# Integration Styles: File Transfer

Das im Example verwendete Integration Style Pattern entspricht einem File Transfer Pattern.  
Es werden XML Dateien konsumiert und in POJO (**P**lain **O**ld **J**ava **O**bject) und mit Hilfe der JPA (**J**ava **P**ersistence **A**PI) transformiert.

Um verschiedene Programme miteinander kommunizieren und Informationen austauschen zu lassen, muss jede dieser Applikationen Dateien erstellen, die von anderen Programmen benötigt werden. Das Transformieren von Dateien in verschiedene Formate wird von ‚Integratoren‘ übernommen. [1]

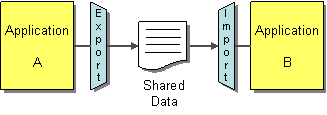


Abbildung 1: Ex- und Import von Daten zwischen zwei verschiedenen Applikationen [1]

# Messaging System: Pipes and Filters

Das Verarbeiten von Daten kann in *Apache Camel* mit Hilfe von verschiedenen, voneinander unabhängigen, miteinander verknüpfbaren Instanzen von Endpunkten realisiert werden. [2]  
Beispielsweise baut der Message Translator auf diesem Prinzip auf.

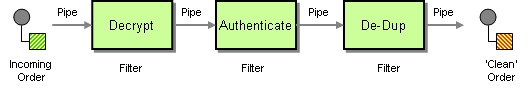


Abbildung 2: Verarbeitungsvorgang einer Nachricht mit Hilfe von Weiterleitungen und Filtern [2]

# Messaging System: Message Translator

Zum Übersetzen von Nachrichten kann in *Apache Camel* der „Message Translator“, ein spezieller Filter eingesetzt werden. Dank ihm können Nachrichten zwischen 2 Filtern bzw. Anwendungen übersetzt und so miteinander kompatibel gemacht werden. [3]  
Im Example findet sich der Message Translator in der **CustomerTransformer** Klasse wieder.

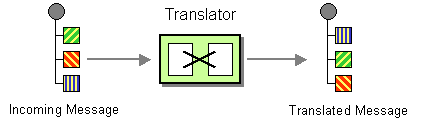


Abbildung 3: Übersetzen einer Nachricht mit Hilfe des Message Translators [3]

# Messaging System: Message Endpoint

In *Apache Camel* können Nachrichten an Clients mit Hilfe eines Message Endpunktes geschickt werden. Anwendungen können mit Hilfe eines solchen Message Endpoints mit einem Messaging Channel (Nachrichtenkanal) verbunden werden. Dies ermöglicht schlussendlich einen Nachrichtenaustausch zwischen Applikationen. [4]  
Angewendet wird dieses Prinzip im Example beispielsweise in der Klasse **EtlRoutes**.

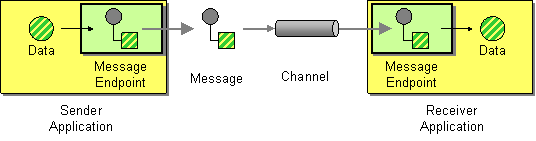


Abbildung 4: Übermitteln einer Nachricht mit Hilfe eines Endpoints über einen Datenkanal [4]

# Technologiebeschreibung

Apache Camel ist ein umfangreiches Framework, welches einem ermöglicht Code zu schreiben, der auf EIP aufbaut. EIPs (Enterprise Integration Pattern) sind Vorlagen bzw. neue Entwurfsmuster, die uns erlauben möglichst effektiv große komponentenbasierte Systeme zu entwerfen. Die Komponenten dieses Systems können dabei sowohl im gleichen Prozess, als auch auf einem gänzlich anderen Rechner laufen. Camel bietet hierfür häufig benötigte Pattern Implementierungen, Debugging Werkzeuge, ein Konfigurationssystem und vieles andere Kleinigkeiten an, die einem Entwickler viel Zeit sparen können. MVC (auch, wenn es nicht wirklich ein Pattern ist) könnte prinzipiell leicht implementiert werden. Mit Hilfe eines Frameworks, welches bereits eine sofort verwendbare Struktur anbietet, werden dem Entwickler bestimmte Aufgaben abgenommen, sodass er sich auf das Wesentliche konzentrieren kann. [5]

# Abbildungsverzeichnis

[Abbildung 1: Ex- und Import von Daten zwischen zwei verschiedenen Applikationen [1] 1](#_Toc412535349)

[Abbildung 2: Verarbeitungsvorgang einer Nachricht mit Hilfe von Weiterleitungen und Filtern [2] 1](#_Toc412535350)

[Abbildung 3: Übersetzen einer Nachricht mit Hilfe des Message Translators [3] 2](#_Toc412535351)

[Abbildung 4: Übermitteln einer Nachricht mit Hilfe eines Endpoints über einen Datenkanal [4] 2](#_Toc412535352)

# Quellen:

[1] Enterprise Integration Patterns – File Transfer, Gregor Hohpe & Bobby Woolf, 18. Jänner 2015,   
*http://www.eaipatterns.com/FileTransferIntegration.html*, zuletzt aufgerufen am 24. Februar 2015

[2] Apache Camel: Pipes and Filters, The Apache Software Foundation, 16. März 2014,   
*http://camel.apache.org/pipes-and-filters.html*, zuletzt aufgerufen am 24. Februar 2015

[3] Apache Camel: Message Translator, The Apache Software Foundation, 16. März 2014,   
*http://camel.apache.org/message-translator.html*, zuletzt aufgerufen am 24. Februar 2015

[4] Apache Camel: Message Endpoint, The Apache Software Foundation, 16. März 2014,   
*http://camel.apache.org/message-endpoint.html*, zuletzt aufgerufen am 24. Februar 2015

[5] What exactly is Apache Camel?, Amr Mostafa, 18. Juli 2012,  
*http://stackoverflow.com/questions/8845186/what-exactly-is-apache-camel*, zuletzt aufgerufen am 24. Februar 2015