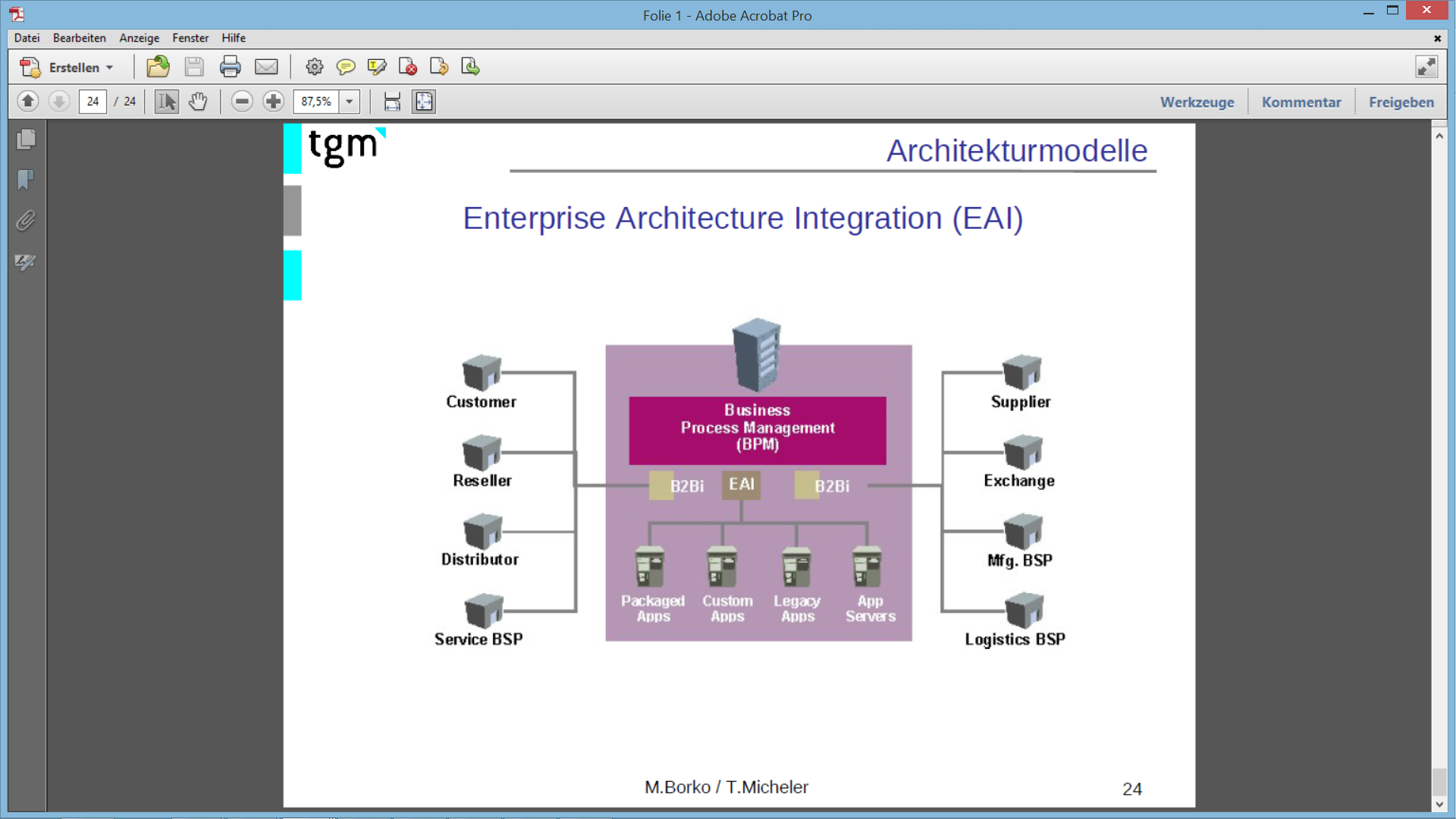


Abb.1: Übersicht EAI Aufbau [4]

## EIP

Durch den Einsatz von EIP (Enterprise Integration Patterns) können Probleme bereits im Vorhinein vermieden werden. Denn die meisten EIP bieten die "best practice" in diesem Bereich und bereits fertig funktionierende Beispiele.

File Transfer: Austausch von Files im standardisierten Format

Message Translator: Baustein der die Übersetzung übernimmt

Content Enricher: Lädt notwendige Daten einfach von eienr anderen Stelle nach

Event Message: Eigener Message Typ

Selective Consumer: Konsumiert genau nur die Daten, die vom Empfänger gebraucht werden.

[3]

## Insbesondere ETL

ETL ist ein Prozess, bei dem Daten aus mehreren auch unterschiedlich strukturierten Datenquellen in einer Zieldatenbank vereinigt werden. ETL bedeutet einzeln:

* **E**xtract: Extraktion der relevanten Daten aus verschiedenen Quellen
* **T**ransformation: Transformation der Daten in das Schema und Format der Zielbank
* **L**oad: Laden der Daten in das Data Warehouse

ETL-Prozess: 
     Extraktion, Transformation, Ladung[5]

Abb.2: ETL Übersicht [6]

# Verwendete Patterns

## File Transfer

Ein File Transfer von XML Files in JPA und POJO findet in dem ETL-Beispiel statt.

## Message Translator

Die Klasse CustomerTransformer dient als Message Translator, da sie die Informationen in ein

Costumer speichert. Siehe Kommentar:

*/\*\*  
 \* A transformation method to convert a person document into a customer entitiy*

*\*/*

## Message Endpoint

Senden von PersonDocument zum JPA-Endpoint in EtlRoutes.java

# Sourcecode-Dokumentation

## PersonDocument.java

In diesem File wird eine Person erstellt.

Eine Person besitzt einen Usernamen, einen Vornamen, einen Nachnamen und eine Stadt.

## Main.java

Diese Klasse ist zum ausführen des Programmes da.

## CustomerEntity.java

In dieser Klasse wird eine Entity-Bean erstellt. Eine Entity-Bean ist eine Komponente von Enterprise JavaBeans. Mit Entity-Beans lassen sich dauerhafte Daten einer Datenbank, zum Beispiel Benutzer oder Adressen, darstellen. In diesem Fall wird ein Customer erstellt. Er besteht aus einem Usernamen, einem Vornamen, einem Nachnamen, einer Straße, Stadt, Postleitzahl und einer Telefonnummer. Dieser wird dann in einer Datenbank abgespeichert.

## CustomerTransformer.java

In dieser Klasse wird ein PersonDocument Objekt zu einem CustomerEntity Ojekt umgewandelt.

## EtlRoutes.java

In dieser Klasse wird eine Route von src/data aus erstellt. Es wird der „noop“-Modus verwendet. Im noop Modus werden bearbeitete Files weder verschoben noch gelöscht. Wenn man das Programm also erneut startet arbeitet es dieselben Files nochmal ab.

# Quellen

[1] Installation Camel Apache

Verfügbar bei: <https://camel.apache.org/camel-2141-release.html> (zuletzt gesehen: 20.02.2015)

[2] Thorsten Horn – „EAI“

Verfügbar bei: <http://www.torsten-horn.de/techdocs/eai.htm> (zuletzt gesehen: 25.02.2015)

[3] Lukas Hörting, Martin Gawlowski – „Enterpise Application Integration“

Verfügbar bei: <https://elearning.tgm.ac.at/mod/resource/view.php?id=32658> (zuletzt gesehen: 25.02.2015)

[4] M.Borko/T.Micheler – „Architekturmodelle“

Verfügbar bei: <https://elearning.tgm.ac.at/mod/resource/view.php?id=29644> (zuletzt gesehen: 25.02.2015)

[5]  [Dr. Klaus Manhart](javascript:void(0)) - „BI-Datenmanagement (Teil 1): Datenaufbereitung durch den ETL-Prozess“

Verfügbar bei: <http://www.tecchannel.de/server/sql/1746250/bi_datenmanagement_teil_1_datenaufbereitung_durch_den_etl_prozess/> (zuletzt gesehen: 25.02.2015)

[6] IT-Wissen - „Extraktion-Transformation-Lade-Prozess“

Verfügbar bei: <http://www.itwissen.info/definition/lexikon/ETL-extract-transfer-load.html> (zuletzt gesehen: 24.02.2015)