

Die Rolle von EDI in einer Service- orientierten Architektur

WAM - Workshop
Hamburg, 12.Nov.2010

Agenda

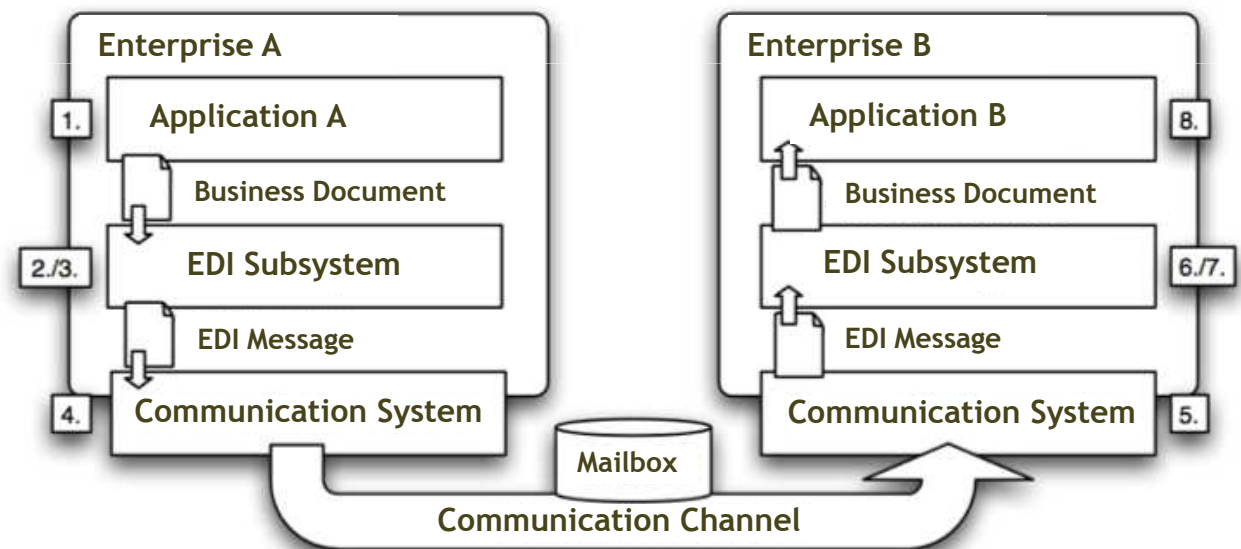
1. Electronic Data Interchange (EDI)
2. Service-Oriented Architecture (SOA) and SOPERA ASP
3. EDI in a SOA
4. tSmooks - SOPERA's choice for EDI integration
5. Outlook

EDI

- **Electronic Data Interchange**
- Exchange of structured business information between business partners
- Standard: ANSI and EDIFACT (grammar, types), with many sub standards, e.g.
 - CEFIX (Chemical Industry)
 - EDIGAS (Gas Industry)
 - EDITEX (Textile Industry)

EDI

- Main focus: Peer-to-Peer, often coupled with contractual aspects
- Communication over Third-Party Value-Added Networks (VANs)
- High effort:
 - Initial development and later change of EDI formats
 - Cost for VAN usage



(Abb. nach B. Safi)

Service-Oriented Architecture

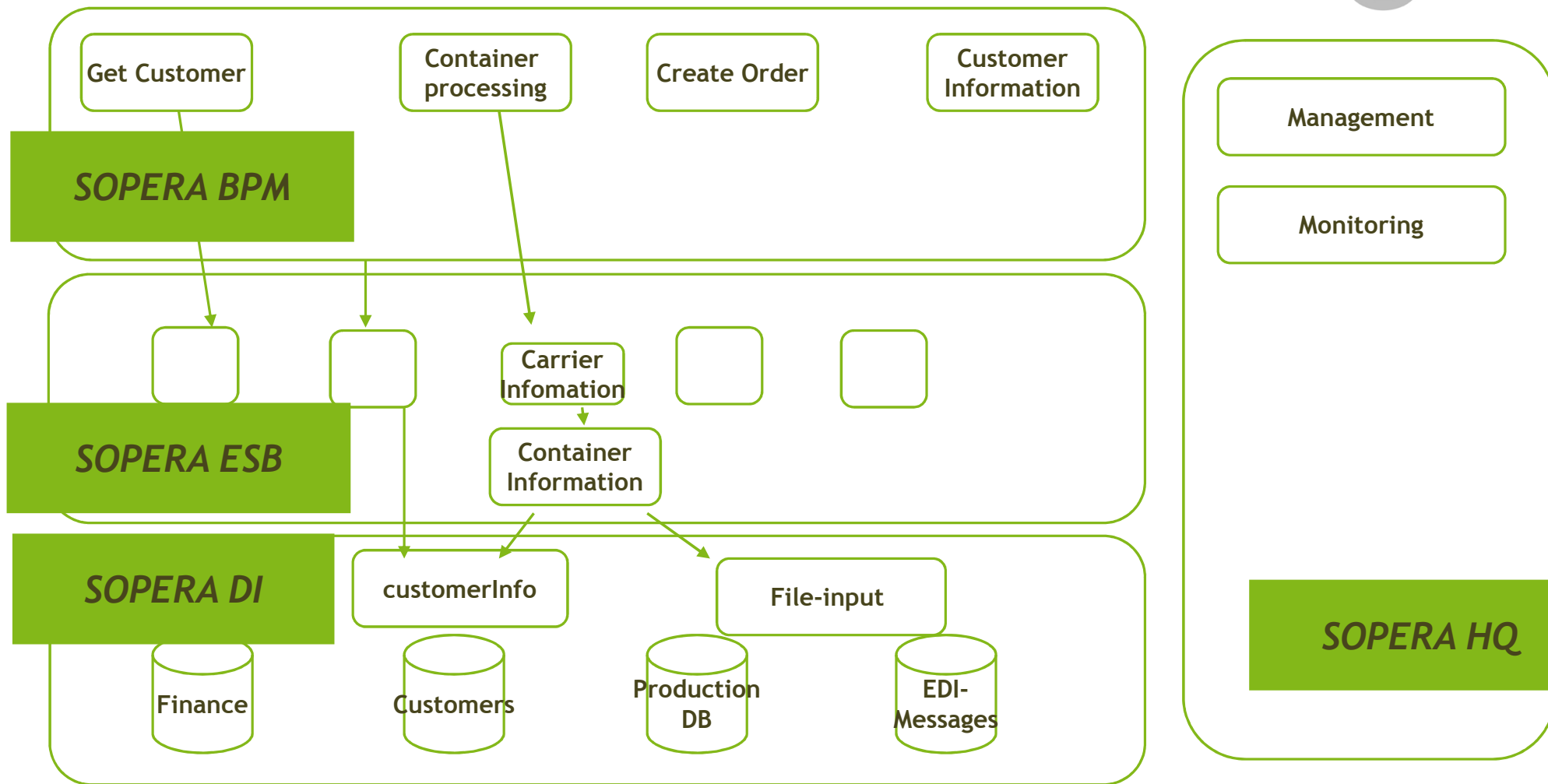
- Domain Services: domain-oriented functional bundles
- SOA Objective:
 - Providing reusable domain functionality
 - Opening up „application silos“
 - Decoupling of 1-to-1 relations between systems
 - Providing the basis for flexible business processes (BPM) on top of services
- Providing an infrastructure (Enterprise Service Bus)
 - Ideally: platform and language neutral (e.g. Web Services, REST)
 - stateless
 - Services as „First Class Citizens“
 - Repository
- SOA as an API for the application landscape



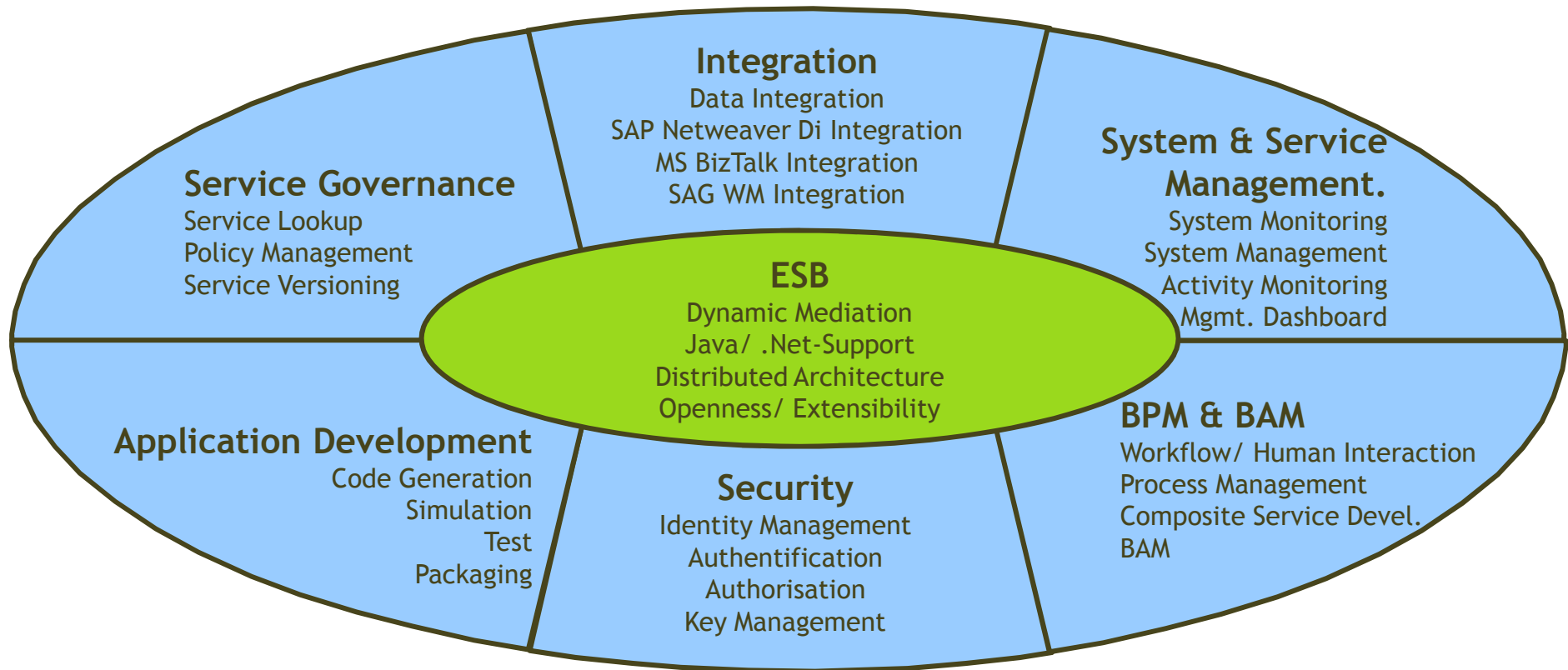
SOPERA



The SOPERA Advanced Services Factory



SOPERA ASF is a complete openSource SOA-Stack



Document Identifier

EDI in a SOA

- EDI is targeted towards inter-company communication
- SOA is (more) targeted towards the internal transactions of a company
- Purely SOA-based EDI would be implemented using Web Services / XML
 - But potential problems are the same as with trad. EDI
 - Specification and changes, legal frame
 - Eventually: VAN or hub or the like
- **Pragmatic approach: Leverage existing EDI formats and contracts for your SOA**
 - Plugging into existing EDI solutions using converters on top of a SOA
 - Implementing EDI interfaces using BPM processes and SOA services
 - Wrapping existing EDI implementations as services, plugging them into the Service Bus

SOPERA DI Studio (4.0.1.r41260) | HHLA | HHLA (Connection: Local)

File Edit View Window Help

100%

Component's ... Design Works...

Repository Navigator

Job EDItoXMLMapping 0.1

Business Models

Job Designs

- CBSSchiffGefundenJob 0.1
- EDIMappingJob 0.1
- EDItoXMLMapping 0.1
- FindSchiffCBS 0.1

Contexts

Code

SQL Templates

Metadata

Documentation

Recycle bin

Designer Code

Job(EDItoXMLMapping 0.1) Contexts(Job EDItoXMLMa) Component Run (Job EDItoXMLMappin) Problems Modules Talend Exchange Scheduler Job Hierarchy

Properties not available.

Palette

Find component...

- Business Intelligence
- Business
- Custom_Code
- tGroovy
- tGroovyFile
- tJava
- Data Quality
- Databases
- ELT
- File
- Input
- Management
- Output
- Internet
- Logs & Errors
- tAssert
- tAssertCatcher
- tChronometexStart
- Misc
- Orchestration
- tFileList
- tFlowToIterate
- tForEach
- Processing
- Fields
- tAggregateRow
- tAggregateSortedRow
- SOPERA
- tSOPERAParticipant
- tSOPERAProviderInput
- tSOPERAProviderOutput
- Smooks
- tSmooks
- tSmooksInput
- System
- Talend MDM
- XML

Diagram description: The diagram shows a data flow process. It starts with a 'tWaitForFile_1' component, followed by an 'Iterate' loop containing a 'tSmooks_1' component. Below this, a sequence of components is shown: 'tFileFullInputRow_1' (row1 (Main)), 'tJavaRow_1' (row2 (Main)), 'tLogRow_1' (row3 (Main)), 'tSOPERAParticipant_1', and 'tFileDelete_1'. There are 'Iterate' labels above the connections between 'tWaitForFile_1' and 'tSmooks_1', and between 'tSOPERAParticipant_1' and 'tFileDelete_1'.

Outline

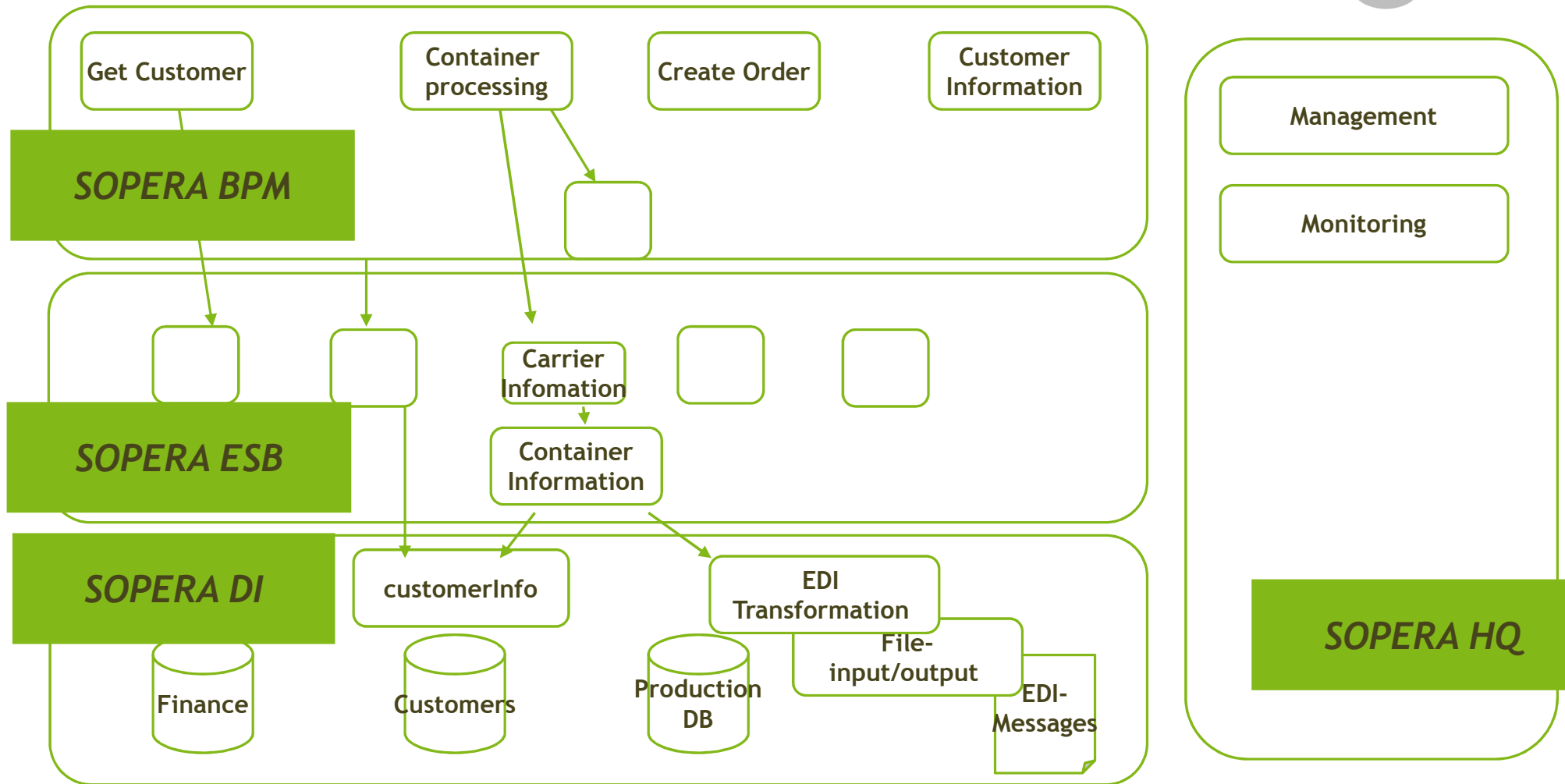
- tFileDelete_1
- tFileFullInputRow_1
- tJavaRow_1
- tLogRow_1
- tSOPERAParticipant_1
- tSmooks_1
- tWaitForFile_1

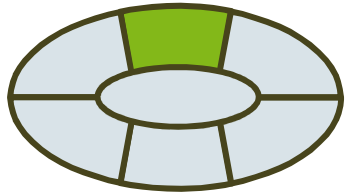


SOPERA



The SOPERA Advanced Services Factory





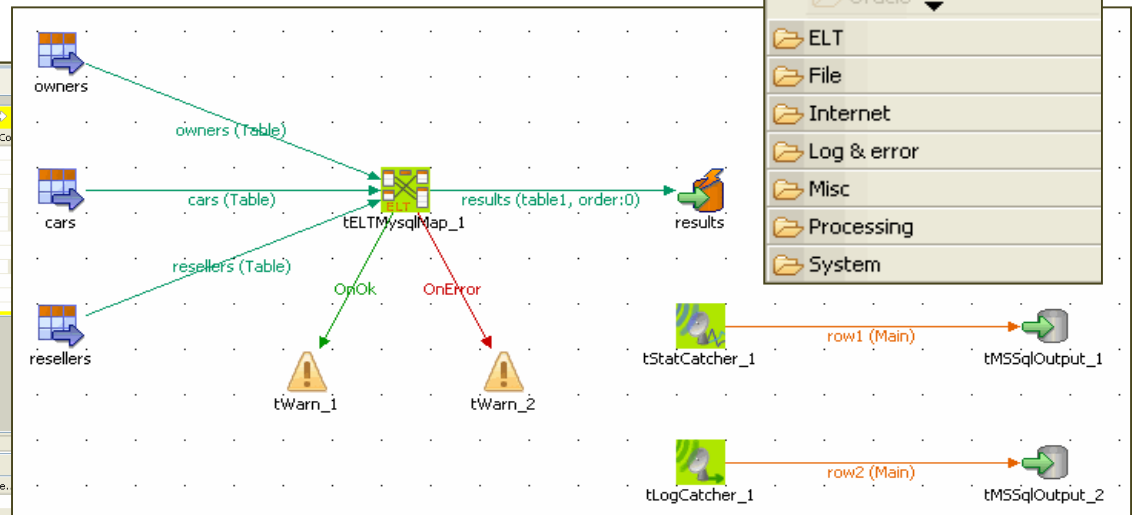
... via graphical development of integration jobs



Key Features

- Graphical development
- Huge set of connectors and adapters
- Repository for configuration, design and execution
- Robust and distributed execution
- Real-time Debugging
- Automatic documentation
- Integration with SOPERA ASF

The screenshot shows the graphical development interface. On the left, the 'Schema editor' displays a table with columns: ID_Owners, Registration, Make, Color, and ID_Reseller. On the right, the 'Expression editor' shows a 'Comprehensive' table with columns: Make, Color, Name, ID_Insurance, Name_Reseller, Address_Reseller, and City. The interface includes various toolbars and a 'Var' section.

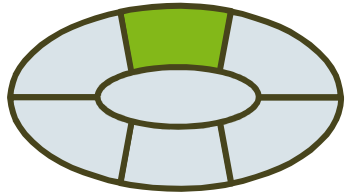


r Service-orientierten Architektur

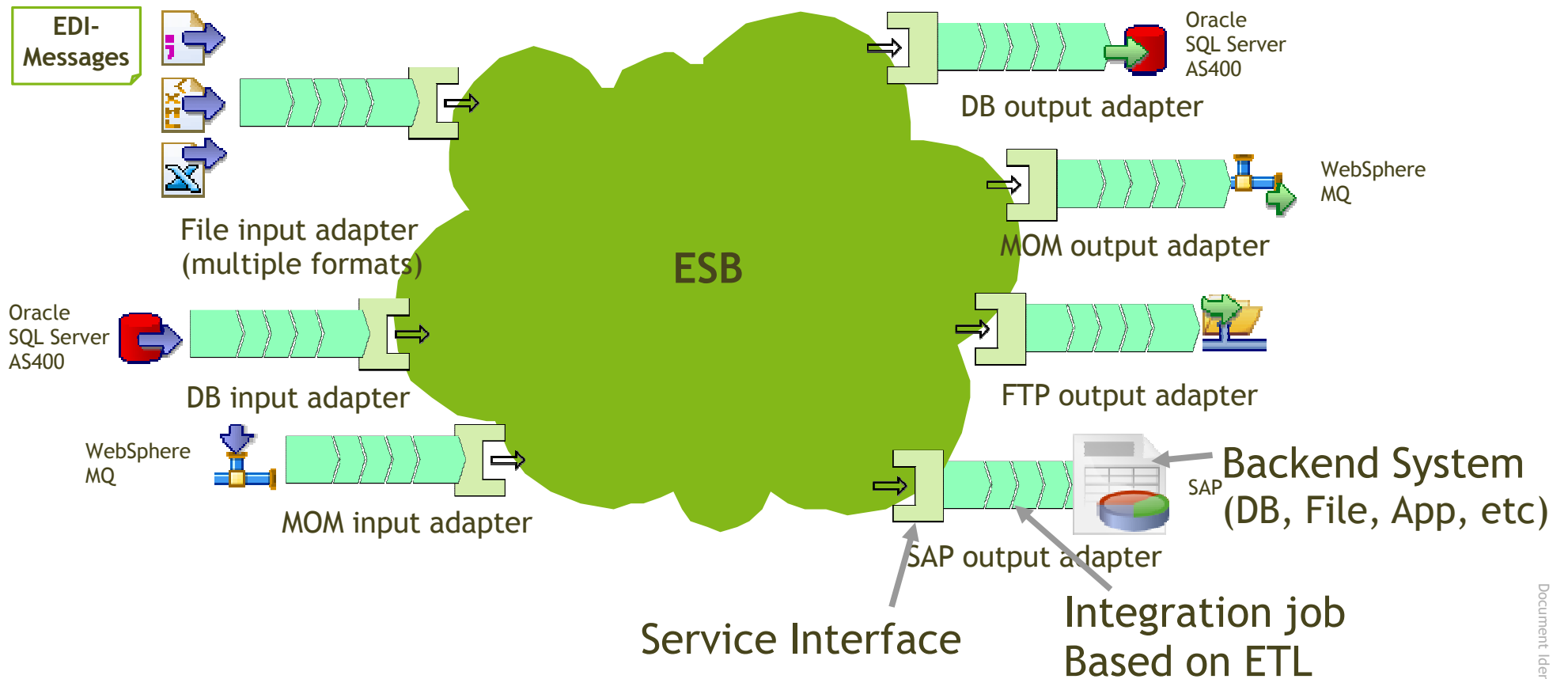
12

WAM-Workshop, 12. November 2010

© SOPERA GmbH, C1 WPS GmbH



SOPERA DI supports service development with backends ...



Document Identifier

Smooks Data Transformation



tSmooks

- Additionally to the already existing data phrasing (tFileInputExcel, tFileInputPositional, tFileInputDelimited and data mapping (tMap, tXSLT, tExtractXMLField, tWriteXMLField, ...) the tSmooksXXX Components allow a flexible and performant data transformation from for even very largen and complex data types:

- XML to XML
- XML to Java
- Java to XML
- Java to Java

- EDI to XML
- EDI to Java
- Java to EDI

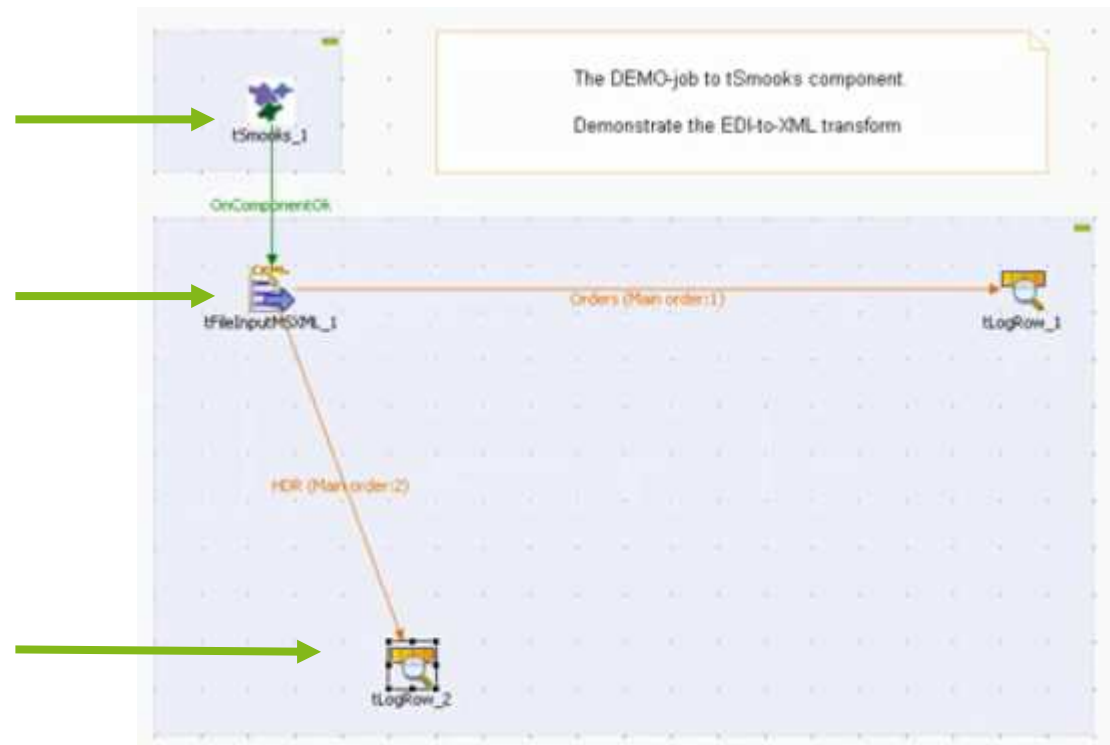
- CSV to XML
- and more ...

- In the context of SOPERA DI we are mostly interested in the
 - Text -> XML -> Text transformations and the specific extensions for the
 - EDI -> XML → EDI transformations

tSmooks: Leverage the full advantage of the Smooks library within SOPERA DI

With the generic tSmooks component it is possible to use any of the Smooks data transformation and write out the result as File which can then be further processed by any of the tFileXX Components of SOPERA DI.

1. Transform data from text (file) to XML using the tSmooks component
2. Read output file(s) provided by the Smooks data transformation into SOPERA DI for further use
3. The tLogRow is just a simple component to allow us to see that the data is available for further processing, e.g. by the tSOPERAParticipant component to send the data to a SOPERA Provider for further processing.



tSmooksInput: allows to get the result of a smooks transformation directly into SOPERA DI Schema definitions - focused on EDI processing



The screenshot shows the SOPERA DI Studio interface. The main workspace displays a job design with components like tLogRow_1, tSmooksInput_1, tLogRow_2, and tLogRow_3. A configuration dialog for tSmooksInput_1 is open, showing a table of columns and their properties.

Column	Key	Type	<input checked="" type="checkbox"/>	N..	Date Patt...	Length	Pre...	D...	Co...
position	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>						
product_id	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>						
quantity	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>						
title	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>						
price	<input type="checkbox"/>	Float	<input checked="" type="checkbox"/>						

The configuration dialog also shows a table of Schema and Loop Path mappings:

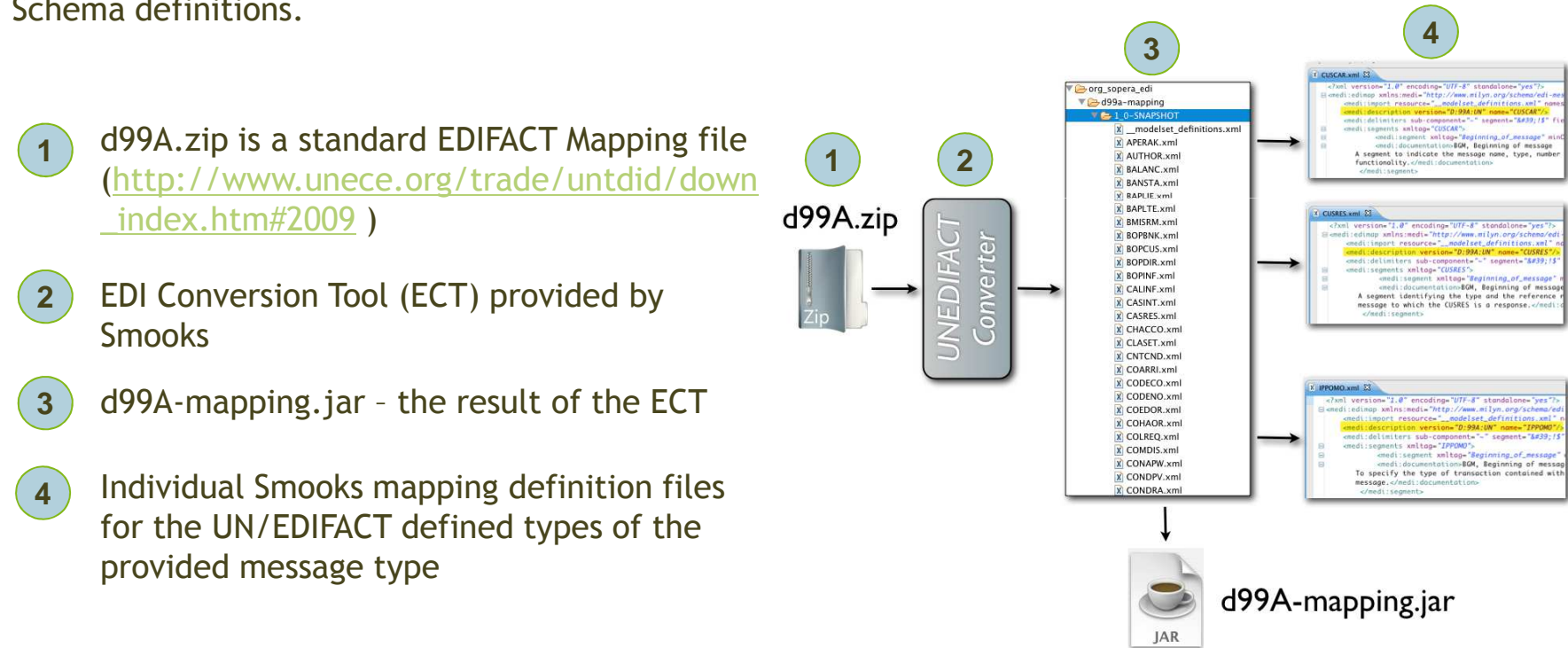
Schema	Loop Path
HDR	"/Order/header"
CUS	"/Order/customer-details"
ORD	"/Order/order-item"
Orders	"/Order/order-item"

tSmooksInput support multiple Outputs from Smmoks to be mapped to SOPERA DI Schemas

Extended EDI Support: UN/EDIFACT Mapping (EDI->XML)

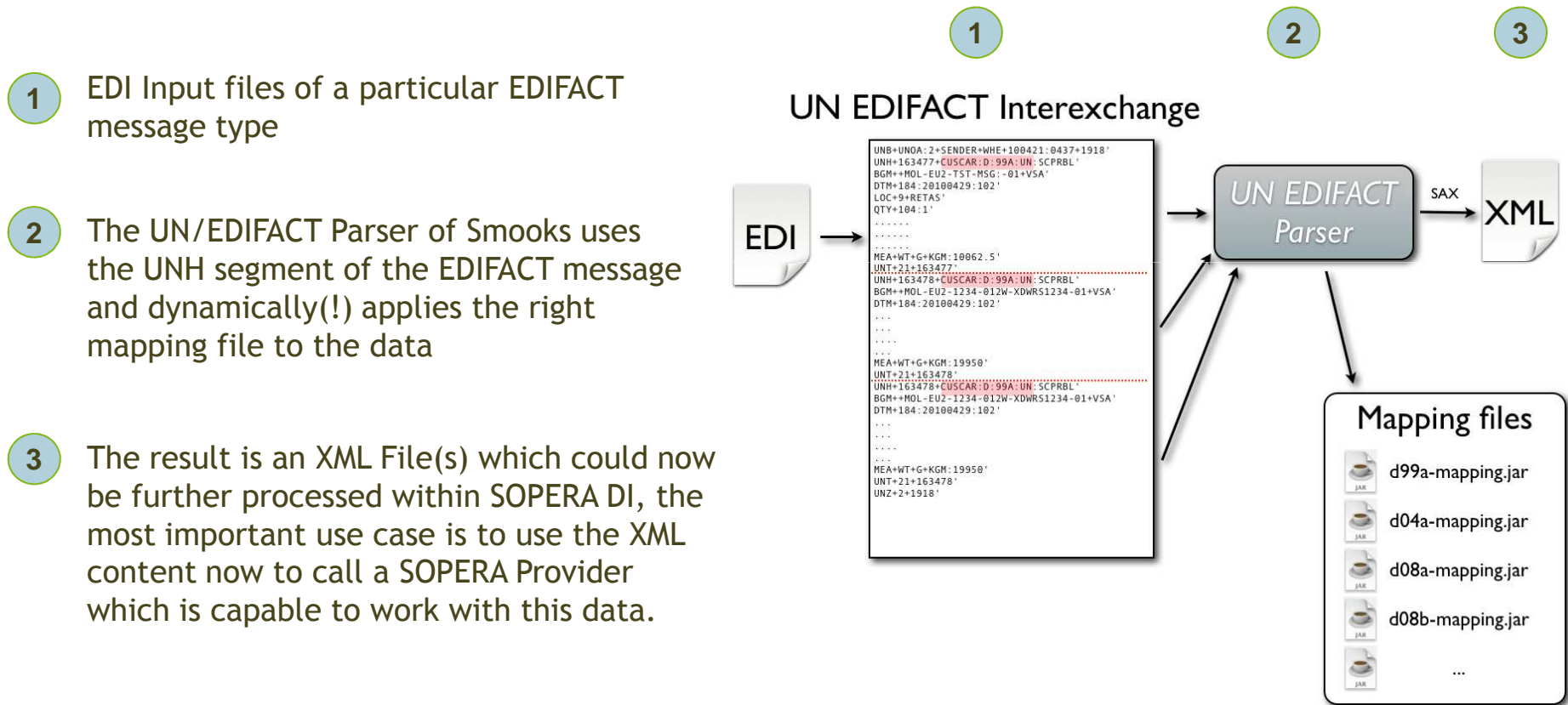
The (manual) definition of the required mapping for EDI are quite time consuming and failure prone. The Smooks team has done a great job to get this important part solved!

The new „EDI Conversion Tool“ (ECT) allows to generate these mapping based on the official UN/EDIFACT Schema definitions.



Extended EDI Support: UN/EDIFACT Processing

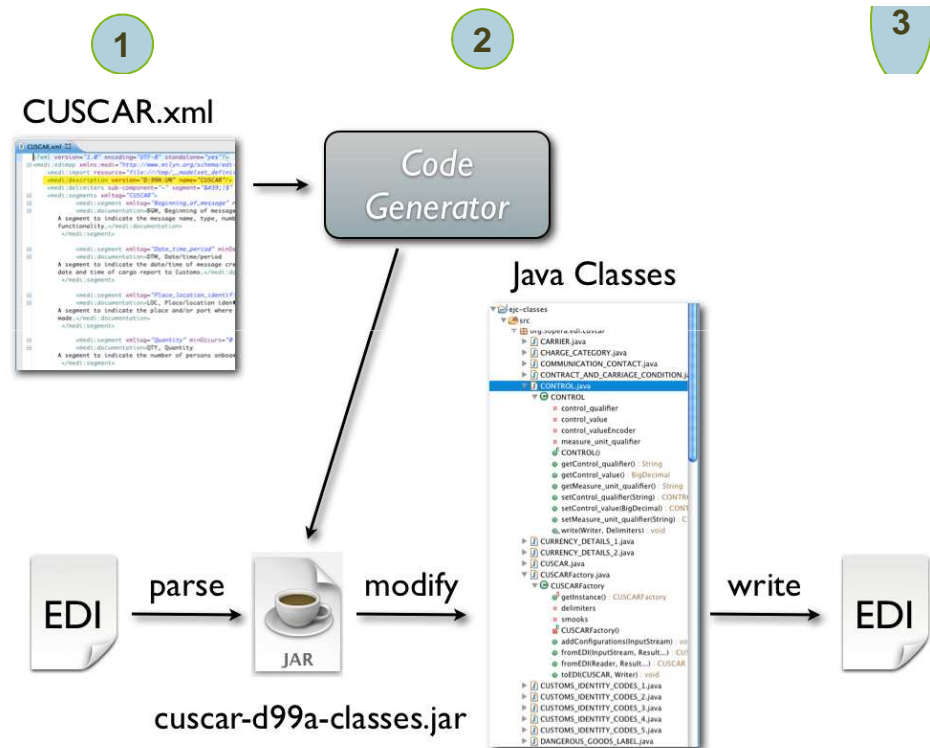
The next after we got the mapping files is to use the mappings to process the EDIFACT messages



Extended EDI Support: UN/EDIFACT Mapping XML-> EDI

As we looked so far to get EDIFACT Data into XML we now like to see how we also could write out EDI Files from data we have in XML available.

- 1 The UN/EDIFACT message definition file could also be used to generate Java Class files to convert XML EDI data back to EDI Text files. (the output case)
- 2 The new Smooks EDI Java Compiler (EJC) generates Java Models which allows to serialize the Java Classes to EDI (similar two what JAXB does for Java to XML)
- 3 The Result is a EDI File which could now be send to your business partner or just be used for interchange between your internal systems.



Outlook & Links



■ Outlook

- we continue to support the Smooks team to get the new features tested and out as a release in Q4/2010
- will more participate in the Smooks project to get the new features out and continue to work on e.g. a Eclipse Based tooling based on the EMF/Ecore model to allow Model2Model transformations e.g. a UN/EDIFACT -> X12 transformation and to allow easy comparison of UN/EDIFACT Versions (and there differences)

■ Links

- www.smooks.org - The official project page
- <http://code.google.com/p/soperadi-smooks> - the stuff SOPERA does currently for the SOPERA DI Integration
- <http://blog.sopera.com/> - we will blog from time to time about the progress we and the Smooks project makes
- http://www.smooks.org/mediawiki/index.php?title=User_Documentation - To find out what the actual Smooks release provides you.