University of Economics, Prague
Faculty of Informatics and Statistics
The Department of Information Technologies

Study program: Applied Informatics
Specialization: Information Systems and Technologies

Data Integration in large enterprises

MASTER THESIS

Student : Bc. Barbora Nagyová
Supervisor : Ing. Jan Kučera, Ph.D.
Opponent : Ing. Dušan Chlapek, Ph.D.

2016

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracovala samostatně a že jsem uvedla všechny použité prameny a literaturu, ze kterých jsem čerpala.

V Praze dne 21. dubna 2016 . . . . . .

Barbora Nagyová

Acknowledgement

I would like to thank my supervisor Ing. Jan Kučera, Ph.D. for his patience, beneficial comments and time, which he has spent on leading my master thesis. I would also like to thank my colleagues, who gave me useful advice and who attended the interviews and enabled me to create the practical part of this thesis.

Abstrakt

Datová integrace je v současné době důležité a komplexní téma, týkající se mnoha společností, protože mít kvalitní a fungující řešení datové integrace může přinést mnoho konkurenčních výhod. Zavádění datové integrace obvykle bývá uskutečněno ve formě projektu, který se snadno může obrátit v neúspěch. Ke snížení rizik a negativního dopadu neúspěšného projektu datové integrace je klíčové mít dobrý projektový management, mít potřebné znalosti datové integrace v rámci týmu a použít vhodnou technologii pro dané řešení. V této práci je navržen framework pro vytvoření kvalitního řešení datové integrace. Framework je vyvinutý na základě současné teorie, dostupných nástrojů datové integrace a zkušeností poskytnutých experty pracující v této oblasti více než 7 let, kteří nasbírali své znalosti v úspěšně fungujícím projektu datové integrace. Tato diplomová práce nezaručuje vyvinutí "správného" řešení datové integrace, avšak poskytuje návod jak se vypořádat s projektem datové integrace pro velké podniky.

Práce je strukturovaná do sedmi kapitol. První kapitola zahrnuje přehled o této práci, především vymezení rozsahu, cíle, předpoklady a očekávanou přidanou hodnotu. Druhá kapitola popisuje datový management a základní teorii datové integrace a následně tyto dva pojmy rozlišuje a specifikuje jejich vzájemný vztah. Třetí kapitola je zaměřená čistě na teorii datové integrace, která by měla být známá každému, kdo se podílí na integračním projektu. Čtvrtá kapitola analyzuje vlastnosti současných řešení datové integrace, které jsou dostupné na trhu a poskytuje seznam a přehled nejběžnějších a nezbytných funkcí. Pátá kapitola vstupuje do praktické části této práce, kde je navržen framework datové integrace, založený na zjištěních a závěrech z předchozích kapitol a rozhovorů s experty v této oblasti. Šestá kapitola aplikuje navržený framework do skutečného a fungujícího (anonymizovaného) řešení datové integrace, vyzdvihuje nesoulad komponent řešení a poskytuje návod, jak se vypořádat s mezerami. Závěrečná kapitola poskytuje shrnutí, osobní názor a výhled do budoucnosti.

Klíčová slova

Datová integrace, podnik velkého rozsahu, framework, řízení dat, oblast datové integrace, nástroje datové integrace, vodopádový model.

Abstract

Data Integration is currently an important and complex topic for many companies, because having a good and working Data Integration solution can bring multiple advantages over competitors. Data Integration is usually being executed in a form of a project, which might easily turn into failure. In order to decrease risks and negative impact of a failed Data Integration project, there needs to be good project management, Data Integration knowledge and the right technology in place. This thesis provides a framework for setting up a good Data Integration solution. The framework is developed based on the current theory, currently available Data Integration tools and opinions provided by experts working in the field for a minimum of 7+ years and have proven their skills with a successful Data Integration project. This thesis does not guarantee the development of the "right" Data Integration solution, but it does provide guidance how to deal with a Data Integration project in a large enterprise.

This thesis is structured into seven chapters. The first chapter brings an overview about this thesis such as scope, goals, assumptions and expected value. The second chapter describes Data Management and basic Data Integration theory in order to distinguish these two topics and to explain the relationship between them. The third chapter is focused purely on Data Integration theory which should be known by everyone who participates in a Data Integration project. The fourth chapter analyses features of the current Data Integration solutions available on the market and provides an overview of the most common and necessary functionalities. Chapter five focuses on the practical part of this thesis, where the Data Integration framework is designed based on findings from previous chapters and interviews with experts in this field. Chapter six then applies the framework to a real working (anonymized) Data Integration solution, highlights the gap between the framework and the solution and provides guidance how to deal with the gaps. Chapter seven provides a resume, personal opinion and outlook.

Keywords

Data Integration, large scale enterprise, framework, data management, Data Integration landscape, Data Integration tools, waterfall model.

Contents

1 Introduction

1.1 Scope of topic and motivation

In today's world, full of innovations, everyone has got in touch with information technologies. Information technologies are expanding from large international enterprises to small enterprises and households so deeply, that most end users tend to become dependent on today's technology. Information has become a key interest to managers, trying to assure the success of their companies. The possibilities of using information technologies are so broad, that they have main impact on enterprise business and accomplishment of enterprise goals.

A common trend has become obvious for both individuals and large companies: People want their gadgets to communicate with each other and to have their data accessible everywhere. The key to this is **Data Integration**. The challenges for Data Integration cover a very wide range: from an individual's phone, which has to know the Facebook contacts to the manager's financial report, which needs to associate key figures to a world map. People prefer to have key information available quickly, comprehensively and without additional manual effort. Data Integration helps to make widespread data centrally available and assure communication across all kinds of systems.

Each company usually uses an entire set of independent applications, which are not always integrated with each other. Over time, the spectrum and complexity of these applications grow and the information dimensions develop even further apart from each other. If no integration is available, this will cause an unwanted effect: As more information becomes available, the user has to deal with more than one system to have a single overview of all needed information. For these reasons, integration has become a key topic for companies and enterprise projects whose key asset is data.

Every task which companies and enterprise projects have tried to accomplish in their history has been made easier through the usage of tools. Even if a Data Integration project starts with no demand for tools in the beginning, it will certainly result (depending on project size) in a fairly large collection of tools for creating and maintaining the final product (Gladden, 2008). Because of wide range of challenges for Data Integration, the tools for solving the problems are still developing as well.

For the reasons mentioned above, it seems necessary to look at Data Integration and the tools supporting it from a theoretical and practical point of view and investigate what decisions and also what kind of tools make Data Integration solutions successful. These reasons have led to the creation of this master thesis. The author of this thesis has been working in one of the largest Data Integration projects worldwide for almost 2 years and

had a chance to observe, influence and participate in the creation and usage of tools, processes and solutions in an evolving and challenging project environment. Therefore a lot of knowledge which was gained during the course of this project served as input for the thesis. This thesis focuses on the high level theoretical needs of the project and put it into the context of a framework.

1.2 Goals, metrics, indicators and definitions of the thesis

Table 1 below describes the list of defined goals, metrics (used measure for goal fulfilment) and indicators (threshold for determining if the metric can be considered as successfully fulfilled).

Table 1: Goals, Metrics and Indicators of this thesis

Name of the goal	Metrics	Indicator
1. Describe the role of Data Integration within the context of Data Management.	Does the description explain the areas of Data Integration and Data Management as well as their relationship?	The areas of Data Integration and Data Management are described in theory with practical examples showing the relationship between each other.
2. Describe the possible approaches to Data Integration in large enterprises.	Are the possible approaches to Data Integration explained from multiple views?	Data Integration is theoretically described using different points of view or dimensions.
3. Analyse the landscape of Data Integration solutions.	Does the analysis reflect the current landscape of Data Integration solutions?	The analysis of the landscape is based on studies from 3 different sources, which are not older than 3 years.
4. Describe the typical functionality of Data Integration tools.	Is the description showing typical functionality on a high level through practical examples from existing Data Integration tools?	Typical functionality of Data Integration tools is described using examples of tools/methods, which are productively used in Integration projects.
5. Propose a framework for evaluation of Data Integration solutions	Does the framework provide recommendations for analysing solutions and does it provide guidance for a typical real world example?	Framework must be based on Data Integration theory, as well as experience and be compatible with requirements of a real world project

6. Analyse relevant Data Integration solutions using the defined framework and propose a Data Integration solution for a large enterprise.	Is the framework applicable to one real world solution and does it have any practical benefits?	Framework must be applied to least in one real world case.

Goal 1 will provide theoretical background on Data Integration and Data Management and describe their relation to each other. Goal 2 will look at different approaches being used in large enterprises to implement Data Integration solutions. Goal 3 will provide a market analysis with current Data Integration trends and solutions. Goal 4 will analyse common sets of functionalities used in tools available in the Data Integration market. Goal 5 will create a Data Integration framework, which provides a structure and content which should help an enterprise to create a Data Integration solution. Goal 6 will evaluate the created framework against an existing solution and describe how the optimal solution would look like.

Definitions

Data Integration solution

This thesis considers a Data Integration solution as the entirety of technological, organisational or human aspects which solve Data Integration related problems in companies.

Data Integration tools

This thesis considers Data Integration tools as software applications, which assist in creating Data Integration solutions.

Framework

According to The Open Group, the definition of a framework is the following:

"A structure for content or process that can be used as a tool to structure thinking, ensuring consistency and completeness." (TOGAF © 1999-2011a)

1.3 Structure of thesis and used methods

The thesis is divided into two parts: A theoretical part, consisting of three chapters and a practical part, consisting of two chapters. In the theoretical part, ***chapter 2*** focuses on the role of Data Integration within the context of Data Management; ***chapter 3*** is dedicated to different approaches to Data Integration – in small as well as in large enterprises. ***Chapter 4*** connects the theoretical with the practical part of this thesis: Typical functionality of Data Integration tools is examined and categorized based on tools currently available on the market.

The practical part, starting with ***chapter 5***, attempts to create a framework – a set of guidelines for creating Data Integration solutions. In order to create the framework, different methods are applied. In the first method, the thesis extracts the most important key points from the theoretical part. Second method is analysis which functional aspects are most commonly used in the Data Integration market. In the third method, key responsible employees working in this project – from technical experts to senior managers - are interviewed with the aim of examining the most common approaches to their Data Integration challenges. The information, gathered through all methods mentioned before, are analysed in detail and used for the creation of the framework. ***Chapter 6*** consequently puts the framework to a test. An anonymised Data Integration solution is tested using the framework and the identified gaps are listed and verified by peer reviewers. A proposal for closing the gaps is given.

Chapter 7 provides a resume of all previous chapters as well as a personal outlook on the topic.

1.4 Restrictions of the thesis

This thesis underlies a few restrictions. The real world example (see chapter *6 Applying the framework to a Data Integration Project (*DIP)), on which this thesis is based, is a living and breathing project environment. Its course may change at any time, possibly invalidating parts of the work already done. The framework creation will include qualitative feedback which will be based on personal opinions. These opinions may not always reflect the opinion of the author or the needs of the framework. Lastly, it should be stated, that some project information is inaccessible, due to confidentiality.

1.5 Outcomes of thesis and expected added value

The main aim of this thesis is the creation of a framework for Data Integration in large enterprises. For the creation of the framework, one of the main outputs is a list of functionalities provided by Data Integration tools, which is created based on analysis of multiple available tools from the market. In order to test the final framework, it is applied to the project which the author has participated in, in order to analyse whether there are any gaps and if so, recommendations for their closure are created.

Summarizing, the outcome and expected added value of this thesis should be the following:

- A Data Integration framework that gives large enterprises a guideline for implementing Data Integration solutions

- A list of common functionality currently available in Data Integration tools, that provides an overview of available features on the market
- A list of identified gaps and a proposed process for their closure, that represents the practical benefit for the project "DIP"

1.6 Literature review

This chapter describes how the search is executed, which key words and which sources are used in order to compare this Master thesis with other available theses and scientific papers, which are related to the same topic.

First, necessary key words related to the topic of this Master thesis were specified in English as well as in Czech language. The list of the examined key words is the following:

- Data Integration in Large Enterprises / Datová integrace ve velkých podnicích
- Data Integration / Datová integrace
- Large Scale Data Integration / Datová integrace velkého rozsahu
- Big Data Integration / Integrace velkých dat

Second, sources for search are specified as following:

- www.theses.cz – a database of all university (mostly Czech ones) qualification works written by students
- www.scholar.google.com – this webpage contains scientific articles which are available to public
- www.vse.cz/zdroje; ProQuest – a commercial database which is available to students of University of Economics in order to help them to find reliable sources for their studies
- www.forrester.com – a company, which provides a research about business and technological topics

In the following table, there is an overview of the search results. In each field, there are numbers of found results for particular key word and source. First number is a result for the English key word, the number after the slash is the result for the Czech key word. Two searches are executed for each key word and source. The first search is done without any advanced search (in the table, the results are highlighted in green colour) and the second search is done with usage of advanced search only for finding results, where the key words are an exact phrase in the title of the document (highlighted by blue colour).

Table 2: Overview of key words and sources with numbers of findings

Source / Key words	www.theses.cz	www.scholar.google.com	www.vse.cz/zdroje - ProQuest database	Forrester research
Data Integration in Large Enterprises / Datová integrace ve velkých podnicích	Std.: 832 (EN) / 976 (CZ) Adv.: 0 (EN) / 0 (CZ)	Std.: 1 200 000 (EN) / 3 310 (CZ) Adv.: 6 (EN) / 0 (CZ)	Std.: 320 711 (EN) / 0 (CZ) Adv.: 3 (EN) / 0 (CZ)	Std.: 3710 (EN) / 0 (CZ) Adv.: 0 (EN) / 0 (CZ)
Data Integration / Datová integrace	Std.: 855 (EN) / 938 (CZ) Adv.: 234 (EN) / 110 (CZ)	Std.: 7 110 000 (EN) / 8 230 (CZ) Adv.: 273 000 (EN) / 44 (CZ)	Std.: 1 683 927 (EN) / 6 (CZ) Adv.: 8 035 (EN) / 0 (CZ)	Std.: 7873 (EN) / 0 (CZ) Adv.: 24 (EN) / 0 (CZ)
Large Scale Data Integration / Datová integrace velkého rozsahu	Std.: 948 (EN) / 981 (CZ) Adv.: 0 (EN) / 0 (CZ)	Std.: 3 960 000 (EN) / 5 210 (CZ) Adv.: 832 (EN) / 0 (CZ)	Std.: 470 958 (EN) / 1 (CZ) Adv.: 266 (EN) / 0 (CZ)	Std.: 1897 (EN) / 0 (CZ) Adv.: 0 (EN) / 0 (CZ)
Big Data Integration / Integrace velkých dat	Std.: 956 (EN) / 961 (CZ) Adv.: 0 (EN) / 1 (CZ)	Std.: 1 870 000 (EN) / 19 000 (CZ) Adv.: 503 (EN) / 0 (CZ)	Std.: 403 532 (EN) / 38 (CZ) Adv.: 248 (EN) / 0 (CZ)	Std.: 2769 (EN) / 0 (CZ) Adv.: 5 (EN) / 0 (CZ)

From *Table 2* it is obvious, that some of the chosen key words are too common, that many results were found and hardly all of them could be investigated by the author of this thesis. However, while searching the first few pages with found results (ordered by relevance), it is obvious that none of the articles and theses are related to the same topic as this Master thesis. The reason could be, that the topic of Data Integration is often being discussed, but not in the context of large enterprise solutions. While doing the search, interesting facts appeared:

- Most of the results were found in English. In Czech language, there were much less results found (in some of the sources, there were no findings at all), for that reason this thesis will primarily use foreign sources in English language.
- Multiple sources related to theory about Data Integration contained slightly different classifications, for example Voříšek, et al. (2015) and Gála, et al. (2006) specifies different levels of integration, however both of the classifications are correct. For that reason the author of this thesis uses both sources, which describes Integration theory but does not compare between them.
- The topic of this thesis - Data Integration in Large Enterprises – did not find any suitable results while doing the search for these key words, which means that this thesis can bring new information about how Data Integration in Large enterprises works.

- Many sources from past few years, containing Data Integration key words, are describing concept of "Big Data Integration", for example studies from Kraska (2013), or Dong and Srivastava (2013), which describe topic of Big Data Integration in today's world. It is obvious, that Big Data Integration belongs to highly discussed phrases and for this reason, this topic will be considered as one of the trends in chapter *3.5.2 Current trends in Data Integration*.

Important sources, which are used in this thesis, are following:

- **Dama DMBOK Framework** for specification of Data Management and evaluating Data Integration in its context (Cupoli, Earley and Henderson, 2014)
- Book **Enterprise Design Patterns** for classification of Data Integration into different design styles (Hohpe and Woolf, 2004)
- Books **Podniková Informatika** [Enterprise Informatics] (Gála, et al., 2006) and **Tvorba Informačních systémů** [Information System Creation] **(**Voříšek, et al, 2015) for specification of integration architecture and levels
- **Documentations** made by Vendors (IBM, Informatica, SAP and Talend) for analysing the current Data Integration market (see chapter *4 Features of Data Integration tools*)
- Thesis from **Tomáš Dohnal (Stav trhu v oblasti Enterprise Information Integration** [Current market status in the area of Enterprise Information Integration]), (Dohnal, 2011) and **Jan Růžička (Integrace entity „cash event" v systémech Sugar CRM a Adempiere** [Integration of entity cash event in systems Sugar CRM and Adempiere]**),** (Růžička, 2014) for helping to gather ideas for the theory part of this Master thesis.

2 Data Management and Integration

The following chapter introduces Data Integration, presents its differentiation from other forms of integration and describes the role of Data Integration within the context of Data Management.

2.1 What is Data Management?

Data Management can be seen as the summary of tasks, designs, processes, roles and implementations that have the purpose of governing data as an asset within a company (Cupoli, Earley and Henderson, 2014, p.5). Following the Data Management Association's (DAMA) **D**ata **M**anagement **B**ody **o**f **K**nowledge (DMBOK) – which represents a major authority in collecting and providing recognized sources for the topic in form of a "Guide" and a Framework – Data Management covers a wide range of aspects, from high level management decisions (like Data Governance decisions) to low level technical implementations (like Database Operations Management).

The DAMA-DMBOK2 Framework suggests 11 main functional areas for Data Management, which cover all main challenges of Data Management as following:

- Data Governance – for controlling the entire data management
- Data Architecture – for designing the global data structure
- Data Modelling & Design – for low level design and maintenance of data
- Data Storage & Operations – for the technical foundation of storing the data
- Data Security – for ensuring proper data protection
- Data Integration & Interoperability – for combining data across heterogeneous sources
- Documents & Content – for operating data across heterogeneous sources as basis for integration
- Reference & Master Data – for standardization and documenting of data content
- Data Warehousing & Business Intelligence – for analytical reporting
- Meta Data – for managing data about data
- Data Quality – for centralized government of data quality

To provide an overview of all 11 main functional areas in more detail, *Figure 1* below is showing all main areas and their descriptions. This picture has been taken directly from the DAMA-DMBOK2 Framework and has been enhanced by quoting the descriptions of the 11 functional areas directly from the same framework.

Figure 1: DAMA-DMBOK2 Framework (Source: Cupoli, Earley and Henderson, 2014, p. 9-10, modified by author)

2.2 What is Integration of Information Technologies?

Integration of Information Technologies can be seen as a business driven effort to combine heterogeneous sources into one operational whole (Rouse, 2015). This chapter describes

the integration levels according to Voříšek, et al (2015) and will introduce data as an aspect of integration.

Voříšek, et al (2015, p. 162) defines a schema for Information Systems (IS) and Information and Communication Technology (ICT) as follows:

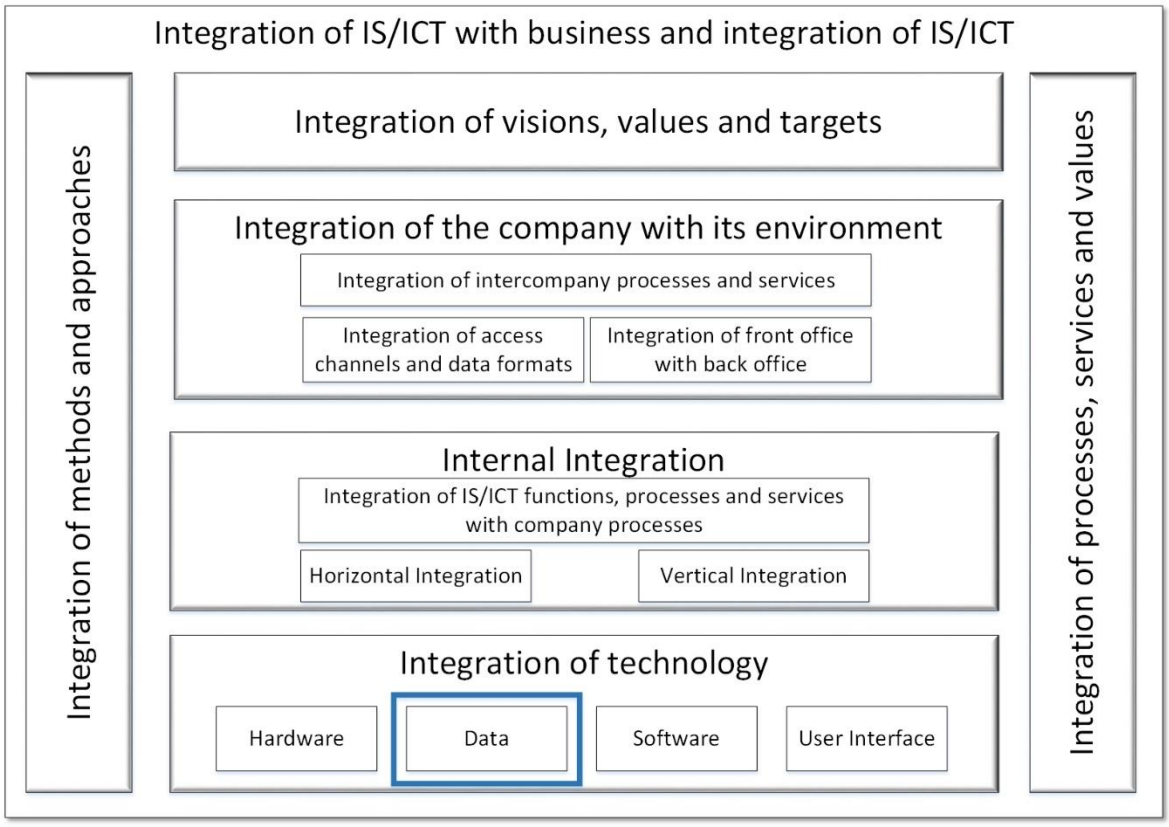

Figure 2: Integration model (Source: Voříšek, et al, 2015, p. 162, translated and modified by author)

The integration model consists of 5 levels, going from highest to lowest level as follows (Voříšek, et al, 2015, p. 162-163):

- **Integration of visions, values and targets** – the aim of this level is to ensure, that all top managers will have consistent opinion on priorities in IS/ICT in the company.
- **Integration of company with its environment** – the aim of this level is to adapt a company and its information system to the communication with its external partners (supplier, customers, banks) and to disseminate company information as well as gathering necessary information with its economic environment in general.
- **Internal integration** – this level enables to make processes in companies more efficient and effective through considering the connection between processes or their consistency with processes in information systems.

- **Integration of technology** – this level is about integration of hardware components, software components, integration of data (which is described in detail in the following paragraphs) and user interface integration.
- **Integration of methods and approaches** – this integration helps to align all methods, techniques and tools which are specified in previous levels in order to create one unite methodology from them.

As this thesis is primarily focused on Data Integration, the following chapter describes Data Integration as one component of technological integration in detail (see blue frame in *Figure 2*).

2.3 What is Data Integration?

When looking at the Integration model by (Voříšek, et al, 2015, p. 162-163) in chapter *2.2 What is Integration of Information Technologies?*, Data Integration is one aspect of technological integration. Data Integration treats the data of a company purely as an asset which needs to be connected together ("integrated"). Instead of having multiple sources providing data, an integrated data solution aims at unifying the data to achieve one common view (Lenzerini, 2002, p. 233). This aim can be reached using different methods and techniques, for example by merging two databases into one or by installing a regular file transfer between the two systems (Schwinn, Schelp, 2005, p. 471-482). The exact methods and techniques are discussed in chapter *3.2 Integration styles* and its subchapters.

The term "Systems" is considered very broad in this thesis when talking about Data Integration and can be everything that contains data: Be it just a couple of files or databases, or entire clusters of applications. Whatever two data sets need to be integrated with each other: They may or may not share the same idea of the syntactic structure of data. Both data sets may have different structures, relations, granularities and purposes. On top of that, another effect called "Semantic Dissonance" can appear naturally: Every system has its own unique model, view or definition of the world, which means that objects which exist in both systems may seem to be exactly the same (meaning they could share the same structure and content), but are actually different from each other, because for example they are interpreted or processed differently (Trowbridge et.al., 2004, p. 57).

The following chapters will examine the details of Data Integration, what it means in the context of Data Management (see chapter *2.4 The role of Data Integration in the context of Data Management*), why companies need it (see chapter *2.5 Necessity of Data Integration*), where it came from (see chapter *2.6 History of Data Integration*) and what the advantages and disadvantages are (see chapter *2.7 Advantages and disadvantages of Integration* and subchapters).

2.4 The role of Data Integration in the context of Data Management

As described in chapter *2.1 What is Data Management?*, the DAMA DMBOK Framework considers Data Integration to be a vital part of Data Management. However, this has not always been the case. In fact, the DAMA DMBOK Framework did not consider Data Integration as part of their framework before they revised it in 2013 with Version 2. (Cupoli, Earley and Henderson, 2014, p. 9)

This indicates, that it is a mistake to underestimate the importance of integration when dealing with Data Management (as the DAMA decided to consider it in their latest version).

Interestingly though, several sources are suggesting that Data Integration by itself cannot stand without the Data Management components defined by DAMA. Data Integration requires a certain set of Data Management activities to be carried out at a bare minimum, in order to succeed. These areas are (Fetsel et al., © 2001, p. 55-56):

- **Data Governance,** because Data Integration requires high level strategic decisions to be consistently carried out, e.g. which system is the leading master data system, where will the integration be carried out physically (source, target or middleware system) and which team takes care of master data management.
- **Data Architecture,** because every integrated data solution requires a solid understanding of the (minimum) two systems which need their data to be integrated with each other. The data architect could also develop a standardized scheme ("mediator") which serves as the basic exchange format of the data.
- **Data Security**, as with integration being implemented, more than one system will have access to the data or data will be sent across multiple systems in form of messages. This raises the question how to secure the communication.
- **Reference & Master Data**, as usually two connecting data sets come with their own representation, classification, model and code-lists for data, a proper master data management is required. This master data management has to assure that in the communication between the two different data sets, a common understanding is shared, either by defining one system as the master or by offering a translation or classification service when sending messages between the systems.
- **Data Storage & Operation,** as Data Integration will always require low level operations directly on the database to be defined, set up and monitored.

2.5 Necessity of Data Integration

With the expansion of information technologies and the rising importance of the Internet, companies need to run a high amount of applications in order to make the business successful. Those applications can range from standard software to customized or even custom applications, which can be developed by company employees themselves or also third parties, using different platforms, having widely different architectures and being deployed to different geographical locations as well (Hohpe and Woolf, 2004, p. 1).

Those applications use a vast amount of data which exponentially increases during the applications lifecycle. Therefore, manual integration made by humans is not possible anymore and sooner or later all data assets in companies need to be maintained on higher level. Moreover, for application and data warehouses support, the data should be integrated from the beginning, because the older a company gets the larger and more complex the data handling becomes (McKendrick, 2014, p. 6).

As mentioned earlier, one possible way how to deal with large and complex data maintenance is Data Integration. However, this solution does not only have advantages, but also disadvantages and companies need to consider, if the integration will bring benefits or not. For further information, please see chapter *2.7 Advantages and disadvantages of Integration*.

This leads to the question, what makes a Data Integration solution a good solution and on what aspects should a company focus, when aiming at a good solution. Hohpe and Woolf (2004, p. 39-41) mention the following criteria:

- **Need for integration** – in case the company is able to utilize a single application that can cover all functionality, an integration solution will not be needed. However, due to the fact that companies usually do not have such an application, this is highly unlikely.

- **Application coupling** – integrated applications should not cause higher dependency on each other and all integrated applications should still be able to work independently from each other. In case one of the applications breaks or changes, it should not break the integration or stop other applications from running.

- **Intrusiveness** – the main focus area for changes should lay outside of the system which is being integrated. Changes to such a system should be avoided, unless this leads to missing integration functionality.

- **Technology selection** – used integration technology should not be too expensive and should generally have a low demand of adopting additional hardware and software.

- **Data format** – each application might use a different data format and when they communicate with each other, the company has to decide about a unified data for-

mat or specify a mediator, who will translate data going from one application to another.

- **Data timeliness** – the integration solution should not delay the traffic of the data. If the data is delivered with delay, applications run into the risk of not being synchronized.
- **Data or functionality** – the company should also decide between Data Integration and usage of the integration solution to wrap application functionality into a shared asset. Sharing functionality between applications brings higher abstraction but also raises the complexity of the integration.
- **Remote Communication** – a good integration solution can work completely asynchronously, however this will make the implementation of the solution more complex.
- **Network Reliability** – as mentioned in Application Coupling, the applications should be able to work independently, also when the integrated applications are not reachable or the network is slow.

Data Integration plays a key role in combining data assets in companies and making them widely available to users of the data (for example employees, management or customers). Data Integration serves as a catalyst to make data and applications "smarter" and serves as a business enabler (see also chapter *2.7.1 Advantages*).

2.6 History of Data Integration

For most companies, the necessity of integrating data appeared shortly after the formation of database technologies in the 1960s. Companies had to deal with a constantly raising amount of data and storing such data has always been a fundamental challenge to IT. The newly born database technologies led to the creation of many supportive business applications, which helped dealing with the raising amount of data (Ziegler and Dittrich, p. 7). Until the 1970s, most of the applications working with data were isolated from each other, which unavoidably led to unwanted effects in the IT business, like higher costs for software development, increasing manual work with data (since almost no technological support for integration existed) and higher probability of making mistakes while executing the mentioned manual maintenance (Bruckner, 2012, p. 57).

The first "real" integration projects were executed between the 1970s and 1980s, when companies started to discover the benefits of Data Integration: Since it involves automation of storing unified information on a very granular level, it avoided the mistakes in manual work and was generally faster than human-based integration. Therefore it saved costs and increased the quality for any enterprise using it. Companies were now able to connect the

required data between their most important applications, like salary information of their employees with accounting data from another specific department or customer master data with customer orders, which could even be transmitted to other countries (Bruckner, 2012, p. 58).

The striking success of these companies led to a raise in demand for higher integration for more complex and integrated support of operations. In the 1990s, the first ERP (Enterprise resource planning) systems were created as a reaction to the rising demand, mainly for areas like logistics, finances and human resources (Bruckner, 2012, p. 58).

From the end of 1990s, companies became more interested in supply chain support and communication with their partners and customers. At this time, new applications (like CRM – Customer relationship management and SCM – Supply chain management) were created. The amount of used applications as well as data inside of these applications started to grow rapidly and they needed to be integrated even across company borders towards other companies, e.g. partners or customers. At the same time, the security of the company data became more important and had to be ensured. (Bruckner, 2012, p. 58)

From 2005, the era of Cloud computing started. According to the NIST (The National Institute of Standards and Technology), the definition of Cloud computing is following: *"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."* (Mell and Grance, 2011, p. 2)

With the emergence of cloud computing, integration has faced a new challenge, as usually a cloud service and an integrated software solution exclude each other, as cloud-based software is harder to integrate into the internal business application landscape of the company. Most companies avoid to move to cloud-based solutions, as many of their applications require data exchange (Oracle, 2015, p. 6).

Due to these circumstances, new cloud-based integration solutions have evolved. These solutions help overcoming the most technical challenges, like integrating a cloud-based application with an on-premise (company internal) environment (which is also called "hybrid integration"), but also consider building integration services that emphasize on the data management perspective:

- Integration Platform as a Service (iPaaS), which means that the cloud service will provide an environment for the easy definition, setup and control of an infrastructure that allows for a uniformed system integration inside and outside of the company (also between cloud and on-premise solutions), (Liaison, © 2015b).
- Data Platform as a Service (dPaaS), which means that the cloud service will provide full integration functionality and therefore serve as an integration middleware, offering tools for creating client-independent maps and flows, as well as controlling

and monitoring message flows between the integrated applications (Liaison, © 2015a).

2.7 Advantages and disadvantages of Integration

Leaving the theoretical perspective aside, when it comes to the implementation of an actual integration solution, one will hardly find a commercial off-the-shelf product that will suit all needs (Boehm and Abts, 1999, p 135); hence integration implementation will mostly be connected with a complex and long lasting project. Considering this, it needs to be stated that there are not only advantages but also some disadvantages regarding integration. The most important advantages and disadvantages for companies are mentioned in this chapter.

2.7.1 Advantages

Table 3 shows an overview of Data Integration advantages, which will be described in detail below.

Table 3: List of common Data Integration advantages (Source: Author)

List of Data Integration advantages	Source
1. Common view of data	Schwinn, Schelp, 2005, p. 473, 475-480
2. Data redundancy decrease	Schwinn, Schelp, 2005, p. 473, 475-480
3. Synchronization control	Schwinn, Schelp, 2005, p. 473, 475-480
4. Overall data quality increase	Schwinn, Schelp, 2005, p. 473, 475-480
5. Creation of “Intelligence”	Halevy, et al., 2006, p. 3-4
6. Workload and costs reduction	Guess, 2012

As mentioned above, Data Integration is about combining data from different sources and synchronizing data between systems. In the process of designing a Data Integration solution, different data models will be harmonized and faulty data can be spotted more easily, which raises the overall data quality (see points 1.,2.,3.,4. in *Table 3*). (Schwinn, Schelp, 2005, p. 473, 475-480).

There is another field in Information Technology which is facing the challenges of combining data: Artificial Intelligence is not only a field which has a lot in common with Data Integration, it can also be considered to be equally difficult in terms of complexity, compared to Data Integration. Data Integration can profit from artificial intelligence methods.

For example, the mapping of multiple heterogeneous sources carrying data of similar semantics into one common schema often involves repetitive mappings to be defined by the person implementing the Data Integration. Therefore, it is possible to use machine learning techniques to automate this process. See point 5. in *Table 3*; (Halevy, et al., 2006, p. 2).

Vice versa, this implies that when artificial intelligence is seeking to create more "intelligence" through combining data, then Data Integration (which is also combining data) should be considered as a field which is creating "intelligence" as well. This "intelligence" can very often be seen by the output of Data Integration projects, for example in form of the setup of automated processes, which will lead to a decrease in workload for employees: The system is basically automating what the user would have to do manually (in form of intelligence). The creation of "intelligence" increases the efficiency by requiring a smaller employee count, enabling the reduction of costs (see point 6. in *Table 3*). (Guess, 2012).

2.7.2 Disadvantages

The following table brings an overview of Data Integration disadvantages. These disadvantages are described in detail in this part of the chapter.

Table 4: List of common Data Integration disadvantages (Source: Author)

List of Data Integration disadvantages	Source
1. Effects are long-term rather than short- or medium-term	McKendrick, 2014, p. 6
2. Logical system dependency	Schwinn, Schelp, 2005, p. 476
3. Technical and business skill sets are required simultaneously	McKendrick, 2014, p. 6

One of the biggest disadvantages for companies is, that the success of the integration implementation cannot be measured in a short time, like weeks or months, but usually it takes years or decades until measurable positive effects become visible (see point 3. in *Table 4*). (McKendrick, 2014, p. 6). These effects could be cost reduction, revenue increase or other „business enabler", which are described in chapter *2.7.1 Advantages*.

Taking the step into the world of Data Integration in a company, by deciding to implement it for some systems or scenarios, also means accepting higher dependency on integrated components (applications, systems, interfaces, etc.), also see point 2. in *Table 4*. (Schwinn, Schelp, 2005, p. 476). If one component fails, other components might get affected. Therefore, companies often strive to minimize the impact of dependencies between components (see also chapter *2.5 Necessity of Data Integration*), however it can easily happen to create a logical dependency between systems, for example by declaring one system the "master"

holding the source of the data, making the connecting system dependent on this master, or also the creation of a shared database between both systems can lead to such dependencies (Schwinn, Schelp, 2005, p. 476).

From business perspective, there is usually a lack of knowledge of the data architecture which is implemented in the backend of a system, simply because there is no need for architectural knowledge for a business person. This is why data architects are needed, as they can fill the gap by providing the necessary skill. However, a data architect will always be dependent on the knowledge of a business person, to understand the semantics of the data s/he is integrating (see point 3. in *Table 4*). (McKendrick, 2014, p. 6). This means that there will never be one role which will have all the knowledge to implement Data Integration alone and therefore cooperation is always required. This can be considered a risk in integration projects, as miscommunication could lead to wrongly integrated data.

2.8 Summary

Summarizing the previous chapters, one can see that Data Integration has come a long way and has developed naturally, evolving from the 60s, finding its way into focus of today's daily business. Regardless of its long history, it has long been underestimated as a theoretical topic and companies are just discovering its importance, advantages and disadvantages. One finding of this chapter is that Data Integration cannot happen without a proper Data Management being in place and also that Data Management today can no longer live without considering Data Integration as a vital part of managing data as an asset.

3 The landscape of Data Integration

While the previous chapters cover the theory of what Data Integration is all about, the following chapter introduces different approaches being used in large enterprises to implement Data Integration solutions. This chapter also breaks down Data Integration approaches into four dimensions: Integration styles, Integration architecture, solution scale and Application2Application vs. Business2Business. Every Data Integration approach can be measured in each of these dimensions and knowing the consequences of the positioning in each dimension allows for an easier evaluation of the correctness of the chosen approach. Finally, the chapter provides an overview of the Data Integration market and current trends.

3.1 Integration architecture

Integration architecture is a very important component of integration, because it defines how the connections between applications will be organized. There are three basic approaches (Gála, et al., 2006, p.326-329):

- Point to point
- Hub and spoke
- Message bus

While these architectures actually only refer to integration, they can also be applied as a dimension to measure a Data Integration solution. Any Data Integration solution requires a technical platform to be run on. This thesis understands a platform as (for example) a PC that locally executes an application, which can create files for Data Integration (see also *3.2.1 File transfer*) or a server which is serving as a messaging bus (explained in this chapter). While the platform should stand independently from the data itself, they do have an important relationship, as Data Integration cannot happen without any platform.

3.1.1 Point to point

In point to point integration architecture, each application needs to implement an interface which enables the communication with other applications (one interface for each application). The number of connections can be up to *n(n-1)*, where *n* stands for the number of applications, which need to be integrated.

This architecture is easy to implement, however it is used mostly for small systems, where fundamental changes or large system growth are not expected.

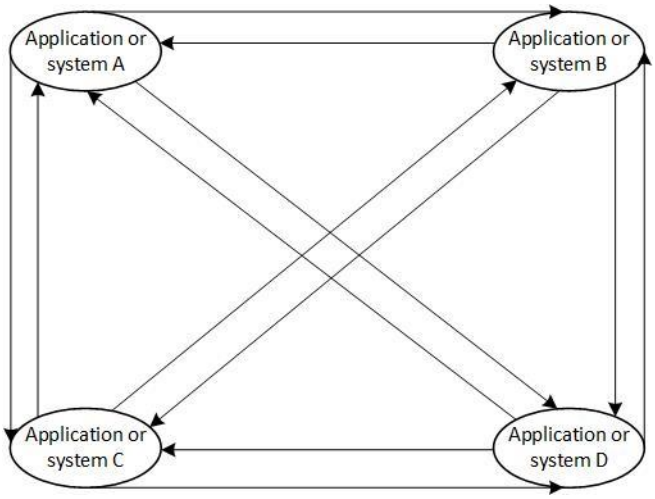

Figure 3: Point to point architecture (Gála, et al., 2006, translated by author)

3.1.2 Hub and spoke

The hub and spoke architecture uses a logical middle-layer component (called "Broker" or "Integration Broker"), which contains the entire integration logic and where all required applications are connected to. This means that it is not required to specify one interface for each set of two applications as in point to point architecture, because each application can communicate through the integration broker by specifying one interface (adapter) only.

This approach is most frequently used, because an integration broker enables a combination of multiple approaches and integration styles.

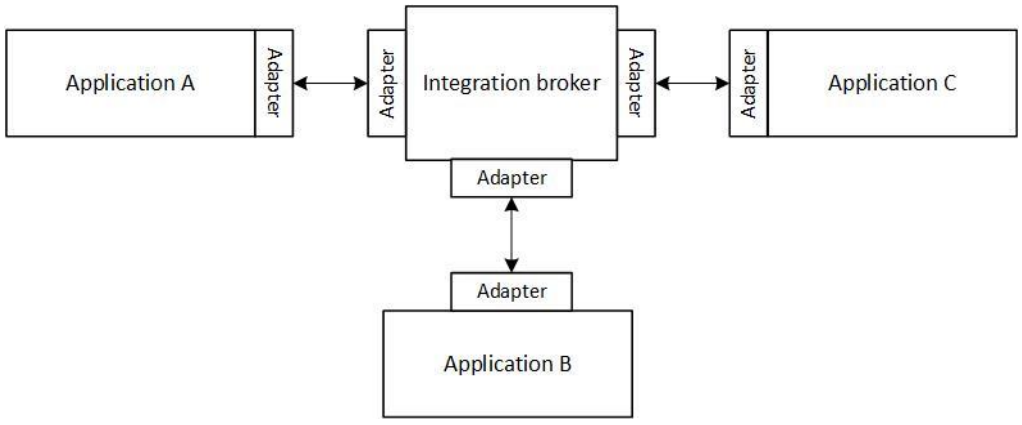

Figure 4: Hub and spoke architecture (Gála, et al., 2006, translated by author)

3.1.3 Message bus

The architecture of a message bus is similar to the hub and spoke architecture. A message bus uses adapters for each connected application and the communication is led through a middle component, which is called "bus" in this case. However, the role of adapters and bus are different than in the previous architecture. The transformation (usually to canonical schema e.g. standardized structure of data element) is ensured by the application adapter. The intelligent information routing between applications is executed by the bus. The message routing is then executed by a "publish-subscribe" mechanism, which means that communication is "published" by a source application and it is accepted ("subscribed") by a set of target applications.

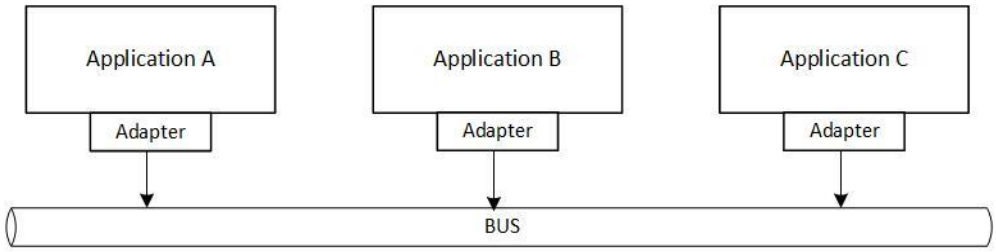

Figure 5: Message bus architecture (Gála, et al., 2006, translated by author)

3.2 Integration styles

The following chapter provides an overview of the different fundamental logical models for integrating data, also called "integration styles". All of the below integration styles can solve the problem of combining data from multiple independent applications, even if they are using different languages and platforms (Hohpe and Woolf, 2004, p. 41). Discussed integration styles are following:

- File transfer
- Shared database
- Remote Procedure Calls
- Messaging

3.2.1 File transfer

"Files are a universal storage mechanism, built into any enterprise operating system and available from any enterprise language. " (Hohpe and Woolf, 2004, p. 44). In this method, applications produce files containing data, which are consumed by one or multiple other applications. The files of each application may be produced in different formats, thus it is

necessary to specify one unified format, which all applications will be able to read or write. Also, the time and frequency of the file transfer needs to be setup, as the production and consumption of the files may have an impact on application performance and the content of the data might have a time critical aspect for the business (Hohpe and Woolf, 2004, p. 44).

The advantage of file transferring is, that integrators do not need to have knowledge about the internal functionality of the integrated applications, because this knowledge is usually provided by the application team itself. Also, the integrated applications are still independent (decoupled) from each other, therefore if a change on one application is needed or if the application fails, this will not affect the rest of the integrated applications (Hohpe and Woolf, 2004, p. 44).

This style does not require any additional tools or integration packages, as most work is carried out by the developers on the application side. There is a need to specify naming conventions for sent and received files to make sure, that these names will be unique for each file. Furthermore, it needs to be decided, which system will carry out the transformation of the messages (if needed): This can either be handled by the integrators or carried out by the applications themselves. (Hohpe and Woolf, 2004, p. 45).

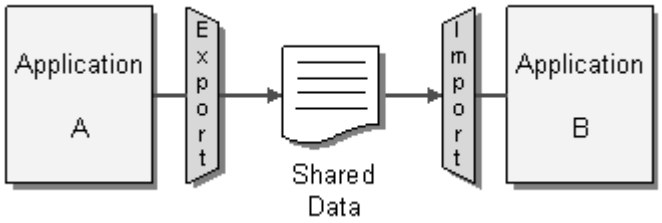

Figure 6: File transfer integration style (Hohpe and Woolf, © 2015a)

3.2.2 Shared database

Another integration style is to have a shared database across multiple applications. The applications do not need to produce, send and consume data as in file transferring; all activities are done through one shared central database, which is accessible to all applications. When looking at shared databases, it appears to have a lot of advantages from a data perspective; however it comes with a lot of disadvantages from an integration perspective (Hohpe and Woolf, 2004, p. 47).

Data in shared databases should be consistent all of the time. This is true for two reasons: There are almost no delays in the data maintenance compared to file transferring times and there are no transformations needed between the data views of the applications. All applications need to have one common understanding of the underlying data and its semantic meaning, even before the applications can be productively used. In scenarios, where data is

time critical from a business perspective, a shared database offers a good solution, as all data will be up to date in all applications in real time (Hohpe and Woolf, 2004, p. 47-48).

However, the list of challenges and downsides that come with a shared database seems to be much longer. First, applications using this database can do changes on the same data at the same time in the database, so there needs to be a locking strategy in place. Then again, when data is locked, the advantage of being a real-time solution can be diminished, as other time critical components might suffer delays from this. When integrating multiple applications into one large common database format, the result often takes longer to design, is much harder to understand and process from application side, more difficult to maintain and less flexible for future adaptations: Adding more existing applications to the same schema will cause huge changes to all applications and create dependencies between all of the applications (Hohpe and Woolf, 2004, p. 47-48).

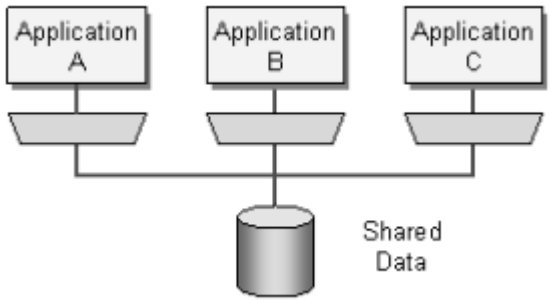

Figure 7: Shared database integration style (Hohpe and Woolf, © 2015b)

3.2.3 Remote Procedure Calls

Remote procedure calls send data across different systems directly from code to code; the interfaces are „hardwired“ directly from application to application, usually by the usage of services on one side of the applications (for example Simple Object Access Protocol, SOAP) and a utilizing component on the other application. The signatures (data structures) in these calls are usually defined by the service itself, meaning that there is no negotiating middle-layer between the applications – this type of integrations usually happens independently from any integration platform (Hohpe and Woolf, 2004, p. 51).

Remote procedure calls are used primarily in scenarios, where a change of data should also trigger a sequence of actions in the data or surrounding applications. For example if one application changes the address data of a customer, another system might need to change the same address as well and would at the same time inquire the user, if a change of invoicing address data for open orders is necessary as well. Such actions could be carried out as well by the file transfer method, but would be unfeasible for shared database scenarios. The reason is that in a shared database the application are pulling the data when an action is executed by the user, whereas in other scenarios data is being pushed and can trigger

actions automatically. Furthermore, remote procedure calls can become useful in scenarios where data needs to be available in real-time, but from a foreign data source (Hohpe and Woolf, 2004, p. 51).

Remote procedure calls are useful, when data with complex semantic meaning need to be transmitted, as the remote interfaces need to abstract and define the data clearly and without ambiguity. However, remote procedure calls always induce tight coupling between applications, no matter how hard one tries to keep them separate: If the service fails, it is likely that their subscribers will be unable to operate, especially when an operation requires a sophisticated sequence (Hohpe and Woolf, 2004, p. 52).

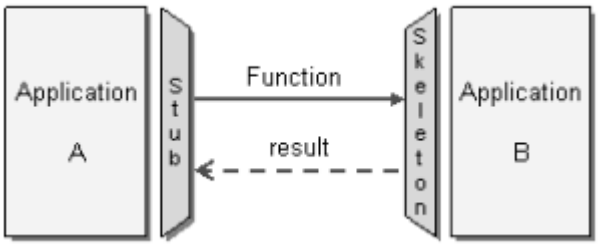

Figure 8: Remote procedure calls integration style (Hohpe and Woolf, © 2015c)

3.2.4 Messaging

"We consider messaging to be generally the best approach to enterprise application integration" (Hohpe and Woolf, 2004, p. 55).

The previous sub-chapters have shown that there can be more than only one feasible approach in the Data Integration landscape. All of the above methods might be realistic in some context, as the following chapters will show. However, messaging seems to be the one method which can handle most issues and disadvantages on a conceptual level.

Messaging is basically a file transfer taken to the next level: An application decides to send a message containing a data update, which can either be scheduled or triggered ad-hoc by an action in the application. This message does not necessarily need to know its recipient, it can be specified in an arbitrary data format, it does not need to be written into a common database and it does not need to wait for a reply: All of these actions will be executed asynchronously by a messaging bus connecting all integrated applications.

The idea here is that each application will provide only a set of messaging "End Points" for sending and receiving data, in a previously defined format and structure. This structure can be completely independent of the entire messaging bus (but should of course follow a valid business scenario – e.g. expecting weather data from a stock exchange structure is rather unfeasible). The messaging bus will take care of collecting and sending these messages, routing them to the right recipients, transforming them to the receiving formats and will also feature retry mechanisms in case of failure.

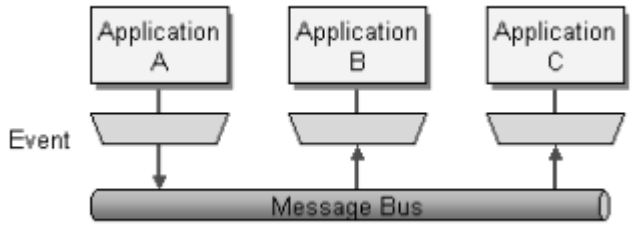

Figure 9: Messaging integration style (Hohpe and Woolf, © 2015d)

3.3 Approaches in small vs. large enterprises

The following chapter compares small scale and large scale Data Integration solutions. In order to be able to compare, the chapter concentrates on the differentiation of approaches in small (less than 50 employees) and large companies (more than 250 employees) (BusinessInfo.cz, 2009).

As the following two sub chapters will show, the overall requirements of small scale vs. large scale solutions are distributed as shown in the *Table 5* below:

Table 5: Comparison of small scale vs. large scale solutions (Source: Author)

	Small Scale	Large Scale
Need for Data Integration	Very high need for integration due to isolated applications	High need for integration
Preferred Integration Style	File Transfer, Remote Procedure Call, Shared Database	Messaging
Internal vs. External	Mixture of self-developed solutions and external tools	Mostly external, proven solutions
Impact	Quick medium term results, but messy implementation for long term maintainability	Projects usually take longer to finish, but are easier to maintain

3.3.1 Small scale solution

For small companies, the investment into information systems as well as Data Integration solutions presents a high percentage of overall costs. Therefore managers have to choose the right solution in order to prevent project failure, which could noticeably weaken the company. (Smith, Simon, 2009, p. 36)

There is a "trend", expanded across small companies. A large percentage of them are still purchasing pre-packaged applications, which can deal only with one function in isolation and these applications usually cannot read data and work with the data, which are available in different sources out of the box. This means, that these applications can't work efficiently until a Data Integration project is executed. (Smith, Simon, 2009, p. 37)

When a Data Integration project is successful, the company can profit from increased information availability, because all information are available from one source, which eases work of employees regarding long searches for the right information. Information availability leads to decision correctness increase as well. (Smith, Simon, 2009, p. 36)

Small companies usually do not use robust solutions from market leaders in integration fields, because these solutions usually contain wide set of functionalities, which would be useless for small businesses. Also, the price for maintaining these solutions can be on an unreachable level. However, this does not change the fact, that Data Integration is needed at some point, because the cognitive capacity of managers in small firms cannot face manual Data Integration forever, especially in today's era of massive data increase. (Smith, Simon, 2009, p. 36)

Small companies have more options, how to deal with integration by other ways. One of these options is to realize Data Integration internally, through internal tools. This has an advantage, that the solution does not need additional employees for Data Integration maintenance as for example the company needs only one employee to implement one solution for one system. Also, the integration does not require previous integration experience and the output would largely depend on business needs. Feasible integration styles for such an approach (based on chapter *3.2 Integration styles*) could be file transfer, shared database and remote procedure calls.

The other approach is to use external solution for Data Integration, which seems to be the more future proof option compared to internal integration, because companies can more concentrate on business processes than on Data Integration, which is not their primary business (MuleSoft, © 2015). Examples of small scale solutions available to companies are MuleSoft (Mule ESB), SAP (SAP Business One) or Insightly (Insightly for Sales).

3.3.2 Large scale solution

The wording „Large scale technology" already suggests that solutions based on this approach always cost a lot of money, but surprisingly, the tools for these solutions (which are only part of the costs though) can be cheap, sometimes even for free. As the price is usually for companies one of the crucial criteria for the selection of right solution, managers often chose exactly one of these. However, these solutions also require additional costs for employee training and business insight, because without proper knowledge, Data Integra-

tion will never fully work. Also, facing big data challenge without previous knowledge of small data solutions can be very frustrating for a company (McKendrick, 2014, p. 6).

From a Data Integration style perspective, large scale solutions in enterprises, as well as Data Integration market leaders[1] would profit most from focusing primarily on Messaging solutions, as this is considered the best approach for Data Integration (see chapter *3.2.4 Messaging*) and large scale enterprises and vendors should have the necessary manpower for setting up and maintaining a message bus. Other integration styles are supported by integrators as well, most probably because these integration styles might be easier to implement. However they can become much harder to maintain, create stronger system dependencies and are therefore more costly in the long term (see also chapter *3.2 Integration styles*).

Summarizing, when comparing small vs. large scale solutions, one can see that in case of small companies, the desire for having integrated solutions is higher, however the used styles focus mostly on simpler and less future proof methods. Large enterprises should prefer doing integration in a "proper" way, meaning that they choose more complex but future proof solutions.

3.4 Business2Business vs. Application2Application

The definition of Business2Business (B2B) is not clearly specified, as it has been used in many companies within different contexts, having developed into some sort of "Buzzword" (Bussler, 2003, p. 3). This thesis proposes to differentiate between B2B and A2A (Appilcation2Application) by assuming that B2B is about sending messages between systems of different companies, while A2A is solely centred around integrating applications within the same company (Adusupalli, 2013).

Both methods share their advantages, disadvantages and challenges and have already been discussed from a technical standpoint in chapter *2.7 Advantages and disadvantages of Integration*, which remain the common challenges for both types of Data Integration. In the following paragraphs the major differences between B2B and A2A challenges will be highlighted.

[1] Currently, the leaders in market of big data integration solutions are well-known companies like Oracle, IBM, SAP and Informatica (DAO Research, 2015, p. 1; Gartner, 2015a)

B2B Integration

This thesis classifies B2B Integration as any flow of data crossing the "border" between companies. The main challenge of B2B integration is to incorporate the most heterogeneous data landscapes with each other: As every business defines their own data architectures, finding common models even for the same kind of business (Wende, 2007, p. 317) can be very hard to achieve. A company can of course decide to integrate their systems manually with each system of their various partner, however this can quickly turn into a nightmare, as the partner companies can also change their systems and interfaces at any time. To accommodate this problem, several organizations, like Microsoft, IBM, SAP, Intel, Hewlett Packard and even the United Nations have developed multiple sets of standards which can be used by any company to exchange electronic messages with each other. Known formats include examples like "RosettaNet", "e-speak" and "UN/EDIFACT" (Kim, et.al, 2003, p.318).

To give an idea, what these standards look like and not to exceed the scope of this thesis, only UN/EDIFACT will briefly be explained in the following. The UN/EDIFACT (United Nations/Electronic Data Interchange for Administration, Commerce and Transport) defines a set of rules and guidelines for the implementation of EDIFACT, as well as a so called "Directory", which is effectively a large list of possible EDIFACT message types and a detailed definition of the fields in such messages, as well as descriptions and technical attributes, such as cardinality (how often a field can appear); (UNECE, 2016a).

One practical example of such a message is the message type "IFTMIN" (UNECE, 2016b), which is supposed to be used as an instruction message for the transportation of goods between two trading partners. It contains detailed information about the goods to deliver, surrounding containers, origin, destination, involved parties. The information in an EDIFACT message is usually stored in multiple lines, which could look like this:

```
GDS+4++3+4711+++Eau de Cologne’
NAD+ST+0815+++HP Headquarters+Hewlett-Packard+1st Street [...]’
```

The two lines from a fictional IFTMIN message above represent two so-called "Segments": GDS (Nature of Goods) and NAD (Party Name and Address, segment is shortened in the example). Each segment can store different amounts of information, of which most are optional. Each field is separated by a "+" delimiter. The information that is represented by the above example is well defined in the IFTMIN standard and informs the receiver about the following information:

Table 6: Extracted Fields from an IFTMIN message (Source: Author)

Position	Field meaning	Value (and Meaning)
1st segment, 1st field	Segment Identifier	GDS (Nature of Goods)
1st segment, 2nd field	Cargo Type	4 (High value consignment)
1st segment, 4th field	Responsible Agency	3 (IATA)
1st segment, 5th field	Product Code	4711 (System code for product identification)
1st segment, 8th field	Product Name	Eau de Cologne
2nd segment, 1st field	Segment Identifier	NAD (Party Information)
2nd segment, 2nd field	Party Function	ST (“Ship to” Party, Receiver of goods)
2nd segment, 3rd field	Party identifier	0815 (System code for party identification)
2nd segment, 6th field	Name and Address	HP Headquarters
2nd segment, 7th field	Party Name	Hewlett-Packard
2nd segment, 8th field	Address Line	1st Street

Having a standard like EDIFACT brings a lot of advantages with it, as it allows companies to send structured information to each other, in a pre-defined, durable format (EDIFACT is ISO certified since 1988); (ISO, © 2016). However, it also only solves a few of the integration challenges. For example, the “Party identifier” in *Table 6* might have a meaning to the sending system, but not necessarily for a receiving system, especially not when codes are being sent between companies. This means that the free text field containing the name of the party might contain more crucial information than the actual (possibly machine readable) code.

Furthermore, fields in EDIFACT are mostly kept optional to allow for high reusability (the given example above also has many fields, which are simply empty), so when a new company joins a B2B network, knowing the EDIFACT standard alone is pretty useless, as every company can use it differently. On top of that, the author of this thesis has also witnessed EDIFACT messages, which are simply used wrongly, storing for example location codes (for example IATA airport codes) in the field for Party Name – which is double wrong, as a location in EDIFACT is something else then a party (for example it could be stored in segment LOC) and storing an internationally defined code in a free text name field is not improving the situation. This is of course not the fault of the standard itself, but

rather of its usage. However it shows that a high amount of message flexibility increases the effort a company has to put into assuring that the quality of the data is sufficient.

A2A Integration

This thesis classifies A2A Integration as any flow of data between systems of the same company. For A2A flows, as mentioned earlier, the company will mostly benefit from higher efficiency due to more "intelligent" systems (see chapter *2.7.1 Advantages*). Here the integration process can focus more on the internal needs of the company, largely ignore external business models and can integrate systems based on much more strict assumptions. Since the company also has control over its own internal systems, it is much easier to achieve a closely bound integration than it is the case with B2B integration. Also, there is nothing wrong with using standards like EDIFACT for pure inhouse A2A communications.

A2A and B2B solutions do not exclude each other, as usually a business can benefit from both. The concepts however differ in detail and it is therefore a common practice to keep the platforms for both solutions apart from each other (Bussler, 2003, p. 22).

3.5 Data Integration market and trends

This chapter aims to complete the picture of the Data Integration landscape. While the previous chapters clarify in what kind of technical dimensions a Data Integration solution can be measured, this chapter provides an overview of the Data Integration market and trends. Chapter *3.5.1 Data Integration tools* market focuses on available standard solutions in the Data Integration market while chapter *3.5.2 Current trends in Data Integration* provides an overview of the current trends.

3.5.1 Data Integration tools market

The market of Data Integration tools offers a wide variety of commercial off-the-shelf (COTS) tools. Gartner, Inc., which is a market observing company centred in the U.S., has provided an overview of the currently biggest Data Integration vendors and has categorized them by "ability to execute" and "completeness of vision" and has drawn a map called the "Magic Quadrant" to visualize it:

Figure 10: Gartner Data Integration tools Magic Quadrant 2015 (Source: Gartner, 2015a)

According to Gartner, the top three commercial vendors in terms of vision completeness and execution ability should be considered as Informatica, IBM and SAP. A special mention should go to Talend, which is the top commercial vendor in the open source market. These vendors are taken as an example for a market overview in this thesis. When looking at the offers from these vendors, most of them offer products for all previously mentioned integration styles, are suitable for large scale integration and include A2A as well as B2B solutions. *Table 7* below provides a detailed overview of what the vendors (Informatica[2], IBM[3], SAP[4] and Talend[5]) are offering to customers directly. The table has been created by visiting the websites of the mentioned vendors and searching them extensively for any keywords that are connected to the category.

[2] Sources: Informatica, © 2016a; Informatica, © 2016b; Informatica, © 2016c

[3] Sources: IBM (© 2016a); IBM (© 2016b); IBM (© 2016c); IBM (© 2016d); IBM (© 2016e)

[4] Sources: SAP (© 2016a); SAP (© 2016b); SAP (© 2016c); TechTarget (© 2016a); Raju and Wallacher (2008, p.141)

[5] Sources: Talend (© 2016f); Talend (© 2016g); Talend (© 2016c); Talend (© 2016b); Talend (© 2016e)

Table 7: Overview of Top-4 Data Integration vendors and covered functionality

	File Transfer	Shared Database	RPC	Messaging	Scale	A2A/B2B
Informatica	Yes	Only supportive tools	Yes	Yes	Small to Large	Both
IBM	Yes	Yes	For selected products	Yes	Small to Large	Both
SAP	Yes	Yes	Yes	Yes	Small to Large	Both
Talend	Yes	Only supportive tools	Yes	Yes	Small to Large	Both

The conclusion from this overview is that when a company of any size seeks for an external vendor for Data Integration, at least the three biggest vendors (plus Talend) try to cover all possible requirements.

Data Integration always requires a project and full Data Management support in order to work (as it was mentioned in chapter *2.4 The role of Data Integration in the context of Data Management*). While all vendors have good offers for getting started in Data Integration (some of the solutions above require only 1000$ for a license, some basic versions even come for free[6]), the true money in Data Integration will be made in a different field: An article from TDWI, an education and research company that focusses on Data states the following:

"When it comes to the cost of a [Data Integration project in the world of] BI deployment, it's not the software that will get you; it's the miscellany -- the miscellaneous integration work, in particular. […] Why so expensive? Well, it's Data Integration. It's hard, it's complex, it's perhaps one of the most difficult jobs in the world of BI, and it's often unsung." (TDWI, © 2016).

The successful vendors of Data Integration technology have understood that and while they are advertising Data Integration solutions as affordable COTS products, what they are actually selling is a long-term partnership with their company. For example, when trying to actually purchase IBM products, the customer gets an offer for an "IBM PartnerWorld" Membership (© 2015), which can provide trainings to the customer, easier access to sup-

[6] Sources: Informatica (© 2016d); SAP (© 2016d); IBM (© 2016f)

port knowledge and will also offer IBM employees as resources for configuring or even adapting the actual integration solution, which gives advantages to both sides, provider and customer.

Some vendors go even one step further and offer their solutions as so called "Commercial Open Source Software" (COSS), the most prominent, leading example being Talend (© 2016a). The idea of COSS is to give the customer full access to the Data Integration Software and its source code, free of charge. While this is a very tempting offer, especially when the interested company has developers available in house, who could potentially enhance the open source code, these offers come with a few disadvantages: As with any free product, COSS are shipped without any support (however it is of course possible to buy additional support). On top of that, most COSS products represent only a basic version; more advanced features have to be bought commercially as an add-on (Astera, © 2014, p. 2). Therefore, COSS solutions have most advantages for companies having employees with high technical skills.

3.5.2 Current trends in Data Integration

When the author of this thesis was doing a literature review (see chapter *1.6 Literature review*), one of the findings was, that while searching for the phrase "Data Integration", the results often referenced to topic of "Big Data Integration" and generally "Big data" (for example Dong and Srivastava, 2013). The conclusion is that these topics should be considered as one of the trends in Data Integration. This chapter will focus on describing these topics. Also, an indispensable amount of search results for Data Integration were related to medical topics. It seems like Data Integration plays a big role in medicine, hence this topic is highlighted as one of the current trends as well. The next trend in this chapter is the so called "Internet of things", which is considered as one of the latest trends (Tata Consultancy Services, © 2015).

Big Data has become one of the biggest trends in the Information Technology industry and is often considered to be a "buzzword" and Kraska (2013, p. 84) even considered it as a "Buzzword of the Year [2013]". The reason for that is that there is a large variety of definitions available which quickly puts the word out of context (De Mauro, 2015, p. 97). To summarize, Big data is characterized by large volume of data, ranging from Terabytes to Exabytes (According to Munroe, 2014, Google has been estimated to store an unconfirmed total data size of 10-15 Exabytes) with wide variety of types (like audios, videos, tables) and request for high velocity and high volume processing (like real-time, batch), (Rouse, 2014a). The challenges of Big Data and Integration have a lot in common and one could say that integrating two Big Data sources with each other is the most extreme form of integration, as it will provide the largest set of challenges in all dimensions. To cope with such challenges, new paradigms and technologies are evolving; one of them is Apache Hadoop. Hadoop is a free open source platform based on Google Inc.'s MapReduce programming

model (Jeffrey and Ghemawat, 2004), which allows parallel processing of large amounts of data using distributed computing (Hadoop, © 2014). Both Hadoop and MapReduce have been mentioned as the most popular programming models and most commonly used implementation for Big Data Integration (Assuncao, et al, 2014, p. 14). Since Hadoop is considered to be the most prominent technology associated with Big Data, it is a strongly required skill for integrators when looking for job offers.

Another trend, found in search for Data Integration topic and which is also related to the trend of Big data, is its role within life and medical sciences. One interesting example is the Abcam company. Abcam is a scientific company evolving since 1998, which collects large amounts of data from multiple sources, largely focused on medical researches, studies, tests, results and data sheets (which can easily grow into the category of "Big Data"). The gathered data is analysed to get information about current viruses, cancer proteins, etc. This data is then used for producing antibodies which will defeat the actual human sicknesses. The big advantage for Abcam's business is that they are trying to produce the highest quality proteins possible and for that they do not need only research data from their own institute but also the accumulated knowledge that is being produced by researches worldwide. Whenever an external research is being conducted on one of Abcam's proteins, they collect all the data about that and try to extract the most important information for that - through Data Integration they basically can gain knowledge easily and for "free" (Abcam © 1998-2015; Internal source).

Another subject which is currently trending in IT community discussions is the “Internet of things”. The Internet of things is more a future trend in IT, which is considering the trend of daily-life objects (like phones, watches, TVs, light switches) to become “intelligent gadgets” by connecting them and enabling communication with the Internet (Shancang, et al, 2015). The Internet has so far been greatly dependent on humans, because the majority of the data needs to be first created and inserted by them. The time and costs which result from this, lead to new approaches, how to automate the collection, integration and analysis of the data. The idea of the Internet of things is that with all connected devices, data can be gathered without human intervention and people will just get results of the data (Rouse, 2014b). Since gadgets are being produced by a multitude of companies, Data Integration for all such devices will become more and more important in the future, as it will open up new business possibilities. For example when a gadget breaks and needs some reparation, it could automatically send data to the store in which the gadget was bought and a shop assistant can call the gadget owner and offer him good service immediately.

3.6 Summary

To summarize all previous subchapters, this thesis has described multiple Data Integration approaches, which might be used by large companies and provided an analysis of the Data Integration landscape. Several sources which are not older than 3 years have been taken into consideration (Bruckner, 2012; McKendrick 2014; Guess 2012; Oracle, 2015; Mulesoft, © 2015; Assuncao, et al, 2014; Kraska, 2013). Concluding from the analysed sources, Data Integration can be split into 4 possible approaches (File transfer, Shared database, Remote Procedure Calls and Messaging). Considering the findings of the second part of the chapter, which compared possible Data Integration solutions (Small scale vs. Large scale, Internal vs. External solution), the conclusion is that small scale internal solutions usually use File transfer, Shared database and Remote Procedure calls approaches. Large scale solutions on the other hand are usually so complex, that the Messaging solution seems to be the most advanced, but also most sophisticated solution that can cope with all challenges – and therefore it is also used by current market leaders.

4 Features of Data Integration tools

This chapter analyses multiple Data Integration tools available on the market with the aim of providing a list of categories for functionality which can be commonly found in most Data Integration tools. Chapter *4.1 Analysis of common functionality in Data Integration tools* describes which tools were chosen for analysis, which sources were considered for analysis and summarizes the results on a high level, chapter *4.2 Common functionality in detail* and its sub chapters provide the detailed description of functionalities and chapter *4.3 Summary* provides a short summary and findings resulting from the analysis.

4.1 Analysis of common functionality in Data Integration tools

To find out which functionalities in Data Integration can be most commonly found across multiple Data Integration tools, the most helpful sources that can be found are marketing material, mostly in form of video-tutorials, white-papers and webinar recordings.

Additionally, some companies offer trial versions of their products (Informatica, © 2016e) or, as mentioned in chapter *3.5.1 Data Integration tools* market are completely open source and therefore free to download including their documentation (Talend, © 2016b). These applications have been downloaded and analysed as a part of this examination.

Since most of the tools are very feature rich, it takes a long time to learn all available functionality and most vendors only offer training in their Data Integration applications for a high training course fee, it would exceed the focus of this thesis to explore all available tools on the market in detail. Therefore, this chapter primarily focuses on the four leading Data Integration tool providers in detail mentioned in chapter *3.5.1 Data Integration tools* market (Informatica, IBM and SAP for COTS and Talend for COSS). To provide an idea how big the market for Data Integration really is, the below list provides an overview of the tools which are being offered by the vendors in connection with Data Integration and Data Management. The list has been reduced to the most relevant findings related to Data Integration, including short descriptions of each tool:

Table 8: Overview of tools offered for Data Integration purposes by Informatica, IBM, SAP and Talend

Product	Description	Source
Informatica		
Informatica Advanced Data Transformation	Tool for ETL Processing	Informatica (© 2016f)
Informatica B2B Exchange	Tool for ETL Processing, Monitoring and also a portal for business partners	Informatica (© 2016j)
Informatica Connectors (PowerExchange)	Enhanced template repository for easier connection to heterogeneous sources	Informatica (© 2016k)
Informatica Data Integration Hub	Enterprise Service Bus with focus on hybrid Integration (between cloud and on-premise)	Informatica (© 2016b)
Informatica Real-Time Integration (RulePoint Complex Event Processing and Ultra Messaging)	Technology using special protocols and tools for faster file transfer	Informatica (© 2016h)
Informatica PowerCenter Express	Free ETL Tool, based on PowerCenter, which is used to administer and govern Data Integration end to end	Informatica (© 2016e)
IBM		
IBM InfoSphere DataStage	ETL Platform, near real-time and hybrid capable	IBM (© 2016g)
IBM Integration Bus	Enterprise Service Bus	IBM (© 2016h)
IBM InfoSphere Master Data Management	Tool for governance of master data	IBM (© 2016i)
IBM InfoSphere Information Analyzer	Tool for improvement of data quality	IBM (© 2016j)
IBM InfoSphere Optim Data Privacy	Data Masking Tool	IBM (© 2016k)
IBM Sterling B2B Integrator	B2B Integration Tool	IBM (© 2016l)

Product	Description	Source
SAP		
SAP BusinessObjects Business Intelligence	Data Warehouse Tool, used mostly for BI and Reporting, strong ETL capabilities	SAP (© 2016e)
SAP NetWeaver ProcessIntegration	The main integration platform for SAP products	SAP (© 2016f)
SAP NetWeaver Master Data Management	Tool for centralized governance of master data	SAP (© 2016i)
SAP Data Quality Management	Tool for measuring and improving data quality	SAP (© 2016j)
SAP HANA	Hardware and Software platform for in-memory based real-time processing	SAP (© 2016l)
SAP NetWeaver Process Orchestration	B2B integration tool	SAP (© 2016g)
Talend		
Talend OpenStudio for Data Integration	Open source tool for ETL-Processing	Talend (© 2016a)
Talend OpenStudio for Data Quality	Open source tool for measurement and improvement of data quality	Talend (© 2016i)
Talend OpenStudio for Enterprise Service Bus	Open source messaging platform	Talend (© 2016c)
Talend OpenStudio for Master Data Management	Open source tool for central management of master data	Talend (© 2016d)

By looking at the extensive list of products and features above, it becomes clear that there are large similarities between the four vendors. Each vendor is for example offering at least one tool with ETL capabilities, or one tool for data quality improvement. To avoid describing each single individual application in detail, this thesis is clustering the applications into seven categories and describes these categories in detail, on the example of the before mentioned applications. These categories are:

- **ETL Processing**: ETL, standing for Extract, Transformation and Load, is one of the most important functionality, which specifies, what is going to happen between two different applications A and B, when application B wants to get data from application A.
- **Enterprise Service Bus**: ESB or Message bus is one of the possible architectures of Data Integration, mentioned in the chapters before. It serves as a middle layer for communication between applications and it can create, mediate and deploy services.
- **Master Data Management**: MDM is a kind of repository, which enables to unify, store and maintain Master Data of a company.
- **Data Quality Analysis**: contains methods, how to improve overall data quality, for example to determine the nature of data available in a flow and to measure the quality of the individual data contents.
- **Real Time Integration**: it is highly desired that when messages are being transferred from one application to another, that it will not cause delays on the integration layer.
- **Data Masking**: Tools that allow securing data while keeping the structure.
- **B2B Integration**: Data Integration tools that are specialized for B2B scenarios.

Table 9 below gives an overview of how well the individual vendors (Informatica[7], IBM[8], SAP[9] and Talend[10]) fit into this categorization.

[7] Informatica (© 2016f); Informatica (© 2016b); Informatica (© 2016g); Informatica (© 2016l); Informatica (© 2016h); Informatica (© 2016i); Informatica (© 2016j)

[8] IBM (© 2016g); IBM (© 2016h); IBM (© 2016i); IBM (© 2016j); IBM (© 2016k); IBM (© 2016l); IBM (© 2016 m)

[9] SAP (© 2016e); SAP (© 2016f); SAP (© 2016i); SAP (© 2016j); SAP (© 2016h); SAP (© 2016g); SAP (© 2016k)

[10] Talend (© 2016a); Talend (© 2016c); Talend (© 2016d); Talend (© 2016h); Talend (© 2016l); Talend (2015d); Talend (© 2016e)

Table 9: Categorization of tools available by vendors

	ETL Processing	Enterprise Service Bus	Master Data Management	Data Quality Analysis	Real Time Integration	Data Masking	B2B Integration
Informatica	Yes	Yes	Yes	Yes	Yes	Yes	Yes
IBM	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SAP	No discrete tool, but generally available	Only SAP PI with focus on SAP Products	Yes	Yes	Yes	No discrete tool, but available with limited functionality	Yes
Talend	Yes	Yes	Yes	Yes	Yes	Yes	Yes

It is important to notice that:

a) The categories above cover a wide range of functionality, but not all are necessary to use or buy to get a working Data Integration.

b) All four vendors separate the features for each category into individual applications instead of offering only one application for everything, so customers can chose which components to use or buy according to their requirements.

c) Some of the analysed applications are designed to work independently from each other, so they can be easily combined with products from other vendors (see for example Informatica, © 2016f).

4.2 Common functionality in detail

Chapter *4.1 Analysis of common functionality in Data Integration tools* has defined which tools and sources are analysed as well as which categories of shared features between Data Integration are considered for analysis. This chapter describes these categories and their functionalities in detail. Since Informatica is the leader for COTS tools and Talend is the leader for COSS tools, the following subchapters focus on describing functionality from these two vendors.

4.2.1 ETL Processing

Both Informatica ("Advanced Data Transformation") and Talend ("Open Studio for Data Integration") essentially offer an ETL tool as their primary "Data Integration" tool. ETL, which stands for "Extract, Transformation, Load" is the process of getting data from system A (Extract) to system B (Load) while changing the data structure (Transform) so that both source and target systems can understand the data, which is a necessary feature of Data Integration. These Data Integration tools can deploy the code on pretty much any platform – for example on a "hub" (see chapter *4.2.2 Enterprise Service Bus* below), a web service or in form of an executable java applet, meaning that the actual integration code stands independent of the tool itself. This is a general concept that each application seems to follow, as implemented flows[11] should be reusable in other flows as well (Informatica, © 2016f). What is interesting about both tools is that every vendor is offering a visual front end for defining the flow from source to target, without having to write a single line of code.

All analysed ETL tools share three key elements as their core features:

- **Template repository**: For the creation of an ETL flow it is necessary to specify which components are covered inside of the ETL process, because ETL can extract, transform and load many different things. For this reason, there is a high demand for templates which can reduce work. Both Informatica and Talend offer many templates or wizards for connecting to EDIFACT, SAP, SalesForce, DotNET, OLAP Cubes, Microsoft Office, Oracle databases, MS SQL Server, LDAP, email and countless other formats, to allow easy and faster integration. While these templates are useful for industry standard formats as well as DB connections, self-written exchange formats (for example CSV or tab delimited flat files) still need to be defined manually.

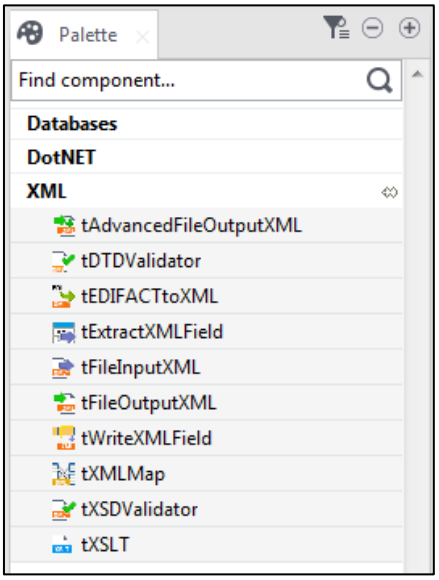

Figure 11: Example of available components in the Talend template repository

- **Extensive testing and debugging**: Both ETL tools provide watch windows for the execution of an ETL flow for monitoring and logging each step in the process. The usual output of the test contains precise timing information (which process takes how long) and can also output the entire data set (or individual lines, if executed

[11] "Flow" is understood in this thesis as a Data Integration scenario connecting a minimum of two system end points with each other.

step by step) for easier debugging. The output options are as countless as the integration flow itself, as the tools can offer the output directly into the watch window (which is usually also provided as a connector in the template repository, see *Figure 12*), as well as external flat files (like log files) or even send logging information to other systems, which makes sense, if a company is utilizing a separate system for monitoring flows.

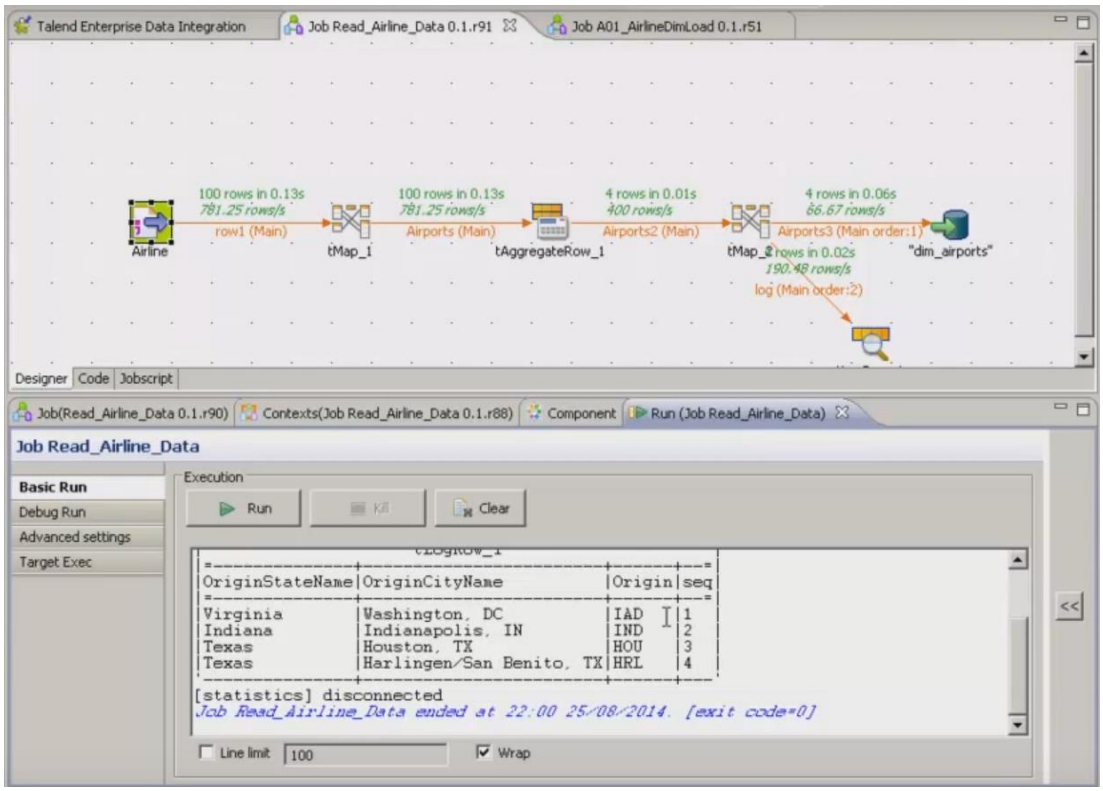

Figure 12: Example of flow debugging with performance data and logging output in the watch window (Source: Talend, 2015a, 02:58)

- **Flexible configuration**: Each template for connecting a system to the ETL flow requires a certain set of parameters to establish the connection. For example, a flat file requires a defined name and a system file path to be saved into and a DB connection requires a connection string. These parameters don't have to be stored directly into the ETL flow (which would be considered "hard coding" as the graphical front ends create the integration code in the background), but can be defined as variables. The *Figure 13* below shows how the user can define an arbitrary amount of variables, for example suffixes for the DB connection string ("_dev", "_prod") and how these variables can be defined in the flow. This allows for easier deployment and reuse of integration code.

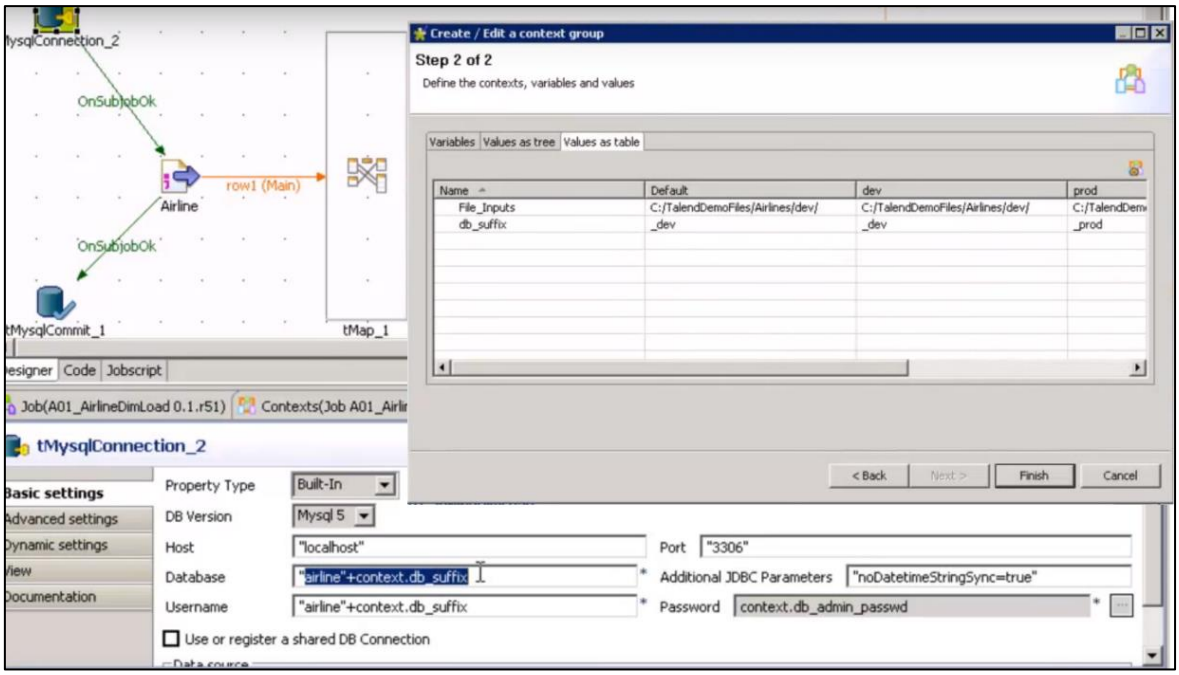

Figure 13: Example of flexible configuration in Talend Open Studio (Talend, 2015b, 01:06 to 02:10, modified by Author)

4.2.2 Enterprise Service Bus

Both Informatica (© 2016b), as well as Talend (© 2016c) offer a messaging bus tool, or Enterprise Service Bus tool, or also simply called “Hub”. These hubs usually serve as the deployment platform and runtime server environment for the actual integration code from the ETL tools. While the codes from these tools can both be compiled as JAR, it is easy to execute them on any machine hosting a JAVA runtime environment, however the easiest solution is always to deploy and run them on the hubs they were created for. For example, code from the Talend Studio should ideally go to the Talend ESB and it would be much harder to deploy it on an Informatica ESB.

Integration hubs serve as a middle layer for communication between applications, as it was specified in chapter *3.2.4 Messaging*. In reality, that means that the hubs can create, mediate and also deploy services, for example for SOAP web services (Talend, © 2016h), set up service policies or also define “contracts” (Loughead, 2014). The idea of a hub contract is that the connected, individual systems are allowed to publish data into the hub using a pre-defined contract and to subscribe to data from the hub, which is under the same contract. The contract may specify a message format or just a set of fields which are mandatory to connect to the contract and also frequency, delivery information, file sizes or simply the message purpose.

Finally, the enterprise service bus also takes care of correctly routing the message based on specified rules. Such rules could for example evaluate specific fields of an individual mes-

sage and could route it based on the parameters inside. *Figure 14* below shows how this Routing is represented in the Talend Open Studio.

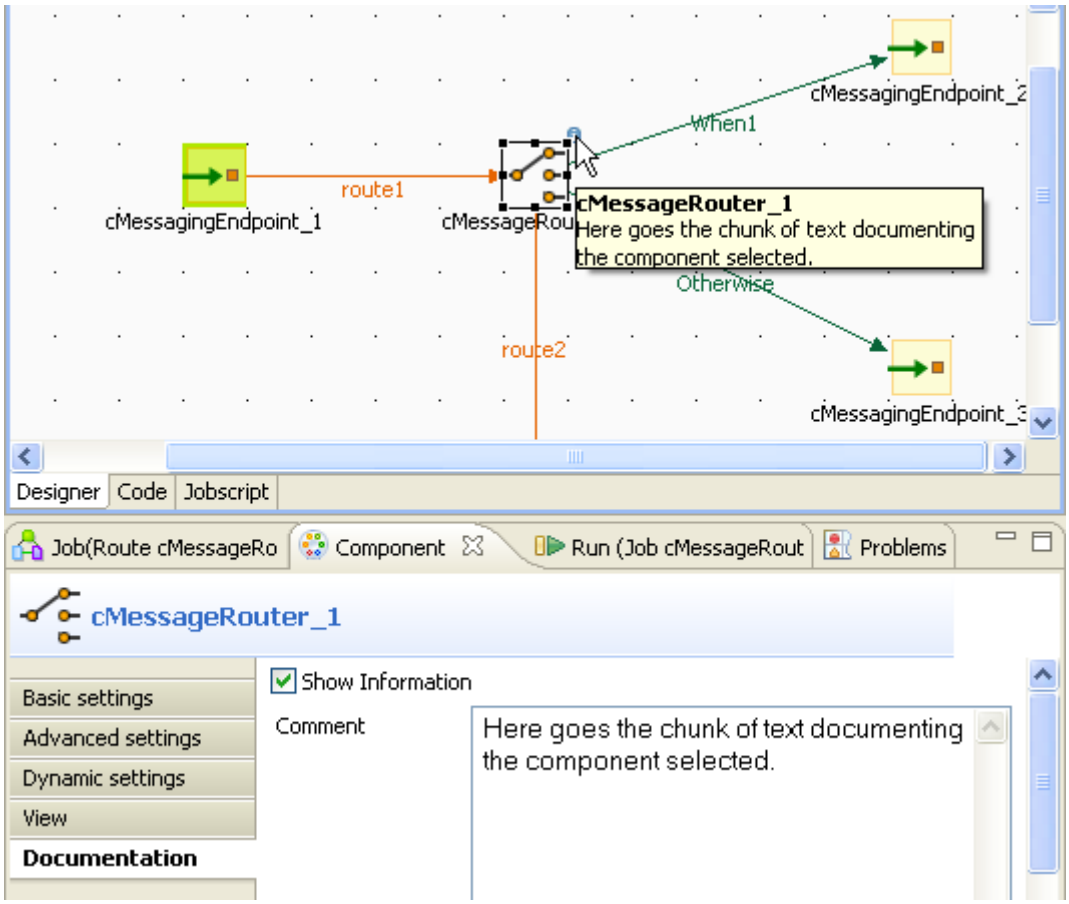

Figure 14: Example of a routing component in Talend Open Studio (Talend, © 2016m)

4.2.3 Master Data Management

MDM Tools in general serve as a repository for unifying, storing and maintaining company Master Data. Utilizing a Master Data Management tool brings conceptual advantages for the entire company, as it allows for five major improvements (Talend, © 2016d):

- **Consolidating**: When data is being present in multiple independent sources and the company tries to consolidate this data, MDM Tools can help to identify and categorize the data, as well as to resolve duplicates between the systems.
- **Synchronizing**: Once the data has been migrated into one common source, the next challenge is to keep all other systems, which utilize their own copy of master data, in sync. As MDM involves techniques from Data Integration, keeping systems in sync with the same set of master data is as well a task of MDM tools.
- **Service Enabling**: Being able to consolidate all data into one source and synchronize it between all systems, enables master data to be handled by MDM services.

The idea is that the MDM tool is offering services to gather data from all available sources and finally offer one source of truth for all available master data. For example, when a person in a company wants to know, which customers the company currently deals with, the MDM tool has to offer the correct answer.

- **Collaborating**: Having different systems which need integration in a company, also means having different departments using each system. When integrating the master data between the systems, it is most important to come to a common agreement between all departments to define the master data correctly. MDM tools take care of the master data "stewardship", meaning they make it easier to watch and decide over data, by offering workflows for approval processes or can set up tasks for data cleansing activities.

- **Reference Data**: The final area helps with automating Data Integration within a company. Whenever data is flowing from system A to system B, these systems might have a different data structure or data content. For example the same customer could have different IDs between two systems and in one system it lacks the phone number, while in the other this is a mandatory piece of information. Reference Data in MDM tools solve this problem: By making an MDM tool an integral part of the Data Integration message bus, every data flow that goes from system A to B has a chance to "ask" the MDM tool, what data it will need to reach system B. It could for example change the customer ID on the fly or it can grab the telephone number directly from the MDM tool and make sure that the target system will receive all required information in one step.

4.2.4 Data Quality Analysis

While MDM tools handle a good part of data quality issues in a Data Integration project, there are also dedicated tools which take care of the specific analysis, categorization and highlighting of data quality issues.

The starting point of Data Quality tools is usually a so-called "data profiler", which allows a holistic analysis of a given data feed. The profiler tries to determine the semantics of each column individually and could give as a result an ontological analysis of the data set given: In case of Talend, the result of the profiler is a percentage match, determining the "nature" of the message.

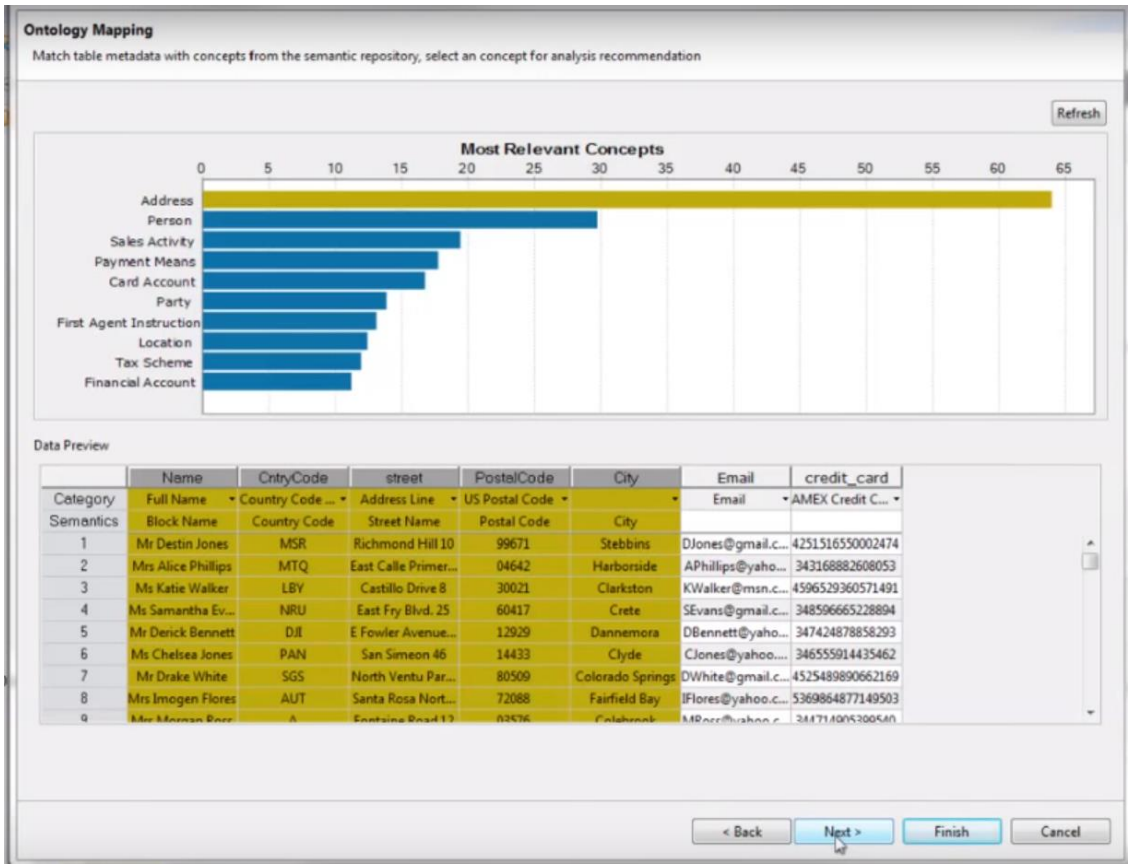

Figure 15: Example of Talend data profiler identifying the ontology of a given data set (Talend, 2015c, 01:46)

According to a report by Gartner (2015b), further common critical features of Data Quality tools include:

- **Generalized Cleansing**: The ability to update data using the tool to meet pre-defined data quality standards of the company.
- **Matching**: The ability to detect and create links between two heterogeneous data sets.
- **Monitoring**: The ability to observe data going through the Data Integration process to assure the pre-defined data quality level and to raise alerts in case the data quality standards are not met.
- **Issue resolution and workflow**: The ability to mitigate any discovered data quality issues by providing processes, interfaces and roles. Interestingly, this feature is very similar to the "Collaborate" feature of MDM Tools, as it is also offering workflows and roles for solving these issues (see chapter *4.2.3 Master Data* Management).
- **Enrichment**: The ability to use Reference Data (see chapter *4.2.3 Master Data* Management) to enhance given data sets.

4.2.5 Real Time Integration

As it has been mentioned many times in this thesis, having data from various sources available in whatever target has to happen quickly (see for example chapter *1 Introduction*). The frequency of a Data Integration flow largely depends on the requirements of the business case behind. Some may require that a data transfer within 24 hours is fast enough, while others might need the data to be transmitted within a few seconds. There are few different methods how to transport data from system A to system B. The two most common ones being "batch" and "real-time" data transfer. Batch data transfer usually transmits multiple data records from source A to target B based on a scheduled plan (for example once per night), while real-time data transfer usually sends only one record at a time and is triggered by an event in the source system (Nadhan and Weldon, 2004). It should be mentioned, that in Data Integration the word "near real-time" is often used. There is no clear definition on what "near real-time" really is[12], as it technically should not differ from a real-time data transfer. The differentiation threshold of "real time" to "near real-time" could be milliseconds, seconds, minutes or even more, depending on business and process.

However, as it was mentioned in chapter *2.5 Necessity of Data Integration*, a Data Integration middle layer should not cause a significant delay when transferring messages across applications. While the speed of the Data Integration middle layer largely depends on the architecture of the connected applications, many companies offer specific tools that help speeding up the real-time integration process. For example, one interesting piece of technology is offered by Informatica under the name "Ultra Messaging Streaming Edition" (Informatica, © 2016h).

Ultra Messaging is based on a flexible protocol called "SMX". The idea is that Ultra messaging creates a fast point-to-point connection (for example to the Data Integration Hub), by being deployed directly on the source application server. Ultra messaging provides an application programming interface for common programming languages (meaning that a lot of manual programming work is involved when using this tool), can communicate using any protocol (TCP/IP, UDP, RDMA, IPC) and can reduce the latency of transmitting messages down to less than 100 nanoseconds (Informatica, © 2016h).

4.2.6 Data Masking

Both Talend and Informatica advertise data masking as a strong security feature in their application portfolio (Informatica, © 2016i; Talend, 2015d).

---

[12] Near real-time is „Pertaining to the timeliness of data or information which has been delayed by the time required for electronic communication and automatic data processing. […] The distinction between near real time and real time is somewhat nebulous and must be defined for the situation at hand", ITS (1996).

The idea of data masking is to alter the stored data in a way that it becomes anonymous for anyone reading it, while still keeping the format of the data correct. This can help to protect sensitive customer information, meet governmental regulations or also to display data to a user in the right format, but with anonymized, masked structure. *Table 10* below shows two data rows, one being the original and one being masked.

Table 10: Example of Data Masking, before and after (Source: Author)

Type	Name	Credit Card	Address	Email
Unmasked	Jon Doe	4244487462024688	337 5th Street	JD@gmail.com
Masked	David Johnson	4724465412113708	880 XX Street	XXXX@gmail.com

Another usage of data masking is for example in test environments of Data Integration projects. Test environments can have different security constraints than productive environments, for example by making the data accessible to all developers. However, in order to properly test data on a test environment, some kind of data source is necessary. Data masking provides an easy way of taking a large amount of production data, anonymizing the information contained, but still keeping the structure so that tests are possible (SAP, © 2016h).

4.2.7 B2B Integration

The final feature that should be mentioned here, is B2B integration. Regarding the features it is very similar to A2A integration (compare chapter *3.4 Business2Business vs. Application2Application*), the tools given by both Informatica and Talend help in developing successful B2B solutions (Talend, © 2016e).

For Talend, there is no separate B2B tool available. Instead, the B2B functionality is given directly in the ETL Tool, as part of the mapping components in the template repository (Talend, © 2016j). Unfortunately, the “advanced B2B data formats” that are supported by Talend as part of the so called “tHMap” component are not available for free and only become available, once the company subscribes to one of the platform products of Talend (Talend, © 2016k).

For their B2B portfolio, Informatica is offering a tool that is not only similar to a normal Data Integration tool, but also offers a portal for the external partners. This can help a company to more easily create file exchange services or to enable new business partners to get integrated into the B2B network more easily (Informatica © 2016j).

4.3 Summary

The analysis in this chapter is based on the landscape of Data Integration and is trying to answer the question “What common functionality are companies offering?” by looking at their product portfolio in detail. As a result, seven main areas of core functionality (categories) have been identified: ETL Processing, Enterprise Service Bus, Master Data Management, Data Quality Analysis, Real Time Integration, Data Masking and B2B Integration.

All of these seven areas are equally strongly advertised, implying that potential customers seek tools in these specific areas. Some companies are also offering free versions, trial versions or (in case of Talend) even publish parts of their source code, which can help small business customers in growing faster and also generates desire for more products from the same company, as no freely or openly available tool covers all aspects or all functionality.

The most basic of all tools seems to be the ETL processing tool. This is even freely offered by Informatica (PowerCenter Express). This kind of tool solves the most fundamental of all Data Integration tasks: getting data from A to B. Running the ETL code modules on a centralized ESB comes next, as this is the most future proof of all solution approaches. Finally, and as it has been identified earlier, Master Data Management plays an important role in getting Data Integration right.

This implies that for carrying out Data Integration, having the right tools at hand is crucial to the success of a project. Chapter *5 A framework for developing Data Integration solutions for enterprise scenarios* goes more into detail regarding this point.

5 A framework for developing Data Integration solutions for enterprise scenarios

While the previous chapters discussed topic of Data Integration purely from a theoretical standpoint, this chapter connects Data Integration theory with the practice. This chapter contains a design of a Data Integration framework, based on theory and findings from previous chapters, as well as on interviews with experts from the Data Integration area. The chapter *5.1 Methods used for building a framework* summarizes all methods which are considered for building a framework and describes each of them in detail. Chapter *5.2 A Framework* for Data Integration designs the framework, based on all used methods, described in first part of this chapter.

5.1 Methods used for building a framework

Because the Data Integration framework should not be based only on one study as this would not provide multiple points of view and it would lead to less reliability of the model, the following four main sources are considered:

- Theory from chapters *2 Data Management and Integration* and *3 The landscape of Data Integration.*
- Findings in analysis of functionalities from four commercial solutions (SAP, IBM, Informatica and Talend), defined in chapter *4 Features of Data Integration* .
- Interviews with Data Integration experts and managers, who have worked at least 7+ years in the Data Integration area.
- As a basis for structuring the process model in the framework, the waterfall model has been selected.

These sources were chosen because on one hand it is necessary to specify basic theory of Data Integration, but on the other hand it is also relevant to have a look at real Data Integration solutions based on years of experience and to find common trends in functionalities. Speaking with experts of the field enhances the findings based on theory and shows what works for them in reality. Also, it is important that the Data Integration project should follow an established project structure, where all activities are categorised into suitable phases. For this reason, the waterfall model was chosen as a template.

5.1.1 Theory covered in previous chapters

In chapter *2 Data Management and Integration* and *3 The landscape of Data Integration*, the fundamental theory about Data Integration is described, like the context of Data Integration and Data Management, Integration levels, Advantages and Disadvantages, Integration architecture, Integration styles, as well as Data Standards. All these aspects are necessary to consider in the framework, because they are the basic attributes of each Data Integration solution. Therefore this chapter provides a short summary of what has been written before and provides an overview of all previously discovered findings. These conclusions have an impact on the Framework designed in chapter *5.2 A Framework for Data Integration*. All findings listed below are marked with a number in square brackets, so that the designed framework can be linked back to the findings.

Chapter *2 Data Management and Integration* defines according to the DAMA DMBOK2 Framework 11 Data Management areas, which cover all main challenges of Data Management. Data Integration belongs to these areas as well.

- [Finding 1]: *In the relationship between Data Integration and Data Management it is discovered that Data Integration cannot work without proper Data Management.*

Voříšek, et al. (2015) defines 5 main integration levels, where Data Integration together with hardware, software and user interface integration belongs to the technology level.

- [Finding 2]: *When looking at levels of Integration it was furthermore discovered that Data Integration needs to resolve syntactical and semantic dissonances between systems.*

Hohpe and Woolf define 9 main criteria which are needed to have a good Data Integration solution.

- [Finding 3]: *For having a good Data Integration solution, it is necessary that the company has a need for Data Integration, there is a low application coupling, low intrusiveness, inexpensive and easy to adapt technology, a standardized data format, timely data delivery, clear distinguishing of data and functionality, clearly defined communication layers, and high network reliability.*

Finally, 6 main advantages and 3 disadvantages are listed, giving an idea of what a company has to expect when choosing to build a Data Integration solution. Summarizing, the findings of these chapters are the following:

- [Finding 4]: *The main advantages of Data Integration are a common view of the data, decreased data redundancy, control over data synchronisation, an overall increase in data quality, creation of more "intelligent" systems and a reduction of workload of employees and company costs.*

- [Finding 5]: *The main disadvantages of Data Integration are that the effects of Data Integration are measurable only in mid- or long-term, the creation of a logical system dependency and the need for both a strong technical, as well as business skill set simultaneously.*

Chapter *3 The landscape of Data Integration* specified three Integration architectures (Gála, et al., 2006), which define architecture for communication between two and more systems.

- [Finding 6]: *Point to Point architecture is used mostly for small solutions, Hub and Spoke is most common one and Message bus is similar to Hub and Spoke, but contains more advanced functionality.*

Another aspect is the specific style of integration, where according to Hohpe and Woolf (2004) the following 4 Integration styles exist: File transfer, Shared database, Remote Procedure calls and Messaging.

- [Finding 7]: *File transfer, Shared database, Remote Procedure calls, are mostly used for small scale integration projects with less complexity and can involve "hardwiring" two applications. Messaging is considered as the optimal approach for a (large scale) Data Integration solution, as it avoids disadvantages of the other integration styles, however it is considered to be the most complex one at the same time.*

Chapter 3 also compares approaches in small and large companies. As this thesis is about large enterprises, the framework could be limited to those approaches only, however even large companies might be having some small scale integration issues, which do not always require a large scale solution.

- [Finding 8]: *Small solutions often have a higher need for Data Integration and prefer simpler integration styles, while large scale solutions are more complex but also better implemented and easier to maintain.*

Chapter 3 also highlights the difference between A2A and B2B and presents UN/EDIFACT as an example for an internationally recognized and widely used standard for electronic data interchange.

- [Finding 9]: *Defining or using existing format standards for a Data Integration solution makes the solution long lasting and durable. For example, a good standard like EDIFACT defines the usage of each field, the format of the exchange and the business use case in a flexible way.*

5.1.2 Functional aspects offered in Data Integration tools

In chapter *4 Features of Data Integration tools*, Data Integration tools which are currently available on market are analysed. For the analysis, tools from SAP, IBM, Informatica and Talend were considered. The main sources which are used for this analysis are whitepapers, video tutorials, marketing material, as well as a detailed examination of trial versions of chosen products and describing, how the functionality works in practice.

- [Finding 10]: *The analysed Data Integration vendors offer multiple Data Integration products, each covering a different set of functionalities, so the customers can chose the ones, which fit to their requirements (instead of offering one tool for everything).*

While doing the high-level analysis of products, it is discovered that all four companies offer a similar set of functionalities, based on which a list of categories is created, covering most common functionalities. These categories were analysed on a detailed level in solutions from Informatica and Talend.

- [Finding 11]: *When building a Data Integration solution the most common features provided by tools include: ETL Processing, Enterprise Service Bus, Master Data Management, Data Quality Analysis, Real Time Integration, Data Masking, B2B Integration.*

- [Finding 12]: *All analysed Data Integration vendors offer solutions having similar or identical functionalities, meaning that these functionalities are highly desired by customers. This also means that customers cannot expect fundamental technical differences, when choosing a particular top-vendor.*

5.1.3 Interviews

During the creation of this thesis, the author had a chance to participate in a large scale, technically advanced Data Integration project, which has been running successfully for several years. Therefore, another important point of this thesis is the gathering of qualitative feedback from key individuals of this project, to find out what their view of a successful Data Integration solution is – independently from looking at other theoretical sources.

In order to gather this feedback, 8 project members, having a minimum of 7 years of experience in Data Integration have been anonymously interviewed. The interview was conducted by sending out a list of questions via email. The project member could freely choose to reply to the questions via email or in form of an interview. In total, 4 of the 8 project members decided for the interview, while the other 4 decided to send their feedback in written form. The roles of the interviewed persons included (anonymized and summarized, some persons shared roles or carried out multiple of these roles):

- Vice President of Data Integration Project – full accountability of the Data Integration project
- Senior Data Architect – definition of mediated schema, as well as data standards
- Data Modeller / Data Mapper – documentation of schemas from connecting systems, definition of mapping to mediated schema, as well as data translations
- Senior Solution Support – monitoring of live Data Integration flows, error handling, solution coordination and deployment
- Project Manager – responsible for any changes to the Data Integration product, including connecting new systems
- External Consultant – can have any of the other roles, this role has been interviewed mainly due to experience in other Data Integration projects
- Data Platform Architect – responsible for the non-data related side of the Data Integration product (server, software, distribution, code or platform standards)

The full anonymized transcript of the interview questions and answers can be found in the *Annexes – Interview questions and full replies* of this thesis. The following list shows the interview questions that the participants were asked:

1. What do you think makes a good Data Integration project?
2. If you had to build a second Data Integration solution in another project, what would you change?
3. Can you name one example from the past years, what really improved in your project a lot? Or an example, what made it worse?
4. How do you think does your current project compare to other current market solutions, like IBM, Informatica, SAP, Oracle (or any other that you might know)? Do you think there are any differences at all?
5. Do you think that there is a technical aspect which makes your project stands out?
6. Do you think that the right tools make the right Data Integration solution?

Due to the nature of these open questions, many answers were given repeatedly across different questions, by different roles. To avoid repeating several points across these questions, the below list represents the most crucial, or most frequently given answers and findings by the participants. To structure the answers, they are divided into four main areas below.

Project Management and Scope

- [Finding 13]: In the beginning of a project, there needs to be a clear understanding of the business case, i.e. what should be achieved by the Data Integration.
- [Finding 14]: A Data Integration project needs to happen under realistic timescales. Too short timescales decrease the solution quality. When mapping System A to System B, 80% of this mapping might be easy and intuitive, however it is the last 20% which take the most time to get it right.
- [Finding 15]: Especially large companies should consider how much money they really want to spend on the Data Integration. If a company asks for having a solution which can do „everything“, they should not be surprised about the costs.

Standards

- [Finding 16]: Standards (Canonical Message Format, Platform Architecture, Code Modules, Data and Document Standards, definition of mandatory fields for a publish/subscribe mechanism) should be defined at the very beginning of the project. It does not necessarily make sense to have only one data model to define the entire business, because a change to the model later on can affect existing Data Integrations. High flexibility of the standards allow for high reusability, however they make governance more difficult. It makes sense to have one data model for each business area and subset models for each business operation.
- [Finding 17]: Defining a Canonical Message Format brings the ability to decode any message going through the bus into one generic standard. This has the advantage that one only needs to manipulate one side of the flow, if the system on that side has changed, the other interface stays stable.
- [Finding 18]: Standards should be followed 100% and always updated, when exceptions from the standard become regular. It also makes sense to revise the Data Integration solution regularly to bring all exceptions that piled up over time up to the latest standard.
- [Finding 19]: It brings advantages to build the integration code very modular, as this increases the reusability of the code. When implementing a new technical component in the Data Integration solution, one needs a clear design first to allow for maximum reusability.
- [Finding 20]: A large scale Data Integration project needs a clear link of mapping specification and implementation, so that for each code package it is clear from which specification it came and vice-versa. The company needs to make sure that the solution is clearly documented and not only in the heads of the people working for it.

- [Finding 21]: Standards should be documented in a meaningful way. The best thing is to have a few, concise guidelines, having less documentation but focussed on the most important content.

Tools

- [Finding 22]: A clearly defined toolset and people with the right skills to use these tools are needed. A project should try to avoid using only standard applications like Microsoft Excel rather rely on specific tools like Altova XMLSpy, IBM InfoSphere or SVN and GitHub for distributed development. Also, a Data Integration project should try to always use the latest technology as in long-lasting projects; technologies can grow old over time.

- [Finding 23]: Data Integration needs a stable platform on which it will run, with high performance, no delays or failures. It makes sense to make this platform distributed, so it can be scaled easily just by adding more servers. Also this increases the stability of the platform, in case one server fails.

- [Finding 24]: It makes sense to build integration tools internally, if the right skillset is present, as this can save a lot of costs, compared to buying a COTS solution. Plus the custom built solution will fit exactly to the companies' needs.

- [Finding 25]: When choosing Data Integration tools from external vendors, there are not many differences between these tools. The difference between these products is in the detail, but overall, they provide similar functionality. Products often get chosen due to preferences, marketing and also skill within the enterprise.

- [Finding 26]: If choosing a Data Integration tool, the aspects which companies look at are the software provider's ability to innovate, scalability of the product, the provider's performance, track record, provided support and price.

- [Finding 27]: It is important to keep track of messages going through the integration layer for trouble shooting, as the conversion history in case of failure needs to be clearly identifiable. A tool that allows end to end monitoring of the flows is highly useful for the live support.

- [Finding 28]: An Integration Platform, which will integrate many different systems at once, will have several projects running on them in parallel. Thus, a good tool for deployment is needed, that can take care of different concurring code versions and branches.

- [Finding 29]: An Integration Project can be largely improved when including proper Master Data Management. Moreover, there should be a tool that allows defining translation of data within messages (manipulation of message content) as they go through the integration layer, to cope with different sets of Master Data (e.g. system A sends "Czech Republic", system B expects to receive "CZ" in the same field).

- [Finding 30]: Right tools don't make the right Data Integration solution, but a good solution cannot be made by using the wrong tools. It is most important to have the right people with the right skills first, and then the right tools can be chosen. Tools are simply there to make the work of people easier.

Team Structure and Responsibilities

- [Finding 31]: Clear Data Governance is required from the top management downwards. When implementing new standards, overcoming reluctance of the staff to adhere to the new standards is one of the biggest challenges. Having top management on board, helps overcoming this challenge.

- [Finding 32]: A Data Integration Project will profit from a dedicated Architecture Design Team, something like a council of Architects that decides about each new feature of the Integration Platform. Every new requirement, newly connected system, every new flow needs to be approved by this group first. This brings a lot of stability and standardization to the platform.

- [Finding 33]: The business needs to stay accountable for the Data Integration and the IT supplier should not be accountable. The whole design and functional aspects of the solution should stay the responsibility of the company and not be outsourced to an external supplier, as the company then loses control over the entire design and technical specifications.

- [Finding 34]: Data Integration involves many roles, like mapping experts, database experts, network experts, system experts, architects (for platform and mapping), top management, developers or testing managers. It is crucial to have good communication between all knowledge areas to assure that projects go successfully. If possible, IT and business should sit physically very close together. Having project members located in different time zones adds to the challenge.

5.1.4 Waterfall model

[Finding 35]: The waterfall model is one of many models, which can be used in the software development lifecycle. The model was created by William Royce in 1970 and has been presented in various variations since then (even in the original document, Royce proposes some variants of the model (Royce, 1970)), however the phases remain very similar between each variation and the waterfall model until today is still one of the most popular IT project models (Balaji and Sundararajan Murugaiyan, 2012). This thesis specifies the process model for the framework as an intersection of various versions of the model (SLDC, 2011; BOEHM, Barry W., 1988, Smart, © 2006-2011).

The phases in the framework will be the following:

1. Requirements
2. Analysis
3. Design
4. Implementation
5. Testing
6. Deployment & Maintenance

The reason for selection of waterfall model is, that based on previous interviews and findings, a Data Integration project can be broken down into the phases of the waterfall model. However, this is not the only model which can be used for this framework, because there are also models like V-model, which can be adapted from the basis waterfall model by any user of the framework.

5.2 A Framework for Data Integration based on messaging

This chapter and its subchapters describe a framework consisting of steps and supplements for developing a Data Integration solution. It is build using the methods described in *5.1 Methods used for building a framework*. If it is unclear why this framework shall be used, the chapter *5.2.1 Part I – Introduction* contains an executive overview describing the usefulness of the framework.

5.2.1 Part I – Introduction

Definition

What is a Data Integration Framework?

This thesis sets the definition of a Data Integration framework like follows: A Data Integration framework should provide the basics of a framework, which (according to the framework definition in chapter *1.2 Goals, metrics, indicators and definitions of the thesis*) includes a structure and content, which might help an enterprise to build or revise a Data Integration solution (under the mentioned limitations). The structure of this framework can be found in Part II in form of the waterfall model and the content in this framework will be represented by “building blocks” which are explained in Part III.

Framework Scope

This framework focuses on the messaging bus integration style and architecture. As the earlier chapters described, there is more than just one integration style and architecture. This style and architecture has been selected, as it is considered the best approach for Data Integration and as it has been mentioned before, can suit best for large scale enterprises [see findings 6, 7, 8].

Structure of the Framework

The structure of this framework documentation classifies all previously mentioned aspects of Data Integration into logical objects, also referred to later as "building blocks". The composition, relationship and interactions of these objects or building blocks are governed by the Data Integration Development Method and described in the Supplements. *Figure 16* below shows the components associated to the structure of the framework.

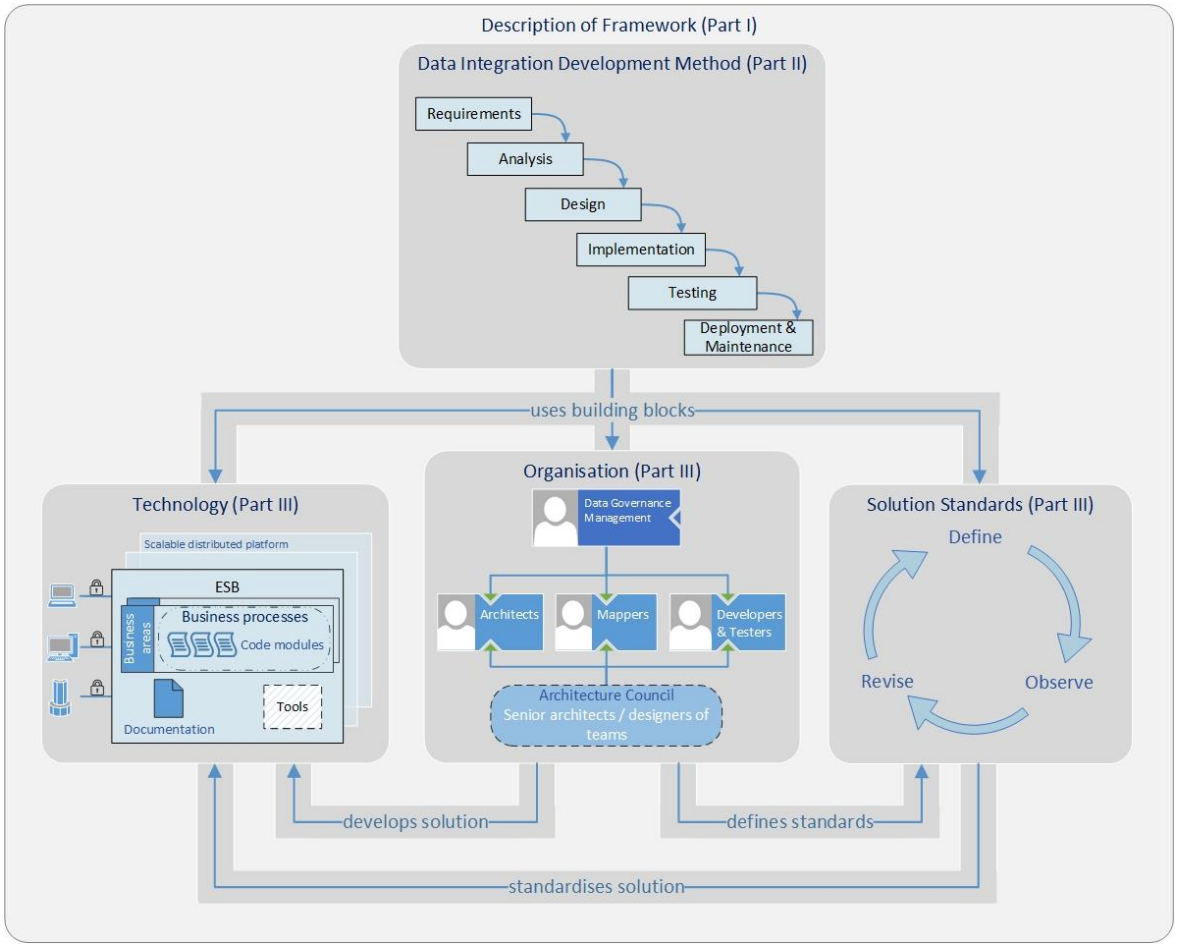

Figure 16: Structure of the framework

This framework is structured into three parts:

- Part I (Introduction) – This part contains a high level overview of the framework, describing the structure, purpose and limitations of the framework.

- Part II (Data Integration Development Method) – This part describes the lifecycle process of building a Data Integration solution step by step. Each step utilizes building blocks from Part III.
- Part III (Supplements) – This part describes the content of Part II in further detail, explaining building blocks of different steps of the process. These building blocks are:
    - Organisation – This building block describes the roles necessary in a Data Integration framework. Roles of the organisation define solution standards and develop the technological solution.
    - Technology – This building block describes the components of the Data Integration solution. The technological solution is built by the organisation and uniformed by the solution standards.
    - Solution standards – This building block describes regulations which should be considered when building a Data Integration solution. This building block is used to standardise the technological solution and executed by the organisation.

Limitations of the Framework

Missing Methodologies

This framework does not recommend any concrete methodologies for executing individual steps in the development method, however, it does contain generic examples in the supplements for different aspects of Data Integration. The reason for not providing (or developing) methodologies is that this has not been in the scope of this thesis, as it should only provide guidance (*Part II*).

Flexibility of this framework

This framework may be modified by its users to any specific enterprise needs. Any user of this framework needs to evaluate if this framework is suitable for structuring the planned Data Integration program.

If this framework is selected, it should be further modified to fit the company's culture, principles or business aims, for example by changing the used terminology to fit the company's vocabulary, or by removing unwanted parts or adding missing parts.

Executive Overview

Why do I need Data Integration?

As mentioned in *2.5 Necessity of Data Integration*, almost no company today has managed to have one single application in place that can cover all required functionalities. It is therefore a usual practice for many companies to have multiple applications in place. Making

these applications "talk to each other" (and this is what Data Integration is ultimately about) brings multiple advantages, as each system might know key information from other systems, without the user having to manually look up every piece of data [see findings 3, 4].

What is the added value of this framework for Data Integration and why should I use it?

So far there are no accepted Data Integration frameworks available on the market or in public sources. This framework is trying to change that.

This framework was developed based on the latest theory regarding Data Integration, considering functional aspects from real world solutions as well as experiences from experts working in one of the largest successful Data Integration projects worldwide. Due to the limitations of this framework, it might not be perfect in every situation, but it helps to develop a strong and proven long lasting foundation for current and future large scale Data Integration projects based on messaging. Furthermore it can be used to analyse existing projects for any potential gaps. As it is written in generic terms and not domain specific, it should be useful across many domain areas (for example economic, governmental or health).

Who would benefit from the Framework?

Any organisation that plans to implement a Data Integration solution or has an active Data Integration solution running, but considers reviewing it.

5.2.2 Part II – Data Integration Development Method

Overview

This part describes the process of creating and maintaining a working Data Integration solution. The Development Method is based on the waterfall model [see finding 35]. This framework highlights specific aspects discovered in the methods described before. These aspects can easily be underestimated or also be executed in the wrong phase. The Data Integration Development Method can also considered to be a part of the building blocks in Part III, however in order to avoid redundancy in this framework, it is only mentioned in Part II.

Each phase is divided into three segments:

- Objectives – defining what is the aim of each phase
- Steps – defining which steps should be taken to reach this aim
- Approach – explaining each step in detail

5.2.2.1 Requirements

Objectives

Develop the high level vision of what the Data Integration should enable to do, what value it should deliver and what is expected to be achieved from the Data Integration program.

Steps

1. Study and understand the business case for identification of requirements
2. Assure ability of business to run a Data Integration program
3. Set up the Data Integration lead
4. Identify roles and establish a Data Integration team
5. Specify realistic timescales
6. Assess company ability of developing Data Integration internally
7. Define business standards

Approach

1. Study and understand the business case for identification of requirements

This step should provide a clear understanding of what the company hopes to achieve by defining the requirements and should also consider possible disadvantages [see findings 5, 13]. The understanding should be advertised to the Data Integration team. Therefore it is necessary to put the Top Management in charge of Data Integration team [see finding 31].

2. Assure ability of business to run a Data Integration program

It might be that the business does not have the necessary competence to run a Data Integration or Data Management project. In that case it seems necessary, that the company builds this skill inside the company, either through training or through hiring, to have a strong link between the technological challenges of the Data Integration project and the business vision [see finding 30].

3. Set up the Data Integration lead

The Data Integration lead has to be tightly bound on the organisational level (see also chapter *5.2.3.1 Organisation building block*). A business leader should be put into the role of Top Management, who communicates with the IT Project Manager, in order to advertise a clear scope, including business requirements and budget. Also, business needs to stay accountable for Data Integration and not to transfer its accountability to the IT Management. As it was revealed in the interviews, it is a possible scenario that large scale Data Integration projects might fail, because business is giving the full accountability and control of the Data Management to an external vendor. When this happens, clear governance

on the vision, business relevant design decisions and approval authority get lost to the external vendor as well – which could lead to a fast deviation of the final product from the original vision [see finding 33].

Summarizing, the business should cover five main areas:

- Strong leadership – in order to decrease the reluctance of staff to accept change [see finding 31]
- Requirements – in order to clearly advertise the vision of the project [see finding 13]
- Scope – in order to prioritize which features are needed most [see finding 14]
- Accountability – in order to stay in full control of the IT project at any time [see finding 33]
- Budget – in order to have a clear financial frame for the execution of the project [see finding 15]

4. *Identify roles and establish a Data Integration team*

The organisation and roles of the Data Integration team should be set up in a way that it covers all the skills and requirements of a Data Integration program [see finding 34]. Further information can be found in *5.2.3.1 Organisation building block.*

5. *Specify realistic timescales*

One of the big mistakes is to specify timescales shorter than it should be since this would lead to big delays and solution quality decrease [see finding 14].

6. *Assess company ability of developing Data Integration internally*

When the company is able to develop Data Integration or parts of it internally, this will lead to saving of costs. If not, the right commercial Data Integration solution needs to be chosen [see finding 24].

7. *Define business standards*

It is necessary that a Data Integration project starts early with the definition of standards to be used. These standards include document standards, data standards, code module standards, technological standards. All these standards have to be derived from clear standards resulting from business processes, for example a standardised business data model [see finding 16]. For more information, see chapter *5.2.3.3 Solution standards building block.*

5.2.2.2 Analysis

Objectives

Define for both data and applications how the Data Integration solution will allow the business to work in order to fulfil the requirements.

Steps

1. Analyse requirements and context
2. Evaluate and choose tool set
3. Define Application and Data Standards
4. Reassess timescales

Approach

1. Analyse requirements and context

The requirements defined in the first phase need to be analysed and further detailed out. The aim is to gain a high level understanding of what should be designed in the next phase. It is also important to consider the context of the requirements, because there might be already an existing Data Integration solution, which requires analysing how the new requirements will fit into the architecture and what components can be reused [see finding 13].

2. Evaluate and choose tool set

The dedicated Architecture Design Council (see chapter *5.2.3.1 Organisation building block*) will select the appropriate tool set for the Data Integration and define the required functionality based on common features of Data Integration solutions (see chapter *4.2 Common functionality in detail* and [findings 10, 25]). It is important that this step is carried out in this phase and not before, as it was revealed in the interviews that the tools for designing the Data Integration should be selected by subject matter experts. As the tools are very similar to each other [see finding 12], the aspects which can be considered for choosing the tool are the software provider's ability to innovate, scalability of the product, the provider's performance, track record, provided support and price [see finding 26].

3. Define Application and Data Standards

High level solution standards (for example for the Canonical Message Format, Platform Architecture, Code Modules, definition of mandatory fields for a publish/subscribe mechanism) should be defined based on the business standards which were previously defined [see finding 16]. Further information can be found in chapter *5.2.3.3 Solution standards building block*.

4. Reassess timescales

Since the timescales were only defined on a high management level and it can have a critical impact on the project's success to have unrealistic time scales, they should be reassessed by subject matter experts once a high level design has been created [see finding 14].

5.2.2.3 Design

Objectives

Develop a detailed low design for the logical and physical implementation of the Data Integration solution, covering applications and data.

Steps

1. Create low level, detailed system specifications
2. Create detailed data standards and models
3. Create mapping, master data and translation specifications
4. Define test cases

Approach

1. Create low level, detailed system specifications

Specify the technology and solution standards to use, how to set up the server landscape, define correct sizing and implement tools [see findings 22, 23]. For more information, see chapter *5.2.3.2 Technology building block*.

2. Create detailed data standards and models

This design phase provides detailed specification of the Canonical Data Structure, which is going to translate data from one system to another and how the data models and standards for each business process look like [see finding 17].

3. Create mapping, master data and translation specifications

The low level field-by-field transformations between source and target systems are defined in this step. Furthermore, this step should collect master data from the connecting systems and define translation between the sets of master data, if there is any difference [see finding 2].

4. Define test cases

Since the previous step ultimately defines how system A is going to talk to system B, the definition of clear test cases for evaluating the syntactical and semantic correctness of the communication should be done in this step [see finding 35].

5.2.2.4 Implementation

Objectives

Develop the Data Integration solution based on the previously designed specifications according to business priorities towards the target solution using previously defined solution standards. Also change requests need to be covered by this step.

Steps

1. Implement physical data models and code
2. Realign documentation and solution standards

Approach

1. Implement physical data models and code

This step implements the design from previous phases, by technically defining the physical data models and creating the actual integration code. It brings advantages to build the integration code very modular, as this increases the reusability of the code [see finding 19]. For more information see chapter *5.2.3.1 Organisation building block.*

2. Realign documentation and solution standards

Realign the previously defined mapping and standard specifications. There might be some changes to the design, because every design may have some gaps which will only become visible during the implementation.

Solution standards should be monitored by the Architecture Design Council and each deviation from the standard should be handled by a proper design through the previous phases. If exceptional designs become the standard, the solution standards need to be revisited (see chapter *5.2.3.3 Solution standards building* block and [finding 18]).

5.2.2.5 Testing

Objectives

Test the previously built solution and work cooperatively to eliminate any issues. Create evidence that the solution is working [see finding 35].

Steps

1. Test internally
2. Test end to end and execute user acceptance testing

Approach

1. *Test internally*

Ideally, first all implemented components are tested independently from each other on a separate environment. If modules have been standardised, the usage of automated tests seems appropriate.

2. *Test end to end and execute user acceptance testing*

The cooperation between the implemented components is tested by doing end to end testing, which means testing a message flow from source through middle layer to target. It makes sense to involve the respective system owners in such a test, as the message can be tracked from end to end. When any of the testing fails, it needs to be fixed in implementation and tested again. In case all tests are successful, the result is being documented.

5.2.2.6 Deployment & Maintenance

Objectives

Make the solution available for productive usage. Ensure that the Data Integration is running without errors, enabling business to fulfil its needs.

Steps

1. Deploy solution to production
2. Monitor Data Integration flows closely and patch issues

Approach

1. *Deploy solution to production*

When everything is tested successfully, the product can be released to production (RTP) which means that the integrated systems can start using the Data Integration. As a large scale Data Integration project may have multiple developments in place at the same time, the usage of a deployment tool might be necessary [see finding 28].

2. *Monitor Data Integration flows closely and patch issues*

This step monitors the deployed flows for any issues. In case of any errors, a detailed analysis needs to be created and routed to the specific teams to fix it [see finding 27].

5.2.3 Part III – Supplements

Overview

This part contains descriptions of the three main building blocks of the framework: organisation, technology and solution standards. Each of the building blocks contains a structured overview of how it can be implemented and a detailed description of how the components work in the context of the framework.

5.2.3.1 Organisation building block

This building block describes which roles need to be set up in a Data Integration environment. Each of the roles is described in detail. Roles might be broken down into further roles and it is also possible that a single resources covers multiple tasks from other roles or even has more than one entire role assigned to itself.

Implementation

Figure 17 below shows an overview of the generic roles required to set up a Data Integration team.

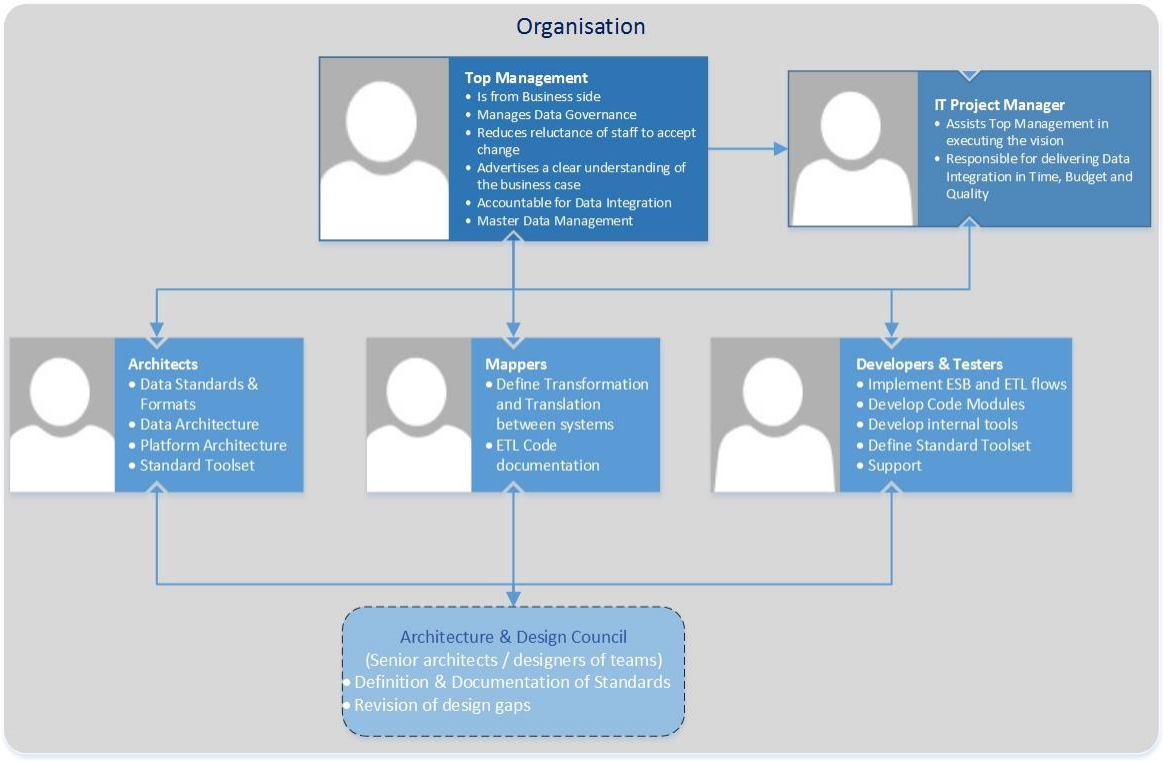

Figure 17: Roles of the organisational building block in detail

General finding

To assure good communication between each team/employee it is very important that IT and business department sit close to each other and avoid being located in different time zones since this makes the communication more difficult [see finding 34].

Components

Top Management

As Data Integration has to happen from top down and be within the accountability of the business, it makes sense to put the business into the Top Management role of a Data Integration project [see finding 33]. From a Data Management perspective, business needs to take care of providing a clear Data Governance and Master Data Management [see finding 1]. The Top Management should also help to reduce reluctance of staff for the acceptance of change. This is because (as it turned out in the interviews), Data Integration project often come along with an organisational change in large enterprises [see finding 31]. The Top Management should also advertise a strong understanding of the vision and business case, or "what they hope to achieve by Data Integration" [see finding 13]. Master Data Management will also involve the business directly. It will not necessarily be executed by the same person, but has to be governed on a hierarchical level with the necessary decision privileges.

IT Project management

The responsibility of delivering the Data Integration solution is with an IT Project Manager. All roles from IT Project Manager and below can be either internal or external roles, depending on the company's budget and skills [see finding 34].

Architects

Architects maintain Data Standards and Formats, design the Data and Platform architecture and define standard toolsets, used for Data Integration as well [see finding 34].

Mappers

Mappers define transformation and translation of data which is sent between two systems and are essentially responsible for documenting the ETL code through the mapping definition [see finding 2].

Developers and Testers

Developers and Testers implement the ESB and ETL flows and develop reusable code modules for the ETL flows, as well as internal tools. They are also responsible for assuring the production support [see finding 27].

Architecture & Design Council

An Architecture & Design Council consists of people from different roles, to decide about the high level architecture and to approve each amendment to the data model, solution standards, code modules or other global design decisions. Each decision from all phases of the process model should run through this council. Having this team can bring a multiple advantages like stability and standardization to the platform [see finding 32].

5.2.3.2 Technology building block

This building block describes which technical components can be part of a technological Data Integration architecture. The components described below are optimized for a messaging bus solution (see also chapter *5.2.1 Part I – Introduction* for restrictions on this framework).

Implementation

Figure 18 below shows the components of the technological Data Integration solution. The overview represents a generic architecture to explain how to structure the components.

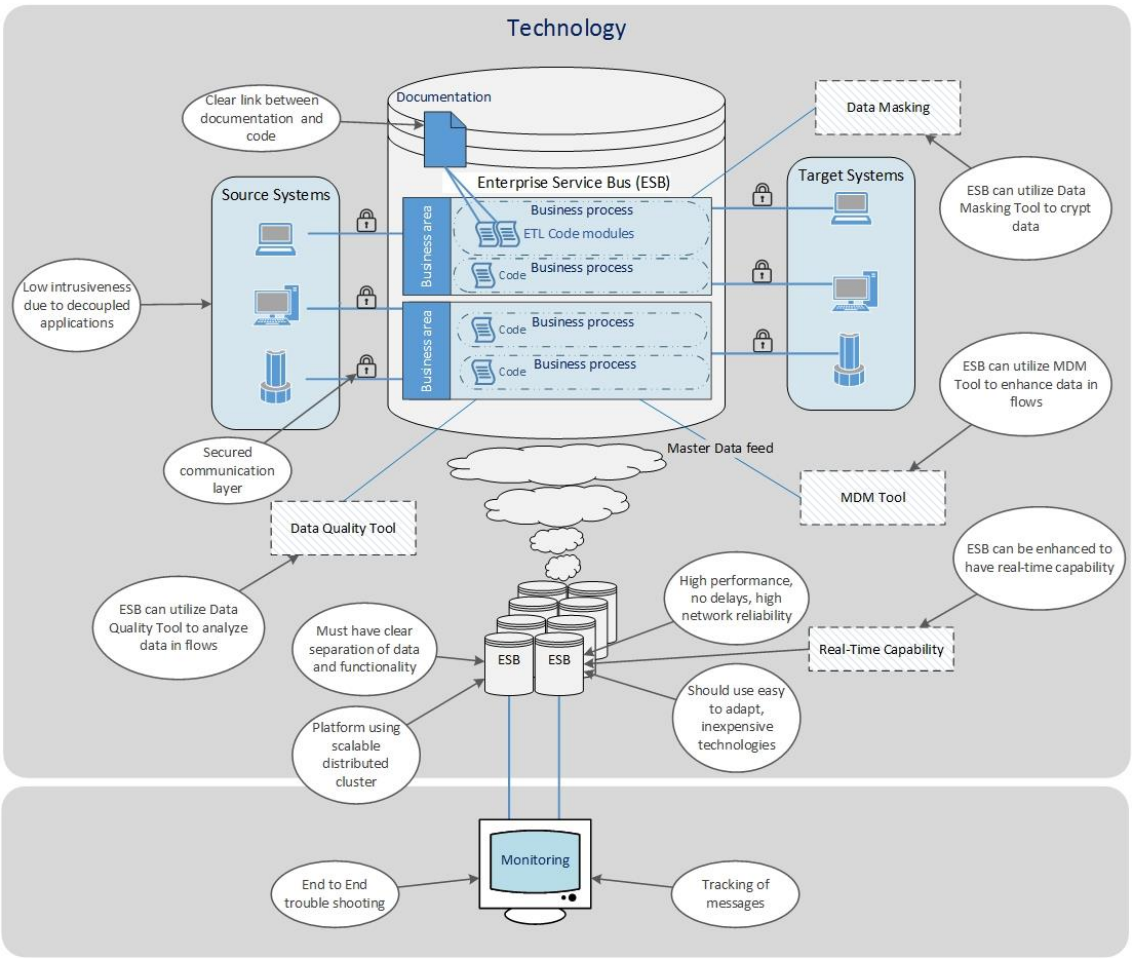

Figure 18: Components of the technological building block in detail

Components

ESB platform

Ideally, the Data Integration platform (if an ESB is going to be used) should be a distributed platform, where servers can be easily added if performance needs to be scaled up according to needs. Also if one of the servers fails, this will not affect the entire system [see finding 23]. There is a wide variety of tools available for implementing an ESB, see also chapter *4.2.2 Enterprise Service Bus*.

Business Areas and Business Processes

Inside the Enterprise Service Bus, the actual integration code should be clustered by multiple business areas like Finances, Customs, CRM, etc., and each of the areas should be clustered into business processes like customs filing, account receivables/payables, account creation, etc. This way, the integration platform can assure that each service is receiving and delivering data that is expected from it – for example stock exchange data is sending stock exchange data and weather data is sending weather data. Each business process contains the ETL code modules, defining exactly what should each business process do [see finding 16].

ETL Code modules and documentation

Each code module should be clearly documented and there must be a clear link between code and documentation, so that the origin of requirements remains clear [see finding 20]. Each module should be assessed for standardisation (see chapter *5.2.3.3 Solution standards building* block) by the Architecture & Design Council (see *5.2.3.1 Organisation building block*)

Secured Communications Layer

According to chapter *2.4 The role of Data Integration in the context of Data Management*, Data Security is a vital part of a Data Integration solution. One viable solution for this is to secure the endpoints of the bus against outside threads. Source and Target systems communicate with the Data Integration layer through a secured communication layer. Data Integration solutions offer multiple templates for connecting to secured channels, like secured DB connection, sFTP or MQ or secured web services (see chapter *4.2.1 ETL* Processing).

Decoupled Applications

Because the ETL code is purely hosted on the platform, the applications do not need any integration code themselves, but only need to provide connecting end points. This causes low intrusiveness into the applications so that the middle layer as well as the applications do not have to change with each new requirement [see finding 3].

Real-Time Integration

Ideally, each Data Integration solution should be implemented without large delays in the message transmission. In case the business scenario calls for a very fast Data Integration architecture, allowing for a real-time data transfer, additional tools can be used to speed up the architecture (see also chapter *4.2.5 Real Time Integration* and [finding 11]).

Master Data Management Tool

An optional part of the technological aspect is the inclusion of a Master Data Management tool to get access to Master Data, so once the data from Source should get transformed to Target, Master Data specifies the translation from Source to Target (see also chapter *4.2.3 Master Data Management* and [finding 29]).

Data Quality Tool

In order to improve the Data Quality, the ESB can be optionally connected to a Data Quality tool which can analyse specific messages going through the bus and can help with identifying and mitigating Data Quality issues. In combination with the MDM tool, this provides a strong foundation for assuring high data quality in all connected systems (see also chapter *4.2.4 Data Quality Analysis* and [finding 11]).

Data Masking

In case any legal or security related constraints identified in the project, specific flows can be manipulated using a Data Masking tool, to assure an information-rich flow of data, while protecting the data from outside threads (see also chapter *4.2.6 Data Masking* and [finding 11].

Monitoring

Each ESB usually provides a Monitoring feature, which is being used in process of Testing as well as RTP and Support. Monitoring enables to track all activities, going through each data flow. Because there might be millions of messages going through the Data Integration Layer, monitoring should ensure that when there is some error, people are able to find the specific error based on message content and are able to explore the entire data history from entering the bus to leaving it [see finding 27].

5.2.3.3 Solution standards building block

This building block describes which solution standards can be part of a Data Integration solution. The topics addressed are recommendations for users of the framework, which topics they need to address for the standardisation of the solution.

Implementation

Figure 19 below shows a generic model of a standardisation lifecycle and its contents.

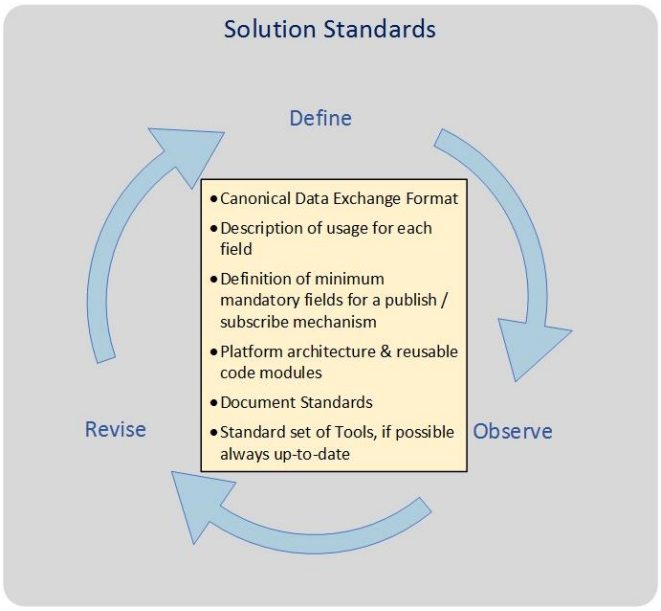

Figure 19: Components of the standardisation building block in detail

Components

Standardisation lifecycle

The *definition* of solution standards starts in the preliminary phase of the Development Method. Due to its early inception, it is necessary to regularly *observe* the usage of the solution standards during each phase and keep the solution standards up to date by *revising* them in each phase [see finding 18].

Responsibility for solution standards and documentation

Solution standards should be in the responsibility of the Architecture & Design Council (see chapter *5.2.3.1 Organisation building block*) or a similar group of subject matter experts [see finding 32]. The documentation of solution standards should also be standardised, so that the information contained in the standardisation guidelines is provided a meaningful way without flooding the company with documentation which will never be read again [see finding 21].

Canonical Data Exchange Format

Since the connecting applications can provide their data in many different formats, it is difficult to standardise for example reporting or test automation on each newly connected format. Introducing one canonical format for the Data Integration layer brings the advantage of further decoupling applications from each other (every message needs to translate to the canonical format when entering the bus and is translated back to a local format

upon leaving the bus). There should be one format defined for each business area and operation [see findings 9, 17].

Description of usage of each field

The standardisation of the Data Exchange Format requires a thorough understanding of each field and its meaning for the company, to allow for a standardisation across all connecting applications [see finding 16].

Definition of minimum mandatory fields for a publish/subscribe mechanism

To fully decouple applications from each other, the usage of a publish/subscribe mechanism helps to shift the view of Data Integration from point-to-point to a service oriented approach. If one application wants to share data, it simply publishes a pre-defined minimum set of data to the Data Integration bus and any other application may or may not decide to subscribe to this information [see finding 16].

Platform architecture and reusable code modules

The architecture and each of the code modules creating the technological solution in chapter *5.2.3.2 Technology building block* needs to be standardised to allow for higher reusability and easier maintenance in the long term [see finding 19].

Standard set of tools

The tools for governing the Data Integration program and the tools for implementing the final solution need to be assessed, chosen and standardised, so that the company will profit from a streamlined set of licenses being used with highest cost-to-value ratio [see finding 22].

5.3 Summary

In this chapter, the framework for Data Integration based on messaging is created based on four specific methods, each providing a set of numbered findings, which were used to derive specific aspects of the framework. The table below summarizes all listed findings from chapter *5.1 Methods used for building a framework* and the components of the framework (beginning in chapter *5.2 A Framework for Data Integration based on messaging*), where the findings were used and also the reason for the usage in this component.

Table 11: Overview of findings and their usage

[Finding number]	Influenced components
[Finding 1]	Part III – Organisation building block: Top Management Part III – Organisation building block: Top Management
[Finding 2]	Part II – Design: 3. Create mapping, master data and translation specifications Part III – Organisation building block: Mappers
[Finding 3]	Part I – Executive overview: Why do I need Data Integration? Part III – Technology building block: Decoupled Applications
[Finding 4]	Part I – Executive overview: Why do I need Data Integration? Part II – Analysis: 4. Reassess timescales
[Finding 5]	Part II – Requirements: 1. Study and understand the business case for identification of requirements
[Finding 6]	Part I – Framework scope
[Finding 7]	Part I – Framework scope
[Finding 8]	Part I – Framework scope
[Finding 9]	Part III – Solution standards building block: Canonical Data Exchange Format
[Finding 10]	Part II – Analysis: 2. Evaluate and choose tool set
[Finding 11]	Part III – Technology building block: Real-time Integration Part III – Technology building block: Data Quality tool Part III – Technology building block: Data Masking
[Finding 12]	Part II – Analysis: 2. Evaluate and choose tool set
[Finding 13]	Part II – Requirements: 1. Study and understand the business case for identification of requirements Part II – Requirements: 3. Set up the Data Integration lead Part II – Analysis: 1. Analyse requirements and context Part III – Organisation building block: Top Management
[Finding 14]	Part II – Requirements: 3. Set up the Data Integration lead Part II – Requirements: 5. Specify realistic timescales

[Finding 15]	Part II – Requirements: 3. Set up the Data Integration lead
[Finding 16]	Part II – Requirements: 7. Define business standards Part II – Analysis: 3. Define Applications and Data Standards Part III – Technology building block: Business Areas and Business Processes Part III – Solution standards building block: Description of usage of each field Part III – Solution standards building block: Definition of minimum mandatory fields for a publish/subscribe mechanism
[Finding 17]	Part II – Design: 2. Create detailed data standards and models Part III – Solution standards building block: Canonical Data Exchange Format
[Finding 18]	Part II – Implementation: 2. Realign documentation and solution standards Part III – Solution standards building block: Standardisation lifecycle
[Finding 19]	Part II – Implementation: 1. Implement physical data models and code Part III – Solution standards building block: Platform architecture and reusable code modules
[Finding 20]	Part III – Technology building block: ETL
[Finding 21]	Part III – Solution standards building block: Responsibility for solution standards and documentation
[Finding 22]	Part II – Design: 1. Create low level, detailed system specifications Part III – Solution standards building block: Standard set of tools
[Finding 23]	Part II – Design: 1. Create low level, detailed system specifications Part III – Technology building block: ESB platform
[Finding 24]	Part II – Requirements: 6. Assess company ability of developing Data Integration internally
[Finding 25]	Part II – Analysis: 2. Evaluate and choose tool set
[Finding 26]	Part II – Analysis: 2. Evaluate and choose tool set

[Finding 27]	Part II - Deployment & Maintenance: 2. Monitor Data Integration flows closely and patch issues Part III – Organisation building block: Developers and testers Part III – Technology building block: Monitoring
[Finding 28]	Part II - Deployment & Maintenance: 1. Deploy solution to production
[Finding 29]	Part III – Technology building block: Master Data Management Tool
[Finding 30]	Part II – Requirements: 2. Assure ability of business to run a Data Integration program
[Finding 31]	Part II – Requirements: 1. Study and understand the business case for identification of requirements Part II – Requirements: 3. Set up the Data Integration lead Part III – Organisation building block: Top Management
[Finding 32]	Part III – Organisation building block: Architecture & Design Council Part III – Solution standards building block: Responsibility for solution standards and documentation
[Finding 33]	Part II – Requirements: 3. Set up the Data Integration lead Part III – Organisation building block: Top Management
[Finding 34]	Part II – Requirements: 4. Identify roles and establish a Data Integration team Part III – Organisation building block: General finding Part III – Organisation building block: IT Project management Part III – Organisation building block: Architects
[Finding 35]	Part II – Overview Part II – Design: 4. Define test cases Part II - Testing

6 Applying the framework to a Data Integration Project (DIP)

This chapter puts the framework to test by examining a working and successful real world solution and by overlaying it with the created framework to demonstrate how the framework might be applied on a real world solution.

The author of this thesis had the chance to participate in a large scale Data Integration project for almost 2 years and was actively participating in building new flows, connecting new applications and supporting projects end to end. Due to restricted information, the presented Data Integration solution in this chapter is anonymized and only presented on a high level.

This chapter examines the real world example based on Data Integration Building Blocks which have been defined in Part III of the Framework, chapter *5.2.3 Part III – Supplements*. Chapter *6.1 Introduction to project DIP*, presents each building block of the solutions independently from the framework. Chapter *6.2 Differences between DIP and framework* goes through all the presented building blocks and uses the framework building blocks to analyse, if there are any gaps between the project and the framework. The reason why only the building blocks are used for analysing DIP is that the Data Integration Development Method is specified based on a generic waterfall model. The framework does not have the sources or scope to specify concrete methodologies which would be recommended in a Data Integration project and is therefore not part of the analysis.

Chapter *6.3 Using the framework to resolve the differences* will demonstrate how to solve gaps identified in the previous chapter by applying the Data Integration Development Method (*Part II*) specified in the framework.

Chapter *6.4 Expert review* contains results from a peer review of chapters *6.2 Differences between DIP and framework* and *6.3 Using the framework to resolve the differences*. Finally chapter *6.5 Summary* provides a resume of the framework test.

6.1 Introduction to project DIP

The project “DIP” is an (anonymized) project which stands for “Data Integration Project” with the aim of helping to completely renew the application landscape of the company “CDI” (“Company Doing Integration”). CDI is a worldwide operating company using a multitude of systems and applications in several countries. CDI has been trying for many years to introduce a new set of applications to run the company in a more controlled, centralized and globalized approach: Instead of having each country run their own set of applications, a set of standardised global applications should be used worldwide. This new

set of applications should cover a wide range of products, for example CRM, Finance, Customer Feedback Platform or a shopping portal. All of these applications should be fully integrated with each other to allow for a seamless work with new technology in all countries.

This project has been running in CDI for several years now. As CDI has been trying to connect a lot of systems with each other, a strong Data Integration platform for connecting all mentioned systems was needed – when the application replacement project started, the idea of DIP was born. DIP is supposed to be the new Data Integration platform for CDI, which should not only handle the integration of the new project but also cope with any new requirements that might come up in the future.

As every Data Integration project, DIP has faced a lot of challenges during its development over the years. But today DIP has evolved to a state where it can proudly present itself as a system which:

- Is the CDI A2A main integration platform
- Integrates any protocol, any format, any messaging pattern
- Runs 24/7 with a Disaster Recovery

According to this thesis' classification of Data Integration solutions (see chapter 3 *The landscape of Data Integration*), DIP qualifies as a large scale A2A messaging bus solution based on external technology (using a third party integration hub and ETL tools).

6.1.1 Organisation

The DIP project team organization consists of nine teams. These teams are responsible for taking care of any new Data Integration project that gets started. As DIP is already an existing and stable platform, the teams' primary tasks focus on getting the requirements from new systems that should be connected, designing the solution and implementing the Data Integration flow. The primary tasks, which the teams deal with, can be split into four groups:

- Transform – defining how data structures should change between system A and B
- Translate – defining how data content should change between system A and B
- Routing – defining to which system(s) a message should get delivered, after it got received on the Data Integration bus
- End Points – defining how each system should be connected to the messaging bus

The below *Figure 20* shows how the nine teams are set up, what their primary task is and which of the above aspects they usually deal with:

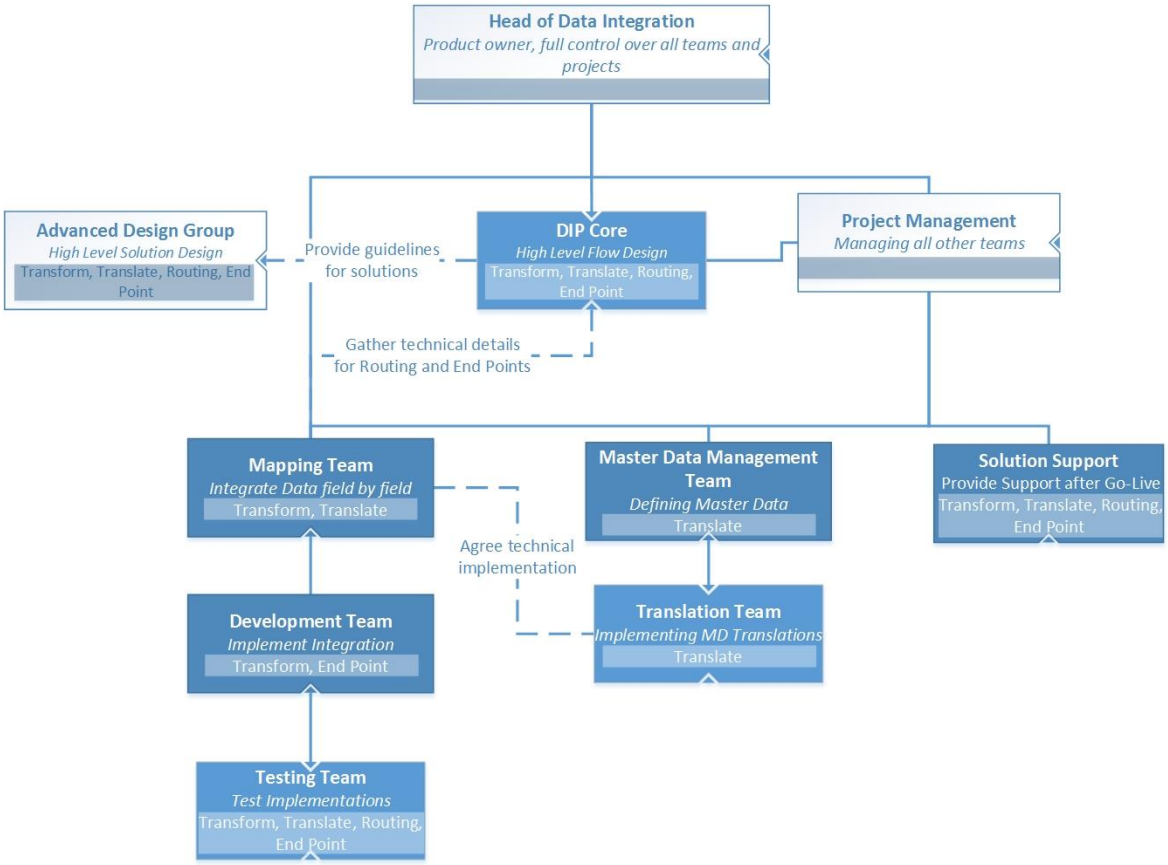

Figure 20: DIP Team structure, primary tasks and relationship to each other (Source: Author)

Head of Data Integration

The Head of Data Integration is a business person, fully accountable of making the Data Integration solution work. All major platform decisions are usually approved by this person, all priorities for each project are defined by this person and all projects risks need to be raised to him / her.

Project Management

The project managers are responsible for starting new projects and coordinating them until the release to production. They usually gather high level information like which system, which scenario, how much data, how complex the integration logic and until when the project needs to be executed.

DIP Core

The DIP core team (basically consisting of managing developers and architects) has to gather the end to end technical requirements for the project. How and how many end points will be connected, what are security constraints, what data loads will be expected, via which protocol will the systems connect to each other, what are connection addresses and credentials. Furthermore this team identifies the "kind" of flow that is being set up here, e.g. "is this a flow for finance or for CRM"? The DIP Core Team usually works together

with the Mapping Team to gather the detailed requirements. The decisions made by this team define the high level architecture of each flow and influence the transform and translation logic.

Advanced Design Group Team

DIP is a living product and there might be edge cases, where the requirements of new systems exceed that standard approach of integration which is covered by DIP. In such a case, the ADG team is called to action, consisting of the top IT architects from all mentioned groups. They do an assessment of the requirement, discuss possible solutions and propose new functionality for DIP, if needed, which will then be handled by the Core Team.

Master Data Management Team

Each newly connected system might come with their own set of master data for potentially identical business objects (e.g. customer data). It is in the domain of the MDM team to collect the master data, the requirements and the translations from one set of master data in system A to a different set of master data in system B.

Translation Team

The Translation Team is the technical support group for the MDM team. They implement the requirements of the MDM team by recording and documenting enumerations of master data, as well as implementing the actual translation tables.

Mapping Team

The Mapping Team represents the “brain” of the entire Data Integration platform. Whenever a new project comes in, the Mapping Team will start documenting all flow relevant information in so called “Paper Maps”. The creation of these maps is one of the most complex tasks of the project, as it requires technical understanding, as well as in-depth business knowledge to understand all consequences of a mapping. Another task of the mapping team is to gather requirements for the routing rules (i.e. which fields of the message can determine the target system) and work with the DIP Core team to find the appropriate Semantic Service for implementing the rules. The primary task of the Mapping Team is therefore to provide the transformation rules for every message field by field and defining whether the translation service for this field should be used or not. They also work together with DIP Core to gather detailed information regarding end points.

Development Team

The Development Team consists of a large pool of resources with technical knowledge for implementing the ETL solution. The basis for the implementation are the previously mentioned Paper Maps, which tell the developers field by field, what they need to do. The developers do not have any architectural or design responsibility. In case of any discrepancies during the implementation, they always have to go back to the Mapping, Core or ADG

team and ask for the solution. While developers mainly implement the transformations within flows, they are also responsible for setting up the technical end points.

Testing Team

The Testing Team is also a large pool of tester resources, which collect messages from the sending system, processes them through the Quality Assurance (QA) instance of DIP and inspect the results for possible failures. The testing will also include direct connections to the integrated systems and test the scenario end to end together with the system owners. In case of any failures during the test, the Testing Team has to work with Mappers, Developers, the MDM, Translation and the DIP Core team to resolve the issues.

Solution Support

Once the project has been finished, the flows through DIP are being monitored and supported by the Solution Support team. Whenever an error occurs, the Solution Support team will work with the available documentation and code or with the former project team members to find solutions for the problems and implement them.

6.1.2 Technology

The overall aim of DIP is to reduce point to point integrations and replace them with one integration bus in the middle, which allows several applications to connect to each other, based on a specified business processes.

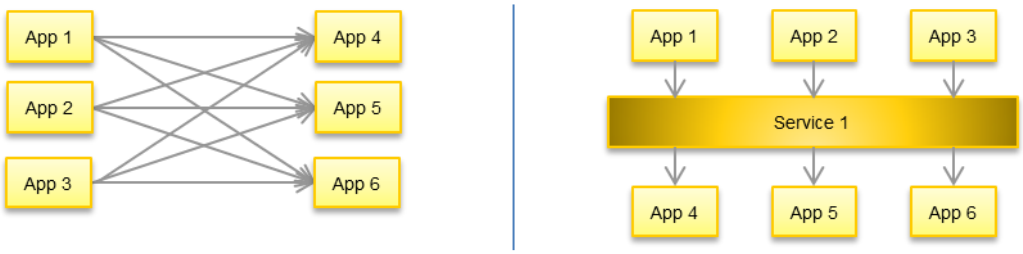

Figure 21: Point to Point vs. Message Bus Integration (Used with permission of CDI)

Even though this doesn't necessarily lead to a reduction of flows between applications, the big advantage of DIP is reusability. Since the integration code is placed in one big Data Integration platform (instead being scattered around between applications), it is not necessary to rewrite large parts of code with each newly integrated system. To enable the reuse of parts, the high level DIP architecture is split into "Services". In general, there are three types of services, which DIP is using to carry out Data Integration:

- **Application Service (AS)** – collects, receives or sends data from or to a connecting system, transforms it into or from the "Canonical Data Model" (CDM, see below) and carries out value translation, if necessary

- **Gateway Service (GS)** – same functionality as AS, but used mostly for Integration with B2B gateways or other Integration Platforms (this makes DIP fully compatible with every kind of integration, as it can connect to all locations)

- **Semantic Service (SS)** – receives a CDM message from an AS/GS, processes it using the specified business process and routes it to the outbound AS/GS based on previously specified routing rules

One rule of DIP is that every fully integrated flow passing through DIP has to be translated into the "Canonical Data Model" or "CDM". The CDM defines a fixed XML structure in form of a huge XSD (containing approx. 10,000 fields) which documents the data architecture of the business. CDM messages are XML messages, which follow this XSD. The CDM is in constant development and receives updates almost with each new requirement or connecting system – while still maintaining full backwards compatibility.

Figure 22 below shows how these services and message formats work together (and also points out further key architectural components):

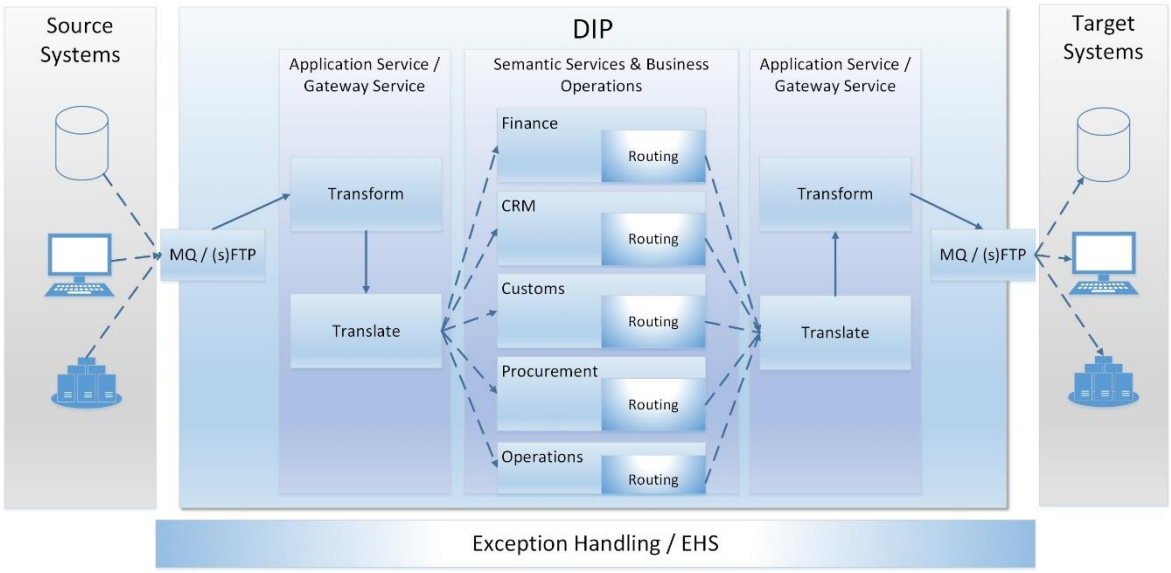

Figure 22: The DIP System Architecture

Figure 22 above summarizes all technical key architectural components and typical functionality, which are necessary for this Data Integration platform to work. To summarize the components, these are (from end to end):

- **Message Queue / (s)FTP (Inbound)** – every message needs to enter the DIP message bus through the use of an end point. A message queue or a (s)FTP are good, generic solutions to assure this. The DIP environment supports both PUSH and PULL services.

- **AS/GS Transformation (Inbound)** – each message going through DIP needs to be translated from whatever source format it is in, into the CDM format. The transla-

tion are usually specified field by field in a technical document (so called "Paper Maps") and then implemented using the ETL Tool.

- **AS/GS Translation (Inbound)** – each system might be using different master data for the same semantic field. The values for this master data are translated here to canonical values.

- **Semantic Services (SS) & Business Operations** – each Semantic Service represents a business area and within each service there are operations which represent a business scenario. For example "Finance" (which is a business area), contains the integration code for all flows that send invoices or customer account information (business scenarios).

- **SS Routing** – the routing services define a set of rules for specific CDM fields, in order to decide to which system a message should get routed. For example, some CDM files for "Operations" will contain a field called "Postbox", containing information about the receiving system. The routing rules read information directly from the CDM (usually from a specific envelope or header section) to decide where a file should be routed to.

- **AS/GS Translation & Transformation (Outbound)** – analogue to the inbound services, the outbound side has to translate a CDM message back to the receiving format and also needs to translate the values inside the message into a consumable form.

- **MQ / (s)FTP (Outbound)** – end point for the receiving side, same as the end point of the sending side.

- **Exception handling / EHS** – the Error Handling System (EHS) catches all errors which might occur during a flow and puts them out to the monitoring tool. These errors can be exceptions (code crashes) or defined errors in business logic, e.g. empty fields, translation errors or validation rule failures. EHS has been integrated with the CDI ticketing system and is capable of raising incident tickets to the resolver groups directly

6.1.3 Solution standards

Project DIP has a high level of standardisation. As it was mentioned earlier, DIP has a dedicated Advanced Design Group (ADG) which can make decisions about each technological aspect that requires standardisation. The Advanced Design Group of DIP is following a passive design approach: Whatever is considered as a design gap can be raised to the team using a separate website. Alternatively, ADG is called for help during the course of integrating new systems. The team investigates gaps which are raised to them once per week and provides the results back to the requestor. *Figure 23* below shows the ADG, the solu-

tion standards which they regularly assess and the process which triggers such an assessment.

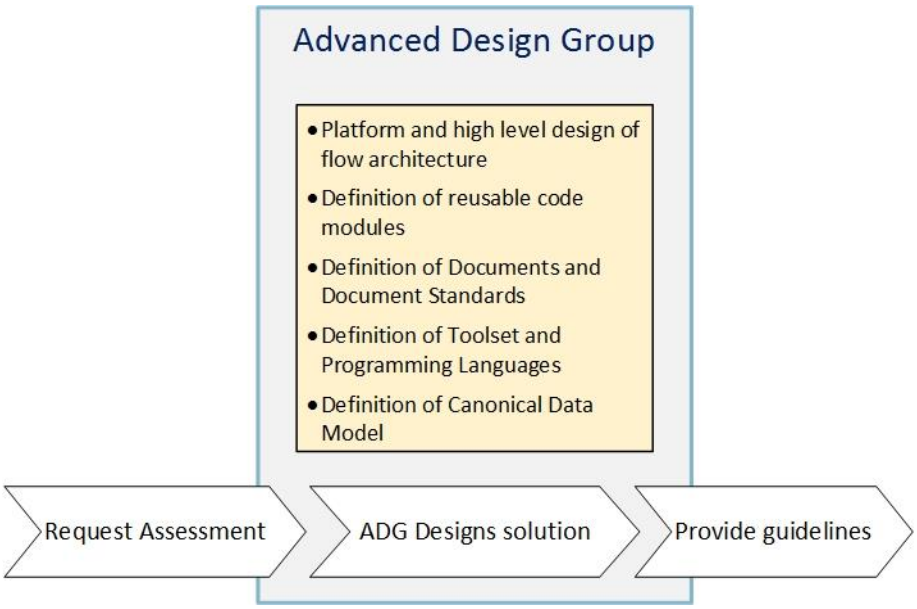

Figure 23: DIP solution standards

The Advanced Design Group takes care of four major areas within DIP, which are described in more detail below:

Platform and high level design of flow architecture

ADG is responsible for standardising the technological components of the platform which executes DIP. Every decision that influences the platform (also for example performance impacts due to new flows) and flow architecture have to be investigated by ADG.

Definition of reusable code modules

Every requirement for a new Data Integration flow that is not covered by standard modules of DIP is being assessed by ADG, with the aim of determining, if the new functionality needs to be standardized or can be implemented as a standalone (not reusable) solution.

Definition of Documents and Document Standards

DIP has a very large document repository that covers all aspects of DIP from high level design to low level technical documentation. Each of the documents is standardized as a template, for example excel files have specified columns with specified content, word documents have specified chapters, some files have special naming conventions, etc.

Definition of Toolset and Programming Languages

Tools are a strategic component for CDI, as every new tool needs to be licensed for many users, due to CDI size. Since the toolset in DIP is already established, this topic rarely gets

discussed, only if new tools join the company or tool upgrades are necessary. ADG is responsible for choosing tools and programming languages.

6.2 Differences between DIP and framework

This chapter analyses each of the presented building blocks by using the framework. This is done by taking the figures shown in the previous chapters and analysing each part of each building block, whether it matches with the components identified in the framework. If components are considered to match, they are shown with an "overlay" box coloured in green. If components are considered to match, but contain differences, then the box is coloured in yellow. If the components do not match at all or one component is missing, the box is shown in red.

6.2.1 Organisation

Figure 24 below shows the same picture as *Figure 20*, including an overlay with the roles specified in the framework example (see Part III, chapter *5.2.3.1 Organisation building block*).

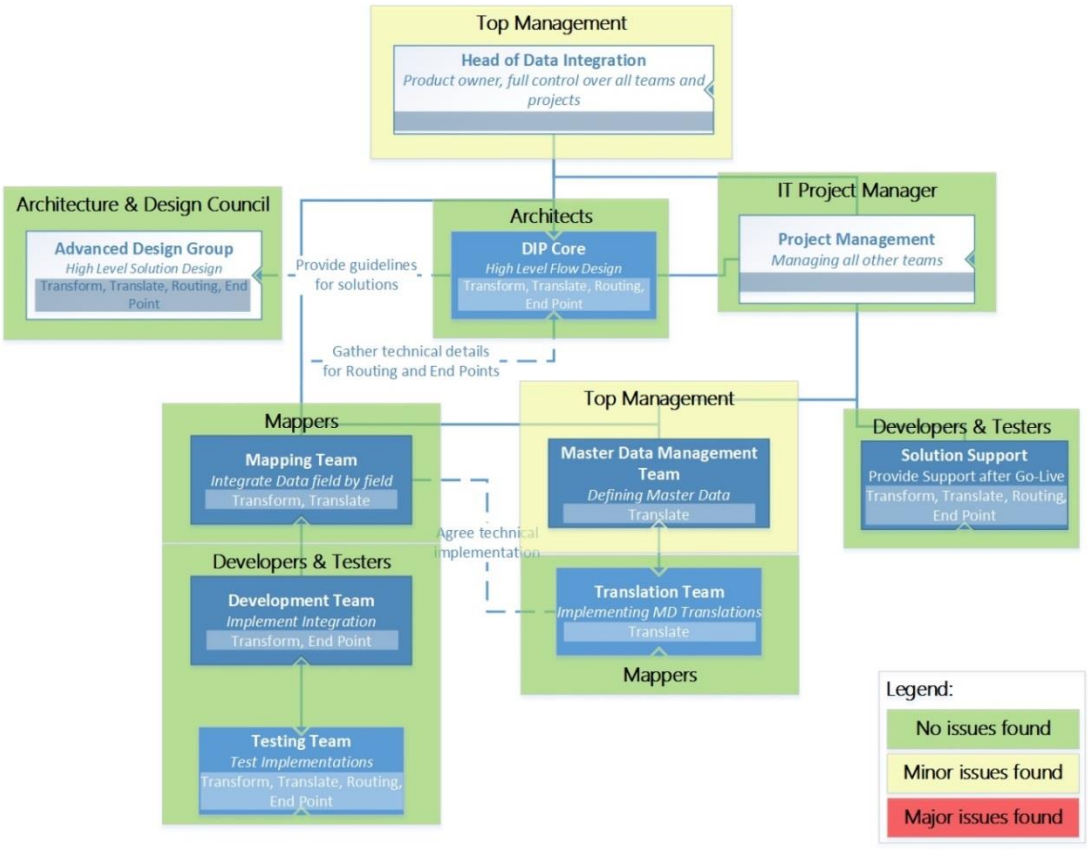

Figure 24: DIP Organisation overlaid with framework

When overlaying the DIP team structure with the organization building block of the framework, there is a large match. *Table 12* below shows an overview of the framework, DIP components and identified issues:

Table 12: Differences between framework and DIP organisation

Organisation		
Framework component	**Matching DIP component**	**Identified issues of DIP**
Not available	Master Data Management Team	Should be covered by Head of Data Integration according to the Framework, however in DIP this exists just on a team level
Top Management	Head of Data Integration	Has no full control over the acceptance of change
IT Project Manager	Project Management	No issues identified
Architects	DIP Core	No issues identified
Mappers	Mapping Team, Translation Team	No issues identified
Developers & Testers	Development Team, Testing Team, Solution Support	No issues identified
Architecture & Design Council	Advanced Design Group	No issues identified

In the following, the issues are described in more detail:

Issue 1: There is no role occupied for Master Data Management in CDI

One of the few differences in this building block is that in CDI, Master Data Management is being carried out by a different (business) role in the company, which is not on the Top Management level. CDI is already aware of this gap and has recently addressed it by calling for a new role in the company (Head of Master Data), who will be tied into the Data Integration also from the Top Management level.

Issue 2: The Top Management of DIP is not in the position to decide, which systems within CDI will use DIP as their Data Integration platform

DIP is treated as a product, which must be "sold" in order to get further budget. The people "buying" the solution (the "customers" of DIP) are the system owners of the various systems that should be integrated (which are usually from the same company). Since these system owners can freely decide how to invest their budget, there is no real pressure on them to use the new standardized Data Integration platform, making it more difficult to force the usage of DIP in the entire company.

6.2.2 Technology

Figure 25 below shows the same picture as *Figure 22*, including an overlay with the technological components specified in the framework (see Part III, chapter *5.2.3.2 Technology building block*).

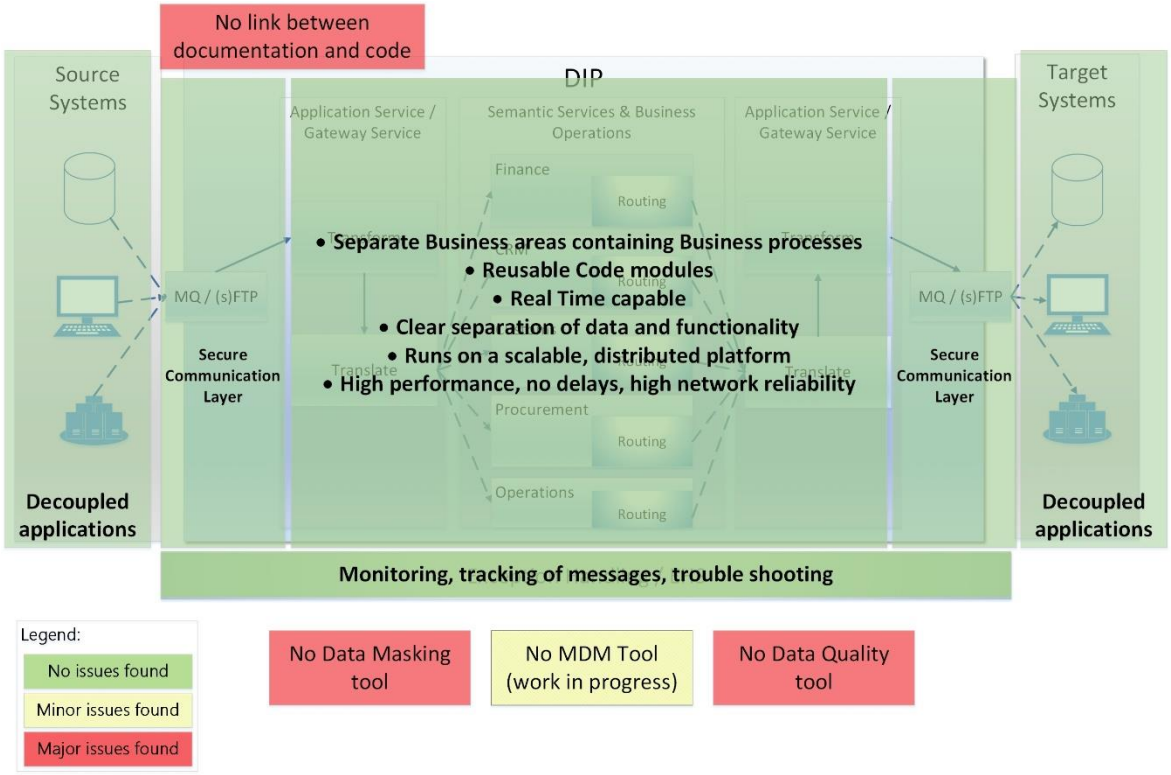

Figure 25: DIP technology overlaid with Framework

When overlaying the DIP technology with the technology building block from the framework, there is a large match between the two. *Table 13* below shows an overview of the framework, DIP components and identified issues:

Table 13: Differences between framework and DIP technology

Technology		
Framework component	**Matching DIP component**	**Identified issues of DIP**
ETL Code modules and Documentation	Code inside of Business Operations, documentation exists outside of DIP	No link between Documentation and Code
Data Masking	Not available	No Data Masking Tool is available
Data Quality Tool	Not available	No Data Quality Tool is available
Master Data Management Tool	Not available	No Master Data Management Tool is available (but work is in progress)

Technology		
Framework component	**Matching DIP component**	**Identified issues of DIP**
ESB Platform	DIP is running on an ESB, supports Routing	No issues identified
Business Areas and Business Processes	Semantic Service and Business Operations	No issues identified
Secured Communication Layer	MQ/sFTP	No issues identified
Decoupled Applications	Application Services doing Transformation and Translation to a canonical format	No issues identified
Real-Time Integration	DIP is real-time capable	No issues identified
Monitoring	Exception Handling	No issues identified

In the following the differences are described in further detail:

Issue 3: There is only a high-level link between documentation and code

One issue that has been called out in one of the interviews is the lack of a clear link between the implemented ETL code and the documentation. Both components (code and documentation) are there and are created with sufficient quality, but there is no direct link between the two elements.

Issue 4: There is no Data Masking tool present

CDI is not utilizing any tool for Data Masking as of now, however it is unknown to the author, if there are any business cases within CDI that would require a Data Masking tool.

Issue 5: There is no Data Quality tool present

CDI is not utilizing any of the introduced Data Quality tools from chapter *4.2.4 Data Quality Analysis*. Data Quality has indeed always been a topic in CDI, however is addressed on a business level and not assisted by any Data Integration tool.

Issue 6: There is no MDM tool present

Currently, there is no MDM tool in use within CDI, however the decision has been made to utilize a new tool for this purpose. The implementation and business processes for correctly using this tool are still pending though.

6.2.3 Solution standards

Figure 26 below shows the same picture as *Figure 23*, including an overlay which compares the building block to the components identified in the framework (see Part III, chapter *5.2.3.3 Solution standards building block*).

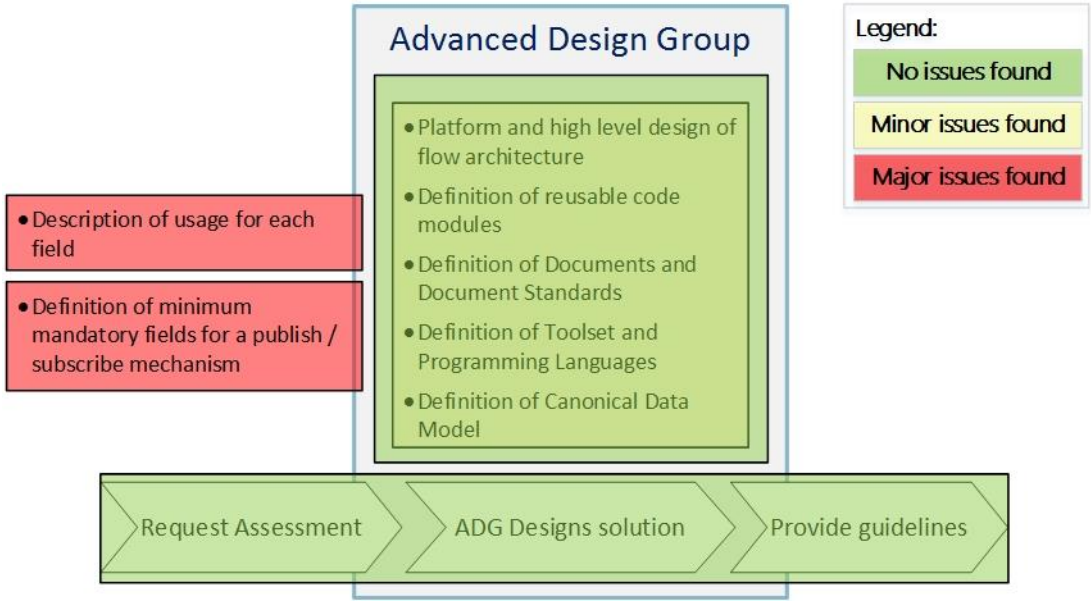

Figure 26: DIP solution standards overlaid with Framework

When overlaying the DIP solution standards with the solution standards building block from the framework, there are a few matches and mismatches between the two. *Table 14* below shows an overview of the framework, DIP components and identified issues:

Table 14: Differences between framework and DIP solution standards

Solution Standards		
Framework component	**Matching DIP component**	**Identified issues of DIP**
Description of usage of each field	Not available	There is no description for the usage of each field in CDM
Definition of minimum mandatory fields for a publish/subscribe mechanism	Not available	There is no definition of minimum fields needed
Standardisation lifecycle	The lifecycle is triggered by out-side assessment requests and produces guidelines[13]	No issues identified
Responsibility for solution stand-ards and documentation	Responsibility is with ADG	No issues identified
Canonical Data Exchange Format	Definition of Canonical Data Model	No issues identified
Platform architecture and reusable code modules	Platform and high level design of flow architecture, definition of reusable code modules	No issues identified

[13] ADG is not proactively reviewing solution standards, but assesses standards on request by other teams. This process is working very well for CDI and many solution standards are getting reviewed repeatedly, therefore the process model is judged as a full match, even though it looks differently.

Solution Standards		
Framework component	**Matching DIP component**	**Identified issues of DIP**
Document Standards	Definition of documents and document standards	No issues identified
Standard set of tools	Definition of Toolset and Programming Languages	No issues identified

In the following the differences are described in further detail:

Issue 7: There is no description regarding the usage of each field in the Canonical Data Model

The Canonical Model defines more than 10000 individual fields which are handling all data for all business areas and business processes – but without any business relevant descriptions assigned to them. Therefore it is very challenging for the mapping team to map any message to the Data Format, for example because a "customer" data object can have "attributes" and "references" as child data objects. However without any descriptions it is very hard to decide what kind of data goes into which object.

Issue 8: There is no definition of minimum fields needed for a publish / subscribe mechanism

As described in chapter *3.1.3 Message bus*, a message bus usually implements a publish / subscribe mechanism. In order to go for this approach, it is important to define, what data is necessary in order to publish a message, so that subscribers can rely on receiving this information as well. CDI has not yet started any activity to define the relevant fields for such messages.

6.3 Using the framework to resolve the differences

As there is already a large match between DIP and the framework, there are not many things which CDI needs to do. The following list summarizes the potential issues that were identified in DIP using the framework.

1. There is no role in the Top Management for Master Data Management (but the work on that is in progress).
2. The Top Management of DIP is not in the position to decide, which systems within CDI will use DIP as their Data Integration platform.
3. There is only a high-level link between documentation and code
4. There is no Data Masking tool present
5. There is no Data Quality tool present

6. There is no MDM tool present (but work on that is in progress)
7. There is no description regarding the usage of each field in the Canonical Data Model
8. There is no definition of minimum fields needed for a publish / subscribe mechanism

CDI can use the Data Integration Development Method to fix the issues highlighted above. In the following, it will be shown how the framework could be applied, so that CDI is in the position to close all gaps identified in the analysis. The following subchapters show based on the Part II of the framework, how the issues can be closed by taking the indicated steps.

To keep the descriptions shorter and since some issues are already being addressed by CDI, issues 1 and 6 are not considered in the following sub chapters. In this example it is shown, how CDI could answer four questions based on the issues:

- How can we use DIP as our new company wide standard Data Integration platform? (Issue 2)
- How can we improve the way we document our solution? (Issue 3)
- How can we implement a Data Masking and Data Quality tool in order to improve our business? (Issue 4, Issue 5)
- How can we improve the way we standardize our Data Integration platform? (Issue 7, Issue 8)

6.3.1 Requirements

In this phase, CDI should generally evaluate if the issues represent an issue for the company and if the closure of any of these gaps can be supported by a vision or justified by a business case.

Actions to take:

- CIG needs to evaluate, if they want to force more systems to use DIP. In case the answer is “yes”, they need to commit to a budget and a roadmap with realistic timescales for implementing it, governed from top-down.
- CDI needs to evaluate, how big the problem of a missing link between documentation and code is (for example, if for new employees understanding of documentation and code takes too long or if fixing support issues is too complicated due to documentation). If CDI wants to fix this issue, it should decide on a budget and a business standard for the approach (should the documentation be tool based, which

would require skills in that specific tool, or based on Microsoft Office products, as it is now).

- CDI needs to evaluate, if a Data Masking or Data Quality Tool is needed. CDI needs to decide how to govern these aspects from a Data Management perspective be creating a business process for each future tool.
- CDI needs to decide, if the scope of ADG needs to increase to cope with field descriptions for the Canonical Exchange Format, and minimum set of fields. CDI needs to calculate extra resources for this and also define their business standards, from which these data standards would be derived.

6.3.2 Analysis

The DIP team should analyse in detail how the scoped issues can be handled in the best way, by creating a high level plan.

Actions to take:

- There needs to be a plan how to integrate further systems with DIP without overloading the system or the team. Timescales need to be reassessed together with CDI.
- The ADG team should evaluate available tools for documentation of code and how far the link between code and documentation can be automated.
- Data Masking and Data Quality tools need to be chosen by experts of these fields. They need to be assessed by ADG and have to be integrated with the rest of the tools and processes landscape.
- ADG needs to define the new data standards based on a clear business data model, which they have to develop in cooperation with operational experts in CDI.

6.3.3 Design

Based on the outcome of the analysis, a detailed design, including the logical and physical definition of the implementation is created.

Actions to take:

- Newly connected systems should be integrated into existing Data Integration ESB architecture, while following existing standards. The high level flow design, transformation, translation, routing and end points need to be defined in detail, based on

the specific integration requirements. Test cases for verifying the correctness of the flows should be defined.

- The process for the usage of new documentation tools should be designed. In case new tools will be implemented, a detailed design for the features, architecture, application integration, front end, or backend database (if some database is used at all) should be written.
- The Data Masking and Data Quality tools have to be assessed in details for any possible customisation, like integration interfaces for connecting them to the DIP platform according to standards defined by ADG.
- The standards for the data model should be defined logically on a field by field basis, taking into account standardised and business-approved descriptions for each individual field. Each business scenario should be fully understood by the end of this phase, as it should also identify the minimum amount of fields necessary for each type of message.

6.3.4 Implementation

According to the results of the design phase, the implementation of all planned activities is being carried out.

Actions to take:

- The new systems are connected with DIP using the available platform tools. The Transformation rules should be physically implemented, the translation table should be uploaded to the databases, the routing rules should be deployed on the (development and test) ESB and the end points should be technically set up.
- The documentation tools should be developed according to their specifications or installed based on the chosen product and approach.
- The Data Masking and Data Quality tools should be deployed and connected to the DIP platform. Since these products can be quite complex, end users should start to receive training on how to correctly use them in the context of the Data Integration flows.
- The physical representation (XSD in this case) of each existing data model and field (including descriptions and mandatory/optional constraint) should be implemented and published for further use.

6.3.5 Testing

CDI should verify that the requirements that were originally raised will be covered by the steps that have been taken and produce test evidence for each topic.

Actions to take:

- The integration flows should be tested internally (to validate the function according to design) and fully end to end, from source system to target system (to validate that the Data Integration requirements will be fully met).
- The first code modules should get documented by using the new tools as well as read and understood by users of the documentation, to prove that the new way of documenting is delivering the expected improvements.
- Data from selected flows is masked and analysed for data quality issues. The results should be assessed by business experts in order to validate if the tools are creating a reliable output and can be used to anonymize data or to improve the overall data quality.
- The updated XSD should be tested on an existing flow to verify if the business requirements have been correctly understood and converted into the correct descriptions and set of mandatory fields.

6.3.6 Deployment & Maintenance

In this stage, the gap between the identified issues and the solution will be closed by making all produced application available to the entire company as specified in the previous phases.

Actions to take:

- The integration flows should be enabled to connect the new source applications to the target applications. The flows should be monitored for any errors and any issues should be resolved by patching the individual components of the flow.
- The new documentation method should be applied on all newly implemented integration code modules. If time permits, existing documentation should be updated to match the new standard.
- The Data Masking and Data Quality tool are made available to all eligible users for productive usage. In case the applications show bugs or limitations, this should be recorded and handled using bug reports and new feature requests.

- The updated XSD can be used on all newly created flows. If time permits, they can also be rolled out to existing flows, but have to make sure first, that every flow provides the minimum number of required fields.

6.4 Expert review

To proof that the designed framework creates added value, two experts from the previous interviews were asked to assess the analysis results presented in this chapter, as well as the recommendations for fixing them. As the experts are working in DIP for several years, they are aware of the major issues and can comment on the suggested solutions.

6.4.1 Review on the Building blocks

In order to validate that chapter *5.2.3 Part III – Supplements* can be used to analyse a real life Data Integration solution and to identify critical issues, one of the experts from the interview was asked to assess all gaps found in chapter *6.2 Differences between DIP and framework* in order to confirm, if the gaps are valid.

The expert was given the thesis and received a short explanation of the meaning of each issue. The replies for each issue were written down and agreed with the expert. To assure anonymity of the person, the role is not being mentioned here.

The below text repeats the identified issues as well as the expert's answers for each issue.

Issue 1: There is no role in the Top Management for Master Data Management (but the work on that is in progress).

Expert: *"We do have a new guy, who is head of MDM since 2 weeks."*

Issue 2: The Top Management of DIP is not in the position to decide, which systems within CDI will use DIP as their Data Integration platform.

Expert: *"Yes, this is true and this is a big issue. Not many people know about DIP and also the big competitor is another internal Data Integration system (maybe because of lower price). However our IT program aims to fix that and is working on a plan how."*

Issue 3: There is only a high-level link between documentation and code

Expert: *"Correct. We try to address this with one of our internal tools. The only link is in the mapping documentation and in the mapping build packages which are linked together with hard coded links which is not at all ideal!"*

Issue 4: There is no Data Masking tool present

Expert: *"This is true, there is no Data Masking tool, but we do encryption using SSL almost everywhere. Also, we don't know what happens afterwards in other applications, but it is not our concern."*

Issue 5: There is no Data Quality tool present

Expert: *"There is no tool like that and the only data quality we do is an automatic structure validation for example of country codes, when the message is being translated."*

Issue 6: There is no MDM tool present (but work on that is in progress)

Expert: *"Correct. There is a new global MDM system being built as we speak…"*

Issue 7: There is no description regarding the usage of each field in the Canonical Data Model

Expert: *"This is true and currently description is in Excel sheets and almost never available. We also try to address this with an "Implementation Guideline". This is a document for each interface which we can send to the systems which DIP integrates with. This way they know exactly which field of canonical means what and which ones they need to fill mandatory for their interface to work…"*

Issue 8: There is no definition of minimum fields needed for a publish / subscribe mechanism

Expert: *"Publish subscribe is not part of DIP yet. We have this planned though."*

6.4.2 Review on the Data Integration Development Method

In order to validate that chapter *5.2.2 Part II – Data Integration Development Method* can be used to fix issues in a real life Data Integration solution, one of the experts was asked to assess, if the steps proposed in chapter *6.3 Using the framework to resolve the differences* make sense, are complete and valid steps in the context of DIP and CDI.

The expert was given this thesis and received a short explanation of the found issues, as well as of the entire process as suggested in chapter *6.3 Using the framework to resolve the differences*, and was asked to freely comment on each step. The answers were noted down and agreed with the expert. To assure anonymity of the person, the role is not being mentioned here.

In the following, the individual feedback for each of the steps is given. The expert commented directly on each of the suggested steps in each phase and gave an overall opinion regarding the correctness of each phase.

1. Requirements

Expert: *"I absolutely agree, however we will not force applications to use our service but to make it more attractive, but still a budget and timeline need to be planned. Documentation needs to be indeed improved, the better the documentation the less iterations, the easier it is to attached countries and local applications to use our service. This implies descriptions on the CDM and hence enforcing data quality"*

2. Analysis

Expert: *"The analysis definitely reflects the need to improve procedure on setting up new Interfaces. We already started to evaluate the number of messages and it size. This hasn't been done in past. Documentation should be indeed automated out of the system to ensure system coding is 100% aligned with documentation. Data Standards will be more defined using implementation guidelines."*

3. Design

Expert: *"The Design improvements have been correctly identified and actions will be implemented accordingly for future system onboarding activities. This includes implementation guidelines for data quality / data standards routing requirements and documentation."*

4. Implementation

Expert: *"Implementations actions have been lined out correctly. Once platform tools are available or have been approved this will be part or the standard integration. XSD will be used for validation and documentation which influences immediately data quality and quantifying it."*

5. Testing

Expert: *"Absolutely right newly implemented interfaces are already tested internally before approaching the application owners and will be validated against existing documentation to ensure both are aligned."*

6. Deployment & Maintenance

Expert: *"Deployment and Maintenance will be indeed considers by e.g. updating the XSD, adjusting the standard e.g. to strict or to forgiving and will be documented accordingly."*

6.5 Summary

The framework in this thesis can be applied to a real world solution. This can be seen on the analysis of DIP, building block by building block. Eight issues in project DIP have been identified and verified as correct by a subject matter expert from the project.

It was furthermore described how the Part II of the framework can be used to fix the issues identified in this chapter. The suggestions have been verified by one of the experts from the interview and reviewed for correctness. The feedback did not suggest any extensions or changes to the framework, but was only related to details of individual steps.

When looking at these positive results, a few questions are still open though:

- Although the framework can identify issues very reliably, it was not proven, if it can identify all issues of a typical DI project.
- The framework makes no statement, whether an identified issue is critical to the project or not. This needs to be analysed by the enterprise by using Part II of the framework.
- The framework was only applied to one real world solution, so it needs to be proven that it also works in other contexts.

7 Conclusions

In this final chapter, the results of this thesis are summarized and reviewed against the aims in chapter *7.1 Summary of the results*. Furthermore, this chapter contains a list of benefits resulting from this thesis. In chapter *7.2 Future work* further topics for future research, continuation of the thesis and the framework are given.

7.1 Summary of the results

Table 15 below shows the goals which were specified in chapter *1.2 Goals, metrics, indicators and definitions of the thesis*, shows the chapter which is fulfilling the listed goal and evaluates if the goal can be considered fulfilled or not.

Table 15: Overview of goals, respective chapter numbers and fulfilment

Name of the goal	Chapter number	Fulfilled
1. Describe the role of Data Integration within the context of Data Management.	*2 Data Management and Integration*	Yes
2. Describe the possible approaches to Data Integration in large enterprises**.**	*3 The landscape of Data Integration*	Yes
3. Analyse the landscape of Data Integration solutions.	*3 The landscape of Data Integration*	Yes
4. Describe the typical functionality of Data Integration tools.	*4 Features of Data Integration tools*	Yes
5. Propose a framework for evaluation of Data Integration solutions	*5 A framework for developing Data Integration solutions for enterprise scenarios*	Yes
6. Analyse relevant Data Integration solutions using the defined framework and propose a Data Integration solution for a large enterprise.	*6 Applying the framework to a Data Integration Project (DIP)*	Yes

The following text contains a short description how each goal was fulfilled:

1. Describe the role of Data Integration within the context of Data Management.

The thesis described the role of Data Integration (chapter *2.3 What is Data Integration?*) and Data Management (chapter *2.1 What is Data Management?)* and showed how these topics may be connected with each other (chapter *2.4 The role of Data Integration in the context of Data Management*). It was discovered that Data Integration cannot exist without a proper Data Management.

2. Describe the possible approaches to Data Integration in large enterprises.

This goal is fulfilled (together with goal 3) in chapter *3 The landscape of Data Integration* by breaking down Data Integration into multiple dimensions (*3.1 Integration architecture, 3.2 Integration styles, 3.4 Business2Business vs. Application2Application*) and describing typical approaches to Data Integration in large (as well as small) enterprises (*3.3 Approaches in small vs. large enterprises*).

3. Analyse the landscape of Data Integration solutions.

This goal is fulfilled (together with goal 2) in chapter *3 The landscape of Data Integration* by describing multiple approaches in several dimensions, which might be used by enterprises as approaches to Data Integration, using several sources which were not older than 3 years. Furthermore, an overview of the Data Integration tool market as well as future trends are described (chapter *3.5 Data Integration market and trends*). It was discovered that the top vendors (Informatica, IBM, SAP, Talend) all offer tools of similar categories.

4. Describe the typical functionality of Data Integration tools.

Typical functionality of tools in the Data Integration market were shown by doing a broad range analysis of the tools available from four of the top vendors shown in the Gartner Magic Quadrant (chapter *4 Features of Data Integration tools*). The most common functionalities were grouped into categories and presented, showing examples from individual applications (chapter *4.2 Common functionality in detail)*. The groups that were discovered are: ETL Processing, Enterprise Service Bus, Master Data Management, Data Quality Analysis, Real Time Integration, Data Masking and B2B Integration.

5. Propose a framework for evaluation of Data Integration solutions

Based on these theoretical aspects, as well as interviews with experts from the field, a basic Data Integration framework for messaging solution has been developed and described (chapter *5.2 A Framework for Data Integration based on messaging*). The result is a framework consisting of three parts (Part I – Introduction, Part II - Data Integration Development Method and Part III – Supplements).

6. Analyse relevant Data Integration solutions using the defined framework and propose a Data Integration solution for a large enterprise.

To test the framework, it has been applied on an existing (anonymised) real world solution to identify potential gaps (chapter *6.2 Differences between DIP and framework*) and to propose steps in order to close these gaps (chapter *6.3 Using the framework to resolve the differences*). To assess, if the analysis and proposed steps were executed correctly, the results have been peer reviewed by members of the project team (chapter *6.4 Expert review*). In total, eight gaps have been identified and a process for closing them has been suggested and validated by two experts.

Benefits

To underline the benefits of this thesis, the below list summarizes the added value:

1. This thesis provides concise overview of theoretical aspects which form the foundation of Data Integration in large enterprises. It might be used as an overview for someone who is new to the topic of Data Integration.
2. This thesis provides an analysis of the features available in the Data Integration tools of four high ranked vendors from the Gartner Magic Quadrant and lists the most common functionality available in these tools. This overview might be helpful for anyone who is planning to start a Data Integration project (including small companies, as parts of these tools are for free), to get an idea what features might be needed, with which tools to start and how they relate to each other.
3. The thesis provides a solid, expert-reviewed and partially tested framework for Data Integration, which might help large companies to develop, analyse and improve Data Integration solutions.
4. The framework provided in the thesis might also serve as a basic framework, which can be extended by future research into a more solid, detailed framework for Data Integration.
5. The analysis of DIP provided a list of gaps as well as a proposed approach how to close these gaps. This list was given back to CDI for review and can be seen as a practical output of this thesis.

7.2 Future work

In this last chapter, gaps from the framework are highlighted and further research work is suggested.

What should be highlighted is that it was not proven if the framework can reliably identify all issues that occur in a Data Integration project. One way to prove this could be to start a research to collect a list of all issues that are present in another project and then compare if all of these issues are covered by this framework. Also, since this framework is based on experiences from DIP and was only tested on DIP, it is very important that future research will also test it on other projects, if this framework should be continued.

Also, the DAMA framework was introduced however the Data Integration framework only uses a few roles from the DAMA framework in the Organisation part. Further links to DAMA could be part of future improvements to the framework.

Furthermore, the framework is aiming at messaging solutions, however other solutions might also be viable for large and especially smaller size companies. Integrating these styles into the framework, could greatly improve it and make it more usable for a larger audience and more scenarios.

Lastly it should be mentioned that the framework does not suggest any concrete methodologies for solving the phases of the Data Integration Development method (which is the reason why DIP was not evaluated based on Part II of the framework). This could be another great extension of this framework.

I. Glossary

Term	Meaning	Source
CDI	"Company Doing Integration", an anonymized name for a large scale company which is running a large scale IT program, including a Data Integration Platform ("DIP").	Author
CEO	Chief Executive Officer, highest ranking leader of a company	Author
COTS product	Commercial off-the-shelf product is a software product, which is available for sale to public companies.	Morisio and Torchiano, 2002, p. 3
CRM	Customer Relationship Management is a system, whose aim is to maintain data regarding associations with customers	Rababah, et al, 2011, p. 1-2
DAMA	Data Management body of knowledge is an international foundation, whose aim is to provide up-to-date information related to data management area, by offering a platform for their members to gain knowledge from researches, education and publications.	DAMA International, 2015
DIP	"Data Integration Project", an anonymized name for a large scale A2A messaging bus Data Integration solution used in CDI.	Author
DMBOK	"Data Management Body of Knowledge is a collection of processes and knowledge areas that are generally accepted as best practices within the Data Management discipline"	Cupoli, Earley and Henderson, 2014, p.5
End Point	In the context of a messaging solution, End Points represent the technical link between a system and the Data Integration platform, for example in form of a file server or a Message Queue.	Hohpe and Woolf, 2004, p. 55, 56
ERP	Enterprise Resource Planning, software packages covering multiple organisational areas, to integrate business processes with transactional data.	Esteves de Sousa, 2004, p. 15
ESB	Enterprise Service Bus, a central integration layer, or "hub" for hosting ETL code and centrally controlling the integration of surrounding systems	Informatica, © 2016b

Term	Meaning	Source
ETL	Extraction, Transformation, Loading, used by integration tools to process data from system A to system B.	Informatica, © 2016f
ICT	Information and communication technology, an area that revolves around any technology that includes human or data interaction in form of communication	Lloyd, 2005,p. 3
IPC	Inter-Process Communication is a method coming from the Linux operating system and defines multiple methods of communication between programs for developers to use	TLDP, 1996
IS	Information Systems, a network of hardware and software used to work with data in order to achieve company goals	Azeemi, et al, 2003, p. 738
MDM	Master Data Management – an area, which defines and manages key data for the company, so called Master Data	White, et al, 2006, p. 3
MQ	Also called “WebSphere MQ”. MQ stands for “Message queue” and is a messaging middleware by IBM for secure and controlled transfer of data in different styles, for example file transfer	IBM, © 2016n
RDMA	Remote Direct Memory Access, a technology that allows fast read/write operations on foreign systems while bypassing the remote processor, memory, cache and operating system	Curry, 2002, p. 1-2
SCM	Supply Chain Management, operation of all stages involved to fulfil a customer request from manufacturer to customer	AELP, 2012
TCP/IP	Transmission Control Protocol/Internet Protocol, the basic transmission protocol that is used in the in the world wide web.	Rouse, © 2016a
UDP	User Datagram Protocol, a very simplistic and minimalistic transmission protocol, suitable for communications that do not require error checking	Rouse, © 2016b

II. Literature

Abcam, © 1998-2015. *Antibodies, Ice-Buckets and Molly the Sheep - the Story behind Abcam.* [online]. [cit. 2015-12-18]. Available from: http://www.abcam.com/index.html?pageconfig=story

ADUSUPALLI, Neeraja, 2013. *Convergence of B2B (Business to Business) and A2A (Application to Application) to drive business value*. WIPRO [online] [cit. 2016-04-19]. Available from: http://www.wipro.com/blogs/convergence-of-b2b-and-a2a-to-drive-business-value/

AELP, 2012. *Supply Chain Management: A good practice guide for the post-16 skills sectors*. Association of employment and learning providers [online] [cit. 2016-04-16]. Available from: http://www.aelp.org.uk/file/?id=1391&type=item

ASSUNCAO, Marcos D., et al, 2014. *Big Data Computing and Clouds: Trends and Future Directions* [online]. 2014-08-22, 1-44 [cit. 2015-12-21]. Available from: http://arxiv.org/pdf/1312.4722.pdf

Astera, © 2014. *Whitepaper: Commercial Open Source vs. Proprietary Data Integration Software.* [online]. [cit. 2016-01-31]. Available from: http://www.astera.com/media/66282/open%20source%20wp2.pdf

AZEEMI, Imran K., et al, 2013. *Migrating To The Cloud: Lessons And Limitations Of 'Traditional' IS Success Models*. Procedia Computer Science. [online] 2013 (16): 737-746 [cit. 2016-04-16]. Available from: http://www.sciencedirect.com/science/article/pii/S1877050913000781

BALAJI, S., and SUNDARARAJAN MURUGAIYAN, M, 2012. *WATEERFALLVs V-MODEL Vs AGILE: A COMPARATIVE STUDY ON SDLC*. International Journal of Information Technology and Business Management [online]. **2**(1): 26-30 [cit. 2016-4-5]. ISSN 2304-0777. Available from: http://jitbm.com/Volume2No1/waterfall.pdf

BOEHM, Barry and ABTS, Chris, 1999. COGS Integration: Plug and Pray? *CSSE | Center for Systems and Software Engineering* [online]. University of Southern California, 135-138, [cit. 2015-11-27]. Available from: http://csse.usc.edu/TECHRPTS/2000/usccse2000-510/usccse2000-510.pdf

BOEHM, Barry W., 1988. *A Spiral Model of Software Development and Enhancement*. IEEE COMPUTER SOCIETY [online]. 61-72 [cit. 2016-04-05]. Available from: http://www.dimap.ufrn.br/~jair/ES/artigos/SpiralModelBoehm.pdf

BRUCKNER, Tomáš, 2012. *Tvorba informačních systémů: principy, metodiky, architektury*. 1. vyd. Praha: Grada, 357 p. Management v informační společnosti. ISBN 978-80-247-4153-6.

BusinessInfo.cz, 2009. *Uplatňování definice malého a středního podniku (MSP).* BusinessInfo.cz - Oficiální portál pro podnikání a export [online]. [cit. 2016-02-20]. Available from: http://www.businessinfo.cz/cs/clanky/uplatnovani-nove-definice-maleho-a-3760.html

BUSSLER, Christoph, 2003. *B2B Integration Concepts and Architecture*. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-366-2051-696.

CUPOLI, Patricia, EARLEY, Susan and HENDERSON, Deborah, 2014. *DAMA - DMBOK2 Framework* [online]. [cit. 2015-11-30]. Available from: https://www.dama.org/sites/default/files/download/DAMA-DMBOK2-Framework-V2-20140317-FINAL.pdf

CURRY, Brent, 2002. *RDMA CONSORTIUM COMPLETES PROTOCOL SPECIFICATIONS* [online] 1-2 [cit. 2016-04-16]. Available from: http://www.rdmaconsortium.org/home/PressReleaseOct30.pdf

DAMA International, 2015. *About Us.* The Global Data Management Community [online] [cit. 2016-04-16]. Available from: https://www.dama.org/content/about-us

DAO RESEARCH, 2015. *Whitepaper: Data Integration Platforms for Big Data and the Enterprise: Customer Perspectives on IBM, Informatica, and Oracle* [online]. [cit. 2015-12-13]. Available from: http://www.oracle.com/us/products/middleware/data-integration/di-oracle-informatica-ibm-wp-2438402.pdf

DE MAURO, Andrea, et al., 2015. *What is big data? A consensual definition and a review of key research topics*. [online]. 97 - 104 [cit. 2015-12-18]. DOI: 10.1063/1.4907823. ISBN 10.1063/1.4907823. ISSN 0094-243X. Available from: http://scitation.aip.org/content/aip/proceeding/aipcp/10.1063/1.4907823

DOHNAL, Tomáš, 2011. *Stav trhu v oblasti Enterprise Information Integration.* Prague. Master thesis. University of Economics, Prague, Faculty of Informatics and Statistics. Suppervisor Ing. Libor Gála. Available from: https://isis.vse.cz/auth/lide/clovek.pl?zalozka=7;id=41644;studium=102646;zp=27464;download_prace=1

RŮŽIČKA, Jan, 2014. *Integrace entity „cash event" v systémech Sugar CRM a Adempiere*. Prague. Bachelor thesis. University of Economics, Prague, Faculty of Informatics and Statistics. Suppervisor Ing. Jan Kučera. Available from: https://isis.vse.cz/auth/lide/clovek.pl?zalozka=7;id=87818;studium=116020;zp=46412;download_prace=1

DONG, Xin Luna and SRIVASTAVA, Divesh, 2013. *Big Data Integration* [online]. 1188-1189 [cit. 2016-01-31]. Available from: http://www.vldb.org/pvldb/vol6/p1188-srivastava.pdf

ESTEVES DE SOUSA, José M, 2004. *DEFINITION AND ANALYSIS OF CRITICAL SUCCESS FACTORS FOR ERP IMPLEMENTATION PROJECTS*. Barcelona, Spain [online] 1-313 [cit. 2016-04-16]. Thesis. Universitat Politècnica de Catalunya. Advisors: Joan Antoni Pastor and Josep Casanovas. Available from: http://jesteves.com/Tesis_phd_jesteves.pdf

FETSEL, Dienter, et al., © 2001. *Intelligent E-Business: Product Data Integration in B2B E-Commerce* [online]. 54-59, [cit. 2015-11-30]. Vrije Universiteit Amsterdam. Available from: http://www.cs.ucf.edu/~kienhua/classes/COP6730/E-Commerce2.pdf

FEUER, Sven, 2007. Enterprise Architecture – An Overview. SAP DEVELOPER NETWORK [online]. [cit. 2016-04-01]. Available from: http://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/90a744a3-55d9-2910-96b9-8a1513fefb1f?QuickLink=index&overridelayout=true&12962211344732

GÁLA, Libor, POUR, Jan and TOMAN, Prokop, 2006. *Podniková informatika* [online]. Praha: Grada Publishing, a.s. [cit. 2016-02-14]. ISBN 80-247-1278-4.

GARTNER INC, 2015a. *Magic Quadrant for Data Integration Tools*. [online]. [cit. 2016-01-28]. Available from: http://www.gartner.com/technology/reprints.do?id=1-2KDMO20&ct=150731&st=sb

GARTNER, INC, 2015b. *Gartner Reprint: Magic Quadrant for Data Quality Tools*. [online]. [cit. 2016-02-14]. Available from: https://web.archive.org/web/20160416083852/https://www.gartner.com/doc/reprints?id=1-2SGV9XI&ct=151118&st=sb&mkt_tok=eyJpIjoiTUdRMk1UbGtOamd4WXpVeSIsInQiOiJzXC9CRFwvOG1hZGN4OUpGOUtPSDZJd2xXTkxMYkZORmZcL1Vja2Q0cWZ6STQwZjlxU2hFQzRXU01CeUhPamFpSGdlRjg2QlFJUkxrcGFuSmpQdENIdGp2c1NFclNPOE9SYm02NEFpSElYYUJzZz0ifQ%253D%253D

Gartner, Inc., © 2016a. IT Glossary: Enterprise Architecture (EA). [online]. [cit. 2016-04-01]. Available from: http://www.gartner.com/it-glossary/enterprise-architecture-ea/

GLADDEN, Risto, 2008. *Tools for Complex Projects*. Project Management Journal. United States: Wiley Subscription Services, Inc. [online]. 2014-12-20, **39**(3): 1 [cit. 2015-11-16]. ISSN 87569728. Available from: http://search.proquest.com.zdroje.vse.cz/docview/218767074/B26394E6CD104FF2PQ/1?accountid=17203

GUESS, Angela, 2012. *5 Benefits of Enterprise Data Integration*. *DATAVERSITY*. [online]. 2012-02-16 [cit. 2015-11-27]. Available from: http://www.dataversity.net/5-benefits-of-enterprise-data-integration/

Hadoop, © 2014.*Welcome to Apache™ Hadoop®!: What Is Apache Hadoop?* [online]. [cit. 2015-12-18]. Available from: http://hadoop.apache.org/

HALEVY, Alon, et al, 2006. *Data Integration: The Teenage Years* [online]. University of Washington Computer Science & Engineering, 1-8 [cit. 2015-11-27]. Available from: https://homes.cs.washington.edu/~alon/files/halevyVldb06.pdf

HOHPE, Gregor and WOOLF, Bobby, © 2015a. *Enterprise Integration Patterns: File Transfer*. [online]. [cit. 2015-12-22]. Available from: http://www.enterpriseintegrationpatterns.com/patterns/messaging/FileTransferIntegration.html

HOHPE, Gregor and WOOLF, Bobby, © 2015b. *Enterprise Integration Patterns: Shared Database*. [online]. [cit. 2015-12-22]. Available from: http://www.enterpriseintegrationpatterns.com/patterns/messaging/SharedDataBaseIntegration.html

HOHPE, Gregor and WOOLF, Bobby, © 2015c. *Enterprise Integration Patterns: Remote Procedure Invocation*. [online]. [cit. 2015-12-22]. Available from: http://www.enterpriseintegrationpatterns.com/patterns/messaging/EncapsulatedSynchronousIntegration.html

HOHPE, Gregor and WOOLF, Bobby, © 2015d. *Enterprise Integration Patterns: Messaging*. [online]. [cit. 2015-12-22]. Available from: http://www.enterpriseintegrationpatterns.com/patterns/messaging/Messaging.html

HOHPE, Gregor and WOOLF, Bobby, 2004. *Enterprise integration patterns: designing, building, and deploying messaging solutions*. San Francisco: Addison-Wesley, 683 s. Addison-Wesley signature series. ISBN 03-212-0068-3.

IBM, © 2015. *PartnerWorld: It's a brave new world out there*. IBM - United States [online]. [cit. 2015-12-14]. Available from: http://www-304.ibm.com/partnerworld/wps/servlet/ContentHandler/partnerworld-public

IBM, © 2016a. *Data Integration: Understand, cleanse, monitor, transform and deliver your data*. IBM - United States [online]. [cit. 2016-01-30]. Available from: http://www-03.ibm.com/software/products/en/category/SWB50

IBM, © 2016b. *WebSphere MQ File Transfer Edition*. IBM - United States [online]. [cit. 2016-01-30]. Available from: http://www-03.ibm.com/software/products/en/wmq-fte

IBM, © 2016c. *Data Management Platform: Capture, store, process, distribute, backup, recover and protect structured data*. IBM - United States [online]. [cit. 2016-01-30]. Available from: http://www-03.ibm.com/software/products/en/category/SWB00

IBM, © 2016d. *IBM Integration Bus*. *IBM - United States* [online]. [cit. 2016-01-30]. Available from: http://www-03.ibm.com/software/products/en/ibm-integration-bus

IBM, © 2016e. *B2B integration: Modernize your B2B integration platform*. IBM - United States [online]. [cit. 2016-01-30]. Available from: http://www-03.ibm.com/software/products/en/category/b2b-integration

IBM, © 2016f. *View Pricing and Buy*. IBM - United States [online]. [cit. 2016-01-31]. Available from: https://www-112.ibm.com/software/howtobuy/buyingtools/paexpress/Express?P0=E1&part_number=D0L5KLL,D0L63LL&catalogLocale=en_US&Locale=null&country=USA&PT=html&S_TACT=none&S_CMP=none&brand=ws

IBM, © 2016g. *InfoSphere DataStage*. IBM - United States [online]. [cit. 2016-03-09]. Available from: http://www-03.ibm.com/software/products/en/ibminfodata

IBM, © 2016h. *Enterprise Service Bus (ESB).* IBM - United States [online]. [cit. 2016-03-09]. Available from: http://www.ibm.com/middleware/integration/en-us/enterprise-service-bus-esb.html

IBM, © 2016i. *Master Data Management.* IBM - United States [online]. [cit. 2016-03-09]. Available from: http://www.ibm.com/analytics/us/en/technology/master-data-management/

IBM, © 2016j. *InfoSphere Information Analyzer: Data quality assessment, analysis and monitoring*. IBM - United States [online]. [cit. 2016-03-09]. Available from: http://www-03.ibm.com/software/products/cs/ibminfoinfoanal

IBM, © 2016k. *IBM Leverages DataMirror Acquisition for Real-Time Change Data Capture*. IBM - United States [online]. [cit. 2016-03-09]. Available from: http://www-01.ibm.com/software/data/integration/dm/

IBM, © 2016l. *Sterling B2B Integrator*. IBM - United States [online]. [cit. 2016-03-09]. Available from: http://www-03.ibm.com/software/products/en/b2b-integrator

IBM, © 2016m. *InfoSphere Optim Data Privacy*. IBM - United States [online]. [cit. 2016-03-09]. Available from: http://www-03.ibm.com/software/products/en/infosphere-optim-data-privacy

IBM, © 2016n. *WebSphere MQ File Transfer Edition*. IBM - United States [online] [cit. 2016-04-16]. Available from: http://www-03.ibm.com/software/products/en/wmq-fte

Informatica, © 2016a. *Secure File Transfer*. Informatica US. [online]. [cit. 2016-01-30]. Available from: https://www.informatica.com/products/data-integration/b2b-data-exchange/managed-file-transfer.html

Informatica, © 2016b. *Data Integration Hub: Enterprise Data Tool*. Informatica US. [online]. [cit. 2016-01-30]. Available from: https://www.informatica.com/products/data-integration/data-integration-hub.html

Informatica, © 2016c. *Cloud Pricing & Editions: Cloud Integration*. Informatica US. [online]. [cit. 2016-01-30]. Available from: https://www.informatica.com/products/cloud-integration/editions-and-pricing/us-pricing.html

Informatica, © 2016d. *Informatica PowerCenter Express Redefines Entry-Level Data Integration*. Informatica US [online]. [cit. 2016-01-30]. Available from: https://www.informatica.com/about-us/news/news-releases/2013/06/20130605-informatica-powercenter-express-redefines-entry-level-data-integration.html

Informatica, © 2016e. *PowerCenter Express - ETL Tool.* Informatica US [online]. [cit. 2016-02-08]. Available from: https://marketplace.informatica.com/solutions/pcexpress

Informatica, © 2016f. *Advanced Data Transformation Solutions*. Informatica US [online]. [cit. 2016-02-08]. Available from: https://www.informatica.com/products/data-integration/advanced-data-transformation.html

Informatica, © 2016g. *MDM: Master Data Management Software.* Informatica US. [online]. [cit. 2016-02-14]. Available from: https://www.informatica.com/products/master-data-management.html

Informatica, © 2016h. *Low Latency Trading: Quick Messaging*. Informatica US. [online]. [cit. 2016-02-14]. Available from: https://www.informatica.com/products/data-integration/real-time-integration/ultra-messaging/ultra-messaging-streaming-edition.html

Informatica, © 2016i. *Data Masking: Protect sensitive data from unauthorized access.* Informatica US [online]. [cit. 2016-02-14]. Available from: https://www.informatica.com/products/data-security/data-masking.html

Informatica, © 2016j. *B2B Data Exchange: Partners Data Integration.* Informatica US [online]. [cit. 2016-02-14]. Available from: https://www.informatica.com/products/data-integration/b2b-data-exchange.html

Informatica, © 2016k. *PowerExchange Connectors: Data Connectors*. Informatica: Data Integration leader for Big Data & Cloud Analytics | Informatica US [online]. [cit. 2016-03-09]. Available from: https://www.informatica.com/products/data-integration/connectors-powerexchange.html

Informatica, © 2016l. *Data Quality: Because great data is good business*. Informatica US [online]. [cit. 2016-04-01]. Available from: https://www.informatica.com/products/data-quality.html#fbid=1r-_jR4oITQ

ISO, © 2016. *ISO 9735:1988 - Electronic data interchange for administration, commerce and transport (EDIFACT) -- Application level syntax rules*. ISO - International Organization for Standardization [online]. [cit. 2016-01-31]. Available from: http://www.iso.org/iso/catalogue_detail.htm?csnumber=17592

ITS, 1996. *Near real time*. Institute for Telecommunication Sciences [online]. [cit. 2016-03-09]. Available from: http://www.its.bldrdoc.gov/fs-1037/dir-024/_3492.htm

JEFFREY, Dean and GHEMAWAT Sanjay, 2004. *MapReduce: Simplied Data Processing on Large Clusters.* Google, Inc. [online]. 1-13 [cit. 2015-12-18]. Available from: http://static.googleusercontent.com/media/research.google.com/en//archive/mapreduce-osdi04.pdf

KIM, Dan Jong, et al, 2003. *A Comparison of B2B E-Service Solutions*. COMMUNICATIONS OF THE ACM [online]. 2003, **46**(12), 317-324 [cit. 2016-01-31]. Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.95.441&rep=rep1&type=pdf

KRASKA, Tim, 2013. *Finding the Needle of the Big Data Systems Haystack.* IEEE COMPUTER SOCIETY. [online]. 84-86 [cit. 2015-12-18]. Available from: http://database.cs.brown.edu/papers/t-needlehaystack.pdf

LENZERINI, Maurizio, 2002. *Data Integration: A Theoretical Perspective.* ACM PODS [online]. 233-246 [cit. 2016-01-28]. ISBN 1-58113-507-6. Available from: http://delivery.acm.org.zdroje.vse.cz/10.1145/550000/543644/p233-lenzerini.pdf?ip=146.102.19.70&id=543644&acc=ACTIVE%20SERVICE&key=D6C3EEB3AD96C931%2EF07030D8BB94E8BC%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&CFID=768333432&CFTOKEN=12130061&__acm__=1460016104_900fbe06f1bc95c6c247cc3027a213ea

LIAISON, © 2015a. *ALLOY: Data-Inspired Future.* Liaison Technologies [online]. [cit. 2015-12-01]. Available from: http://www.liaison.com/liaison-alloy-platform/

LIAISON, © 2015b. *On Demand, Any-to-Any Integration.* Liaison Technologies [online]. [cit. 2015-12-01]. Available from: http://www.liaison.com/solutions/cloud-services-integration/cloud-adapters/

LLOYD, Margaret, 2005. *Towards a definition of the integration of ICT in the classroom*. AARE, Eds. [online] [cit. 2016-04-16]. Available from: http://eprints.qut.edu.au/3553/1/3553.pdf

LOUGHEAD, Kim, 2014. *Informatica Data Integration Hub Demo*. In: Youtube [online]. Released 2. 8. 2014 [seen 2016-02-14]. Available from: https://youtu.be/RbBQ6CSdtq4

MCKENDRICK, Joe, 2014. DATA INTEGRATION IN THE ERA OF BIG DATA. Database Trends and Applications [online]. Chatham, **28**(1): 4-6, 8-9 [cit. 2015-11-22]. ISSN 1547-9897. Available from: http://search.proquest.com.zdroje.vse.cz/docview/1506952937/8B011D1820764549PQ/1?accountid=17203

MELL, Peter and GRANCE, Timothy, 2011. *The NIST Definition of Cloud Computing: Recommendations of the National Institute of Standards and Technology*. Computer Security Division: Computer Security Resource Center [online]. [cit. 2015-11-21]. Available from: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

MORISIO, Maurizio and TORCHIANO, Marco, 2002. *Definition and classification of COTS: a proposal*. International Conference on COTS Based Software Systems, Orlando. [online] 1-10 [cit. 2016-04-16]. Available from: http://softeng.polito.it/torchiano/papers/ICCBSS2002.pdf

MULESOFT, © 2015. *SMB Integration Draws on Mule ESB*. MuleSoft | Integration Platform for Connecting SaaS and Enterprise Applications [online]. [cit. 2015-12-13]. Available from: https://www.mulesoft.com/resources/esb/smb-integration-solution

MUNROE, Randall, 2014. *Comics that ask "what if?"* [video]. TED Ideas worth spreading [online]. Filmed in March 2014 [seen 2015-12-18]. Available from: https://www.ted.com/talks/randall_munroe_comics_that_ask_what_if#t-2599

NADHAN, E. G. and WELDON Jay-Louise. 2004. *A Strategic Approach to Data Transfer Methods*. Learn to Develop with Microsoft Developer Network | MSDN [online]. [cit. 2016-02-14]. Available from: https://msdn.microsoft.com/en-us/library/aa480064.aspx

ORACLE, 2015. *Demystifying Data Integration for the Cloud*. Oracle White Paper [online]. August 2015 [cit. 2015-12-01]. Available from: http://www.oracle.com/us/products/middleware/data-integration/data-integration-for-cloud-1870536.pdf

RABABAH, Khalid, et al., 2011. *A UNIFIED DEFINITION OF CRM TOWARDS THE SUCCESSFUL ADOPTION AND IMPLEMENTATION*. Academic Research International. [online]. **1**(1): 220-228 [cit. 2016-04-16]. ISSN 2223-9553. Available from: http://www.savap.org.pk/journals/ARInt./Vol.1(1)/2011(1.1-20).pdf

RAJU, Sam and WALLACHER, Claus, 2008. *B2B Integration Using SAP NetWeaver® PI*. SAP PRESS [online]. p. 137-166 [cit. 2016-01-31]. Available from: http://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/4026e6f2-4869-2c10-3dad-fe868a7b8c3c?overridelayout=true

ROUSE, Margaret, © 2016a. *TCP/IP (Transmission Control Protocol/Internet Protocol).* TechTarget - Global Network of Information Technology Websites and Contributors [online] [cit. 2016-04-16]. Available from: http://searchnetworking.techtarget.com/definition/TCP-IP

ROUSE, Margaret, © 2016b. *UDP (User Datagram Protocol).* TechTarget - Global Network of Information Technology Websites and Contributors [online] [cit. 2016-04-16]. Available from: http://searchsoa.techtarget.com/definition/UDP

ROUSE, Margaret, 2014a. *What is big data?*. TechTarget - Global Network of Information Technology Websites and Contributors [online]. [cit. 2015-12-18]. Available from: http://searchcloudcomputing.techtarget.com/definition/big-data-Big-Data

ROUSE, Margaret, 2014b. *What is Internet of Things (IoT)?*. TechTarget - Global Network of Information Technology Websites and Contributors [online]. [cit. 2015-12-18]. Available from: http://whatis.techtarget.com/definition/Internet-of-Things

ROUSE, Margaret, 2015. *What is integration?*. TechTarget - Global Network of Information Technology Websites and Contributors [online]. [cit. 2015-12-22]. Available from: http://searchcrm.techtarget.com/definition/integration

ROYCE, Winston W., 1970. *Managing the development of large software systems*. [online]. University of Maryland. 328-338 [cit. 2016-04-05]. Available from: http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf

SAP, © 2016a. *File Transfer*. SAP Software Solutions | Technology & Applications [online]. [cit. 2016-01-30]. Available from: http://help.sap.com/saphelp_em70/helpdata/en/b4/79223dc5b54b36899ea4f731a712f6/frameset.htm

SAP, © 2016b. *SAP Business One*. SAP Software & Řešení | Technologické a podnikové aplikace [online]. [cit. 2016-01-30]. Available from: http://go.sap.com/cz/product/enterprise-management/business-one.html

SAP, © 2016c. *SOAP Adapter Overview*. SAP Software Solutions | Technology & Applications [online]. [cit. 2016-01-30]. Available from: https://help.sap.com/saphelp_nwpi711/helpdata/en/44/8c4756224a6fb5e10000000a155369/content.htm?frameset=/en/43/951aceb1146353e10000000a11466f/frameset.htm

SAP, © 2016d. *Get a complete view of your information assets with powerful Data Integration software*. SAP Software Solutions | Technology & Applications [online]. [cit. 2016-01-31]. Available from: http://www.sap.com/pc/tech/enterprise-information-management/software/data-integrator/index.html

SAP, © 2016e. *Provide easy, self-service access to decision-ready information with our BI platform*. SAP Software Solutions | Technology & Applications [online]. [cit. 2016-03-09]. Available from: http://go.sap.com/product/analytics/bi-platform.product-capabilities.html

SAP, © 2016f. *Service Bus-based Integration*. Sap Community Network [online]. [cit. 2016-03-09]. Available from: http://scn.sap.com/docs/DOC-8860

SAP, © 2016g. *B2B Integration with SAP Process Orchestration*. Sap Community Network [online]. [cit. 2016-03-09]. Available from: http://scn.sap.com/community/b2b-integration

SAP, © 2016h. *Data Masking*. Sap Community Network [online]. [cit. 2016-03-09]. Available from: https://scn.sap.com/thread/1021980

SAP, © 2016i. *SAP NetWeaver Master Data Management: Elevate performance with consolidated, synchronized data – with master data management from SAP*. SAP Software Solutions | Technology & Applications [online]. [cit. 2016-03-09]. Available from: http://www.sap.com/pc/tech/enterprise-information-management/software/master-data/index.html

SAP, © 2016j. *Integrate, Transform, and Improve your enterprise data – with SAP Data Services*. SAP Software Solutions | Technology & Applications [online]. [cit. 2016-03-09]. Available from: http://www.sap.com/pc/tech/enterprise-information-management/software/data-quality/index.html

SAP, © 2016k. *Landscape Transformation Replication Server from SAP – Real-Time Data Integration for SAP HANA*. SAP Software Solutions | Technology & Applications [online]. [cit. 2016-03-09]. Available from: http://www.sap.com/asset/detail.2012-10-oct.sap-it-replication-server-real-time-data-integration-for-sap-hana-12-ov-us-mp4.html

SAP, © 2016l. *What is SAP HANA?* [online]. [cit. 2016-03-09]. Available from: https://hana.sap.com/abouthana.html

SCHWINN, Alexander and SCHELP, Joachim, 2005. Design patterns for Data Integration. *Journal of Enterprise Information Management* [online]. Bradford, **18**(4): 471-482 [cit. 2015-11-21]. ISSN 1741-0398. Available from: http://search.proquest.com.zdroje.vse.cz/docview/220031836/9C0D36305DD442BDPQ/1?accountid=17203

SHANCANG, Li, et al, 2015. *The internet of things: a survey*. Information Systems Frontiers [online]. April 2015, **17**(2) [cit. 2015-12-18]. ISSN 1387-3326. Available from: http://search.proquest.com.zdroje.vse.cz/docview/1664982373/E66E588246A64B1DPQ/3?accountid=17203

SLDC, 2011. *Waterfall Model*. Software Development Life Cycle (sdlc) Tutorials [online]. [cit. 2016-04-05]. Available from: https://www.sdlc.ws/waterfall-model/

SMART, M., © 2006-2011. *The Waterfall Development Methodology*. [online]. [cit. 2016-04-05]. Available from: http://learnaccessvba.com/application_development/waterfall_method.htm

SMITH, Benjamin and SIMON, Mark, 2009. *HOW DATA INTEGRATION SYSTEMS AFFECT STRATEGIC DECISION MAKING IN SMALL FIRMS*. *Journal of Small Business Strategy* [online]. Oakland University, **20**(1): 35 - 51 [cit. 2015-12-13]. ISSN 1081-8510. Available from: http://search.proquest.com.zdroje.vse.cz/docview/201475682/fulltextPDF?accountid=17203

Talend, © 2016a. *Talend Open Studio: Open Source ETL & Data Integration*. Talend Real-Time Open Source Data Integration Software [online]. [cit. 2016-01-31]. Available from: https://www.talend.com/products/talend-open-studio

Talend, © 2016b. *Talend Downloads: Download Integration Software.* Talend Real-Time Open Source Data Integration Software [online]. [cit. 2016-02-08]. Available from: https://www.talend.com/download

Talend, © 2016c. *Download Talend Free Products: Open Studio for ESB*. Talend Real-Time Open Source Data Integration Software [online]. [cit. 2016-03-15]. Available from: https://www.talend.com/download/talend-open-studio

Talend, © 2016d. *Download Talend Free Products: Open Studio for MDM: 5 Easy Uses of MDM to Improve a Data Integration Project*. Open Source Integration Software for the Enterprise [online]. [cit. 2016-02-14]. Available from: https://www.talend.com/download/talend-open-studio

Talend, © 2016e. *Levolor Streamlines its B2B Integration.* Talend Real-Time Open Source Data Integration Software [online]. [cit. 2016-02-14]. Available from: https://www.talend.com/customers/customer-reference/levolor-streamlines-its-b2b-integration

Talend, © 2016f. *tFileCopy - Talend Open Studio Components v5.2.1 - Reference Guide (EN)*. Talend Help Center - Talend Online Documentation & Knowledge Base [online]. [cit. 2016-02-20]. Available from: https://help.talend.com/display/TalendOpenStudioComponentsReferenceGuide521EN/13.6+tFileCopy

Talend, © 2016g. *Best Practice: Talend ESB*. Talend Help Center - Talend Online Documentation & Knowledge Base[online]. [cit. 2016-02-20]. Available from: https://help.talend.com/pages/viewpage.action?pageId=261422451

Talend, © 2016h. *Download Talend Free Products: Open Studio for ESB: Save Time and Avoid Headaches with a Concrete Services Governance Policy*. Talend Real-Time Open Source Data Integration Software [online]. [cit. 2016-03-15]. Available from: https://www.talend.com/download/talend-open-studio

Talend, © 2016i. *Talend Products: Data Quality*. Talend Open Source Data Integration Software [online]. [cit. 2016-03-09]. Available from: https://www.talend.com/products/data-quality

Talend, © 2016j. Talend Help Center: What's new in v5.4. Open Source Integration Software for the Enterprise [online]. [cit. 2016-02-08]. Available from: https://help.talend.com/pages/viewpage.action?pageId=190513443

Talend, © 2016k. Talend Help Center: tHMap. Open Source Integration Software for the Enterprise [online]. [cit. 2016-02-08]. Available from: https://help.talend.com/pages/viewpage.action?pageId=263003112

Talend, © 2016l. Why Talend?. Open Source Integration Software for the Enterprise [online]. [cit. 2016-04-01]. Available from: https://www.talend.com/why-talend

Talend, © 2016m. *Getting started with a basic Route - Talend Open Studio for ESB v5.2.1 - User Guide (EN).* Talend Help Center - Talend Online Documentation & Knowledge Base [online]. [cit. 2016-02-14]. Available from: https://help.talend.com/display/TalendOpenStudioforESBUserGuide521EN/5.3+Getting+started+with+a+basic+Route?thc_login=done&_ga=1.78438812.902337637.1454231925

Talend, 2015a. *Day In The Life of a Data Integration Developer Series - Part 4: Run Test Debug*. In: Youtube [online]. Released 24. 11. 2015a [vid. 2016-02-14]. Available from: https://www.youtube.com/watch?v=HTA7BYB5jRY&feature=youtu.be&list=PL0aRSCaII9DdhjMFq5l47d2O3LUc-VuKi

Talend, 2015b. *Day In The Life of a Data Integration Developer Series - Part 6: Basic Features Context Var*. In: Youtube [online]. Released 24. 11. 2015b [vid. 2016-02-14]. Available from: https://www.youtube.com/watch?v=iNO37x5VuSE&index=6&list=PL0aRSCaII9DdhjMFq5l47d2O3LUc-VuKi

Talend, 2015c. *Get started with data profiling through automatic semantic discovery, Part II - Talend 6 Features*. In: Youtube [online]. Released 10. 11. 2015c [vid. 2016-02-14]. Available from: https://www.youtube.com/watch?v=EltwXhaPASM

Talend, 2015d. *Secure your data with data masking - Talend 6 Features*. In: Youtube [online]. Released 9. 11. 2015d [vid. 2016-02-14]. Available from: https://www.youtube.com/watch?v=YTl8IgaoZv8

Tata Consultancy Services, © 2015. *Internet of Things: The Complete Reimaginative Force*. TCS: IT Services, Consulting and Business Solutions [online]. [cit. 2016-02-20]. Available from: http://sites.tcs.com/internet-of-things/companies-begin-to-make-a-big-thing-of-the-internet-of-things/

TDWI, © 2016. *The True Cost of Integration in the World of BI.* TDWI | Advancing all things data. | Business Intelligence, Data Warehousing, Analytics | Education & Research [online]. [cit. 2016-01-31]. Available from: https://tdwi.org/articles/2013/08/20/true-cost-of-integration.aspx

TechTarget, © 2016a. *Building a foundation with SAP Data Services*. TechTarget - Global Network of Information Technology Websites and Contributors [online]. [cit. 2016-01-30]. Available from: http://searchsap.techtarget.com/feature/Building-a-foundation-with-SAP-Data-Services

TLDP, 1996. *6.1 Introduction.* The Linux Documentation Project. [online] [cit. 2016-04-16]. Available from: http://www.tldp.org/LDP/lpg/node8.html#SECTION00710000000000000000

TOGAF © 1999-2011a. Part I – Introduction: 3. Definitions and Abbreviations: 3.37 Framework. The Open Group [online]. [cit. 2016-04-01]. Available from: http://pubs.opengroup.org/architecture/togaf9-doc/arch/

TROWBRIDGE, David, et .al, 2004. *Microsoft: Integration patterns* [online]. [cit. 2016-01-28]. ISBN 0-7356-1850-X. Available from: http://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0ahUKEwj6_4rAzMzKAhVkEHIKHae3BU0QFggqMAE&url=http%3A%2F%2Fdownload.microsoft.com%2Fdownload%2Fa%2Fc%2Ff%2Facf079ca-670e-4942-8a53-e587a0959d75%2Fintpatt.pdf&usg=AFQjCNETQXJI_MXTOFA0JcdnIJZN-MVlNw&bvm=bv.112766941,d.bGQ

UNECE, 2016a. *Introducing UN/EDIFACT*. The United Nations Economic Commission for Europe [online]. [cit. 2016-01-31]. Available from: http://www.unece.org/cefact/edifact/welcome.html

UNECE, 2016b. *UN/EDIFACT D.15B - Message [IFTMIN]*. The United Nations Economic Commission for Europe [online]. [cit. 2016-01-31]. Available from: http://www.unece.org/fileadmin/DAM/trade/untdid/d15b/trmd/iftmin_c.htm

VOŘÍŠEK, Jiří, et al, 2015. *Principy a modely řízení podnikové informatiky*. Second edition. Praha: Oeconomica, published by VŠE, 2015. ISBN 978-80-245-2086-5.

WENDE, Kristin, 2007. *A Model for Data Governance – Organising Accountabilities for Data Quality Management*. 18th Australasian Conference on Information Systems [online]. Toowoomba, 417-425 [cit. 2016-01-25]. Available from: http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1079&context=acis2007

WHITE, Andrew, et al, 2006. Mastering Master Data Management. Gartner Research. [online] [cit. 2016-04-16]. Available from: http://kona.kontera.com/IMAGE_DIR/pdf/MDM_gar_060125_MasteringMDMB.pdf

ZIEGLER, Patrick and DITTRICH, Klaus R, 2011. *THREE DECADES OF DATA INTEGRATION - ALL PROBLEMS SOLVED?* [online]. University of Zurich, Winterthurerstrasse 190, CH-8057 Zürich, Switzerland, 2011-05-31 [cit. 2015-11-20]. Available from: http://www.cin.ufpe.br/~if694/artigos/Three%20Decades%20of%20Data%20Integration.pdf

III. List of used figures and tables

List of figures

List of tables

Annexes

Interview questions and full replies

In this annex all interview questions and replies from all eight participants are protocolled. Since the answers of the experts could potentially be used to identify each participant, the answers have been randomly mixed together and are not clearly marked where they begin and end, however, the integrity of the answers has not been touched and any altered statements are clearly marked in square brackets [].Text was only altered in order to keep anonymity of DIP project. Answers which clearly stated "I don't know the answer" have been removed are not shown below.

What do you think makes a good Data Integration project?

Have a goal, have a clear plan, ideally standards will be in place beforehand like data standards, document standards, if you are integrating something, you should already have something existing and not make shit up as you go, have the right skillset, a clearly defined set of tools to work with. Know what are you working to, know what you are working from, knowing that you have right skill set, right project plan, timescales must be consistent to the task. If standards are not in place, then you have to have a company which will accept those standards [sic]. One of the biggest challenges in integration is making all the business units adhering to the same standards. There are good reasons for that, e.g. financial goals, pay scales, bonuses, so their interests collide with global interests, acceptance of change is hard. On the way to putting a new system in, you might even lose customers.

It is not only the staff, integration has to happen from director down. We started to include the directors who were responsible. We made it their responsibility for not having delays. This decreases reluctance of the staff, when all key players are on board. Everybody thinks that DI is responsibility of IT, but in reality data is owned by business, not IT. The business puts rubbish into the systems. So when you integrate this, IT is suddenly responsible for making sure that telephone fields only contain numbers and not addresses and to clean up the mess, but in reality the business is responsible that this doesn’t happen in the first place. Standards and governance have to happen first. When there is no solution there from begin with and IT should just make up shit then the problem will be quality. Data Integration project is not only about the data side, it is a lot about work ethics and the staff that will use it, most of that comes from change.

Data Integration is about Architecture and Design of Data structure. You need a good design of platform and data architecture. They are independent, but they have some relation-

ship. Data Integration is transformation and manipulation. And you need to process the data with high performance, no delays or failures. The platform needs capacity management, performance testing, to see if your platform will be able to sustain the volume, to process the full volume of data.

Many platforms have a Canonical Data Exchange Format. It's very good for design, because it allows standardisation, instead of having random formats. If you have 100 different types of formats, on the output you will have 100 other formats. Meaning the middle needs a standardised design format, so you can measure performance, volume, CPU utilization, data validation on ONE standard instead of having one for each different standard. This gives more flexibility for future, the Canonical Format allows structuring of the business model into one single format, which can be governed or versioned into one repository, which you like. Good version control is necessary to allow for parallel development in branches, for example SVN because it is free and very solid. Or also GitHub, you don't necessary need an expensive one.

Best is also not to keep things only in excel, Excel is dirty. Better is to use a database, or official repository. IBM InfoSphere for example. Data Integration is both about the quality and efficient design on the data structure itself and the platform on which it will run.

The biggest challenge is that technologies are advancing very fast. But a lot of companies are still sitting with their old systems and for them to go from the old systems to the new systems is a huge task. One of the biggest troubles we have is how do you get the data model to work, how do you make the old meet the new, how do you make them compliant then. So for me from a Data Integration perspective for a framework is to give tools and methodologies of how you can easily bridge this gap. How do you take two heterogeneous systems and make them talk to each other? This is the biggest challenge. So everybody has to trial and error. A framework should tell them "this is what you do, you start doing A and B" – that is what is missing I think. Everybody has their own methodology, everybody gives some bullshit, but I don't see a very robust framework right now and every company has to go through this, either they are bought or they buy somebody, typically this is what happens today. And then in every integration, Data is one of the biggest challenges, as it has everything to it, it defines how things work, from a legal perspective, from a financial perspective. For me Data Integration should be a list of steps of how you go across the challenge of making systems talk to each other.

Integrate the data in the right quality, in the right time to the right price. Being designed in such a way that any system can be integrated without redesign and without high effort.

Clear understanding of what you expect to achieve by Data Integration. A staged approach with clear scope of what is, and (even more importantly) what is not, included. Not trying to do too much in one go and avoiding scope creep. A well-documented plan with a lot of contingency for the many unknowns. Having skilled consultants, not cheap labour.

Having a stable integration framework with defined standards for use which will allow easy and automated ways of integration.

In my opinion it is - especially in large enterprises – well written and sound business case because it will get so much needed support from the top management. Then it is close collaboration from business specialists and IT data architects for definition of business requirements and its representation into master data and derived canonical data model (if company selects to use it). I would say it is very useful if the business and data architect are based in one location at least during design phase and start from core data and then add in details. Data architects with positive or negative experience from similar size project even from other company implementation is definitely a plus.

To make customers satisfied, Customers have to specify their requirements clearly, Customer expects that the Data Integration is an easy project, to have some backup like 4 pairs of servers, where everything is running in twice in case somewhere it would stop working, ability to react to problems and solve the problems (sometimes it is difficult to find the problem), problem can be e.g. that everyone talks by different language (not English vs. Czech, but e.g. syntax and semantic, applications are very different and do not understand each other, for example when one application is sending some document in specific order and something happens in the middle and the second application receives the documents in different order, it can be a problem).

If you had to build a second Data Integration solution in another project, what would you change?

Having the right tools in place. We have old versions of software, not the newest software being used on. We are still working on XML Spy version 2011. You should always try to stay up to the latest technology. Also people need to understand the tools which you are giving them. We never designed the canonical model properly, because we never had the timescales. There was too much pressure to get things out, so many things got out of the door wrongly. 80% of mapping are easy, however the last 20% are the ones which take most time. Also it does not make sense to have only one canonical model for everything, because changing can literally affect everything else, only adding is not a solution, because then the structure becomes „ghosted“. It makes sense to look at data first from a high level. There are [many] business operations in [DIP], the majority go through the main canonical model, but this is bad. A canonical model is good and reusable in other scenarios, but you should have less flexibility. If you have too much flexibility, it is too difficult to have governance and standards.

It is a big and complex project, most important is to set up standards how they should be and not to have half of the things according to old standards and half of them according to new standards. To think of standards in the beginning, how it should be. Architect should say how the standard look like and developer should develop according to the standard and

not to decide himself how the standard will be. For example someone sends multiple messages, each one in one row, someone as a separate messages and when there are thousands messages, this could be a problem e.g. with performance.

I joined [DIP] after its framework has already been built but from the various source and comments from other integration platform within [our company]. The downside of [DIP] is its too big complexity as designed. [DIP] was built as quite a scalable solution with idea there would be separate [DIP] instance setup in eg. [CDI] regional offices. [DIP] has been running few years now but this idea has never come into realization […]. Personally what I would put more emphasis on new [DIP] setup would be clear project structure from very beginning as well as naming convention of [DIP] artefacts. Also would involve much more internal staff, there have been many contractors involved so our design processes were not quite followed.

Use of industry best practices and tools that would allow message transformation requirements gathering and mapping a lot easier and automated. Use of such tools that would probably avoid need of a separate mapping team and build team. Efficient use of vendor partners in a better way for delivery execution which will help in quality deliverables and reduced cost; who could bring in industry level best experiences and practices.

Better documented business and interface requirements. Commercial mapping tool. Business ownership of the interfaces. Close interaction between business users and business analysts/mappers. Having business analysts/mappers and business users in close physical proximity if possible. Keeping business constantly involved with constant reviews to ensure not diverging from requirements. Having test plans specified by business analysts/mappers not by developers/testers. Ensuring developers and testers had full understanding of architecture. Creating a data model (or preferably using an Enterprise data model already created in the company) for each business area. Defining subsets of the data model for each business operation.

Force of better documentation of the flows, governance of master data, governance and definition of what if the minimum data set to trigger a message, proper [monitoring] from beginning on, the current [master data translation] tool right at the beginning, use proper integration tool suite (which done not exist yet) so we can avoid using XLS, agnostic data model and proper canonicals, no split between MDM team and Integration.

To be honest with you, I would actually define the Canonical Data structure better. Today our model is very, very generic and not based on our business, so not a business centric model. I would define it in such a way that every single field, every single attribute we define what is our business data model first. We need to define as a company “what is it that we want to do”. It is not always the same across companies. For me, we are too generic and we are struggling when different people want to understand it, they have very different understandings of it. It should be easy enough to understand, but it should be relevant for our business. So as an example “Payment Terms”: We should make a standard which is

saying “what is the standard of our company in terms of, we are now prepaid / collecting next to the payment terms, this is our standard”. If anybody maps from the outside, they map from their standard to our standard. We are missing that. So I would get the data model sorted correctly first. Once you get the data model correct, integration is just “maps”. You know A to B, B to C, but the whole intelligence is in the Canonical Data structure. You will understand the business better, if you have the right model.

I would also like to build more self-services. For example integrating is not something that is so complicated. People should be able to do it themselves. You provide them a Framework which says, for example, if you want to send me a file, here is your data structure and then it just works. It should not require so much manual interference. Because it is not rocket science. If you have a very well defined data model and you are able to write things into it, then the code can automatically generate that and it should work. You don’t need a developer who does A to B, because today you have tools in the market which do that for you. We still use developers, we still have build, so I would just do a technical design and then everything else should be done by the tools. If you want to do a project with us today, you have to come to our team and ask them for it. So people are hesitant to use it, because they don’t know how our system works, they can’t use it, it’s not theirs. So you need to sell it more like “this is your integration layer, anybody can use it”. I would like more plug and play and self service.

You need a flexible and reusable software that would allow us to store maps and efficiently change them in the project life. Better stored relationship between the map itself and the mapping specification. Anytime somebody can know which map is doing this from the mapping spec or opposite which mapping specification is on that code. Now it’s not that easy, you have to go to technical design document, which is again a word document, it’s not so straightforward to link both of them. Either use the same tool or use some kind of relationship that would connect the two together. Main problem is the push by stress, we had to push in 1000 maps, unrealistic timescales, we did not have time to do it properly. Instead spend more time on the standards of the platform. We had a requirement in one of the interfaces to do debatching. For that project we developed a debatched method and send it to production. What we SHOULD have done is to build a debatch component that will enable to do any kind of debatching and then deploy it and apply only to that interface, for reusability. Spend more time on quality and flexibility, reusability. This is very basic in every company. Knowledge sharing was also bad in the beginning. You need mapping experts, database experts, network experts, system experts, architects (for platform and mapping), top management, mapping experts, developers, testing managers, we did not communicate enough, we did not spend enough time on communicating these things. It is cooperation and collaboration between all the subject matter experts, like knowledge sharing sessions.

Can you name one example from the past years, what really improved in your project a lot? Or an example, what made it worse?

Having directorship involved in the project. Finance director, manufacturing director. Not only integration team. If you have those inside, everything else will become a lot easier.

Unrealistic timescales make things much worse. The business doesn't really know what they want, also you should consider the amount of money.

Something which improved is the Architecture and Design Group, it is a council of Architects, which assess new requests. This was a good, because it prevented from doing ad-hoc changes to the platform or to the data. Without looking at the overall impact and assessment. Basically before, if we had a new interface request, these interfaces needed a function, which we didn't have in [DIP]. So what happened before: One guy was told to create the function, one guy was told to create the interface and they just then put the function in the interface and RTP'd to production. Then we found out that there is this impact on the function, that it impacts maybe three other interfaces. If we had gone to ADG first, we would see that we cannot build this function this way, because it would impact three, so we would advise to do it that way. So the council is architects, experts of the platform and data that can assess together as a group, kind of what is happening in politics, like a Senat, Deputies and they all vote and it is the same here: You cannot simply change your Data Integration platform without assessing risk. So ADG is good, it is basically like a police. We don't change anything without going through ADG.

Example of what went bad: we bought a product for in memory caching. Two and a half years ago we knew we had problems without data translation tool, so we had lots of quantity of translation requests to properly manage. What we learned from that: Do not let senior management do a choice on the product, without experts, like ADG for example. Because what went wrong is that the product did not fit our use case, so they just bought it, because the product got sold with good marketing. But as it did not fit the use case, we had to throw it away afterwards. We lost a lot of money, time, build consulting resources, etc. So choice of product must be meticulously done by SMEs (Subject Matter Experts) [sic]. The product itself can be very good, but it was just not for us. This did not impact the project so much, but it was a loss of money and time, but for sure it was not the biggest failure. A far bigger mistake: It was a mistake to outsource to our external vendor the whole design of our IT program, we should have done this inhouse. Our vendor could have done implementation, but they didn't do only implementation, they did the entire design, functional specifications and so on, all of that was there side. So we had no control over that, even though we were the ones to use the product. It's like if you are a company and you are building a new car and it is a special car with futuristic functions with flying like that and you ask somebody else to design it for you and then you realise that it doesn't do what you wanted, if you had designed it yourself and we basically did opposite of what we did before: Our old system was build inside by internal experts, knowing our own business.

Well in terms of improvement, one of the biggest things that we changed were in the beginning people in this project thought that integration is data agnostic. They didn't care about data. They said "we are the integration hub, we only do messaging, we don't need to understand what we are moving, we just move A to B, we don't need to understand A, we don't need to understand the content of the messages". This is where we changed a lot and started more asking for "why". What is it that we are trying to integrate? Don't just integrate, because someone tells you "put a customer name into the street", first try to understand why. So I think we have become more data aware. There were no standards in the beginning, we are a little bit better now, because we have a standard way of doing things now, but I still think we are lacking a few things, like a proper data model.

What went wrong is that we as an organisation struggle to define our priorities. We have gone through a lot of change in the last year and changed our scope a lot. This makes it harder for the entire team to settle. I think it is important to set a goal for a team and go for it. We have too many goals which keep changing. This is not coming so much from the team, more from the management. We never took the time to do what is strategic, we always just did what was required immediately. Data and Integration is a lot about a strategy. If you define a strategy well, you can get a huge value out of it. We have to do more innovation and more value addition. We are always after satisfying a need, not truly innovating, doing something "cool". We did not have enough scope for innovation, in one word.

Another issue is documentation. It makes sense to document standards for example, but I honestly prefer to have smaller documents, which just give the right information. This is something a lot of companies are struggling with. All have huge libraries with millions of documents which make no sense, whatsoever, nobody ever reads them, they are not value adding.

What else went right: New [data translation] tool and integrated push to [our database], documentation of all enumerations source target, proper mapping template, four eye principle between mapping and development.

Having a skilled SAP consultant who knew the originating application and the business area very well improved things a huge amount. Having developers off-shore and even worse, in a different time zone was bad. Having developers that did not know the business was bad. Having developers that were not experts in the tool used to create the mappings was bad. Lack of communication between business analysts/mappers and developers was bad. One-sided mapping (i.e. from application to Canonical message, but not E2E) was a major failing. Expecting that 70% mapping was a good enough to develop mappings from – it just meant many iterations of the map/develop/test cycle.

Last couple of years saw a lot better management of the program, in terms of execution and cost effectiveness. And most importantly, all process etc were evolving and improving throughout. What really improved is support processes – split between technical support team and functional support team. What is also better defined and implemented is tracking

and monitoring of messages. Where I see place for improvement is not updated documentation of [DIP] framework, some processes are couple years old and knowledge lies rather in peoples brain than on Wiki. There is also quite a lot of changes still happening in architecture, this makes supportability a bit difficult.

Nothing what would be worse, just it is slow, mostly when we leave the standards, when it happens once, it is okay, but when 5, 10 times, it is a problem. 80% things works standardized and the rest works as well but completely different. What became better: [revising standard], where most of the exceptions (the 20%) to transform to the same standards as the 80% working stuff. Second thing which could help – tools, mainly deployment tool, which would take care of environment and which version of code is used. It is already implemented and soon there will be a project about deployment to production. It will make the projects faster and it will make the project management easier.

How do you think does your current project compare to other current market solutions, like IBM, Informatica, SAP, Oracle (or any other that you might know)? Do you think there are any differences at all?

[DIP] is not an end point solution. We have a good thing, [master data translation]. IBM supplies also one of those [tools] for 1 000 000, building our own solution made us save a lot of money. Informatica is an ETL Tool, only used in really large companies, because it is bleeding expensive. Flipside: It is really slow. Our solutions works almost like a point to point application, while Informatica is focussed on ETL, we more act like a very quick messaging bus, Informatica seems to only handle point to point really well. SAP is better for ERP, while we are more of a middle ware, Oracle is a backend DB. You can take data out of the DB and send it somewhere else. We are also using Oracle, lots of error messages end up in a Oracle DB. But our application is pretty standalone. Data Integration usually gets rid of something old to replace it something new. Informatica can take data from a stock exchange, from SAP into a database. The problem was they were enhancing the data on the way, we generally don't do that. I was working in another company, with a different focus, they didn't integrate, they just had masses of datasets. I also worked for a […] company, which kept transactional records down to [a very atomic] level, that's mass amounts of data, not really used for trend analysis, it was used most for legal reasons, what they needed was a CRM to analyse the data and target customers based on [how they were using the services]. Most of that was just number crunching.

I have some experience with these products [mentioned in the question], I used them before. The difference is that our product is customized to our business case, which is unique in the world. So there are other companies doing the same business as we do, they have a similar business strategy and nature, but completely different business use cases and implementations, so the difference of all those products is that they address different business cases. In our project we couldn't just buy something, we had to either buy something cus-

tomized or create from scratch. If you compare SAP with Oracle for example, it's kinda like you ask someone to buy a PC or a Mac, it's the same. It is preference. Marketing and preference. Depends on the relationship between the companies. It looks like our company did not want to buy [one of the companies mentioned in the question], but they never said why. Maybe it was political, maybe there was a contract. There are not so much differences between the products – of course experts would tell you that there are, but if you look overall, they provide similar functionality, it is just one is better to somebodies eyes and the other is better to the other person's eyes. It depends on the use case. For example for our company, [SAP] might be better than [Oracle], but for let's say Adidas [Oracle] could be better than [SAP]. It depends how customized your business strategy is.

In terms of the software, all of them provide you an integration framework to do things. But that is not the heart of the solution. I am going back the data model: The heart of your solution is what you want to do with it. In terms of the Gartner Quadrant, if you see it and all the companies on it, they all provide really good frameworks, some work in some companies, some don't work in some companies, some are well scalable, some not so well for the kind of business model and data volumes that you have. But essentially there is not much difference across it. People think that a software can help you define the heart of your project, but software is a tool. And all of those companies are just integration tools. So if you have a framework, they just give you a playground where you can go and play but how you play is what matters. Just "who plays, what sport you play", that matters. Just because you have a different playground it doesn't mean you can do whatever. This is why we simply use Gartner and we just ask them "who is in the top of the market right now". But what really helps when we go for these vendors is that we can look, how mature is the software as in how people are using it. What is their customer base? What scale do they have? What kind of support do they offer? Because these kind of offers, once you buy them, you attempt to use them the next 20, 30 years. And if you buy a fantastic software from a small company and they don't do their support properly and they don't do their updated properly, within a few days you are stuck with a white elephant, you can't deal with it. On the other hand, you may have a software which is very good, looks good, nobody is using it. Then you got some more software where the enhancements and innovation comes very slow, then you're back in the market for example today you do this, tomorrow you want to have a new feature, which the market demands, if the company doesn't do that, then you are inherently behind the market. So you have to look at the software provider's innovation, its scalability, his performance, his track record and in the end always: the price. But in the end of the day all of this doesn't matter. Some people say product A is better, some people say product B is better. It just depends on what you use and what your enterprise has skills at. If we buy a certain piece of software and you have no talent to use it, what do you do with it?

If I understand it correctly, [DIP] is a […] integration platform from supplier […], and has particular strength in application-to-application integration and is positioned as leader in Gartner magic quadrant. IBM and Oracle are very close to it so would core functionality

would be similar but details would differ. Since Oracle and IBM also supply database and OS products I would expect tighter collaboration/integration with those products there.

Do you think that there is a technical aspect which makes your project stands out?

Beautiful about our project is: It is not point to point. Point to Point limits you to the data structure of this one stream. But we decode a message to a generic structure. So you only need to manipulate one side, if one side changes and it is bleeding fast, very modular: It is using a simplified decorator pattern (on java architecture side), in essence, it makes it easier to just put modules into a process chain. Also we are bloody fast. We support multiple business areas. The internal processing is the same. Companies are paranoid about „you must understand how this business works". Data is data. When there are rules and regulations in place, you need someone who understands data. If you have the right functional consultants in place, then that data person doesn't need to understand how applications work, because the functional consultants will support, the business persons will understand the business. But not everyone understands what is clean data. You need skills from all areas. You need a person who looks after the data, the data architect. „If I build this structure like this, it will work how it is supposed to", you need mappers, you have an integration architect, technical development architect and coders. „Data is like blood": It flows around, you need to keep it healthy. When the blood is ill the body will die.

This is huge project – integration of [many] countries with local applications. Regarding to the data it is not so huge, there are processed roughly millions of messages, it is not so much, for example telephone operators have to process messages from all customers, which are much higher. There is the backup (Disaster Recovery) in [another country], so when something happens […], we are able to switch over everything to [the other country]. [DIP] architecture and its integration aspects are quite good and scalable.

What stands out: Size, Scope if [sic] the project, Proper Disaster Recovery, Teamwork.

Our project is built on a complete reasonable framework. The way everything was designed is that everything is a service, which you build only once. Let's say you want a service to convert a flat file to an XML. Build that service once and everybody else uses that service. So you might require this to integrate with application A and with application B, you don't do it twice. So the way we have built our platform is, everything is like a jigsaw puzzle. If you put two, three things together, you will get one result, if you put other things together you get a different result. Everything is reusable. If you build a document handling service, everyone can use it. If you build a service which converts everything to capital letters, everyone can use it. So the concept that everything is a service is a very unique factor. Most big projects with many people in different locations tend to build things again and again.

Well for the data part we have managed to represent our business case into one canonical model. So we were able to consolidate 100 message types into one standard, this is for me something special, which for example other solutions doesn't have. For our platform, we designed a distributed pattern. There are two main integration patterns [for platforms] which you can implement: Distributed or centralized platform. We have done distributed and it is a big plus, because we are able to scale the platform horizontally. In a centralized platform, you have a central processing component and you have other nodes, so it is like a star schema. The problem is: if the central component fails, the entire platform fails. We have done it a distributed way, the more we go in time the more transaction we process. We have central components, but we have distribution. Let's say you are today processing one million messages here, but in one month it will be 100 million. For a centralized solution, you cannot simply add another central node. You wither have to build again the whole thing or completely migrate the platform and turn everything off and build a bigger one somewhere else. In a distributed platform, you do not have to do this. You just can add more distributed nodes. The metadata, code, master data, error processing systems is replicated though all nodes with each deployment. So basically this is only about deployment. All in all, this makes the solution robust and performing.

Do you think that the right tools make the right Data Integration solution?

Yes of course. For example, take our translation tool. We built that ourselves to our own needs. If you have the right staff, you can build it yourself or you can buy it off the shelf. 50.000 to build it yourself, or 100.000 for each year, if you buy it off the shelf. However, from the shelf you at least have a right to get support, while internally you have dependency on your skillset. Also if there is no tool available on the market, then you have to go down the route to write it individually, then you also need to know exactly what you want. The challenge is to design a data model that doesn't require you to know the system you are going to.

I think tools contribute, for sure, but it's not itself alone sustainable... it is not enough just to have the right tools. But you NEED the right tools. So for example to do a distributed deployment, we have the [special Database], which is the right tool to do this kind of solution (there are also others). But the right tools are not the most important thing. Like, you need the right people and the right expertise and the proper understanding of the business. What tool you buy after, is always up to you, there is always a lot in the market and there is always more than one option. For example, even though CouchDB [might be] a good option, MongoDB [might also be good for the same]. The tools really don't matter so much as long as you have good designers, good management, good people with the proper knowledge.

I think that in general: I see Data and Integration as two separate things. One feeds the other. The Data is processed by the Integration. The Integration handles the Data. It can do

anything with the Data. It can transform the data, it can move it around, it can manipulate it, like value conversion, so it is like the Data is information that your brain processes. And the brain is the integration that contains synapses and nerves and all that, which are the Data. So look at the platform: If the brain is too small, it doesn't matter how good your data is, it will not get properly processed. Or when you have a really big brain, but it is full of bad data, or bad thoughts, it will also not produce good results. Both need to be designed properly, independently, but then also they work together. Platforms are about volume, capacity, quality, like a map that performs in 10 seconds... if you have a million of those every couple hours, you need to make the map much faster. You need to use the proper languages, like JAVA or C++.

We have a severe lack of tools. We need to have a good mapping tool. We need to have a good data modelling tool. We need to have a good testing tool, which we have built ourselves now. Again the thing is whenever you scale, tools really help you. When you want to ramp up, imagine if we would hire 50 more mappers into our team, we would suffer to train so many people. If you have a tool which provides you a standard process, then you just say: "here this is what we are using, this is what we do, this is how it works". For example now we have a release management tool, it took us five years to get to that, we used to manually release everything. We still don't have a mapping tool, we don't have a data glossary. So it is very, very important that you identify those key aspects, because software and IT is always about making life easier. If you go and get a glass of water every day from a shop that is let's say 10 minutes walk, then after five days you simply keep a bottle of water with you, because you want to make life easier for yourself. Tools help you to do that. Because for example when we started we had so much trouble with our translation tables, now with our self-made master data translation tool it makes it so easy for us to make updates. Some people for some reason don't get this. I always like lazy developers, because lazy developers find the shortest and the fastest way of doing things. What is really missing in the market is a good data mapping tool. Each company is selling integration services, but none of them offer a tool that makes documenting such maps easier.

Our tools were build out of necessity. Where we had some problems, we built some tools. Like our translation tool, release management, testing automation. We always build something to cover our biggest pain points. Now that the pain is gone we should focus more on "how do I do my work in half the time?" Sometimes just buying a tool can save you a lot of money.

[Tools are a] substantial part of [the solution] but if not used or designed property [sic] tools remain tools. They help to ensure a better quality and better documented solution. The right tools can be sometime a bit expensive. Eg. XMLSpy, one of favourite XML editors from Altova supplier cost about 800-1000 EUR per user. It is though a very powerful tool with many functionalities. As for the integration framework or middleware for enterprise needs this has to be powerful and reliable enough to handle lots of data. Also it has to have a stable supplier for support in case of any difficulties and also development roadmap

to implement modern functionality – eg. mobility features like webmethods have. But it will still be just a tool. There needs to be as well good cooperation between integration and data architects and business users so the middleware design and data model design support business processes well.

Regarding to commercial tools - yes, the tools which we have are on a very good level and they are important (like […] the new tool for deployment). When you have good solution but the tools do not support the solution, it is useless. Internal tools – we do not use internal tools so much but we customize the commercial tools, one good internal tool is for monitoring. Advantage of internal tools is, that people write them how they need and they are not forced to use the functionality, which they do not need. Internal tools do not have aim to be universal and to have a lot of customers and make a lot of money. [We also use] internal tool for monitoring entire landscape, one guy is writing code for that and his boss already starts thinking about to who to sell that to make some profit.

Index