# *Selecting a BI Architecture that fits Organisation's Requirements*

*17 July 2014*

UNIVERSITY OF TWENTE.

# *General Information*

## Master Thesis
## Selecting a BI Architecture that fits Organisation's Requirements

Zwolle, 17 July 2014

### Author

*Robert Spruijt*
Institute: University of Twente
Programme: Master Business Information Technology
Student Number: s1009583
Mail: r.spruijt@alumnus.utwente.nl

### Supervisors University of Twente:

*dr. C. Amrit*
Faculty: School of Management and Governance (SMG)
Department: Industrial Engineering and Business Information Systems (IEBIS)
Mail: c.amrit@utwente.nl

*dr. ir. M.J. van Sinderen*
Faculty: Electrical Engineering, Mathematics and Computer Science (EEMCS)
Department: Information Systems (IS)
Mail: m.j.vansinderen@utwente.nl

### Supervisor PwC:

*drs. M.P. Veensma-Hof RE*
Department: PwC Accountants N.V., Risk Assurance
Mail: marjan.veensma-hof@nl.pwc.com

UNIVERSITY OF TWENTE.

pwc

# *Management Summary*

Ever increasing amounts of data and business environments that keep changing more rapidly every day are two developments that stimulate organisations even more to implement a business intelligence (BI) solution. Business intelligence can help organisations to analyse their data, support them in decision-making and create reports and dashboards to provide first-rate business information and knowledge. The challenge many organisations face today is that they have a hard time selecting the right BI architecture given their specific situation. This master thesis aims to help organisations with their choice of a BI architecture.

The main research question is *'Which BI architecture fits best to organisation's requirements?'* In order to answer this, we first looked at BI itself; what it is exactly, but also its relevance and the benefits, costs and risks associated with it. In short, we can say that BI has both organisational and technological aspects, where the analysis of data, report generation and decision-making support are three of the most important tasks. BI is very relevant and will only become more relevant in the future. Two main reasons for this are the need for faster reporting due to faster changing business environments and the increase in data that is available to organisations for analysis, which can be a source of competitive advantage. There are a lot of benefits to the usage of BI. Examples include increased information sharing, improved efficiency and faster decision-making. However, there are also different types of costs associated with BI: hardware costs, software costs, implementation costs and personnel costs. The most important risk to BI is data quality. The quality of the output of the BI tool can only be as good as the underlying data on which it is based; 'garbage in, garbage out'.

Secondly, we defined a general structure of a BI solution, existing of four layers: 1) the data sources/operational systems, 2) the integration services, 3) the data repositories and 4) the analytical facilities. In this thesis we focus on the integration services layer and the data repositories layer. The main reason that we did not include the first layer is that the data sources/operational systems are usually already in place and are not likely to be changed for a BI implementation. The main reason that we did not include the fourth layer consisting of the analytical facilities is due to scoping considerations, since that could be the subject of an entire thesis itself. We described all the available options in these two layers that we found in the literature. Next, the possible combinations of the data repositories and the integration services were described, which we call 'BI architectures'. After that, the differences between these BI architectures were described.

Thirdly, we used the literature to identify a list of requirements that organisation may have, which may influence the architectural choice.

After that, a connection was made between the identified requirements and BI architectures. This was done by developing a framework (see table 6 on page 44) that maps the requirements to the different BI architectures. Scores were assigned to each architecture for every requirement on a 5-point scale. The higher the score, the better the architecture is suited to support that requirement.

The framework can be used as follows: an organisation can indicate which requirements are important, but also the degree of importance by assigning a weight to those requirements (on a 5-point scale). For every requirement that is indicated by the organisation as important, we multiplied the assigned weight with the score of an architecture on that particular requirement. This is done for each architecture. Then, by adding up the multiplied scores of each requirement for every architecture, we get a total score for each architecture. The architecture with the highest score is the 'best' architecture given those requirements and weights. It is important to keep in mind that the results of the framework should still be interpreted with care. For example, there may be organisational specific, unique requirements which are not considered in the framework.

Lastly, the developed framework was validated. Three interviews were performed and a total of 21 responses to a distributed questionnaire were received. We can conclude that the integration service used by organisations is not always in-line with our framework. We identified possible reasons why this may be the case, none of which would invalidate our framework. Generally, we can say that the results support the recommendations by the developed framework, but more research is required to come with convincing evidence.

# Acknowledgements

This thesis is the final result of my research on Business Intelligence Architectures, performed at PwC. This research concludes my study 'Master Business Information Technology' at the University of Twente, but also my time as a student. I look forward to applying in practise, the knowledge I gained during my years of study.

First of all, I would like to thank Chintan Amrit and Marten van Sinderen as supervisors from the University of Twente for their valuable feedback and advice over the past months.

Secondly, I would like to thank PwC for giving me the opportunity to write my master thesis and providing me with the resources and the network to help conduct my research. Special thanks go out to Marjan Veensma-Hof as external supervisor. She gave me valuable feedback, but also gave me advice on how to approach and conduct the research. Next to that, I would also like to thank all the other colleagues of PwC for their suggestions, advice, practical experience and for bringing me into contact with customers with a BI solution.

Third, I would also like to thank the organisations that were willing to help me with my research; either via interviews or by responding to the questionnaire. Without their answers about their experiences in practise, I would not have been able to validate my framework.

Last, I would also like to thank my parents for their support and help, not only during this research, but also during my years of study.

I hope that you enjoy reading this thesis and that it gives what you are looking for. In case you have any questions, please feel free to contact me.

# Table of Contents

# 1. Introduction

## 1.1. Practical Problem and its Background

More than 90% of all data in the world has been generated over the past two years [72]. Next to that, the amount of data stored and acquired by organisations is ever increasing. This development has been fuelled by the declining costs of acquiring and storing very large amounts of data [16]. These large amounts of data offer organisations a lot of opportunities.

One area with a lot of potential - enabled by this development - is business intelligence (BI). BI has become the top priority for a lot of organisations who want to make the most of their data and want to use BI to facilitate decision-making, reporting and querying [40][60].

There are multiple BI architectures available when selecting and implementing a BI solution. Several choices can be made between integration services, the data warehouse architectures and the analytical facilities. For example, one could choose for a centralised data warehouse, populated using Extract, Transform, Load (ETL) and the solution Oracle BI performing the analysis. However, is this the best choice?

Gartner [25] stated the following: *'When building capabilities for business intelligence, most enterprises focus on the elements that are visible to the business users: functionality in query/reporting tools and BI applications, training on these tools and applications, and the impact of BI on critical business processes. Far too little time is spent on "behind the scenes" or "hidden" aspects of BI: the critical underpinnings that ensure a robust implementation capable of delivering insight in a reliable, scalable and flexible manner. The architecture of the individual components, as well as the overall BI solution, can make or break a BI effort.'*

In this thesis, we aim to help organisations in the selection of a BI architecture, paying attention to all aspects of BI. We identified three possible causes that can make the selection of a BI architecture difficult for organisations. There may be other causes, but these three are the ones we most often heard when speaking with experts on this topic.

1. It is unclear which architectural options are available.
2. It is unclear what the differences between the architectures are and what the resulting organisational impact is.
3. It is unclear which architecture should be selected given particular requirements.

By dealing with these three causes, this thesis will help organisations in the selection of a BI architecture.

## 1.2. Problem Statement

Many organisations already have a BI solution in place or plan to implement one in the near future. However, there are multiple BI architectures available and selecting the right architecture for your organisation can be a daunting task. Different architectures have different advantages and disadvantages, but which architecture is the best option given organisation's requirements?

One of the issues when choosing a sub optimal BI architecture is that it can limit the potential that BI can offer an organisation. In order for organisations to make the most of their data, organisations should select a BI architecture that best fits their requirements.

## 1.3. Objective

The main objective of this master thesis is <u>*to help organisations with the selection of their BI architecture*</u>. This is done by providing a framework which maps the different BI architectures to particular requirements. Given the requirements of the organisation, the framework suggests a BI architecture that would fit 'best' to that organisation.

All three possible causes of a difficult selection process identified in section 1.1 can be described as a general lack of knowledge. In this thesis, we aim to provide this knowledge by first presenting an overview of the different architectures and the differences between those architectures. After that, we create a list of requirements that may have implications for the architectural choice. Lastly, we provide a framework which maps the BI architectures to the different requirements. The developed framework is validated using the results of performed expert interviews and the results of a questionnaire.

## 1.4. Question Formulation

The main research question of this master thesis is: *Which BI architecture fits best to organisation's requirements?*

To answer this question we will answer the following sub questions:

1. What is business intelligence?

The goal of this sub question is to get background information on the topic of business intelligence. With this sub question we aim to establish a clear understanding of BI before embarking on actual research. We aim to provide an overview on the topic of BI. There are a lot of different definitions available in the literature. We discuss these different definitions and introduce the definition that fits best to our perception of BI. Next to that, we discuss the relevance, costs, benefits and risks associated with BI.

2. Which BI architectures exist?

In the answer to this sub section, we aim to describe the general structure of a BI solution and list the different options that can be chosen for the different parts of the BI solution. After that, we discuss the different combinations, which form the BI architectures.

3. What are the differences between these architectures?

This sub question aims to clarify the differences between the different BI architectures identified in sub question 2. We aim to do this by listing a number of characteristics and scoring the different BI architectures on these characteristics.

4. What are the requirements that influence the choice for a particular BI architecture?

In order to recommend a particular BI architecture to an organisation, we need to understand the requirements that are important to that organisation which influence the architectural choice. In this sub question, we do not aim to provide an exhaustive list of requirements that may influence the architectural choice, but we do aim to cover the most important requirements that may play a role in the selection process.

5. What does the framework that maps the different BI architectures to the requirements look like?

In this sub question, the framework which maps the requirements to the different BI architectures is developed. This allows organisations to assign a weight to particular requirements and, given those weights, help the organisation decide which BI architecture should be selected.

## 1.5. Structure of this Thesis

This master thesis has the following structure. In chapter 2, the research plan is given, which contains a description of the activities, methodology and data collection techniques that were used. Chapter 3 provides an overview of BI (RQ1). Chapter 4 discusses the different BI architectures (RQ2) and the differences between those architectures (RQ3). In chapter 5, the different requirements that may influence the decision for selecting one architecture over the other are discussed (RQ4). Chapter 6 provides the framework which maps the different requirements to different architectures (RQ5). Chapter 7 shows and discusses the results of the distributed questionnaire and the performed interviews in order to validate the framework. In chapter 8, we discuss the limitations of this research and in chapter 9 a conclusion is drawn, the contribution to both practise and literature is discussed and suggestions for future research are made.

# 2. Research Plan

In this section, we describe the activities that were performed, the methodology that was followed and the data collection techniques that were used in order to perform this research.

## 2.1. Activities

In order to carry out this research we performed a systematic literature review (SLR), conducted expert interviews, distributed a questionnaire and performed interviews with PwC customers who already went through the process of selecting a BI solution.

To answer RQ1, RQ2, RQ3 and RQ4 we performed a systematic literature review. This gave us insight into the topic of BI, the architectures, the differences between those architectures and the different requirements. We also performed expert interviews at PwC. To answer RQ2 and RQ4, we additionally used the results of the distributed questionnaire and the performed interviews at organisations that have a BI solution in place, to identify additional requirements and architectures.

To answer RQ5 we used the gained knowledge of the previous research questions and performed additional expert interviews within PwC. This gave us the knowledge to create a framework which maps the different BI architectures (identified in RQ2) to the specific requirements (identified in RQ4). Finally, we validated our framework by distributing questionnaires and performing interviews with customers of PwC. By doing this, we identified their requirements and the choices they made and checked whether these were in line with our proposed framework. We did this by performing interviews and by distributing a questionnaire to organisations that already went through the process of selecting a BI solution.

## 2.2. Methodology

We used a questionnaire and interviews in order to check whether the decisions made in practise are in line with the choice our framework suggests. We also performed expert interviews within PwC. The rest of the data was gathered by performing a systematic literature review. Next to that, we also used literature sources and project documents recommended to us by experts.

The systematic literature review was performed by using Scopus, Google Scholar and the Global Knowledge Gateway of PwC. The reason for using the Global Knowledge Gateway of PwC is that it gives us access to additional resources that are not available via Google Scholar. Next to that, according to Falagas et al. [23] Scopus offers about 20% more coverage than Web of Science. So that's why we chose Scopus instead of Web of Science. Falagas et al. [23] also state that Google Scholar offers results of inconsistent accuracy. To overcome this, we used a combination of these three search engines. That way, we think that we cover the most relevant articles for our research.

Since BI is a very hot topic, a lot of results were shown for some of our search queries. We could not really make these queries more specific without excluding some very relevant results. However, a lot of the other results did not really provide us with the information we were looking for. This was especially the case with results that were not on the first five pages of the search engine when sorted on relevance.

To deal with this, we filtered the results in the following way. In Google Scholar, we excluded patents and citations and only looked at the 50 most relevant results (the first five pages) when there were more than 50 hits. We read the abstract and title of these results and those that – in our opinion – could be useful for our research were stored for future reference. In the Global Knowledge Gateway, we did more or less the same. We filtered the results by requiring it to be either Dutch or English and looked at the 50 most relevant results (the first five pages) when there were more than 50 hits. Here, we scanned the document itself and determined whether this could be useful to our research, if so, we stored it. In Scopus we looked at the first 100 results in case we had more than 100 hits (first five pages) and read the title and abstract of these articles in order to determine whether they would be relevant for our research or not. Again, we stored the articles that we deemed relevant. Unfortunately, we could not access every result of Scopus, since the University of Twente has access to a lot of journals, but not all. By searching on those articles via Google, we were able to find some more, but unfortunately not every single one.

Appendix B shows the keywords we used, the date of the search, the number of hits for each query per database and the number of articles we deemed relevant. In addition to these articles, we also used some sources that were recommended to us by interviewees or by experts from either the university or PwC.

## *2.3. Data Collection Techniques*

As described in the previous subsections, we distributed a questionnaire and interviewed organisations that have a BI solution in place. We did this in order to validate our work. We chose to perform semi-structured interviews. The reason for this is that we had a set of standardised questions that were necessary to validate our framework, but besides that we also wanted to leave room to discover important and relevant information that we otherwise might not have gotten when we would have chosen for a completely structured interview.

Before distributing the questionnaire, we asked a couple of experts to take a look at it. By doing that, we tried to minimise the chance of misinterpretation. Based on the expert feedback, we made a couple of improvements to the questionnaire to make it more clear. When possible, we tried to distribute the questionnaire via PwC employees who are in contact with organisations that have a BI solution in place. There are two reasons for this: 1) we expected that people were more likely to respond when someone who they know asks them to fill in a questionnaire and 2) the PwC employees often had contact information of the right person to fill in the questionnaire. Since the number of responses was rather low, we approached a number of additional organisations by ourselves. To do that, we searched for organisations with a BI tool in the Netherlands (since our questionnaire was in Dutch) and send a personal mail to the right person.

We received a total of 21 responses to our questionnaire and performed three interviews. Based on these results we drew our conclusions on the validity of our framework.

# 3. Business Intelligence

This chapter aims to provide an overview of the current state of BI by performing a SLR. We analysed the literature in order to 1) get an understanding of what BI exactly is and to define the term BI, 2) describe the relevance of implementing a BI solution in an organisation, 3) identify the benefits of BI, 4) identify the type of costs of a BI solution and 5) identify the risks associated Business Intelligence.

## 3.1. Definition

There seems to be no generally agreed-upon conception of BI, but rather each author has promoted his or her own idea of its connotations [31]. In the literature, numerous definitions have been used. Öykü [56] listed fifteen definitions and concluded that some authors approach the term from a technical perspective, while others take a more organisational view and see BI as a cross-organisational decision-support system.

Olszak [53] did the same and listed 21 different definitions of BI and concluded that the articles, papers and reports that she examined mainly associated BI with:

1. Tools, technologies and software products
2. Knowledge management
3. Decision support systems
4. Dashboards
5. New working culture with information
6. Process
7. Analytics and advanced analysis
8. Competitive Business Intelligence

In order to get a common understanding of what exactly BI comprises, we also looked at several definitions [7][16][31][51][49][60][61], in addition to those listed by Öykü and Olszak [53][56].

Before we describe what we consider as business intelligence, we will explain our reasoning for selecting that specific definition. We will do so by commenting on the analysis of Öykü and Olszak.

Remark on the analysis of Öykü:

- In our opinion, BI has technological as well as organisational aspects: it is the organisation that has specific questions and BI can help in answering those questions. However, technology is required in order to answer those questions. Some definitions, however, only focus on one of the two aspects. In our opinion, those definitions only give a partial view of what BI comprises.

Remarks on the analysis of Olszak:

1. Due to the vast amounts of data available to organisations nowadays, one cannot work without automated processes to analyse the data. Tools, technologies and software products are an essential part of BI and thus - in our opinion - should be included in the definition.

2. As pointed out by Herschel & Jones [36], although knowledge management and BI are closely related to each other, these are two different, but overlapping things. Knowledge management is described as a systematic process of finding, selecting, organising, distilling and presenting information. It helps an organisation in gaining insight and understanding using its own experience. BI on the other hand is described as a set of technologies that gather and analyse data to improve decision-making, create reports and dashboards. However, in order to be able to do so, many parts of the knowledge management process should be performed. So there is a lot of overlap, but the terms are not exactly the same. If we were to include the knowledge management term in the definition of BI, this would introduce some ambiguity unless we also define knowledge management. Therefore, we decided to keep it simple and did not include knowledge management in the definition.

3. While an important goal of BI is to help with decision-making, it is not solely a decision support system. We therefore think that using DSS in the definition is inappropriate, since it narrows down BI too much.

4. A dashboard is a way to show information to the user. However, this is only one of the many ways to do so; we therefore suggest not using this in the definition as it is too specific.

5. A new working culture with information does not say much about BI itself, but more about the people working with it. We therefore think that in the definition of BI this should be excluded. The reason for this is that we aim to sharply define BI; possible influences that it may have on the people working with it are – in our opinion – too broad and ambiguous and do not define BI itself.

6. The term process is so broad that we prefer not to use this term in itself. A combination with other words to make it more specific/concrete may be possible.

7. Analytics and advanced analysis is very core to BI; therefore we think that this should be part of the definition.

8. Competitive Business Intelligence is one of many BI subareas and since we cannot include all subareas in a comprehensive definition we rather leave this out. Next to that, this would still require us to define competitive business intelligence and all other subareas of BI.

Taking into account the several remarks made above, we looked at all definitions in order to see if we could find one that fits our perception of BI. Eventually, we decided to use the definition of BI by Raisinghani [60] since we think this covers all aspects of the term in a very good way:

*"BI is a general term for applications, platforms, tools, and technologies that support the process of exploring business data, data relationships, and trends. BI applications provide companies with the means to gather and analyse data that facilitates reporting, querying, and decision making."*

## 3.2. Relevance of BI

Olszak & Ziemba [54] describe several management information systems (MIS) such as expert systems (ES), decision support systems (DSS) and executive information systems (EIS) and how they have not always met decision makers' expectations, such as helping with:

- Making decisions under time pressure
- Monitoring competition
- Possessing information that includes different points of view on their organisation
- Carrying out constant analysis of data and consider different variants of organisational performance.

These systems were simply not fit to handle integration of different, dispersed and often heterogeneous data. They were unable to effectively interpret such data in broad contexts and they are incapable of discovering new data interdependencies. Reasons are to be found in improper techniques of data acquisition, analysis, discovery and visualisation. Figure 1 shows the development of MIS.

BI differs from MIS, DSS, ES and EIS on the following aspects, 1) it has a wider thematic range, 2) it is able to perform multi-variate analysis, 3) it can deal with semi-structured data originating from different sources and 4) facilitates multidimensional data presentation [29]. BI can support decision-making on all levels in an organisation [52].

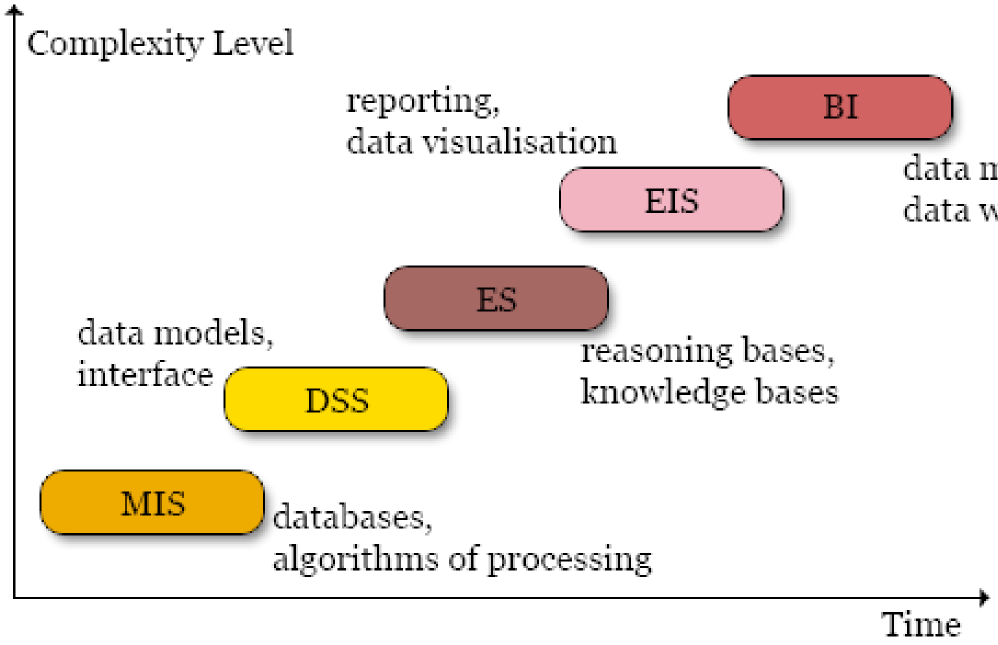



**Figure 1. Development of Management Information Systems**

***Source Olszak & Ziemba [55]***

In order to react quickly to changes that take place on the market, organisations need a MIS that would make it possible to carry out different cause and effect analyses of organisations themselves and their environments [59].

Hannula & Pirttimäki [33] describe the importance of BI in a very interesting way. They suggest that, in order to stay competitive, an organisation should be able to quickly respond to a changing business environment. In order to do so, management requires timely, first-rate business information and knowledge.

This kind of information and knowledge (BI) will only become more important. The reason for this is that business environments change and will continue to do so in an even faster pace in the future. In addition, the cost of data acquisition and data storage has declined significantly [16]. Therefore, organisations have access to more data and filtering the relevant parts out of these large amounts of data is getting more important as well. BI software can help in doing so.

On top of that, competing organisations also implement BI solutions. If organisations do not keep up with this development, they risk being left behind due to the fact that their competitors are able to make better and faster decisions.

## 3.3. Benefits

Hannula & Pirttimäki [33] performed an empirical research on 50 Finnish companies in order to find out what BI represents for those companies. The companies that were interviewed pointed out which benefits of BI were most relevant to them. A lot of these benefits were also identified by other authors [51][75][76][79]. Below, we show the benefits identified as most relevant by Hannula & Pirttimäki [33]. By no means we claim that this list is exhaustive, other benefits may be identified.

- Better quality information
- Better observation of threats and opportunities
- Growth of knowledge-base
- Increased information sharing

- Improved efficiency
- Easier information acquisition and analysis
- Faster decision-making
- Time-savings
- Cost-savings

However, most of the benefits of BI are considered intangible by organisations. The hope is that a good BI system will lead to a big bang return at some time in the future. It is not possible to state these big bang returns beforehand, since they are serendipitous and infrequent [51].

## 3.4. Costs

Naturally, there are also costs which an organisation has to incur when it wants to put a BI system in place. These include the following [51][76]:

- Hardware costs: depending on what is already in place, one would need to implement a data warehouse specifically for BI. Updates may be required to the existing infrastructure in order to support the BI system.
- Software costs: Next to the costs of the BI software package itself, additional software may be required, such as subscriptions to various data sources.
- Implementation costs: The system needs to be put into place, but also maintenance costs and training costs need to be taken into account.
- Personnel costs: Lastly, one will need employees who will work with the system.

## 3.5. Risks

Although BI has a lot of potential benefits as described above, it is important to also identify the risks. Unfortunately, the literature is very limited on this topic. Although there are a lot of articles that discuss the benefits, the risks are underexposed. There is one risk, however, that does come forward in a lot of articles, for example in the article of Strong et al. [74] and in the article by Gartner [25], which is data quality.

The reason that data quality is a risk is due to the fact that a BI solution is only as accurate and efficient as the data being analysed, so ensuring high data quality is essential. In other words, the quality of your operations and analyses are only as good as the quality of the underlying data on which they are based; 'garbage in, garbage out' [66]. It is very important to realise this when dealing with information generated by BI tools. Relying on inaccurate, incomplete, imprecise, irrelevant and non-coherent information for decision-making is very harmful [2][7]. Moreover, Ballou & Tayi [9] state in their article that *'nothing is more likely to undermine the performance and business value of a data warehouse than inappropriate, misunderstood, or ignored data quality'*.

Therefore, the data quality should be carefully evaluated prior to using BI to support decision-making [73]. Loss of information, insufficient information (ambiguity), meaningless data, and incorrect data have been identified by Wand & Wang [77] as the most observed data problems. It is estimated that more than half of the BI projects fail due to data quality issues and that customer data quality issues cost U.S. businesses over $600 billion dollars a year [41].

Strong et al. [74] describe high-quality data as data that is fit for use by data customers. This means that usefulness and usability are important aspects of quality. Wand & Wang [77] describe in their article that the notion of data quality depends on the actual use of the data. According to them, data quality is relative in that the quality of the data might be sufficient in one case, while in another case it is not. They give the example of the analysis of the financial position of a firm compared to auditing. While in the first situation precision may only be required in the units of thousands of dollars, in the latter case precision to the cent is required.

So one should, before using the output of a BI tool, think about the quality of the data that is used to generate this output. Is the quality of the data source enough for the goal it serves? If not, one should be very careful with the output of the BI tool. By not paying attention to this issue, this may lead to erroneous decision-making which can be very harmful to the organisation.

# 4. Business Intelligence Architectures

In this section, we first describe the architectural movement over time from isolated BI solutions to more and more integrated solutions. After that, we describe the different parts that usually make up a complete BI solution. By doing this, it becomes clear what parts of the BI solution can differ. We then describe what choices can be made for each of those parts and last, we show the different possible combinations and the differences between those combinations.

MicroStrategy [50] describes the development of a BI solution over time by distinguishing between three different 'eras'. Each era describes a different period in time.

*First era: Isolated departmental Islands of BI are an Initial Success*
At first, BI was mostly a departmental concern. Each department that required BI developed or bought a BI solution tailored to their specific requirements. This led to a multitude of different technologies, supporting different user bases and databases. These 'islands of BI' satisfied the individual requirements of the departments. However, the early success with isolated solutions led to an increasing number of applications and this resulted in an increasing number of problems.

*Second era: Overlapping Disparate Islands of BI have become an Enterprise Liability*
Due to the increasing number of applications and the expansion of successful applications, departmental BI solutions were no longer 'islands of BI'. They started overlapping in user bases, data access and analytical coverage. This caused a number of problems: there no longer is a 'single version of the truth', synchronisation efforts between systems have to be made and users may have to use different tools in order to get what they want.

*Third era: Enterprise BI Standardisation delivers a single Version of the Truth with lowest Cost of Ownership*
The third era solves the problems of the previous eras by providing a 'single version of the truth' using one interface to access the data of the entire enterprise. This reduces administrative efforts and reduces time to deploy new BI applications.

## 4.1. The Layers of a BI Solution

Before identifying which choices can be made and what the differences are between the different BI architectures, we need to define a general structure of a BI solution. That way, it is clear of what parts a solution usually consists and what parts can actually differ. Multiple authors try to give a general structure of a BI solution.

We chose for the following structure identified by a number of authors [7][48][63][68][69], which consists of four different layers:

1. Data source(s)
2. Integration services
3. Data repositories
4. Analytical facilities

There are other structures described, for example, Baars & Kemper [8] distinguish a BI framework consisting of three layers:

1. Access layer - responsible for bringing all relevant components and functions from the logic layer together and to present them to the user in an integrated and personalised fashion.
2. Logic layer – focuses on the compilation, processing and distribution of management support data. Two types of systems are distinguished at this layer: systems for data analysis and components for knowledge distribution.

3. Data layer – containing the data for the analysis, often a data warehouse.

Below these three layers, Baars & Kemper show the operational systems that deliver the data to the data layer.

The classification of Baars & Kemper is very similar to the 4-layer structure. We have the opinion that the integration services are also an important part of BI, which integrate the data from the source systems in a data repository. We therefore think that operational systems as well as integration services should be included as separate layers. The data layer of Baars & Kemper is very similar to the data repository layer while the logic layer of Baars & Kemper is very similar to the analytical facilities, though the latter also contains presentation, which is a separate layer in the classification of Baars & Kemper.

Concluding, although there are different structures suggested [8][21], the components are often the same, only the grouping of those components differ. We chose for the 4-layer structure, because that – in our opinion – includes the most important parts of a BI structure in separate layers.

This 4-layer structure of a BI solution is graphically represented in figure 2. In the following sections, we will describe these layers, the most common options for each layer and how these options differ from each other.

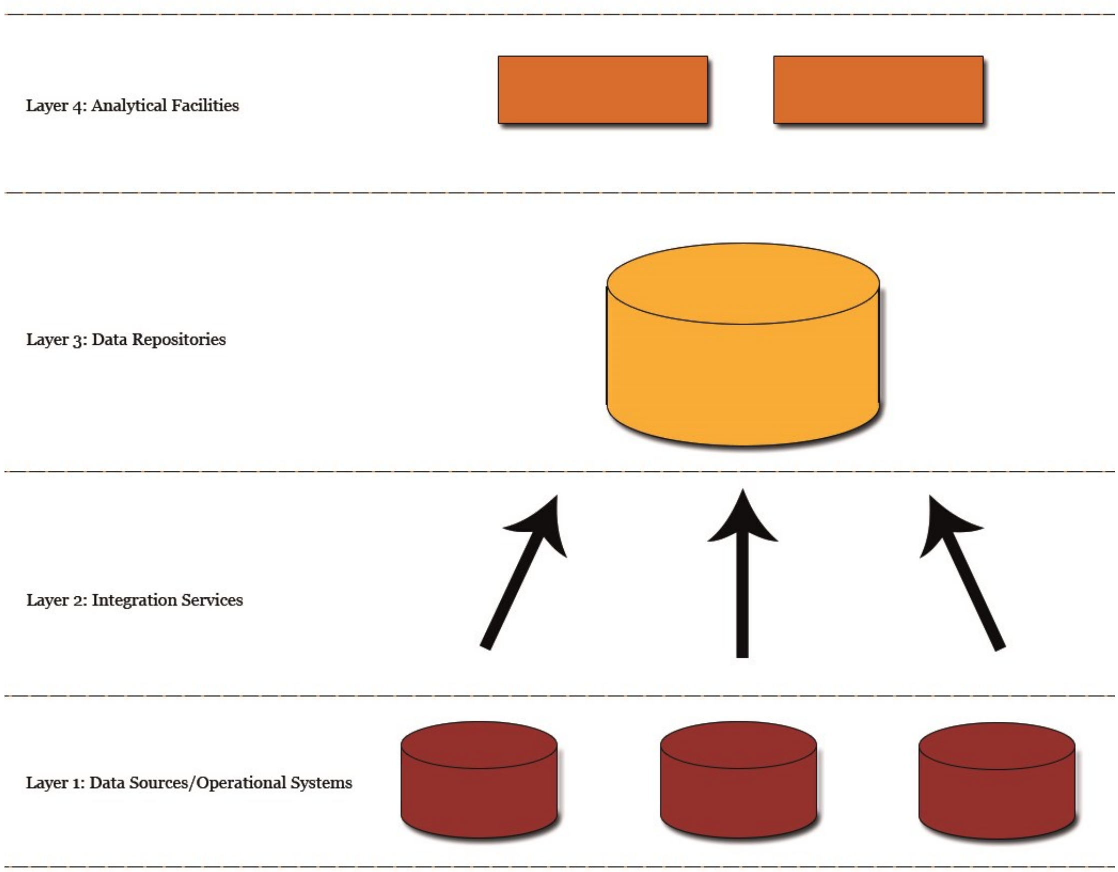



**Figure 2. Layered Structure of a BI Solution**

## 4.2. Data Source

The first layer consists of the systems containing the data that will be used for analysis. Example systems include but are not limited to CRM, ERP, HRM and SCM. In almost every organisation nowadays, there are multiple operational systems. These may be functional or regional specific systems.

The type of the data stored in these source systems may have an impact on the choice of a higher-layer BI option. The data in these systems can be stored in very different ways and can range from structured to unstructured data. Unstructured data is data that does not fit into relational or flat files, unlike structured data. Examples of unstructured data are emails, images, business processes, memos etc. [49][63][82].

Most operational data such as CRM, ERP and SCM systems can be categorised as online transaction processing (OLTP) systems. Transaction processing systems store transactions, such as cash register scans at the supermarket, ATM withdrawals, bank deposits etc. These transaction processing systems are constantly involved in handling updates to what we might call operational databases. These OLTP systems handle a company's routine on-going business [76].

## 4.3. Integration Services

The second layer integrates the data from the data source systems. Gartner [25] said the following about data integration: *'A significant majority of the IT effort expended in a BI project is consumed by data integration issues. Designing a repeatable process by which data is acquired from operational systems, transformed, integrated and delivered to the data warehouse is technically challenging. In addition to issues of data security, ownership and quality, the proper selection of technology for data integration is critical, but not obvious.'*

There are several approaches and techniques available to integrate data [63][82]. We will discuss the most common approaches first, followed by the most common technologies.

One could see approaches as general classifications of technologies. Technologies are actual implementations that one can see within organisations. So each integration approach has one or more technologies that implement that approach.

### 4.3.1. Integration Approaches

**I. Data consolidation**

The consolidation approach captures data from multiple source systems and integrates the data into a single persistent data store. The data in the data store is often not entirely up-to-date, due to the fact that there is some time between the updates on the data in the source system and the updates on the data in the data store. Depending on the implementation this delay can range from a few seconds to a multiple days.

Data consolidation with zero latency; also known as real-time data, is possible, but very expensive. Next to that, only very few situations require real-time information [63][82]. One can argue what is meant with real-time data [7][80]:

- Zero latency
- Access to information when it is required
- Ability to derive key performance measures that relate to the situation at the current point in time and not just to some historic situation.

This is also the reason why people sometimes refer to real-time data as right-time data; in order to be more clear. In this thesis, we speak of real-time data when we mean data with zero-latency and right-time data when we mean data that is available at the right time, unless stated otherwise.

**II. Data federation**

The data federation approach creates a virtual view of the data, in the form the requestor defined. It eliminates the need for an additional data store. It works on an on-demand-basis, which means that it will only pull data away from the source systems when a query is performed [63][82].

Data federation can be used instead of data consolidation when the costs of consolidating the data would outweigh the benefits achieved. Data federation can also be used as a short-term solution after a merger or acquisition [63][82].

Next to that, it can also be a good solution when the data is not allowed to be duplicated from the source systems due to license restrictions and/or security policies. This is often the case when dealing with syndicated data [63][82].

There are a number of drawbacks to the federated approach. First, it may have considerable overhead and an impact on performance since it has to access multiple databases during runtime. In addition, data federation is not well suited for retrieving and reconciling large amounts of data or data of a low quality [63][82].

**III. Data propagation**

Simply said, data propagation is copying the data from the source system to a data repository. After this is done for the data already in the source systems, updates can happen in a synchronous or asynchronous manner. When updates are synchronous, a single transaction will update both the source and target data, which is not the case for asynchronous updates. Nevertheless, both synchronisation types guarantee the delivery of the update to the target system. Data propagation is in most cases transaction- or message-based and has trouble dealing with the transfer of large amounts of data from point A to point B. It is, however, suitable to provide real-time data support [63][82].

**IV. Data access**

The data access approach is more or less a search engine that searches the different data sources. One could compare this approach with search engines that search the internet. The application of search technology in an enterprise is known as Enterprise Information Access (EIA) [63][82].

**V. Data virtualisation**

Similar to the data federation approach described earlier, the data virtualisation approach does not duplicate the data of the source systems. It can communicate with numerous data base management systems (DBMS) and allows the user to access multiple data sources without them knowing exactly where the data resides [63][82].

**VI. Data mash ups**

As with the data federation and data virtualisation approaches, the data mash up approach leaves data at its current location. Based on service-oriented architecture (SOA) principles, it combines data virtually, often via web services [63][82].

**VII. Hybrid approach**

It is not unusual for organisations to have a combination of data integration approaches within their business. This may be a good choice when an organisation has specific requirements that cannot be satisfied in a proper way by using one approach alone [63][82]. Nevertheless, using multiple integration services often makes things more complicated, so unless there are significant advantages to do so, one should aim to use a single integration service.

### *4.3.2. Integration Technologies*

The various approaches that we discussed in the previous sub section are ways to integrate the data from different source systems. However, in order to do this, a specific technology should be chosen. We will discuss eight common technologies below [63][82].

Integration technologies that perform operations (e.g. cleansing, transformation) on the data from the source systems often temporarily store the data in an operational data store (ODS). After extracting the data from the source systems, data is moved to the ODS for additional operations, after which it is passed in a uniform format to the data warehouse [76].

**I. Enterprise Application Integration (EAI)**

EAI is a technology that supports data propagation. Next to that, it allows applications to access data transparently without knowing its location or format. In most cases, it works on a 'push' basis. This means that at the moment a transaction or update occurs, an 'EAI listener' notices the change and 'pushes' this information - via a bus or centralised queue - to the other applications that need to be notified. The major drawback of EAI is that it does not support communication to 'non-applications' such as legacy systems, data warehouses, Excel spreadsheets, unstructured data etc. Some vendors do, however, have custom 'readers' to solve this issue [20][63][82].

As an integration service, EAI can support communication between applications, or it can provide real-time input for another data integration application. EAI is message- or transaction-centric and was designed to synchronously or asynchronously propagate small amounts of data between applications, although in almost every case it occurs within the scope of one transaction [17][63][76][82].

### II. Extract, Transform, Load (ETL)

As opposed to EAI which performs best when dealing with small amounts of data that are message- or transaction-centric, ETL performs best when dealing with scheduled, repetitive tasks that involve very large amounts of data. ETL is sometimes also referred to as ELT (extract, load and then transform) or ETLT (extract, transform, load, transform) [63][82].

ETL supports data consolidation, is a 'pull' based technique and excels at moving large amounts of data from point A to point B. A 'pull' based technique means that it requests data from the source systems at scheduled intervals. So unlike a 'push' based technique, it does not automatically send and receive notifications when data is updated. Typically, an ETL tool provides some standard transformation functionality, which can be further extended by users [63][76][82].

The big drawback to ETL is that it does not work well with transaction-based messages and therefore performs poorly when real-time data integration is required. An additional problem is that, due to the fact that ETL was designed to support warehouse loading and not specifically for data transformation, warehouse data can be transformed only during physical load, not while querying. This prevents users from having different logical views of the data. To illustrate this, imagine that sales data in a data warehouse was converted into dollars during loading the data warehouse. It is then impossible for users to directly query the sales data in euros. This would require an additional step [35][63][82].

### III. Right-time Extract, Transform Load (RT-ETL)

RT-ETL is an upgraded version of ETL. It is an upgrade of the ETL data consolidation technology in such a way that it supports event-driven data propagation. This allows for near real-time data. Varying per product, different benefits of data propagation can be achieved, such as guaranteed data delivery, two-way data propagation, load balancing, back-up and recovery and disaster recovery. The drawback is that RT-ETL is a lot more expensive than ETL [63][82].

### IV. Enterprise Information Integration (EII)

EII is a technology that supports data federation. As discussed in the previous sub section, the advantage of data federation is that it does not require an additional data store. EII provides access to the different data sources on a pull basis and any transformation that is required is done at the moment the data is retrieved from the source systems [31][32][63][76][82].

EII works as follows: it contains a metadata layer, with consolidated business definitions, which define how and where to get the data. When it receives a query, it looks where the data is located and, if necessary, splits the query into separate parts for the different source systems. The results of these separate queries are then merged using the metadata layer containing integration rules. At this point in time, EII is not suited to move large amounts of data from point A to point B [63][82].

### V. Enterprise Data Replication (EDR)

EDR is a technique that makes it possible to copy data from one system to another. This can be done for the entire data set, but also, for example, by only copying the data that has changed since the last time. This technology is also very suitable to store data conveniently, make back-ups and quick access to that data in case of disaster. EDR products are usually very suitable to transfer large amounts of data [63][82].

EDR can be used to replicate data in real-time, sporadically, or at scheduled intervals. EDR does not alter the copied data in any way, but simply copies it [62].

### VI. Enterprise Content Management (ECM)

ECM is the management of information in all its forms across an organisation. ECM extents traditional records and content management and deals with content storage, retrieval, delivery and management. It manages both structured and unstructured data and prepares it for meaningful use by filtering, routing and creating search pathways, corporate taxonomies and semantic networks [14][45][63][82].

### VII. Service-oriented Architecture (SOA)

Although the term SOA is relatively new, its fundamentals are based on a very old and established school of thought, namely separating things into independent and logical units[10][34][75].

In a SOA architecture, processes, applications, data activities and operations are made available as services. These can then interact with each other using a standards-based language. It allows for easy integration between different data sources. In addition, it allows for easy reuse of existing functionalities. The goal is to manage complexity and achieve architectural resilience and robustness through ideas such as loose coupling, location transparency and protocol independence [3][22][24][30][34][57][75][82].

### VIII. Enterprise Service Bus (ESB)

While performing our literature study, we found the paper of Menge [49] who, in our opinion, describes very well and comprehensively what exactly an ESB is:

*"An Enterprise Service Bus is an open standard, message-based, distributed integration infrastructure that provides routing, invocation and mediation services to facilitate the interactions of disparate distributed applications and services in a secure and reliable manner."*

Very much in line with this definition, Schmidt et al. [67] describe the essential characteristics of an ESB as follows:

*"the meta-data that describes service requestors and providers, mediations and their operations on the information that flows between requestors and providers, and the discovery, routing, and matchmaking that realise a dynamic and autonomic SOA."*

An ESB provides a shared messaging layer for applications and services that are connected to it. It also supports SOA by making sure that the different services can communicate with each other in a reliable way. Complex data transformation, however, is not provided by most vendors. This means that custom data transformation should be developed and maintained. ESBs are optimised for XML-oriented data transformation and rely on XSLT-based tooling for defining transformations [63][82].

There is a major difference between ESBs and earlier approaches to EAI. ESB is not monolithic and centralised. Instead, it is distributed throughout the organisation to provide enterprise-wide flexibility. Each department deals with its own part of the bus. Overall interoperability is achieved while still maintaining local responsibility and resource control. This causes an ESB to be much cheaper to build and maintain than traditional EAI models [63][82].

## 4.4. Data Repositories

The third layer consists of the data repositories which store the data that is integrated in the second layer. Some integration services, however, do not require a physical data repository. Instead, these are used in combination with a federated architecture, which will be discussed at the end of this section where we introduce the five different data warehouse architectures. Before we do that, we first give some more information about a virtual data warehouse, the relational data model and the multidimensional data model and we introduce online analytical processing (OLAP) systems and the different types of OLAP.

A setup in which no physical data warehouse is required is often called an in-memory system. Such a setup mainly relies on main memory for computer data storage. This is contrasted with systems that employ a disk storage mechanism. In-memory systems have much faster reaction speeds, because they do not have disk input/output to slow them down. Modern physical systems make extensive use of caching to store frequently accessed data in main memory but for queries that process large amounts of data, disk reads are still required. In-memory systems are sometimes also used in combination with ETL tools so that complex transformations and joins can be executed in main memory without the need to create temporary tables [26][46][76].

The very design that makes an OLTP system (most of the source systems) efficient for transaction processing makes it inefficient for end-user ad-hoc reports, queries, and analysis. For that purpose, OLAP systems are used, which contain all data of an OLTP system, including historical data, but reorganised and structured in such a way that it is fast and efficient for querying, analysis and decision support [76]. If we were asked to assign a layer to the OLAP system, we would say it is a combination of the third and fourth layer, which are the data repositories layer and the analytical facilities layer. Using the second layer, the integration services layer, the data of the source systems is transferred to the data repositories and used in multidimensional analysis.

One can distinguish between relational models and multidimensional models for a data repository. In a relational data model, all data is orderly stored in proper tables, and each table is typically joined with at least one other. These relationships can be visualised using an entity relationship diagram (ERD). A simple example is shown in figure 3. However, this example is nothing you would see behind the scenes in most large enterprises. It is common for enterprise systems to contain thousands of individual tables. Although it is possible to query as many data sources and tables as you like (assuming they are properly joined), queries involving a high number of huge tables take very long to complete [70].

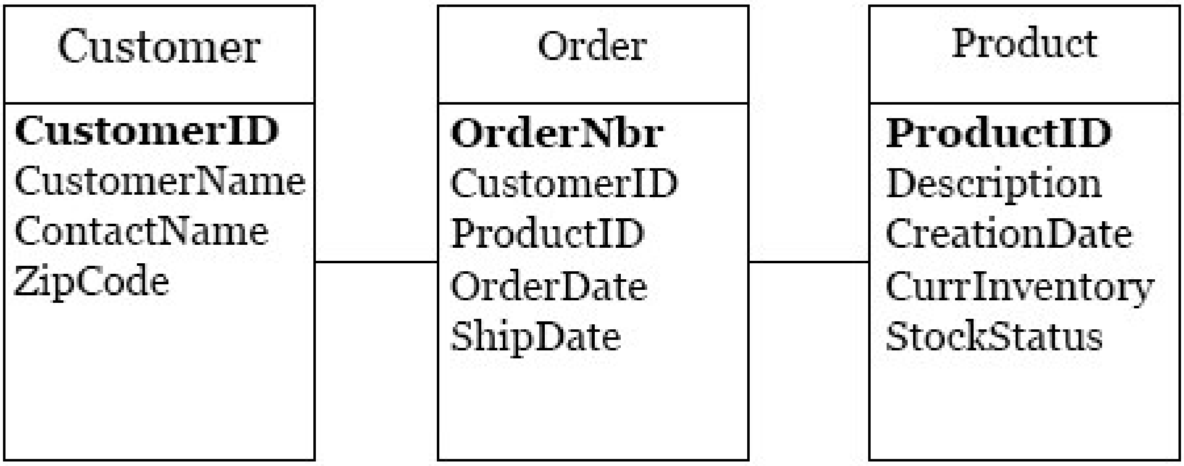



**Figure 3. Example ERD**

The multidimensional data model is much better suited for the task of analysis. It can be physically realised in two ways, one is by using trusted relational databases in a star/snowflake schema, and the other is by using a specialised multidimensional database [2].

In the multidimensional data model, data is modelled as a fact with multiple dimensions in the form of data cubes. An example of the multidimensional model would be the fact sales, with the dimensions book, city and the date of the purchase. An example fact is the number of e-Business books sold in Amsterdam on 7 May, 2014. An example cube is shown in figure 4.

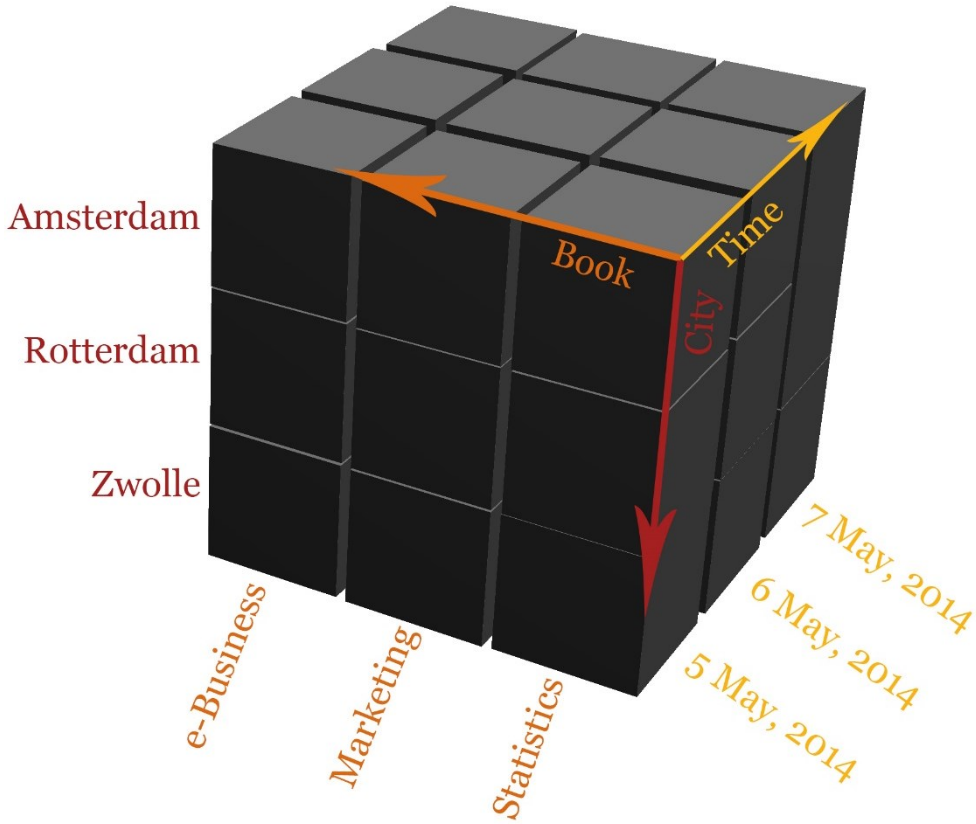



**Figure 4. Example of a Cube**

The multidimensional data model requires fewer disk space and is a lot quicker for OLAP applications compared to the relational data model which is often used in OLTP applications [12][13][18][58].

The differences between relational data modelling and multidimensional data modelling are listed in table 1.

One can categorise OLAP systems in the following way [1]:

- Multidimensional Online Analytical Processing (MOLAP)
  - Cube is precomputed and stored in proprietary format
  - Fast query response times, data duplication, size restrictions, investment in extra technology
- Relational Online Analytical Processing (ROLAP)
  - Use relational tables to store cube's aggregations
  - Slower, no duplication, scalable
- Hybrid Online Analytical Processing (HOLAP)
  - Only aggregation data is stored in MOLAP cubes; facts are stored in relational table.

*Table 1. Relational Data Modelling vs. Multidimensional Data Modelling*

| Relational Data Modelling | Multidimensional Data Modelling |
|---|---|
| Data is stored in relational database management systems (RDBMS). | Data is stored in RDBMS or Multidimensional databases. |
| Tables are units of storage. | Cubes are units of storage. |
| Data is normalised and used for OLTP. Optimised for OLTP processing. | Data is de-normalised and used in data warehouse and data mart. Optimised for OLAP. |
| Several tables and chains of relationships among them. | Few tables and fact tables are connected to dimensional tables. |
| Volatile(several updates) and time variant. | Non-volatile and time invariant. |
| SQL is used to manipulate data. | MDX is used to manipulate data. |
| Detailed level of transactional data. | Summary of bulky transactional data (Aggregates and Measures) used in business decisions. |
| Normal Reports. | User friendly, interactive, drag and drop multidimensional OLAP Reports. |
| Typical data design used for business transaction systems. | Data design used for analysis systems. |
| Goal – reduce every piece of information to its simplest form – a debit transaction, a customer record, an address. | Goal – break up information into 'Facts' – things a company measures and 'Dimensions' - how we measure them: by time, region, or customer. |
| Suited for concurrent handling of many small transactions by many users. Only a limited amount of data history is normally kept. | Suited for reading or analysing large amounts of data by a modest numbers of users. Many years of data history may be kept. |
| User is usually constrained by an application that understands the data design. Users are typically operations staff. | This simpler data design makes it easier for users to analyse data in any way they choose. Users are typically analysts, company strategists, or even executives. |

***Source: Boyina [12]***

We will now introduce the different data warehouse architectures that are identified in the literature. Authors distinguish five architectural styles for data warehouses. There are also some other architectural styles discussed, but those seem to be variations of the original five [4][6][78].

The architectural choice of a data warehouse (DW) is a key factor in determining the abilities and limitations of that data warehouse. Therefore, it is important to understand the differences between these architectures [17].

Ariyachandra & Watson [6] distributed a questionnaire in order to see which of the five architectures was most common. We present their findings below. The percentage represents the percentage of companies that had a particular architecture in place:

1. Hub-and-spoke architecture (39%)
2. Data mart (DM) bus architecture with linked dimensional data marts (26%)
3. Centralised data warehouse architecture (17%)
4. Independent data marts architecture (12%)
5. Federated architecture (4%)

The data repository layer is typically a distinct data storage system that will be used by the fourth and last layer, namely the analytical facilities. Below, we describe in short, the five data warehouse architectural styles.

## 4.4.1. Independent DMs Architecture

Independent data marts are often the result of historical organisational efforts to create decision-support systems. These independent data marts may be very well suited to local requirements. One may see independent data marts in functional silos or in a specific region [4]. Independent data marts are arguably the most simple and least costly data warehouse architecture alternative [76].

The problem with this architecture is that this architecture is not suited to enterprise-wide requirements. There is no 'single version of the truth' and the data is often stored in a different way compared to other systems, using different data dimensions, measures and data definitions. Analysing data across multiple independent data marts is therefore very difficult [4][5][11][76].

Every stand-alone data mart is developed independently from other systems, which means that if an organisation has multiple independent data marts, each data mart also has its own integration service [37]. Figure 5 graphically represents this architecture.

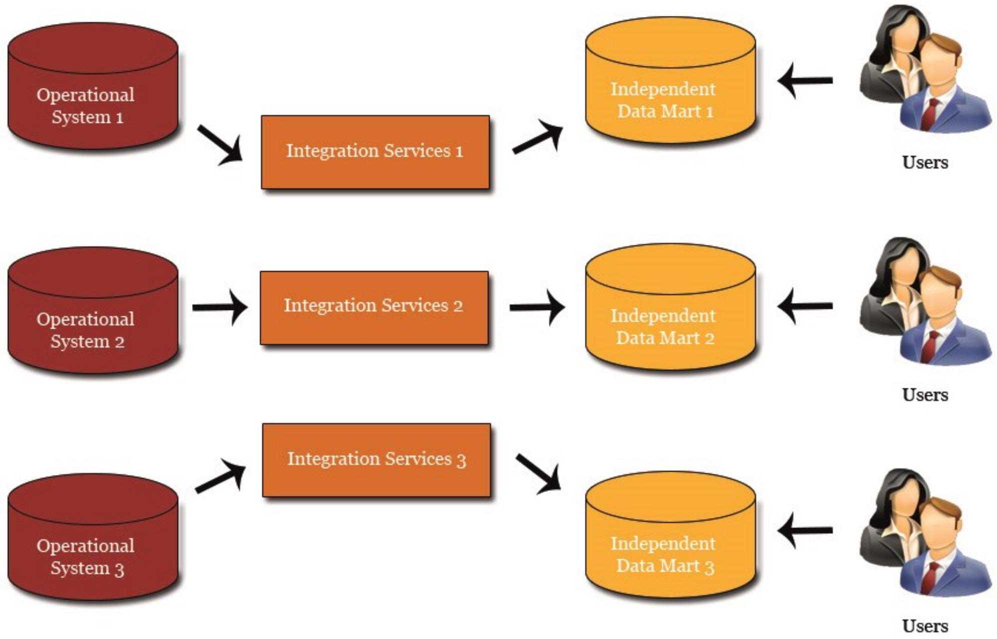



**Figure 5. Independent Data Mart Architecture**

This architecture is not formally advocated in the industry and almost the entire data warehouse community agrees that this architecture is inferior to the others. The major shortcomings are thus the inability to perform enterprise-wide analysis and the unnecessary repetition of integration service efforts [4][11][37].

Still, a significant number of projects turn out to be a collection of independent data marts, often due to a lack of enterprise-wide focus at the start of the project. For example, departments may start data mart projects to fit their own analytical requirements, not incorporating the requirements of other departments [37][71].

Concluding, the independent data marts architecture is in almost all cases not deliberately chosen, but the result of historical organisational efforts or the result of isolated activities of specific departments with a lack of enterprise-wide focus [37].

### *4.4.2. DM Bus Architecture with linked Dimensional DMs*

Together with the hub-and-spoke architecture which will be discussed in the next sub section, this architecture is the most common approach. There are two 'gurus' in the data warehouse community, Kimball and Inmon. Kimball is a strong advocate of the data marts bus architecture with linked dimensional data marts, while Inmon is a strong advocate of the hub-and-spoke architecture. When well executed, both approaches lead to an integrated enterprise-wide data warehouse. Compared to the Inmon methodology, Kimball's methodology is bottom-up [38][39].

Weir [81] describes the different approaches in the following way:

*"Inmon prescribed a data driven approach, which is predominantly a technical exercise of integrating data, subject area by subject area, from disparate operational systems with little input from the business. [...] Kimball, on the other hand, has a reputation for having a development approach that is closer to the business.*

*This is due to the fact that business users feature more strongly via the requirements driven methodology, which allows users to articulate their information needs to support the business process that they are aligned to."*

The approach to developing this type of solution is as follows. A first data mart is created which supports a particular process with particular dimensions and measures. Additional data warehouses then use the same dimensions (structured as a star schema), which leads to an integrated whole and an enterprise-wide view of the data. So the data warehouse is 'conceptual', connected by the 'bus' of the conformed dimensions [4][5][17][37][43][47][71][78].

This architecture is shown in figure 6. It is similar to the independent data mart architecture; the difference is that these data marts are linked by conformed dimensions, unlike the independent data mart architecture.

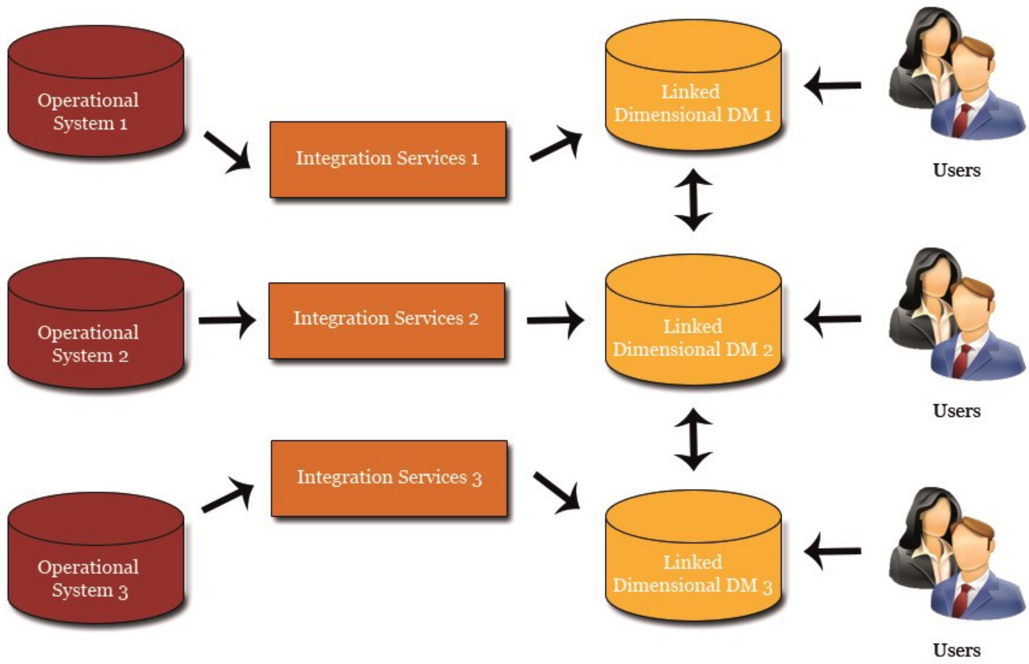



**Figure 6. Data Mart Bus Architecture with linked Dimensional Data Marts**

Due to the fact that the data marts are individually linked to each other, there is a better chance of maintaining data consistency and timeliness (at least at meta-data level). Although this architecture does allow for complex data queries over multiple data marts, the performance of this kind of analysis may be unsatisfactory [76].

### *4.4.3. Hub-and-Spoke Architecture*

The hub-and-spoke architecture is currently perhaps the most famous data warehouse architecture. A strong advocate of this architecture is Inmon who called it the Corporate Information Factory (CIF). Compared to the Kimball methodology described in the previous section, Inmon's methodology is considered top-down [38][39][68].

This architecture exists of multiple dependent data marts (spokes) and a centralised data warehouse or hub. The centralised hub accepts requests from multiple applications that are connected through the spokes. Users use the spokes to get the information they need. This architecture allows for easy customisation of interfaces and reports, tailored to the requirements of the specific user [38][39]. See figure 7 for a graphical representation.

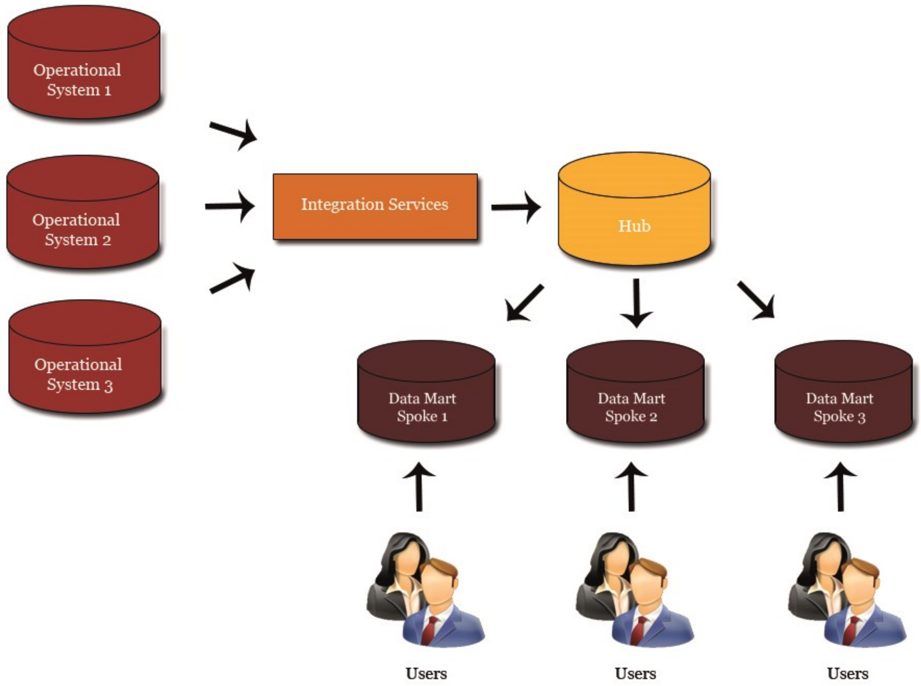



**Figure 7. Hub-and-Spoke Architecture**

According to Inmon, the higher the levels of summarisation, the more usage of the data. He sees the central data warehouse as the database that contains the detailed information. This information is then used and may be normalised, denormalised, or summarised by the dependent data marts, which are in turn accessed/queried by the users [4][38][39][78].

Attention is focused on creating an enterprise-wide view of the data which is scalable and maintainable. This hub-and-spoke architecture is often developed in an iterative way, subject area by subject area. Dependent data marts may be created for specific functional areas, regions or special purposes [4][76][78].

### *4.4.4. Centralised DW Architecture*

The centralised data warehouse architecture is very similar to the hub-and-spoke architecture, except for the dependent data marts. All users use the central data warehouse and are no longer limited to the smaller data marts. The central data warehouse contains atomic level data, some summarised data and logical dimensional views of the data. Data can be accessed from a relational view as well as a dimensional view [76][78].

The advantage of this architecture over the hub-and-spoke architecture is that the amount of data that has to be transferred is much lower, since there are no dependent data marts that require updates. The architecture provides a holistic and timely view of the enterprise, independent of the time and place of that person in the organisation. The disadvantage is that customisation of interfaces and reports is harder to do compared to the hub-and-spoke architecture [76][78]. Figure 8 shows the centralised data warehouse architecture.

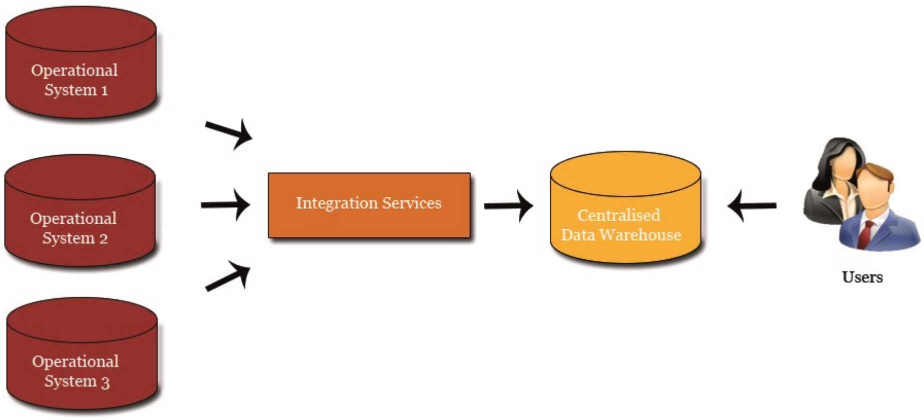



**Figure 8. Centralised Data Warehouse Architecture**

### 4.4.5. *Federated Architecture*

There are different notions of what a federated architecture exactly looks like. Several authors [19][63][78][82] describe their view of a federated architecture. In this sub section we will describe the most common views. We differentiate between virtual/logical and physical/materialised integration, tightly and loosely coupled systems and then we will discuss regional and functional federated data warehouses.

The federated architecture does not re-develop existing data structures (operational systems, data marts and data warehouses). The data is either virtually/logically or physically/materialised integrated, for example by using shared keys, global metadata or distributed queries. This architecture is advocated as a practical solution for firms that have a complex decision support environment in place and do not want to rebuild [71][78]. This architecture is also advocated when organisations want to retain independence, but still want some sort of collaboration [19].

The virtual/logical approach of data integration can be implemented in two different ways. The most naïve way to do so is by pair-wise combining the systems together. However, when the number of components is large, schema translations become very tedious. This is due to the fact that with $n$ components, $n * (n - 1)$ connections are required [19]. This is depicted in figure 9.

Due to the many connections this architecture is sometimes referred to as 'Spaghetti' architecture. An architecture that has a central hub and thus requires fewer connections (only one per application which connects it to the central hub) is structured much clearer and is referred to as 'Lasagne' architecture.

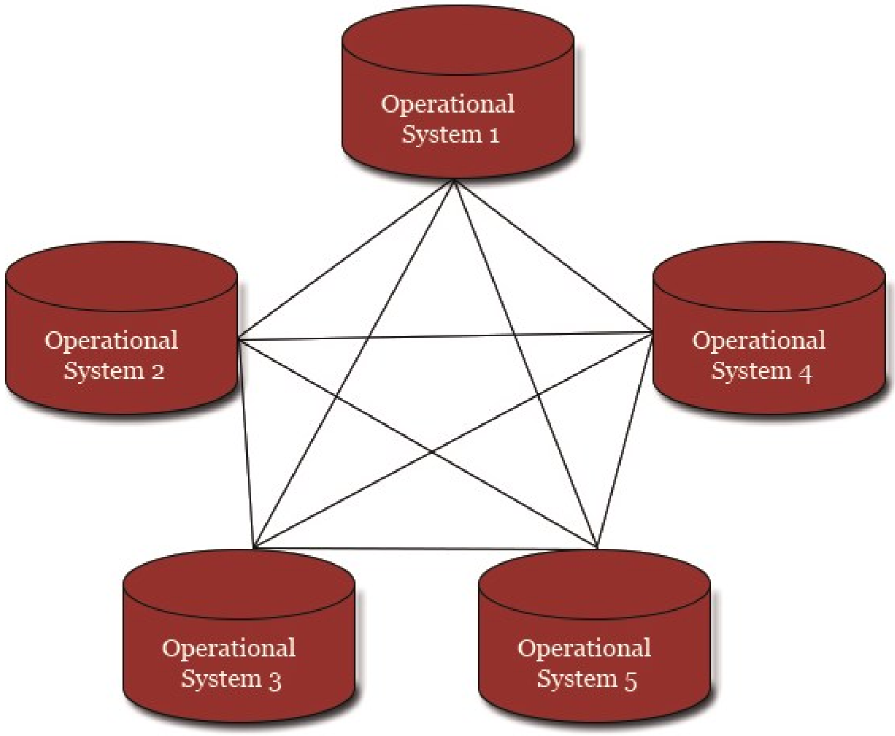



**Figure 9. Naïve Federated Architecture**

The second way to virtually/logically integrate the data is by using a mediation system. This mediation system provides a virtual overview of the data in the different sources and users who query the mediation system do not have to know where the data resides or in what way it is stored. The mediation system provides one global system to the users who structure their queries corresponding to that form. Slightly simplified, the following actions are taken by the mediation system: 1) the mediation system receives a query from the user, 2) the query is split into separate parts for the different source systems, 3) it translates the partial query into the schema of the source system, 4) it receives the result which is translated back into the unified form (these translations are often done by wrappers), 5) this is then combined with the other parts of the query (that were processed the same way) and eventually 6) the complete result is presented to the user [19]. This is graphically represented in figure 10.

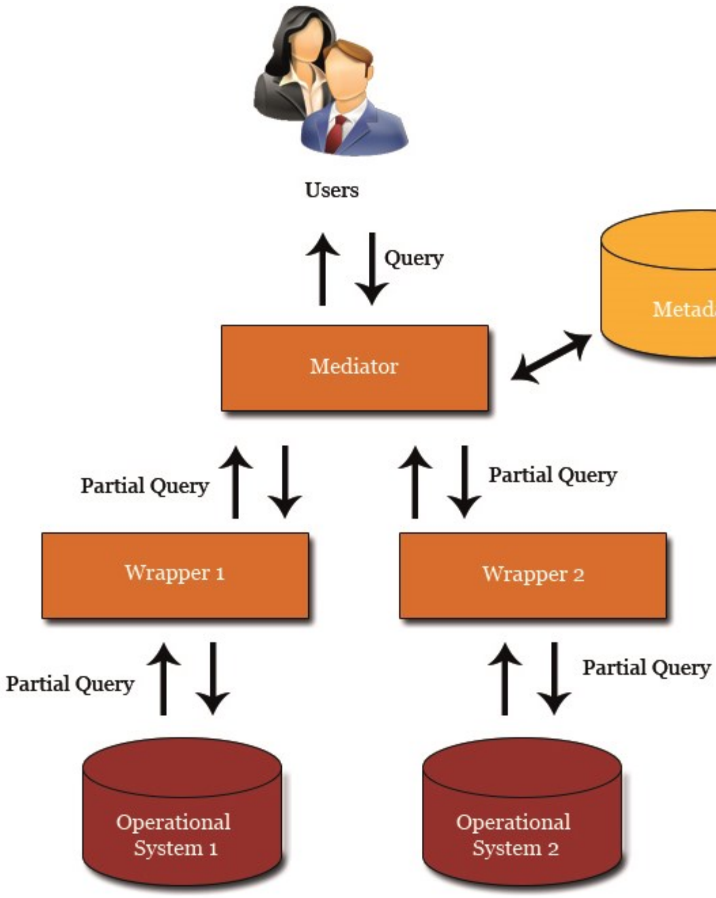



**Figure 10. Mediated Federated Architecture**

Nowadays, mediation is the most common technique of the two virtual approaches. The reason for this is that in the case of a naïve federated architecture, a large number of interfaces have to be written for each source in order to communicate with the other systems [19]. Not only does this require a lot of work, it is also very hard to change one system, since all connections to other systems are affected. These problems do not occur in a mediated federated architecture.

One can differentiate between loosely and tightly coupled federated database systems (FDBSs). Contrary to the loosely coupled FDBS, a tightly coupled FDBS has a unified schema. Changes are often difficult to make because schema integration techniques do not allow easy adding or removing of components [19].

The physical/materialised approach is very similar to the virtual/logical approach. The difference is that in the physical/materialised approach, the filtered data is (temporarily) stored in a data warehouse [19]. The virtual/logical approach is preferred over the physical/materialised approach when the number of different source systems is very large, updates are frequent, or when the kind of queries that users will ask are very unpredictable [19].

Another differentiation that can be made are the regional and functional federations which are variations of the physical/materialised approach. In regional federations, detailed data is in most cases stored in the regional systems, while more summarised data is stored in the global data warehouse. From the regional warehouses fact data is moved to the global data warehouse. This movement of data from regional to global warehouses is called ‘upward federation’.

The movement of data from the global to the local data warehouse is called 'downward federation'. The type of data moving from the global to local data warehouse is reference data, transactional data from corporate systems and summary data [42]. Figure 11 shows the regional federated data warehouse architecture.

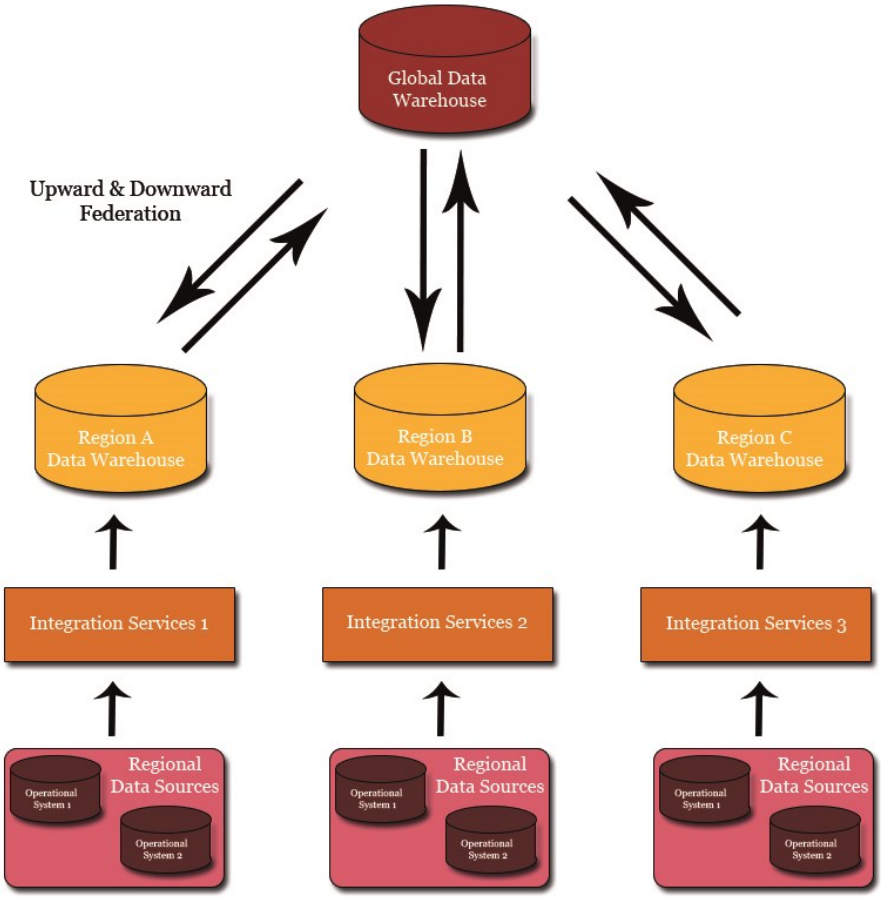



**Figure 11. Regional Federated Architecture**

In functional federations, data warehouses exist for the different functions within an organisation. The federated data warehouse integrates the different functional areas and provides an enterprise-wide view [42]. Figure 12 shows the functional federated data warehouse architecture.

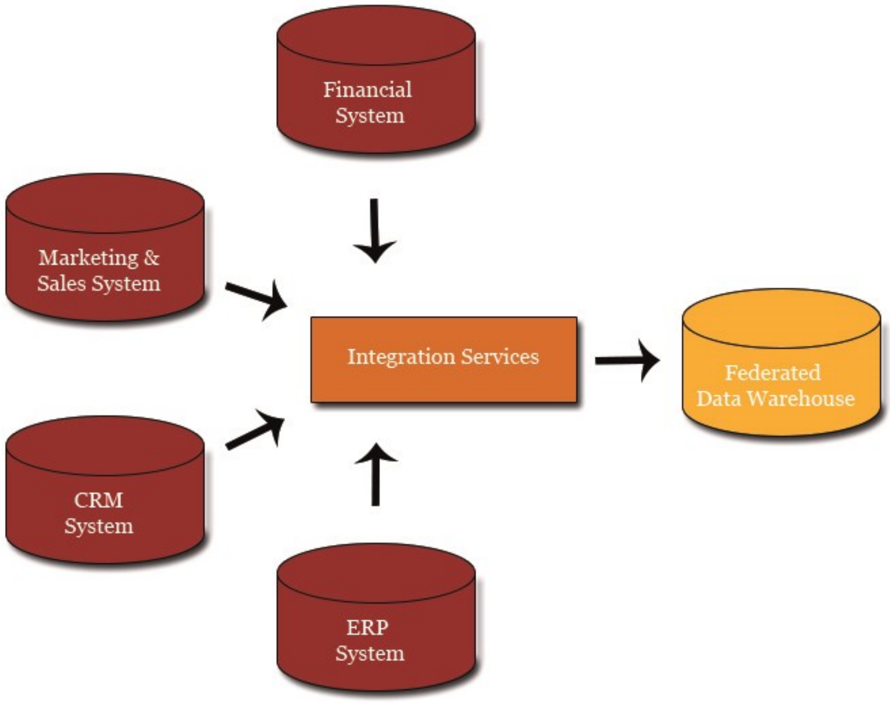



**Figure 12. Functional Federated Data Warehouse Architecture**

What we understand after performing the literature study on the federated architecture is that a lot of different views exist, but all sources we looked at agree that the federated approach does not duplicate all data in a data warehouse. Instead, it integrates heterogeneous data sources in a different way, be it virtually or physically. Often, this is on a query basis, where the query is split in different parts (for the different source systems) and then, after retrieving the data, the results are combined and given back to the user (virtual) and may be stored for future use (physical). The most common view of what people perceive to be a federated architecture seems to be the mediated federated architecture.

## *4.5. Analytical Facilities*

The fourth layer is the final step. Depending on the tool, numerous analytical actions may be performed on the data, in order to e.g. create reports and support decision-making. Often, people refer to this layer when talking about a particular BI tool or solution. However, all layers are required in a BI solution.

The architecture of BI may be influenced indirectly by the analytical facility that is used. MicroStrategy Corp. distinguishes five styles of BI [50][76]:

1. Data Mining and Advanced Analysis — Fully investigate queries with set analysis, statistical and trend analysis, and data mining.
2. Visual and OLAP Analysis — Slice-and-dice analysis with visualisations, drilling, pivoting, and other investigative features.
3. Enterprise Reporting — Print-perfect operational and business reports with interactive content.
4. Dashboards and Scorecards — Highly graphical reports designed to monitor corporate performance.
5. Mobile Apps and Alerting — Business apps on mobile devices and the scanning of data for exception reporting.

Roughly speaking, every analytical facility or BI tool can be grouped into one or more of these five application areas or BI styles.

There are a lot of BI tools on the market. Each year Gartner [27] develops a 'Magic Quadrant' which is a matrix that maps the different vendors on the axes 'completeness of vision' and 'ability to execute'. This is performed for the main software vendors that – according to them – should be considered by organisations seeking to use BI and analytical platforms. The 'Magic Quadrant' for 2014 is shown in figure 13.


**Figure 13. Gartner's Magic Quadrant for BI and Analytical Platforms**

**Source: Gartner [27]**

We decided not to go in depth on each of the tools. There are a lot of tools on the market and it would be the topic of a research itself to analyse each of these tools. Besides, Gartner [27] already discusses in short, the tools adopted in the magic quadrant.

## 4.6. Combinations

In this section, the different possible combinations of the options of the four layers (described in the previous sections) are shown. These combinations form the BI architectures that we use in the remainder of this thesis. First, we need to make some important remarks concerning the combinations that we will adopt:

- We do not incorporate the first layer, consisting of the data source systems/operational systems, in our combinations. The reason for this is that, usually, the source systems are already in place when an organisation would consider a BI solution. This means that we cannot influence this layer. In this thesis, we assume that every source system is able to cooperate with every integration technology. We do realise that there may be (legacy) source systems for which this is not the case. However, there are so many source systems and to discuss every combination of source system and integration technology is simply not possible.

- We do not incorporate the fourth layer consisting of the analytical facilities. Although the choice for a particular style of BI and BI provider may influence the choice in other layers, it does not directly affect the architecture itself. Implicitly, the influence that the choice of a particular BI tool may have on the second (integration services) and third layer (data repositories) is covered by the requirements, so we therefore only look at those two layers.

- The combinations that we adopted are those that are realistic/likely. That is, some integration technology-data repository combinations may simply be very illogical. For example, when a federated data warehouse architecture is used (a virtual database), it is very strange – if not impossible – to use an integration service that provides physical integration such as ETL or EDR.

  Another example: EAI is designed to propagate small amounts of data. So although it might be possible to use this technology to copy large batches of data from one location to the other, there are other technologies much more suitable for this task (e.g. ETL, EDR). In such a case, we leave EAI out of our overview, since it does not make sense to select that technology in that situation.

- Further, we did not include the hybrid form since that can exist of many combinations of other technologies. This means that we cannot say much about this form, since many variations with different characteristics exist.

- We also did not include the ECM integration technology. The reason is that this technology is mainly used for managing the data within an enterprise and not so much as an integration technology. Although it could be used as one, it is not its main strength; better alternatives are available for the purpose of integration. Therefore, unless an organisation has different goals besides data integration, one should not choose for ECM. If an organisation does have different goals besides data integration, ECM could be a good choice, but this is out of the scope of our research.

- Although both SOA and ESB can be implemented without the other, this hardly ever happens. Therefore, we decided to see these two technologies as one integration service.

- EAI and ESB are just as SOA and ESB very closely related. There are two architectures to realise EAI, namely a bus architecture (of which the ESB is a variation) and a hub-and-spoke architecture. Goel [28] describes in a whitepaper the differences between these two architectures. The overview of the differences identified by Goel is shown in Appendix C. For clarity, we decided to only use EAI instead of EAI, SOA and ESB in our combinations. In case EAI is chosen, we refer to Appendix C to decide which EAI architecture is best (e.g. ESB with SOA or hub-and-spoke without SOA).

  Key difference between the hub-and-spoke topology and the bus topology is that for the bus architecture, the integration engine that performs message transformation and routing is distributed in the application adapters and requires the application adapter to run on the same platform as the original applications. In the hub-and-spoke architecture the integration engine is a central ‘hub’ which will also acts as broker for the ‘data repository layer’ [28].

  Very important to realise here is that these architectures are ways to structure the EAI integration service and are unrelated to the data warehouse architectures with the same names.

We first relate the different integration approaches to the different integration technologies. The comparison of different integration technologies by Sahi et al. [63] was really useful for this purpose. We have added his comparison table in Appendix D. The mapping we came up with is shown in table 2.

*Table 2. Integration Approach & Integration Technologies Mapping*

| Integration Approach | Integration Technologies |
|---|---|
| Data Consolidation | ETL, RT-ETL, EDR |
| Data Federation | EAI, EII |
| Data Propagation | EAI, RT-ETL, EDR |
| Data Access | EAI, EII |
| Data Virtualisation | EAI, EII |
| Data Mash Ups | EAI |

After that we matched the different data warehouse architectures to these integration technologies and approaches. We did this by using the knowledge we gained from the performed literature study. The different combinations are shown in table 3. These combinations form the BI architectures that will be used in the remainder of this thesis.

*Table 3. Data Warehouse Architecture & Integration Services Mapping*

| Data Warehouse Architecture | Integration Approach | Integration Technology |
|---|---|---|
| Independent Data Marts Architecture | Data Consolidation<br>Data Propagation<br>Data Propagation & Consolidation | ETL, EDR<br>EAI, EDR<br>RT-ETL |
| Data Mart Bus Architecture with Linked Dimensional Data Marts | Data Consolidation<br>Data Propagation<br>Data Propagation & Consolidation | ETL, EDR<br>EAI, EDR<br>RT-ETL |
| Hub-and-Spoke Architecture | Data Consolidation<br>Data Propagation<br>Data Propagation & Consolidation | ETL, EDR<br>EAI, EDR<br>RT-ETL |
| Centralised Data Warehouse Architecture | Data Consolidation<br>Data Propagation<br>Data Propagation & Consolidation | ETL, EDR<br>EAI, EDR<br>RT-ETL |
| Federated Data Warehouse Architecture | Data Federation<br>Data Access<br>Data Virtualisation<br>Data Mash Ups | EAI, EII<br>EAI, EII<br>EAI, EII<br>EAI |

In order to get a good overview of what the differences are between the BI architectures, we developed a framework. In this framework, we mapped characteristics to the BI architectures. The characteristics that we used show the differences between the BI architectures and are derived from the literature [38][39][63][78]. While there may be other characteristics that can be used to differentiate between architectures, we think that we have covered the most important differences between the architectures with the characteristics we selected.

We rated the architectures on a scale of 1-5 for each of the characteristics using the information we gained by performing the literature review in the previous sections. On this scale, 1 is low and 5 is high. We developed the framework in such a way that a high score of a particular BI architecture on a particular characteristic is always positive. For example, when a particular architecture scores high on the characteristic 'costs', we mean that compared to other techniques, this architecture has relatively low costs. Although this may seem counter-intuitive, this does allow one to see, in a glance, whether an architecture performs well or not when considering a particular characteristic. To make it easier to see which architectures score well on particular characteristics, we coloured the different values from red to green. Red indicates a low score, gradually moving to green for a high score.

There is one characteristic that is an exception, namely real-time data support. Unlike the other characteristics, this either is or is not supported, there is no middle ground. Therefore, this characteristic should be answered with either 'Yes' or 'No', where 'Yes' is green and 'No' is red. Below, we will shortly explain each of the characteristics that we use in the framework. For clarity, we will also describe what is meant with a high or low score per characteristic:

- Level of integration: The level of integration offered by a particular architecture refers to how well that architecture integrates different data sources. A higher score means better integration.

- Real-time data support: Real-time data support has to do with the latency between an update of the data in the source system and the same update in the data repository that is used by the analytical facilities. If this latency is very low (in the order of milliseconds), one can speak of real-time data support. If the latency is high (in the order of hours, days, weeks) real-time data is not supported. It can be answered with either 'Yes' (when an architecture does provide real-time data support) or 'No' (when an architecture does not provide real-time data support).

- Amount of data – With the amount of data we refer to the ability of a particular architecture to deal with large amounts of data. A higher score indicates that an architecture is better able to deal with large amounts of data.

- Costs – This indicates the costs that are associated with a particular architecture. A higher score indicates that an architecture is relatively cheap compared to other architectures.

- Development time – Some architectures may take longer to develop than others. A higher score indicates a shorter development time.

- Maintainability – While some architectures may be relatively easy/cheap to maintain, others may be the opposite. A higher score means that an architecture is better maintainable.

- Data quality – Some organisations have data in their source systems that is not of a particularly good quality. Some architectures are better to deal with this issue than others (for example by offering cleansing and transformation capabilities). A higher score indicates that an architecture is better able to deal with data of a poor quality.

- Scalability – Some architectures are very scalable, i.e. they can easily be extended to support additional users. A higher score indicates a better scalability.

- Performance – Gathering data from the source systems may have a large impact on the performance of those source systems (nightly batches are not as harmful as daily batches or as complicated queries during the day). However, more and more organisations are moving towards 24/7 service, which makes batch processing harder to deal with; there is not enough time available to run the batch processes without affecting the operational systems. When an architecture scores high on performance, it indicates a relatively low impact on the performance of the source systems.

- Difficulty – Some architectures may be harder to develop and maintain than others. A higher score indicates a lower difficulty level.

- Customisability – When a lot of different users are using the system, some customisation may be desired to fit to the requirements of specific user groups. A higher score indicates a higher ability to customise.

Table 4 shows the mapping of these characteristics to the different BI architectures with the corresponding scores.

***Table 4. Mapping Characteristics vs. BI Architectures***

*BI Architectures ->*

| *Characteristics ->* | Independent DMs | | | | | Data Mart Bus Architecture with Linked Dimensional Data Marts | | | | | Hub-and-Spoke Architecture | | | | | Centralised Data Warehouse Architecture | | | | | Federated Architecture | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *Data Consolidation* | | *Data Propagation* | | *Data Consolidation & Propagation* | *Data Consolidation* | | *Data Propagation* | | *Data Consolidation & Propagation* | *Data Consolidation* | | *Data Propagation* | | *Data Consolidation & Propagation* | *Data Consolidation* | | *Data Propagation* | | *Data Consolidation & Propagation* | *Data Federation* | | *Data Access* | | *Data Virtualisation* | | *Data Mash ups* |
| | *ETL* | *EDR* | *EAI* | *EDR* | *RT-ETL* | *ETL* | *EDR* | *EAI* | *EDR* | *RT-ETL* | *ETL* | *EDR* | *EAI* | *EDR* | *RT-ETL* | *ETL* | *EDR* | *EAI* | *EDR* | *RT-ETL* | *EII* | *EAI* | *EII* | *EAI* | *EII* | *EAI* | *EAI* |
| **Level of integration** | 1 | 1 | 1 | 1 | 1 | 3 | 3 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| **Real-time data support** | No | No | Yes | Yes | Yes | No | No | Yes | Yes | Yes | No | No | Yes | Yes | Yes | No | No | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| **Amount of data** | 5 | 5 | 1 | 3 | 5 | 5 | 5 | 1 | 3 | 5 | 5 | 5 | 1 | 3 | 5 | 5 | 5 | 1 | 3 | 5 | 2 | 2 | 3 | 3 | 2 | 2 | 2 |
| **Costs** | 4 | 5 | 4 | 5 | 2 | 3 | 4 | 3 | 4 | 1 | 2 | 3 | 2 | 3 | 1 | 2 | 3 | 2 | 3 | 1 | 5 | 4 | 5 | 5 | 5 | 4 | 4 |
| **Development time** | 4 | 5 | 4 | 5 | 3 | 3 | 4 | 3 | 4 | 2 | 2 | 3 | 2 | 3 | 1 | 2 | 3 | 2 | 3 | 1 | 5 | 4 | 5 | 4 | 5 | 4 | 4 |
| **Maintainability** | 4 | 5 | 3 | 5 | 3 | 2 | 3 | 1 | 3 | 1 | 2 | 3 | 1 | 3 | 1 | 3 | 4 | 2 | 4 | 2 | 2 | 3 | 2 | 3 | 2 | 3 | 3 |
| **Data quality** | 5 | 1 | 3 | 1 | 5 | 5 | 1 | 3 | 1 | 5 | 5 | 1 | 3 | 1 | 5 | 5 | 1 | 3 | 1 | 5 | 1 | 3 | 1 | 1 | 1 | 3 | 3 |
| **Scalability** | 2 | 2 | 2 | 2 | 2 | 4 | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 2 | 2 | 3 | 3 | 2 | 2 | 2 |
| **Performance** | 5 | 4 | 4 | 4 | 4 | 5 | 4 | 4 | 4 | 4 | 5 | 4 | 4 | 4 | 4 | 5 | 4 | 4 | 4 | 4 | 1 | 2 | 2 | 3 | 1 | 2 | 2 |
| **Difficulty** | 4 | 5 | 4 | 5 | 2 | 3 | 4 | 3 | 4 | 1 | 1 | 2 | 1 | 2 | 1 | 2 | 3 | 2 | 3 | 1 | 4 | 4 | 5 | 5 | 4 | 4 | 4 |
| **Customisability** | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 1 | 1 | 2 | 3 | 3 |

# 5. Requirements

In chapter 4 we mapped the different characteristics to the different BI architectures. The purpose of this was to give a comprehensive overview showing the differences between these architectures. In this chapter, we propose a list of requirements that may influence the choice for a particular BI architecture. The purpose of this list is to identify the requirements that an organisation may have, that we can use in our framework in order to recommend the right architecture. Although the list of characteristics and the list of requirements have different purposes, there are a lot of similarities to be found. This is only logical, since an organisation's requirements that influence the choice for a particular architecture is often the result of architectures scoring better or worse on particular characteristics.

When we searched the literature, we came to the conclusion that, unfortunately, there is very little literature available about factors or requirements that play a role in the selection process. We used the literature that we found in our SLR (see Appendix B for the details) to identify a list of requirements. We then validated this list by performing expert interviews at PwC.

Watson & Ariyachandra [78] identified eleven factors that potentially influence the data warehouse architecture selection decision. They came to these factors by looking at the literature and by interviewing experts. It is important to keep in mind that we do not only look at the data warehouse architecture, but the complete BI architecture, of which the data warehouse architecture is only a part. Next to that, we are looking for requirements, which are somewhat different than factors. However, we think that some of the factors that affect the data warehouse architecture decision may also be important in the selection process of a BI architecture. This means that we should critically look at the factors identified by Watson & Ariyachandra and determine whether or not they are also relevant for the choice of a BI architecture, and if so, how we should adjust them in order to adopt them in our list of requirements. Below, we discuss the different factors introduced by Watson & Ariyachandra and whether we adopt them or not.

1. *Information interdependence between organisational units* – When there is a high level of information dependence, the work of one organisational unit is dependent upon information from one or more other organisational units. In this situation, the ability to consistently share and integrate information is important. It is likely that firms with high information interdependence select an enterprise-wide architecture.

   This is definitely a factor that influences the choice for a particular BI architecture and we will adopt this as requirement, by adjusting it to 'high information exchange between departments'.

2. *Upper management's information needs* – Management requires information from lower organisational levels, for example, to monitor progress, drill-down in areas of interest, aggregate lower-level data, and be confident that the organisation is in compliance with regulations. An architecture that supports this is important.

   While the upper management's information needs can influence the choice for a particular architecture, we do not incorporate this in our research. The reason for this is that, as we said before, a list of KPIs would be a research itself. We therefore look at a more technical/functional level, e.g. whether there is a need for real-time data, whether there is a need for enterprise-wide information etc. This still allows us to recommend specific BI architectures, but with a smaller scope.

3. *Urgency of need for a data warehouse* – Some architectures are more quickly implemented than others, which implies that in case that time is an important factor to the focal organisation, the choice can be different compared to when it is not.

   We will adopt this factor. We translated it to the requirement 'low development time'.

4. *Nature of end-user tasks* – Users that perform non-routine tasks may require an architecture that is better suited to analyse the data in different, more creative ways.

This factor will have the most influence on the fourth layer, specifically the 'style of BI'. Since we do not look at specific tools, but only gave a general classification of the BI tools in five styles and did not take the fourth layer into account in our BI architectures overview, we do not include this in our list of requirements. We do include customisation as we will discuss later.

5. *Constraints on resources* – Some architectures require fewer resources to implement and operate than others. This can impact the selection of a BI architecture.

   The constraints on resources, such as a limited amount of time or money available, can also influence the choice for a particular architectural style. We therefore include this in our list. We named this 'few resources required'.

6. *Strategic view of the data warehouse prior to implementation* – Organisations can look very differently to date warehouses. Some see it as a way to provide a 'point solution' to a particular functional unit, while others may see it as a critical enabler to support organisation's strategic business objectives.

   The strategic view of the warehouse prior to implementation may influence the choice for a particular BI architecture. However, there are many strategic views and it is simply not possible for us to incorporate these.

7. *Expert influence* – Consultants, literature, end-users etc. may all influence the decision-making process for a particular architecture.

   We exclude expert influence from our list, since we cannot say anything about this. Different experts may recommend different things, so although this might be a factor influencing the choice, there is no advice to give.

8. *Compatibility with existing systems* – Often, the transactional systems are already in place. This means that the data warehouse that is selected should be compatible with those systems. This may also influence the choice for a particular data warehouse architecture.

   Compatibility with existing systems is also excluded from this research. The reason is that we cannot give any general advice on whether a specific system would work with a specific integration technology. This would require us to look into all different source systems, which is simply impossible to do. Still, most systems nowadays are compatible with the different integration techniques. Exceptions are old legacy systems. Nevertheless, we assumed that compatibility will not be a problem.

9. *The perceived ability of the in-house staff* – Some data warehouse architectures are more challenging than others. Depending on the internal IT staff's technical skills, experience and level of confidence, different architectures may be chosen. Specifically, an architecture that is compatible with what they think can be successfully built.

   The perceived ability of in-house staff definitely can influence the choice for a particular architecture, since some set-ups are easier to maintain/setup than others. Of course, organisations could choose to train their IT staff or hire additional employees or consultants. Since this characteristic covers multiple aspects, we decided to use the more specific characteristics such as complexity and maintainability. We transformed this factor into the following requirement 'low required competence of in-house IT staff'.

10. *Source of sponsorship* – The source of sponsorship can also be an influencing factor. The choice may be different when a single department is the sponsor, compared to when top management is.

    We think that the source of sponsorship could influence the choice for a particular architecture in the following way: If the need for a new BI solution is only supported in particular departments (who for the most part focus on their own department), this may have other implications compared to when it is supported by top management (who focus on the entire enterprise). However, we cannot really compare different architectures on the aspect 'source of sponsorship', since it is not really a requirement. Therefore, we decided to leave this one out.

11. *Technical issues* – Examples are scalability, data volume, performance etc.

    We are, in fact, looking at requirements that influence the choice for a particular architecture, but this is not a requirement itself. We cover this aspect by naming more specific requirements such as scalability, ability to deal with large amounts of data etc.

In addition to these requirements derived from Watson & Ariyachandra, we used Saunders [65] and Schiff & Michaels [66] to complement the list with the following requirements:

- *Ability to deal with low data quality in source systems* – There may be a large difference in the quality of the data in source systems. This can influence the decision for a particular architecture. For example, if the data quality is low, a solution with better cleansing/transformation abilities may be chosen.

- *Need for real-time data* – Some BI solutions may need to deliver real-time information to the user. This influences the architectural choice to be made, since some integration technologies are unable to provide real-time data support.

- *Ability to deal with large amounts of data* – The amount of data the BI solution has to deal with can also influence the choice for a particular BI architecture. Some architectures are better suited to deal with large amounts of data than others.

- *Ability to deal with frequent changes and updates* – Some architectures are better able to deal with changes and updates than others. So this should also be considered when selecting a particular BI architecture.

- *High Scalability* – Another consideration that may be important to organisations is whether a particular architecture is scalable or not. Some architectures are very scalable, while others are not.

- *High Customisability* – Some architectures allow for more customisation than others. This may be important for particular organisations who want a BI solution that is going to be used by a lot of different users. These user groups may have specific requirements and require their own customised view to do their work.

This resulted in the list shown in table 5. By no means we claim that this list is exhaustive, but we do think that the most important requirements that play a role in the selection process are covered. We validated this list by performing expert interviews at PwC, the University of Twente and at organisations that either have a BI solution or that help with the selection and implementation of one. Next to that, organisations that participated in our questionnaire were able to indicate additional requirements, which are discussed in chapter 7.

*Table 5. List of Requirements that may influence the Selection Process*

| Requirements | Source |
|---|---|
| 1. High information exchange between departments<br>2. Low development time<br>3. Few resources required<br>4. Low competence required of in-house staff | Watson & Ariyachandra [78] |
| 5. Ability to deal with low data quality in source systems<br>6. Need for real-time data<br>7. Ability to deal with large amounts of data<br>8. Ability to deal with frequent changes and updates<br>9. High scalability<br>10. High maintainability<br>11. Low difficulty/complexity BI solution<br>12. Low impact on performance source systems<br>13. High performance of BI solution<br>14. High level of integration<br>15. High customisability | Saunders [65], Schiff & Michaels [66] |

# 6. Framework

In this chapter, we create the link between chapter 4 (which describes the BI architectures) and chapter 5 (which lists the requirements that may influence the selection of a particular BI architecture). A framework is developed, which assigns scores to every architecture for every requirement. Scores can vary on a scale of 1 to 5. A higher score indicates better support of that specific requirement by a particular architecture. The 'need for real-time data' is an exception and is answered by either 'Yes' or 'No', since architectures offer this option or not. There is no middle ground. This means that if real-time data is essential for an organisation, then every architecture that does not offer this functionality is no viable option.

We assigned the scores using the information we gathered in our SLR. Inevitably, there is some subjectivity involved when scoring the different architectures. As with the mapping of characteristics to architectures, we tried to be as objective as possible, but one could argue whether a particular architecture should score, for example, 4 or 5 points for a particular requirement. The reason that we chose for a 5-point scale instead of a 3-point scale is because of the fact that we cannot indicate the differences between the architectures well enough on a 3-point scale. Architectures would have similar scores while there are definitely differences to be identified between those architectures.

For example, when we look at the requirement 'few required resources'; if we would use a 3-point scale, with a few exceptions, all BI architectures with the same data warehouse architecture would score the same, although e.g. a data warehouse architecture with EDR as integration service is definitely cheaper than the same data warehouse architecture with ETL as integration service. However, that would not justify a lower score on the 3-point scale, because it still more expensive than EDR in combination with a cheaper data warehouse architecture. This is just one example which shows why we chose for a 5-point scale, since we *can* point out these differences on that scale.

The drawback of larger scales, however, is that it becomes a more subjective matter. We therefore chose for the 5-point scale; in our opinion it has enough detail to differentiate the different architectures from each other, while keeping the subjectivity to a minimum.

To make it easier to see which architectures score well on particular requirements, we coloured the different values from red to green. Red indicates a low score, gradually moving to green for a high score. We will perform a validation of the framework in chapter 7.

## 6.1. How the Framework should be used

The way the framework can be used is as follows. Organisations themselves or advisors can indicate which requirements they deem important. Next to that, they can also indicate exactly how important that particular requirement is for the focal organisation, by assigning a weight on a scale of 1 to 5 (with the exception of the 'need for real-time data'). This can be done in the rightmost column (grey colour).

For every requirement that is indicated as important, the weight that is assigned to that requirement by the focal organisation or advisor is then multiplied with the assigned score of a particular architecture for that same requirement. This is done for every architecture. The outcomes of the multiplications are then summed up, which leads to a total score of a particular architecture in the situation of a specific organisation given their requirements. The architecture with the highest total score is considered 'best' for that organisation. This addition is possible due to the fact that a higher score for each of the requirements always means it is better at that particular requirement than a lower score.

By asking organisations to assign a ratio/weight to each of the requirements they deem important, they can indicate how important they deem that requirement in relation to the other requirements. By performing the multiplication and addition that we described above, we are able to compare architectures that may have very different specialities. This allows us to select the 'best' choice for a particular organisation given their rating of the different requirements. The framework is shown in table 6.

## 6.2. How the Framework should not be used

Although the developed framework can be very useful for organisations that are in the process of selecting a new BI architecture, it should be used the way it is intended. If it is used for different purposes or in another way, it could lead to unexpected or incorrect results. Below, we list some examples of the uses for which the framework is not intended.

1. Do not simply add up all scores to determine a 'best' architecture. The framework is not designed for that. A ratio that indicates how important a particular requirement is should always be included. If one simply adds up all scores, it would mean that each requirement is important and that none of the requirements is more important than another, which is very unlikely.

2. For the same reason, looking at the architectures that are coloured green and then conclude that those are the 'best' architectures is not for what this framework is intended. The reason for this is that although an architecture may score high on a lot of requirements – and thus is indicated as a lot of green cells – it does not imply that this architecture is always better. Maybe those few requirements on which an architecture does not score well on are very important for an organisation; in that case another architecture is probably a better choice. This means that the framework should always be used in the context of organisations or for comparing architectures on the same requirement.

3. When an architecture scores 5 for a particular requirement it does not mean that it is five times as good as an architecture that only scores 1 for that same requirement. Similarly, an architecture with a score of 2 is not twice as good as an architecture scoring 1. One architecture scoring higher on a particular requirement simply means that it is better suited to satisfy that requirement compared to another scoring lower on that same requirement. The larger the difference between scores, the better one architecture is in satisfying that requirement, relative to the other architecture.

## 6.3. Additional Remark

Even if the framework is used as intended, the results should still be interpreted with care. There may be organisation specific requirements which are not considered in the result. For example, one such requirement may be that an organisation states that the architecture of choice must have the highest score on a certain requirement. Even if that requirement is weighted with a score of 5, the framework still takes the total score including all requirements with their corresponding assigned ratios. It may be the case that the architecture with the highest total score does not have the highest score on that particular requirement.

## *Table 6. Framework Requirements vs. BI Architectures*

BI Architectures ->

| Requirements -> | Independent DMs | | | | | Data Mart Bus Architecture with Linked Dimensional Data Marts | | | | | Hub-and-Spoke Architecture | | | | | Centralised Data Warehouse Architecture | | | | | Federated Architecture | | | | | | | Ratio |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *Data Consolidation* | | *Data Propagation* | | *Data Consolidation & Propagation* | *Data Consolidation* | | *Data Propagation* | | *Data Consolidation & Propagation* | *Data Consolidation* | | *Data Propagation* | | *Data Consolidation & Propagation* | *Data Consolidation* | | *Data Propagation* | | *Data Consolidation & Propagation* | *Data Federation* | | *Data Access* | | *Data Virtualisation* | | *Data Mash ups* | *1-5/YES-NO* |
| | *ETL* | *EDR* | *EAI* | *EDR* | *RT-ETL* | *ETL* | *EDR* | *EAI* | *EDR* | *RT-ETL* | *ETL* | *EDR* | *EAI* | *EDR* | *RT-ETL* | *ETL* | *EDR* | *EAI* | *EDR* | *RT-ETL* | *EII* | *EAI* | *EII* | *EAI* | *EII* | *EAI* | *EAI* | |
| **High information exchange** | 1 | 1 | 1 | 1 | 1 | 3 | 3 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | |
| **Low development time** | 4 | 5 | 4 | 5 | 3 | 3 | 4 | 3 | 4 | 2 | 2 | 3 | 2 | 3 | 1 | 3 | 4 | 3 | 4 | 2 | 5 | 4 | 5 | 4 | 5 | 4 | 4 | |
| **Few required resources** | 3 | 4 | 3 | 4 | 2 | 3 | 4 | 3 | 4 | 2 | 2 | 3 | 2 | 3 | 1 | 3 | 4 | 3 | 4 | 2 | 5 | 4 | 5 | 5 | 5 | 4 | 4 | |
| **Low required competence of in-house staff** | 4 | 5 | 4 | 5 | 2 | 3 | 4 | 3 | 4 | 1 | 1 | 2 | 1 | 2 | 1 | 2 | 3 | 2 | 3 | 1 | 4 | 4 | 5 | 5 | 4 | 4 | 4 | |
| **Ability to deal with low data quality** | 5 | 1 | 3 | 1 | 5 | 5 | 1 | 3 | 1 | 5 | 5 | 1 | 3 | 1 | 5 | 5 | 1 | 3 | 1 | 5 | 1 | 3 | 1 | 1 | 1 | 3 | 3 | |
| **Support for real-time data** | No | No | Yes | Yes | Yes | No | No | Yes | Yes | Yes | No | No | Yes | Yes | Yes | No | No | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | |
| **Abillity to deal with large amounts of data** | 5 | 5 | 1 | 3 | 5 | 5 | 5 | 1 | 3 | 5 | 5 | 5 | 1 | 3 | 5 | 5 | 5 | 1 | 3 | 5 | 2 | 2 | 3 | 3 | 2 | 2 | 2 | |
| **Ability to deal with frequent changes/update** | 4 | 5 | 3 | 4 | 4 | 3 | 4 | 2 | 3 | 3 | 3 | 4 | 2 | 3 | 3 | 4 | 5 | 3 | 4 | 4 | 3 | 2 | 3 | 2 | 3 | 2 | 2 | |
| **High scalability** | 2 | 2 | 2 | 2 | 2 | 4 | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 2 | 2 | 3 | 3 | 2 | 2 | 2 | |
| **High maintainability** | 2 | 3 | 1 | 3 | 1 | 2 | 3 | 1 | 3 | 1 | 3 | 4 | 2 | 4 | 2 | 4 | 5 | 3 | 5 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 3 | |
| **Low difficulty/complexity of solution** | 4 | 5 | 4 | 5 | 2 | 3 | 4 | 3 | 4 | 1 | 2 | 3 | 2 | 3 | 1 | 3 | 4 | 3 | 4 | 2 | 4 | 4 | 5 | 5 | 4 | 4 | 4 | |
| **Low impact on performance** | 5 | 4 | 4 | 4 | 4 | 5 | 4 | 4 | 4 | 4 | 5 | 4 | 4 | 4 | 4 | 5 | 4 | 4 | 4 | 4 | 1 | 2 | 2 | 3 | 1 | 2 | 2 | |
| **High performance BI solution** | 5 | 5 | 5 | 5 | 5 | 4 | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 2 | 2 | 3 | 3 | 2 | 2 | 2 | |
| **High level of integration** | 1 | 1 | 1 | 1 | 1 | 3 | 3 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | |
| **High level of customisability** | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 1 | 1 | 2 | 3 | 3 | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

# 7. Validation

In order to validate the framework we proposed in chapter 6, we interviewed three organisations and performed a questionnaire that resulted in a response of 21 organisations. This chapter shows the results of both interviews and questionnaire and the discussion of those results.

We chose for a combination of interviews and a questionnaire for the following reason. We required a lot of results in order to validate our framework. A questionnaire is the best way to get that number of results. However, it is important to realise that the quality of the answers in a questionnaire is usually lower than the answers in an interview. We tried to mitigate this as much as possible by asking respondents whether we could contact them if we had additional questions. This would allow us to contact those organisations that were either not satisfied with an architecture that was recommended by the framework or organisations that were satisfied with an architecture that was not recommended by the framework. In the case of an interview, we already had in-depth knowledge of the choices made, so we could by ourselves identify the cause for choosing a different architecture if that would be the case.

Compared to a questionnaire, interviews usually do give information of a higher quality, but also take more time to perform. For this reason, we performed three interviews next to the questionnaire.

## 7.1. Questionnaire

In this section, we present and discuss the results of the questionnaire, in order to validate our framework. The organisations that were asked to fill in the questionnaire were organisations that had a BI solution in place. We first only approached organisations with a BI tool that were customers of PwC and classified as private companies/public sector (PC/PS), so this does not include, for example, large banks and listed organisations. The reason for this is that these were the organisations that were relatively easily accessible to us, due to the fact that the thesis was written at PwC Risk Assurance PC/PS. We approached these organisations via the contact persons of PwC.

Unfortunately, the number of responses was a lot lower than we expected. This was due to the fact that 1) distribution did not go as smoothly as one would hope, 2) the number of suitable organisations seemed to be rather low and 3) not every organisation was able/willing to cooperate.

Therefore, we decided to distribute the questionnaire via other ways as well. We used Google and LinkedIn to find organisations with a BI solution in place. A lot of these organisations were very large companies, since those organisations were often easier to find on the internet. If PwC was involved with that organisation we tried to distribute it via PwC's contact person. When this was not the case, we approached the organisation ourselves. To maximise the response rate, we tried to approach the right person personally by mail.

This approach gave us numerous responses. We only contacted organisations that were active in the Netherlands, since we developed the questionnaire in Dutch. The reason for this is that we wanted to minimise the number of interpretation errors that can be accounted to language barriers and next to that we expected to receive enough responses from organisations in the Netherlands.

We asked these organisation what requirements played a role in the selection process and for those requirements that did play a role we asked them to assign a weight that indicated how important that particular requirement was. Next to that, we also asked them whether they were satisfied with their current BI solution. For each organisation, we then filled our framework with the requirements that the respondents indicated to have played a role in the selection process, together with the weight of that particular requirement. Using the approach described above, we multiplied these weights with the scores of the different architectures and added up these multiplications to come to a total score. This in order to see if their architectural choice is in line with the architecture that our framework recommends for their situation.

One way to categorise the responses is by using a 'confusion matrix' which is visualised in figure 14. It categorises the responses in four categories. This categorisation allows us to get an indication of the correctness of our framework. In the case that an organisation is satisfied with their current architecture that is recommended by our framework (the green top-left box), or when an organisation is unsatisfied and their architecture is not recommended by our framework (the green bottom-right box), it gives us indications that our framework is correct.

However, in the case that an organisation is satisfied with an architecture that is not recommended by our framework or when an organisation is unsatisfied with an architecture that is recommended by our framework, it gives an indication that our framework may require improvements.

Very important to realise is that one should be very careful with these conclusions, since an organisation might be very satisfied with an architecture that was indicated by our framework as 'second best'. This does not mean that our framework is incorrect. Our framework suggests a 'best' architecture for a particular situation, but that does not mean that a different architecture cannot satisfy an organisation's requirements.

| | User is Satisfied | User is Unsatisfied |
|---|---|---|
| Recommended by Framework | | |
| Not Recommended by Framework | | |

**Figure 14. Confusion Matrix showing the possible Response Categories.**

When an organisation chose for a different architecture then our framework suggests for their situation – given their requirements – this could mean a couple of things:

1. Our framework is incorrect.
2. The organisation made the wrong architectural choice, for example because they were not aware of all architectures and the differences between those architectures.
3. The results of the questionnaire do not correspond to the real situation, or incorrect conclusions were drawn (This could, for example, be caused by the respondent not understanding the questions, or because we asked questions that can be interpreted in multiple ways).

To reduce the chance on the third possible cause we asked multiple people for feedback on the questionnaire before sending it out. We made the necessary adjustments to make the questions as easy as possible to understand and tried to formulate them in such a way that they were not susceptible to interpretation errors. Next to that, we asked – at the end of our questionnaire – whether we could contact the respondent in case we had additional questions. This way, we would be able to contact those organisations that fell in one of the two 'red' categories who did not mind us contacting them and this would allowed us to ask them why they did not choose for the architecture that our framework proposes or why they were not satisfied when that was not clear from the answers they had given. This would give us more certainty during the validation of our framework.

We distributed the questionnaire to 94 organisations of which 21 organisations responded (22.3% response rate). One of the questionnaire responses did give us valuable information, but was not suited to validate our framework. The reason for this is that their BI environment was so complex; all architectures more or less existed and all requirements did play a role in the selection of the different solutions over the past twenty years. For this reason, some of the graphs in this section include one fewer result.

Due to the number of responses, we are unable to generalise the results to the entire population of organisations in the world, or even in the Netherlands. However, it does give us an indication about the validity of the framework, which, in the end, was the goal of performing this questionnaire. Although it would be nice to be able to generalise to the entire population, it would require too much time to gather enough responses to be able to do so. In the responses we have received we see a lot of similarities, so for the purpose of validating our framework it would be unlikely that more responses would lead to a different conclusion. More responses would, however, put more confidence in the results.

### *7.1.1. Questionnaire Results*

As you can see in figure 15, organisations from a number of different sectors participated in the questionnaire. The categorisation used is called the 'Standaard Bedrijfsindeling' (SBI) developed by the 'Centraal Bureau voor de Statistiek' (CBS) [15]. In the figure, we did not include the following sectors, since we did not receive responses from those sectors:

- Agriculture, Forestry and Fishing
- Mining and Quarrying
- Water supply, sewerage, waste management and remediation activities
- Accommodation and food service activities
- Consultancy, research and other specialised business services
- Renting and leasing of tangible goods and other business services
- Public administration, public services and compulsory social security
- Culture, sports and recreation
- Activities of households as employers; undifferentiated goods- and service- producing activities of households for own use
- Extraterritorial organisations and bodies


**Figure 15. Overview of Sector of the Respondents' Organisation.**

Figure 16 shows the segment in which the respondents' organisation is active and figure 17 shows the annual revenue. Except for one participating organisation who had 250-500 employees, every participating organisation had more than 500 employees.

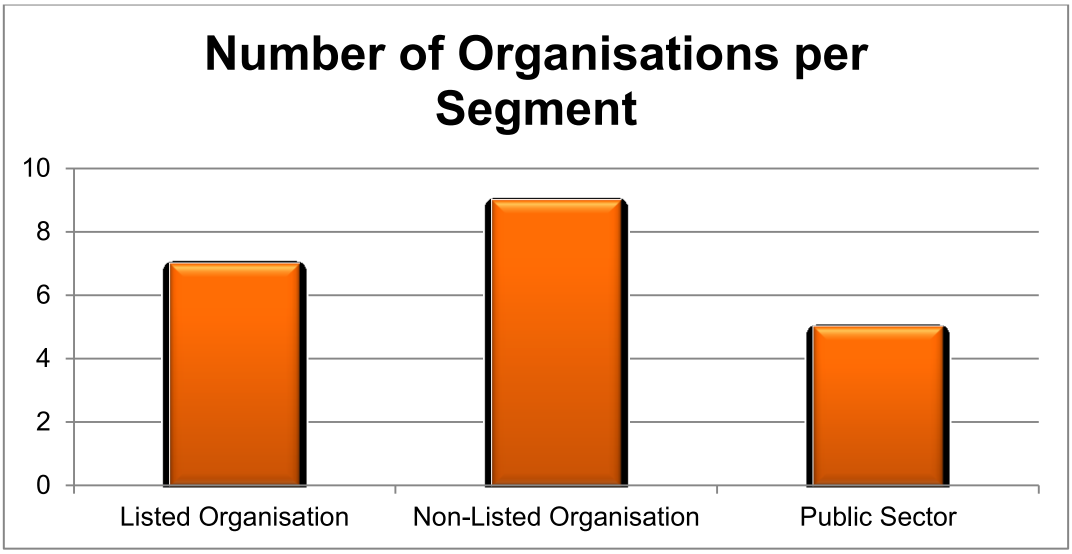



**Figure 16. Overview of the Segment of the Respondents' Organisation.**

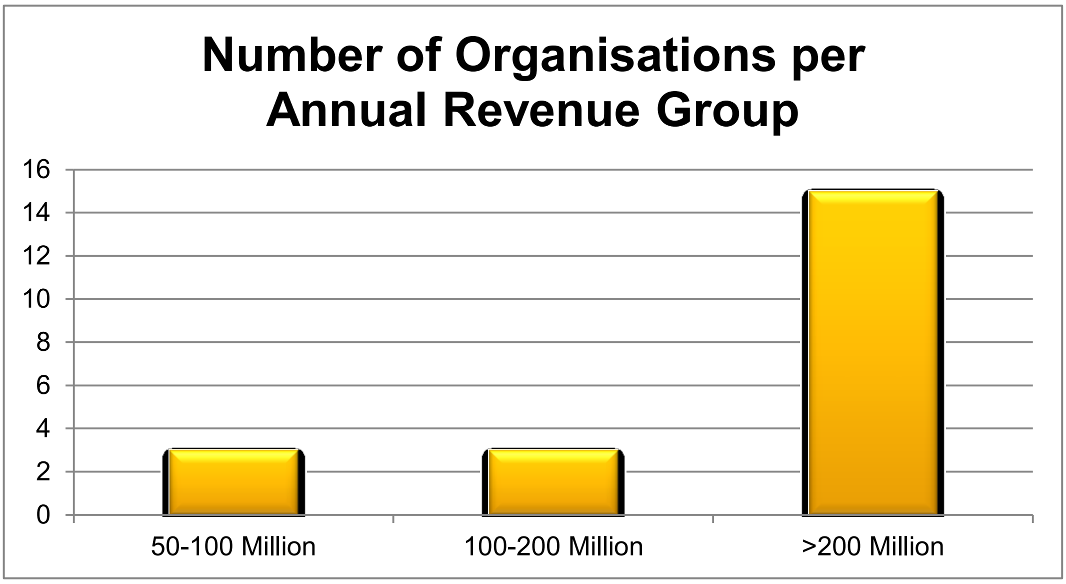



**Figure 17. Overview of the Annual Revenue of the Respondents' Organisation.**

Figure 18 shows the number of respondents that indicated that a particular requirement played a role in the selection of their organisation's BI architecture. Maintainability was most often indicated to have played a role, while none of the respondents indicated that the ability to deal with low data quality was an influencing factor.

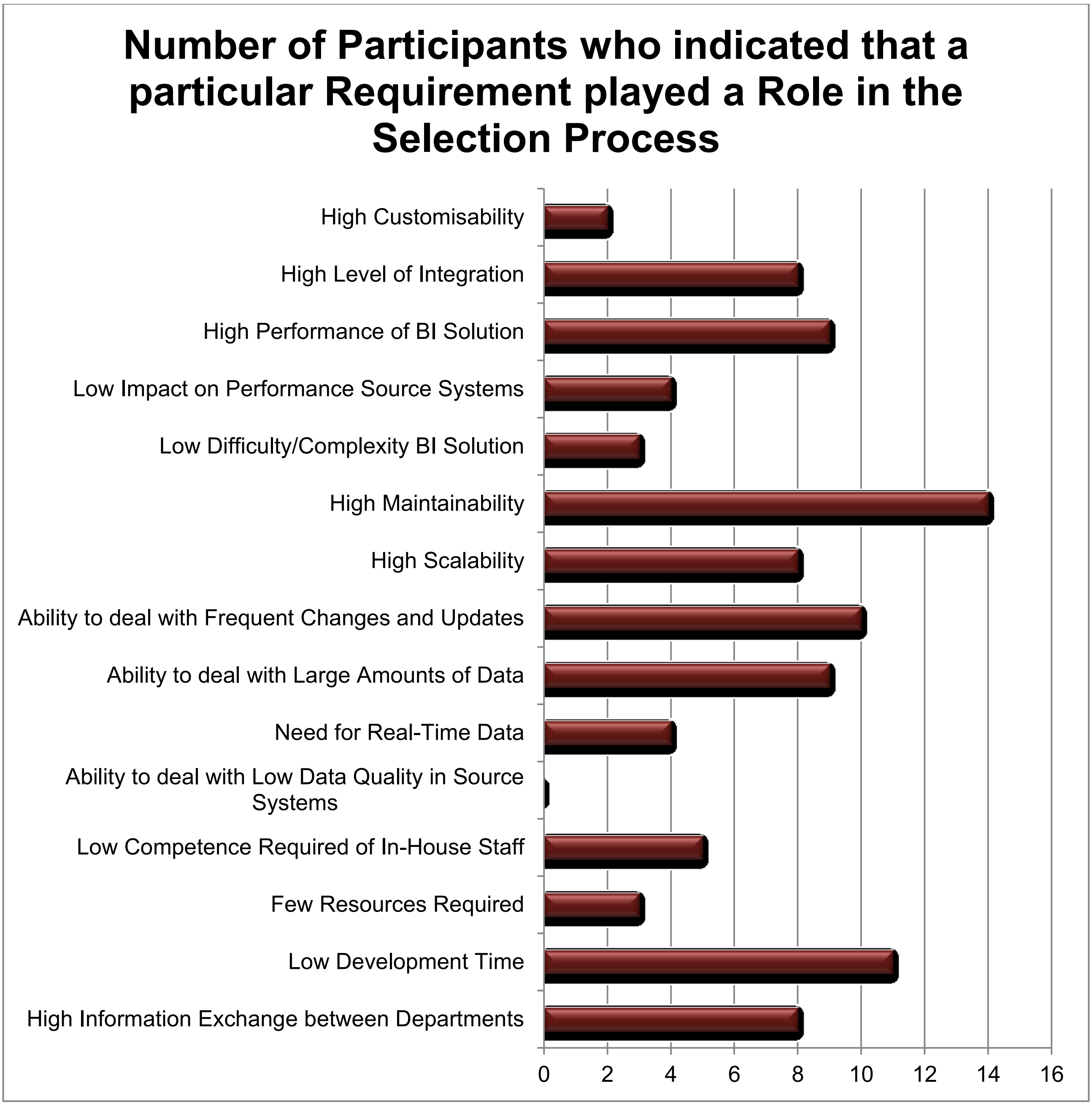



**Figure 18. Overview of the Requirements that played a Role in the Selection Process of the Respondents' Organisation.**

Table 7 shows how important the particular requirements were according to the respondent. In addition, table 8 shows how well their current BI environment satisfies those same requirements. It is important to not only look at average scores here; this may give a biased view, since the number of respondents is relatively low.

*Table 7. Importance of indicated Requirements.*

| Requirement | Min Value | Max Value | Average Value | Standard Deviation | Responses |
|---|---|---|---|---|---|
| High Information Exchange between Departments | 2.00 | 5.00 | 3.50 | 1.07 | 8 |
| Low Development Time | 3.00 | 5.00 | 3.91 | 0.70 | 11 |
| Few Resources Required | 2.00 | 4.00 | 2.67 | 1.15 | 3 |
| Low Competence Required of In-House Staff | 2.00 | 4.00 | 3.20 | 1.10 | 5 |
| Ability to deal with Low Data Quality in Source Systems | 0.00 | 0.00 | 0.00 | 0.00 | 0 |
| Need for Real-Time Data | 4.00 | 5.00 | 4.25 | 0.50 | 4 |
| Ability to deal with Large Amounts of Data | 2.00 | 5.00 | 3.89 | 1.17 | 9 |
| Ability to deal with Frequent Changes and Updates | 3.00 | 5.00 | 3.90 | 0.57 | 10 |
| High Scalability | 3.00 | 5.00 | 4.25 | 0.71 | 8 |
| High Maintainability | 3.00 | 5.00 | 4.07 | 0.73 | 14 |
| Low Difficulty/Complexity BI Solution | 3.00 | 5.00 | 4.00 | 1.00 | 3 |
| Low Impact on Performance Source Systems | 3.00 | 4.00 | 3.50 | 0.58 | 4 |
| High Performance of BI Solution | 3.00 | 5.00 | 4.00 | 1.00 | 9 |
| High Level of Integration | 3.00 | 5.00 | 4.13 | 0.64 | 8 |
| High Customisability | 3.00 | 4.00 | 3.50 | 0.71 | 2 |

*Table 8. Satisfaction with how the Current BI Architecture fulfils indicated Requirements.*

| Requirement | Min Value | Max Value | Average Value | Standard Deviation | Responses |
|---|---|---|---|---|---|
| High Information Exchange between Departments | 3.00 | 5.00 | 4.00 | 0.93 | 8 |
| Low Development Time | 2.00 | 5.00 | 3.73 | 0.90 | 11 |
| Few Resources Required | 4.00 | 5.00 | 4.33 | 0.58 | 3 |
| Low Competence Required of In-House Staff | 2.00 | 5.00 | 3.40 | 1.14 | 5 |
| Ability to deal with Low Data Quality in Source Systems | 0.00 | 0.00 | 0.00 | 0.00 | 0 |
| Need for Real-Time Data | 4.00 | 5.00 | 4.25 | 0.50 | 4 |
| Ability to deal with Large Amounts of Data | 3.00 | 5.00 | 3.89 | 0.78 | 9 |
| Ability to deal with Frequent Changes and Updates | 2.00 | 5.00 | 3.70 | 1.06 | 10 |
| High Scalability | 3.00 | 5.00 | 3.88 | 0.83 | 8 |
| High Maintainability | 2.00 | 5.00 | 4.00 | 0.88 | 14 |
| Low Difficulty/Complexity BI Solution | 4.00 | 5.00 | 4.33 | 0.58 | 3 |
| Low Impact on Performance Source Systems | 3.00 | 5.00 | 3.75 | 0.96 | 4 |
| High Performance of BI Solution | 3.00 | 5.00 | 3.78 | 0.67 | 9 |
| High Level of Integration | 2.00 | 5.00 | 3.88 | 0.99 | 8 |
| High Customisability | 3.00 | 4.00 | 3.50 | 0.71 | 2 |

**Figure 19. Overview of the Importance and the Satisfaction of particular Requirements.**

Figure 20 shows an overview of the different data warehouse architectures and the number of respondents that indicated that their organisation has a particular data warehouse architecture. As you can see, the number of respondents that indicated that their organisation uses a Hub-and-Spoke architecture is highest, while none of the respondents indicated that their organisation was using a federated architecture.

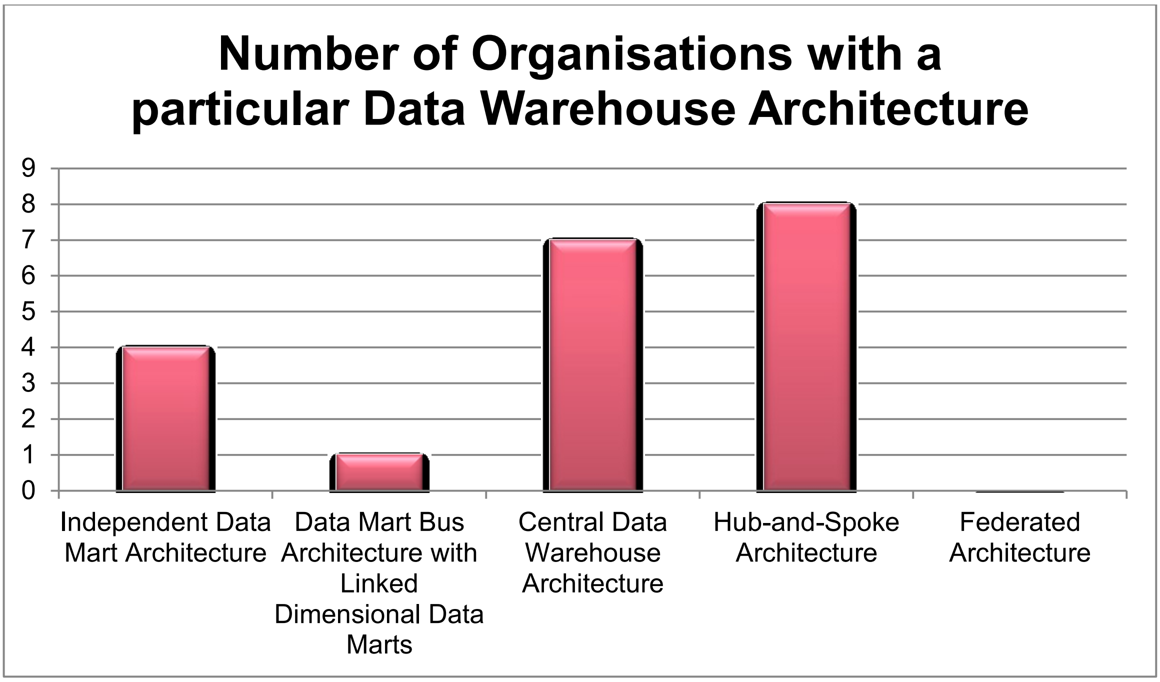



**Figure 20. Overview of the Data Warehouse Architecture of the Respondents' Organisation.**

In figure 21 the different integration technologies that are used by the respondents' organisation are indicated. Very striking in this picture is that every respondent indicated that their organisation uses ETL as integration service. Some organisations use multiple integration services. A couple of the respondents indicated that their ETL tool is also able to deal with real-time data. So although this may be called ETL, we would categorise this as RT-ETL. This is the case in four of the answers, so in the context of this research figure 22 would give the correct view.

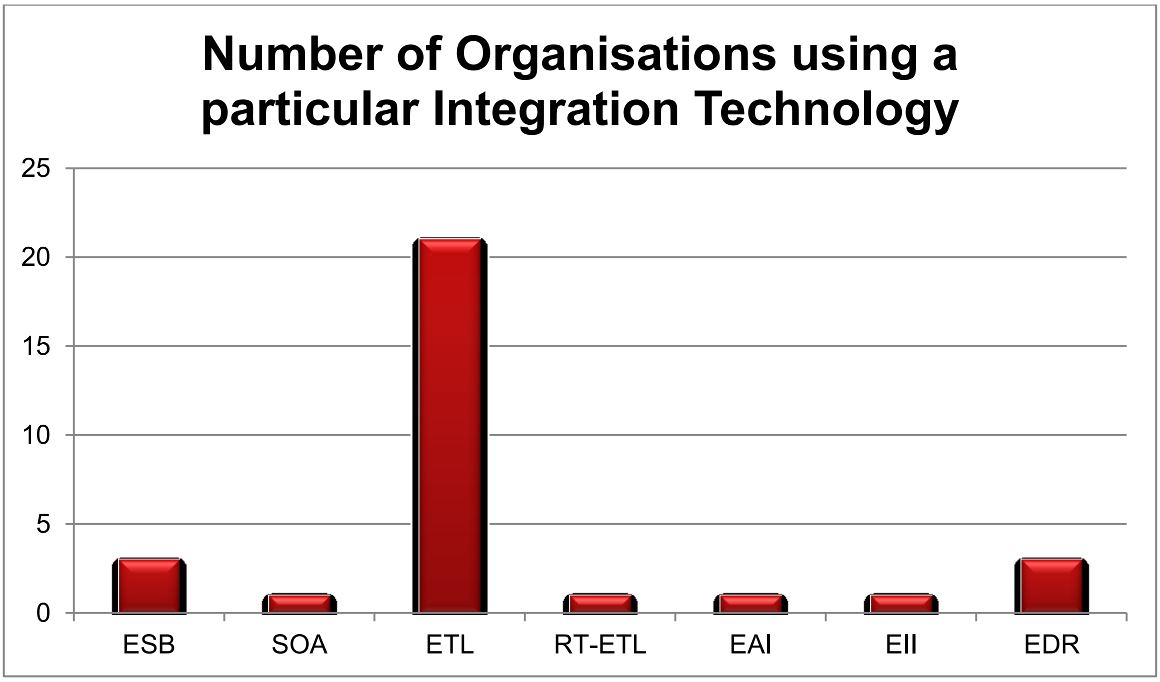



**Figure 21. Overview of the Integration Service of the Respondents' Organisation.**

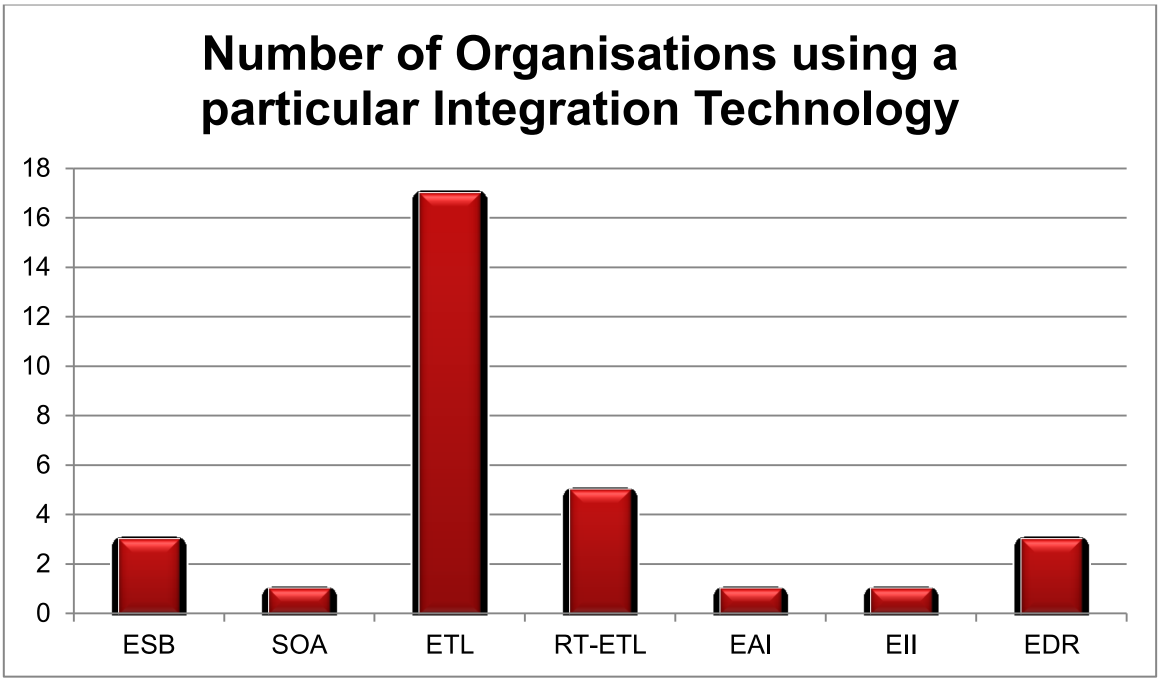



**Figure 22. Overview of the adjusted Integration Service of the Respondents' Organisation.**

In order to get a more complete view of the context of the respondents' organisation and their BI environment, we asked them a couple of more general questions. Figure 23 shows a pyramid that indicates different levels of BI. The lower levels emphasise the measuring and monitoring of activities, while the higher levels emphasise steering and decision-making. We showed the respondents this pyramid and asked them which layers of the pyramid they saw back in their organisation. The results are shown in figure 24.

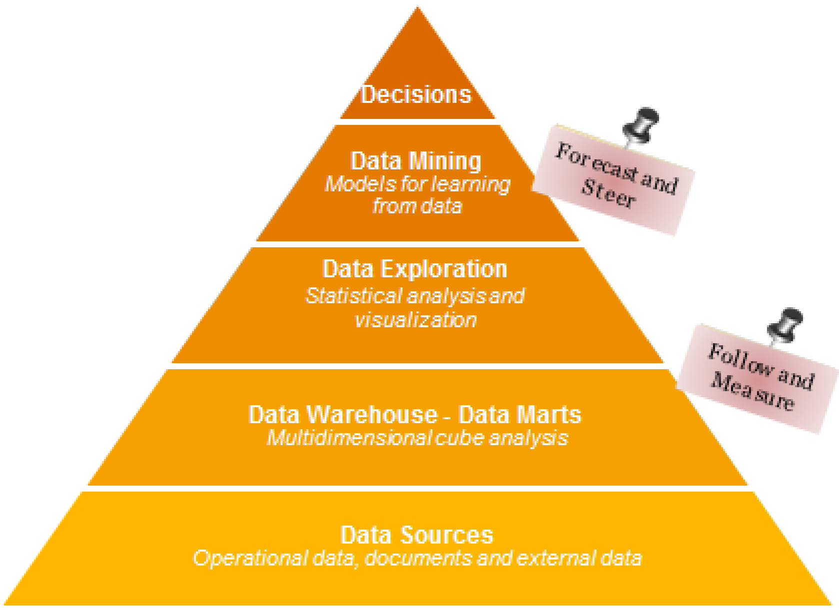



**Figure 23. Levels of BI.**

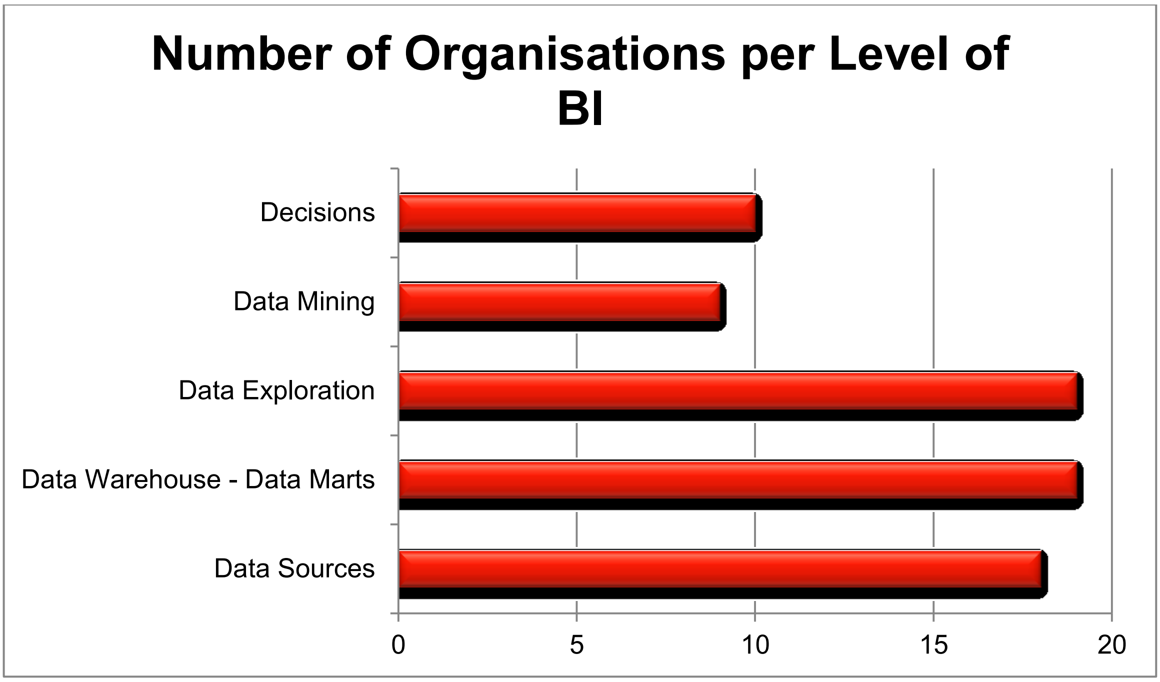



**Figure 24. Overview of the Level of BI of the Respondents' Organisation.**

We also asked which type of BI solution the respondents' organisation had in place. The choices correspond to the styles of BI introduced in section 4.5. See figure 25 for the results.

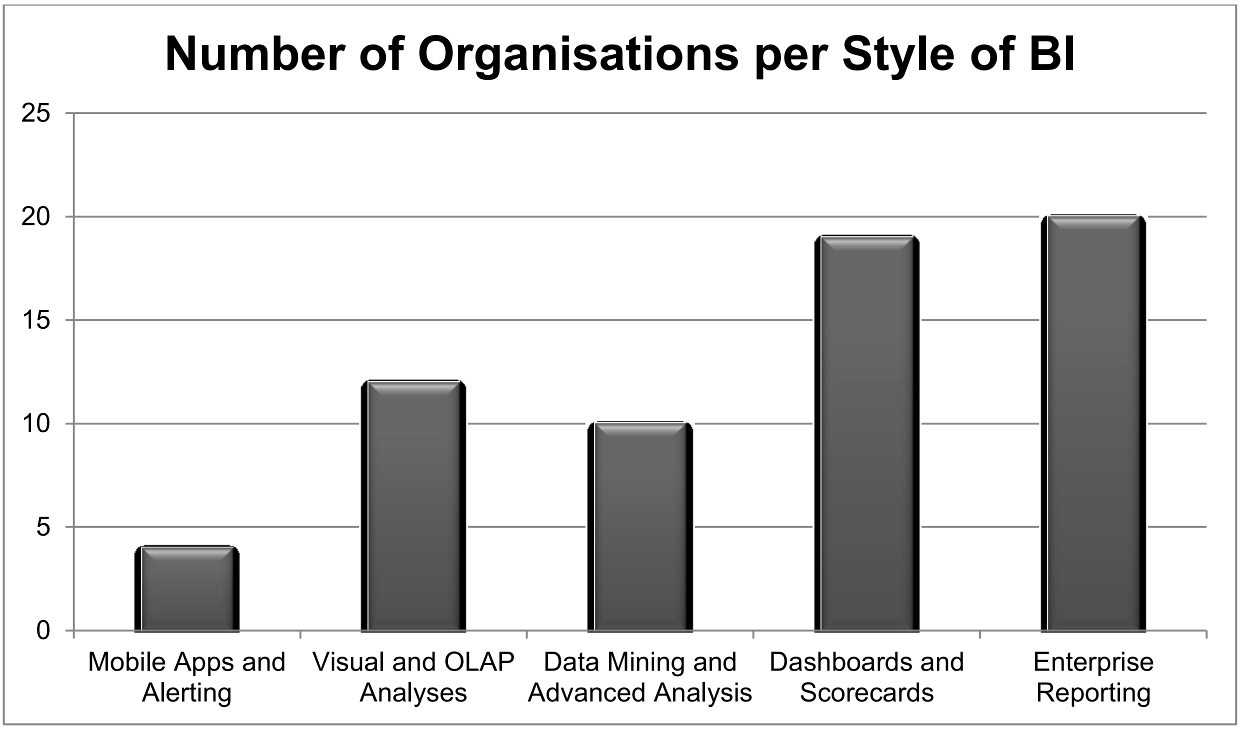



**Figure 25. Overview of the Styles of BI of the Respondents' Organisation.**

Figure 26 shows whether the respondents' organisation is using OLAP, and if so which type of OLAP.

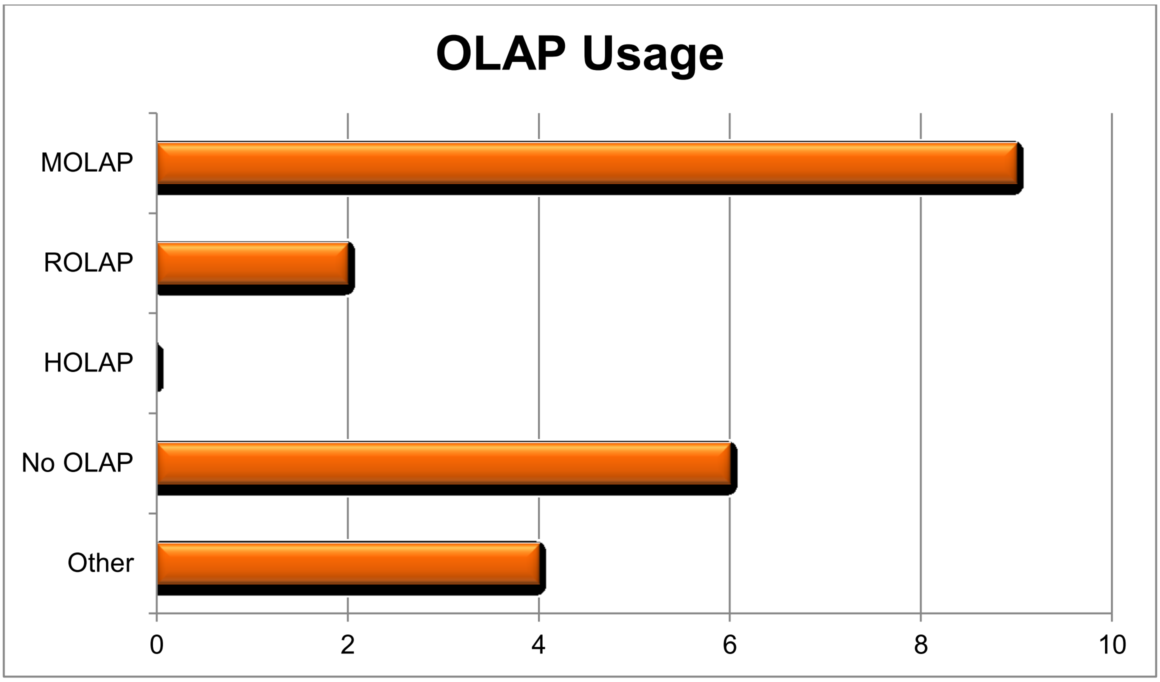



**Figure 26. Overview of whether the Respondents' Organisation is using OLAP.**

Additionally, we also asked organisations some open questions, including questions on why their organisation chose to implement a BI solution, what the largest challenges are in the area of BI and whether they expect the role of BI to become larger or smaller in the future.

There were a lot of different reasons why organisations chose to implement a BI solution. A couple of reasons were named very often: more control, more insight, better reporting functionality, accountability and/or compliance and one integrated data environment.

The challenges that respondents indicated in the area of BI were very different from each other; some indicated that unstructured data was a large challenge; others indicated that their largest challenge was to clean up legacy data warehouses.

All respondents agreed that the role of BI will only become more important in the future, although a lot of different reasons were stated. For example, some indicated that organisations will need to become 'smarter' and BI allows them to do so, others indicated that the amount of data will keep increasing and that BI is needed to help management with controlling and decision-making.

Figure 27 shows the results to the question *'How satisfied is your organisation about the current BI solution?'* Respondents were asked to indicate this for both the IT department and the users.

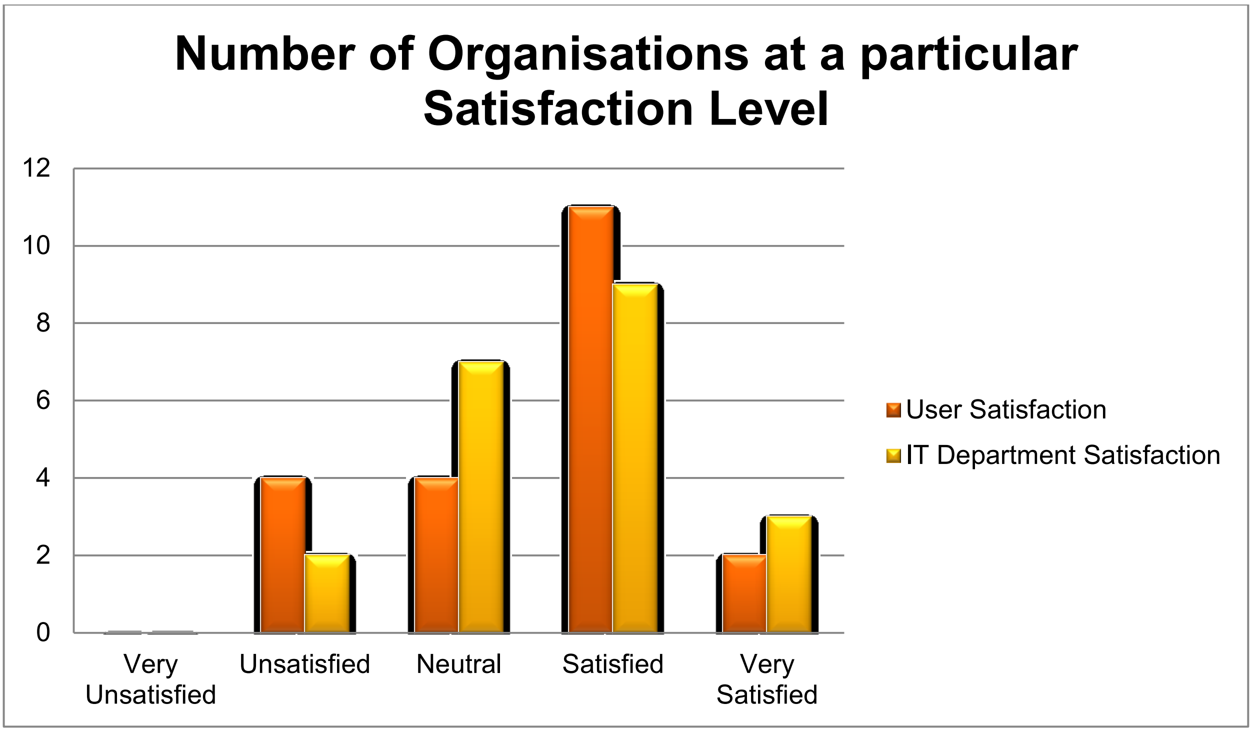



**Figure 27. Overview of the Satisfaction of the current BI Solution of the Respondents' Organisation.**

### *7.1.2. Discussion of Questionnaire Results*

In this section, we discuss the results of the questionnaire that we performed in order to validate our framework. Although the results come from a lot of different sectors and are spread out over the three segments, it is important to realise is that we base this discussion on 21 responses. The responses are relatively consistent, so while we do think that conclusions will not be very different when we would have received more responses, it would have given us more confidence in the results.

**General Remarks**
There are a couple of reasons why relatively many large organisations have participated in the questionnaire. First of all, it is more likely for those organisations to have a BI solution in place. Next to that, when we used Google and LinkedIn to find organisations with a BI solution, the larger ones are more likely to show up in the search results. Because of that, the number of approached organisations that can be categorised as large is also higher.

**Requirements**
We allowed the respondents to enter additional requirements that their organisation may have had when selecting the BI solution. It could be the case that we missed a couple of requirements so this way those requirements may still be identified. Some respondents indeed added requirements that were not included in our list. We decided to not include any of the additional requirements and we will explain our reasoning for that below. We do this for each of the additional requirements that the respondents indicated.

- User friendliness – Although this is a very important aspect for a BI solution, it is mainly influenced by the analytical tool that is selected by the organisation and less so by the BI architecture.
- Time saving – The respondent indicated that saving time in the creation of the report and having more time available to analyse played a role in the selection process. Although important, this aspect is also not really dependent on the architecture. We think that the influence the architecture can have on this requirement is already indicated by the requirement 'real-time data support'.
- Source system supplier – Another respondent indicated that the supplier of the source system was a factor that played a role in the selection process. This could indeed be the case, but we cannot include this

in our framework, since we did not look at suppliers. Next to that, the source system provider is different for different organisations, so there is no general advice to give.

- A data model in-line with the business – The respondent indicated that it was important for them to have a data model that is in-line with the business. That way, it is understandable for the business and also future proof. Again, we cannot give a score to an architecture on whether it is in-line with business, since every organisation is different. So although this could indeed be a factor, we cannot include it in our framework.
- Traceability – This additional requirement is also not really a factor that is implicit to a particular architecture. In other words, any architecture may or may not support traceability; it depends on the actual implementation.
- Low costs – The requirement 'few resources required' also includes the monetary aspect, so this factor is already included in our framework.
- Proven technology -> solution -> architecture – This could indeed be very relevant, but we also have the opinion that each of the architectures that we identified is mature enough to be implemented. This may be a different story for actual BI tools, but since we focus on architectures here, we do not include this in our framework.

These results show that it is important to interpret the architectural recommendation of the framework with care, as we said in chapter 6. The reason for this is that the framework can only take into account the requirements that can be influenced by the architecture, specifically layer two and layer three of the structure shown in figure 2. Needs that are mainly influenced by the analytical facility or requirements that are very specific to an organisation should be separately included in the selection process of a BI architecture.

As one can see in figure 18, the six requirements that were most often indicated to be an influencing factor for the selection of an architecture are:

1. High maintainability
2. Ability to deal with frequent updates
3. High performance of BI solution
4. Ability to deal with large amounts of data
5. High level of integration
6. High scalability

Table 8 shows that all six of these requirements are also rated as important, with average scores close to, or higher than 4 on the 5-point scale.

The requirement 'ability to deal with low data quality in source systems' is not indicated to have played a role in the selection process by any of the respondents. Still, data quality is one of the most important risks associated with BI, so we were surprised that none of the respondents indicated the importance of this requirement. Of course, the best way to deal with data quality issues is at the moment the data is entered in the source system. This, but also the fact that organisations may take this for granted or simply did not think about it could be possible explanations.

**Data Warehouse Architectures**

As we expected to see, the central data warehouse architecture and the hub-and-spoke architecture are the most often used architectures within the respondents' organisation. These are the architectures that offer the most integration, scale very well and have a high performance, which is more and more required by organisations nowadays. These are also the requirements in the top of most indicated requirements in our questionnaire.

The federated architecture is not used by any of the respondents' organisation. It does not score too well on the top six of most indicated requirements, so we think this is the reason why not many organisations choose this architecture. Next to that, it is often only used as temporary solution. We do expect to see this architecture more and more in the future, with the emergence of the in-memory data warehouse, which is gaining in popularity every day.

The independent data warehouse architecture is often chosen when low development time, low costs and low complexity are required. It can also be the result of historical organisational efforts to build a BI solution.

The data mart bus architecture with linked data marts is a good way to gradually integrate data sources (Kimball approach). This approach is easier compared to the hub-and-spoke architecture or centralised data warehouse architecture (Inmon approach). It is also a good architecture when there are legacy systems or databases that need to be integrated. Instead of one large project, it can exist of multiple small projects. We therefore expected to see this architecture more often. The literature [6] also suggested that this architecture is used a lot in practise. However, that article dates from 2006, so it could be the case that this architecture became less popular in the past eight years. It could also be the case that this architecture is more popular at smaller organisations or that due to the size of our questionnaire this architecture seems to be under represented while in reality it is not. Still, when implemented, the hub-and-spoke architecture and the central data warehouse architecture usually do perform better.

**Integration Technologies**

The results of the used integration technology were very striking. We did not expect that every organisation would use a form of ETL (ETL or RT-ETL). Some organisations used other integration technologies next to ETL, but only very few. We thought about possible reasons for this and also discussed these results with experts in the area of BI. Below, we describe some of the possible causes for this:

- Large BI solution providers may only offer ETL – In almost every case, the integration service is included in the product offering. Generally, ETL is more expensive than e.g. EDR. This means that even if an organisation would want an EDR solution, it should incur extra costs while already paying for a package that includes ETL. The expert we spoke to did not know any cases where an organisation would separately buy an integration service.
- Organisations may not have been aware of the different integration technologies – It is also possible that organisations did not specifically think about the integration technology or were not aware of the different options that are available.
- Organisations assume that every technology is able to integrate heterogeneous data sources – It is very possible that organisations have the perception that every BI architecture and thus every integration technique is able to integrate heterogeneous data sources and therefore do not indicate this as a requirement. EDR is often suggested by our framework as the best integration service for organisations, since only very few indicate that 'the ability to deal with low quality data' or transforming the data is important. We think that it may be assumed by organisations that this is standard, while it is not.
- As we will describe in the section where we discuss the interview with the insurance company, the ETL developers are often closer to the BI team. In order to minimise risks, the BI team may choose to work with people they know. Other integration technologies may not even be considered, although they may be a better alternative.
- Another cause that will be described in the section where we discuss the interview with the insurance company is that real-time data is often not required. Showing all the mutations in the BI solution would therefore only result in a higher load. If ETL is used, data is processed in batches which is perfect for the purpose of BI, unless real-time data is required.

These are all possible reasons (and there may be more) why ETL is used by every organisation, but none of these reasons invalidate our framework. We are under the impression that our framework does recommend the integration technology that should be best according to the literature, given an organisation's requirements.

Because of the above, we decided that in the rest of this validation, we would only look at the data warehouse architecture without looking at the integration service. In almost every case, the data warehouse architecture is the most important determinant of the total score. In other words, after filling in the framework, the total scores with the same data warehouse architecture are relatively close together, which makes it possible to analyse whether our framework recommends the right data warehouse architecture, since any integration service that may be chosen would still be used in combination with the same data warehouse architecture.

We filtered the results by the different data warehouse architectures to see if there were patterns to be found. Since we only had 21 responses, this was not possible for all the data warehouse architectures. We looked for patterns for the independent data mart architecture, the central data warehouse architecture and the hub-and-spoke architecture, but did not find anything striking. For example, it may have been the case that organisations with a particular architecture were more satisfied than others, but this was not the case. Maybe if more responses were recorded we would have been able to see patterns.

**Response Categories**
In order to see whether the results are in-line with our framework, we use the 'confusion matrix' that we introduced earlier. Of the twenty responses that we could use to validate our framework, eight (40.0%) indicated that the users were satisfied and their current data warehouse architecture was recommended by our framework. Five (25.0%) indicated that the users were unsatisfied and their current data warehouse architecture was not recommended by our framework. There were no responses (0.0%) in which the respondent indicated that the users were unsatisfied with a data warehouse architecture that was recommended by our framework. These results speak for the validity of our framework.

However, seven (35.0%) of the responses were not recommended by our framework while the respondent indicated that the users were satisfied with their current data warehouse architecture. This does draw our attention, since these results may give us suggestions on how to improve our framework. Figure 28 shows the 'confusion matrix' with the number (and percentage) of the organisations in each category.

| | User is Satisfied | User is Unsatisfied |
|---|---|---|
| Recommended by Framework | 8 (40.0%) | 0 (0.0%) |
| Not Recommended by Framework | 7 (35.0%) | 5 (25.0%) |

**Figure 28. Confusion Matrix Showing the actual Response Categories.**

As we said before, our architecture recommends organisations an architecture that, given their requirements, would best fit to their organisation. This does not mean, however, that the other architectures are by definition bad for their situation. The framework does not say anything about those architectures. So the seven cases where the users are satisfied with an architecture that was not recommended by our framework do not imply that our framework is incorrect. We looked at these seven cases in more detail, and it turned out that in four of these seven cases, the second best architecture according to our framework – given the requirements of the organisation – was their current architecture. This further supports our point in that the second best architecture may still very well fit the requirements of the users.

It would have been problematic for our framework if users were unsatisfied with the architecture that was recommended by our framework, since that would indicate that our framework would require improvement. We are happy that none of the responses fell in this category.

## 7.2. Interviews

In this section, we discuss the results of the three interviews we have performed. One of which was performed at a public transport company and another at an insurance company. Both organisations have a BI solution in place. The last interview we performed was at an organisation that is specialised in BI.

We discuss each of these interviews in a separate sub section below. To get an idea of the context in which the BI solution is placed and what its purpose is, we first give some background information for both the public transport company and the insurance company, then we discuss their BI architectures, the requirements they had when selecting their architecture, whether or not they are satisfied with their current BI solution and finally we took a look at whether these results support our framework or not. For the company specialised in BI, we also give some background information, after which we describe the most important results of the interview.

## *7.2.1. Public Transport Company*

### 7.2.1.1. Background Information

This was the first interview we performed. The interviewee has a lot of experience in the area of BI and has been involved in projects at a number of organisations. Fifteen years ago, he worked at BusinessObjects (BO) and was involved in some of the first BI projects. He saw a lot of the initial problems that have been written about in the literature.

As an example, the first project he was involved in ran an analytical tool directly on the source systems. This had a huge impact on the performance of those source systems and was unworkable. After that, he went to a different organisation that solved this problem by creating independent data marts. Here, the next problem arose, namely, that the data was stored redundantly and no 'single version of the truth' existed. Each department had its own data and when one department changed certain data, it remained unchanged at the other departments. In later projects, this was solved by creating integrated solutions.

He has been hired by the public transport company for ten years now. In the first few years he worked on a part-time basis and also did projects at other organisations. Currently, he is working full-time for the public transport company.

The public transport company uses the BI environment for all kinds of different purposes e.g. financial data reports, budgeting, smartcard (used by travellers) data analysis, but also for determining the amount of passengers using a particular line at a particular time in order to determine the resources required etc.

The reason that we interviewed this organisation is, because colleagues had contacts there and pointed out that this organisation does a lot on the area of BI and would be interesting talking to.

### 7.2.1.2. Architecture & Requirements

The architecture at the public transport company is graphically shown in figure 29. They make use of SAP BusinessObjects to create reports and dashboards. Below this is a meta layer, which integrates the two different data warehouses: Oracle and Microsoft SQL. SAP calls the meta layer 'Universe', which is a semantic layer that resides between an organisation's data warehouse and the end user, but more importantly, it is a business representation of your data warehouse or transactional database. It allows the user to interact with their data without having to know the complexities of their data warehouse or where the data resides. The universe is created using familiar business terminology to describe the business environment and allows the user to retrieve exactly that data that interests them [64].

This meta layer provides different users with views that are relevant to them. This meta layer thus provides 'spokes' in a virtual way. Although slightly different, since two data warehouses are used (as the 'hub') and the meta layer (as the virtual 'spokes'), one can categorise this as a hub-and-spoke architecture.

Both data warehouses use RT-ETL as integration service: the Microsoft SQL data warehouse uses SQL Server Integration Services (SSIS) and the Oracle data warehouse uses SAP BusinessObjects Data Services (BODS). While for most purposes real-time data is not required, it is supported by both of the RT-ETL tools. Currently, the public transport company only uses real-time data for financial information and the budgeting process.

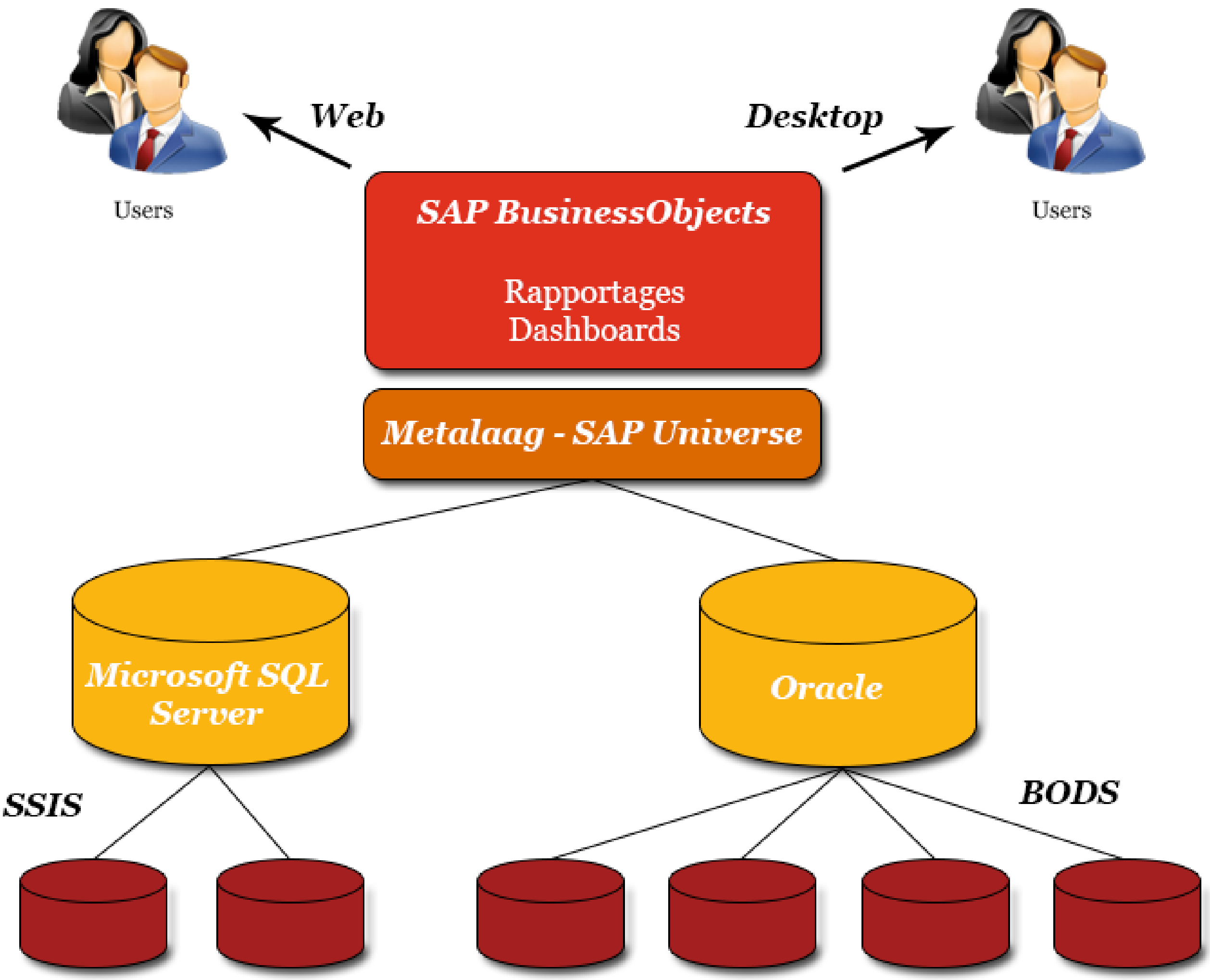



**Figure 29. BI Architecture of the Public transport company.**

The reason that two data warehouses with its own integration services are used is due to the introduction of the smartcard (a new system that travellers use to pay for the services of the public transport company). To incorporate the data of the external system containing this data, an additional server was required. Since the external system was built using SQL, the SQL data warehouse with SSIS integration service was the better (cheaper) choice.

The public transport company is satisfied with their current BI environment. The interviewee would place the public transport company at the top level of the pyramid that shows the levels of BI (see figure 23). In the holding they are part of, they are far ahead on the topic of BI.

A very interesting thing he said was the following:

*'In practise, a lot of organisations choose the BI solution that fits best to their current information requirements. It is not so much an architectural consideration that is made, but more an ad-hoc decision. Most organisations nowadays already have a BI solution in place, so usually they continue to build on existing solutions.'*

This was also the case for the public transport company; it was not so much an architectural decision. This means that there were no particular requirements that influenced the architectural choice. The choice was made for a tool, namely SAP BO *'since that was the best tool available at that time and an Oracle data warehouse* *was chosen since that combination works well'*. This means that we cannot validate our framework using this interview, since it was no architectural decision. Still, it does give us valuable information.

After showing the interviewee our framework, he made the following comment:

*'For an organisation that wants to start with BI or when a new IT manager comes in who decides to start from scratch, this framework could be very useful.'*

He did like the approach we took with our framework and could find himself in the requirements and architectures we identified. One thing he did point out is that in practise, hybrid forms are often present. So with a little effort one could place a particular architecture in one of the architectures identified in the literature, but practise is almost never exactly the same. However, he also said that it was inevitable to not use such standard architectures in order to create a framework, so concluding, according to him the framework does have potential.

The public transport company had no large investment plans in the area of BI at the moment, although there will be a migration of BO to a newer version with more functionalities in the near future. When we asked whether he would choose the same architecture if he was able to start from scratch he said the following:

*'Our current BI architecture has some disadvantages, but also a lot of advantages. The main disadvantage is that we have two data warehouse providers and two ETL tools. The main advantages are that the ETL tools are very stable and that we are very satisfied with both data warehouses. So if I could choose a new architecture I would still go for SAP BO as BI tool, since that is still one of the best tools (if not the best) in the market and we are very satisfied with it. IBM Cognos might perform a bit better, but BO is much better when looking at functionality. I would probably go for only one data warehouse and one ETL tool: When looking at costs we would go for Microsoft, when looking at functionality and user-friendliness we would go for Oracle.'*

### 7.2.2. *Insurance Company*

#### 7.2.2.1. Background Information

This interview was performed with the team leader of the business intelligence architects of the insurance company. The interviewee is working at the insurance company for four years now. The insurance company is an organisation with a lot of different brands. One could say that it is an organisation of mergers. Three years ago, each of those had its own BI team, but there is now one central team that is responsible for the entire BI environment within the insurance company.

The reason that we wanted to interview this organisation specifically is because of the fact that the interviewee responded to our survey and indicated that all requirements and all architectures were present in his organisation. We were interested in, inter alia, why this is the case, how they deal with it, what the requirements were for the different architectures and what architecture the interviewee would select when he could start from a 'greenfield' situation.

#### 7.2.2.2. Architecture & Requirements

The interviewee indicated in his response to the survey that the insurance company has 54 data warehouses, all architectures and all requirements. When we asked him why this was the case, he said that this was mainly due to the fact that the organisation exists of a lot of mergers. Each of those – previously independent organisations –had its own IT environment and its own data warehouse. Since the data is stored differently in those data warehouses, one cannot simply merge those environments. Besides that, each – previously separate – organisation was not very willing to give up its own data warehouse.

Efforts were made to integrate the different data warehouses, but these were not very successful. There are, for example, data warehouses that in turn integrate data warehouses, sometimes up to a stack of four. So although some integration is realised, it is still far from optimal. In the coming years, the goal is to reduce the number of legacy data warehouses.

The interviewee agreed that it would be beneficial to integrate these different sources. For example, currently, customers who want to change their personal information, e.g. a change of address, are required to do so for every brand, since the data is stored redundantly (for each brand). By integrating these different data sources, one call would be enough to change the data enterprise-wide.

We also asked the interviewee whether integrating all the knowledge they have of their customers (of all the different brands) would give them a competitive edge. He answered that, by law, some connections are not allowed to be made. Still, he agreed that there are cases in which it is required to integrate the data, for example for the financial department.

It was hard for the interviewee to say which requirements played a role in the selection of a particular BI architecture, since most architectures were already chosen before the organisation became part of the insurance company. What he did say is that the trade-off between the requirements 'low development time' and 'high level of integration' was probably most important for the choice of an architecture. Some organisations valued a solution that could quickly be realised, but this was often at the cost of the level of integration. Others would value a high level of integration, but that would often be at the cost of a higher development time.

While the independent data mart architecture and the data mart bus architecture with linked dimensional data marts usually have a lower development time, the hub-and-spoke architecture and the central data warehouse architecture provide a higher level of integration. The organisations made different choices, which is probably the cause of such a variety of systems.

We also asked the interviewee why he thought that all organisations that responded to the survey were using ETL (some used another integration technology next to it, but all used ETL). He answered that, for the insurance company, the reason that ETL was used is that the people who knew a lot of ETL were closer to the business intelligence team. When implementing a new BI solution, you want to minimise the risks involved. This means that you try to involve the people that are closer to you, since there is more of a common understanding. So even though another integration technology may be more suited, it is not considered since that would bring additional risks. He also indicated that at the insurance company, no real-time data was required. It is therefore not necessary to see every mutation in the BI solution since that would only result in a higher load. ETL transfers the data in batches and is therefore very well suited for the purpose of BI.

When we showed him the framework, he more or less said the same as the BI specialist of the public transport company; the framework could be very useful for organisations that start from a 'greenfield' situation. However, for organisations like the insurance company who already have numerous BI solutions in place it is much more interesting to know how to integrate these solutions in a good way.

Next to that, he also said that the scores associated with the federated architecture should be higher in the case of an in-memory data warehouse, but that in-memory can be used with other architectures as well, so it may be hard to include this in the framework. We agree that with the introduction of the in-memory data warehouse, the landscape of BI solutions may change in the coming years. We expect that each of the architectures may be able to perform better, but at a higher cost. Since its influence is still not entirely clear (in practise but also in the literature), we decided to not explicitly include the in-memory data warehouse in our framework. Nevertheless, it would be a very good idea to also take a look at in-memory data warehouses when selecting a BI architecture.

The other scores for the requirements that we assigned to the different architectures seem to be in-line with his experience. In his answers, the interviewee often gave examples of architectures that would be better when a particular requirement is important and for those architectures indeed had higher scores for those requirements.

When we asked what architecture the interviewee would select when he could start from a 'greenfield' situation, he said that he would probably choose the hub-and-spoke architecture. The most important reason for this is that the integration is really important, but the different brands would also still require their own view (therefore the central data warehouse is less suited). The independent data mart architecture is undesirable, since its largest drawback – integration – is something that is a large problem now. The data mart bus architecture with linked dimensional data marts and its gradual implementation sounds nice, but the interviewee expected that it would be really hard to enforce the same dimensions. He also said that they were currently looking into an in-memory data warehouse solution.

## *7.2.3. Company specialised in Business Intelligence*

### 7.2.3.1. Background Information

PwC recently acquired a company specialised in business intelligence. This gave us the connection to come in to contact with a specialist in the area of BI. The interviewee's company helps organisations with their BI strategy and BI implementation, looking at both the business side and the IT side. We spoke with someone from their BI strategy team, which tasks include helping organisations with selecting the right BI tool. This person was thus very familiar with the topic of this master thesis.

By interviewing this specialist, we would be able to get another perspective; in addition to that from the literature and from the organisations owning a BI tool.

### 7.2.3.2. Thoughts & Suggestions

We presented our research and asked for the interviewee's thoughts and suggestions, since he has a lot of experience in this area. In this subsection we will summarise the most important results of this interview.

The interviewee pointed out that he really liked the approach taken. He pointed out that it was a very 'scientific way' of approaching the problem. First defining a general structure, then pointing out which parts are within the scope of the thesis. Next, describe for each of those parts that are within scope the different options that are available and then making 'combinations' and explaining why some combinations are not used. A similar approach is taken for the requirements, by filtering out the relevant requirements from the literature. These combinations and requirements together make up the framework. He saw the framework as a helpful tool for organisations that want to implement a BI solution.

The interviewee made several very useful comments when he overlooked the complete research. First of all, he pointed out that we may want to indicate for which organisations in particular, the framework would be useful. He explained it using the three types of situations that he saw at customers: 1) 'green field', 2) merger and 3) chaos. The organisations with one of these three situations are often the ones that request help to his company.

The 'green field' situation is a situation in which no business intelligence solution is in place. It can thus be built from scratch. These organisations often come to the interviewee's company and ask them to help with their BI strategy, selection and/or implementation of a BI solution.

The merger situation is a situation in which two companies with different BI environments are merged and require an integrated solution.

Last, which is probably also the largest group, is the chaos situation. This is a situation in which all kind of solutions and tools are in place in order to help with decision-making and reporting. This situation is often the result of historical organisational efforts to BI.

According to the interviewee, every organisation would love the idea of a hub-and-spoke architecture or a central data warehouse architecture. However, in the chaos situation an organisation would often rather stick with its current situation. The reason for this is that – although they see the potential – often a large investment and/or large amount of time is required to migrate to a completely new solution. This means that even if a business case is created which shows the benefits achieved by a new BI solution, the costs need to be incurred first, while the benefits may only be noticeable in a couple of years. See figure 30 for the three different situations.

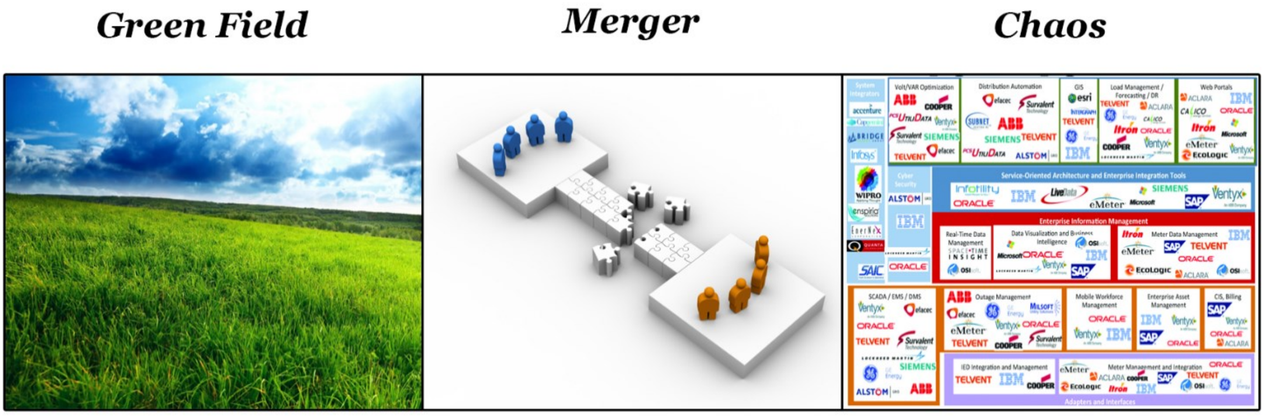



**Figure 30. Three Customer Situations.**

Therefore, he says the following: *'although I do believe the framework could be very useful in a green field situation and also – though to a lesser degree – in a merger situation, since people know that something should happen (somehow the two BI environments should be integrated), it is less suited in a chaos situation. Not because the framework is not useful or that organisations do not see the benefits of an architecture, but because of the fact that it is unlikely that an organisation would completely start over with their BI environment, since that would be an enormous project. So although it sounds really nice in theory, in practise it is not realistic in a chaos situation.'*

The interviewee did not know for sure whether we covered all requirements. He gave an example of a meeting he had with a customer, where only three departments of that customer were involved, but it resulted in a very long list of requirements. However, it is important to note that this also included the front-end, so requirements such as usability, ability to print, nice representation etc. were also listed. This is indeed something we should pay attention to. By trying to be as complete as possible with the requirements from the literature and allowing respondents of the questionnaire to include their own requirements, we agreed that this issue would largely be resolved.

We discussed some of the answers that were given in the other interviews. In particular the response about the situation at the public transport company, where the choice for the current BI solution was not so much an architectural decision, but more an ad-hoc decision where the current BI solution was chosen to fit to the current information requirements. In his response, the interviewee drew the picture shown in figure 31.

He meant the following with this picture: At a lot of organisations where he had been involved, the business intelligence competence centre (BICC) usually reports to the chief executive officer (CEO) or chief financial officer (CFO) instead of the chief information officer (CIO). This reporting structure could be the cause of choosing a BI solution without paying much attention to the architecture, since the choice for a particular BI solution may be primarily based on what the business requires (usually the CEO's or CFO's responsibility), and not so much on what architecture is best in that situation (usually CIO's responsibility).

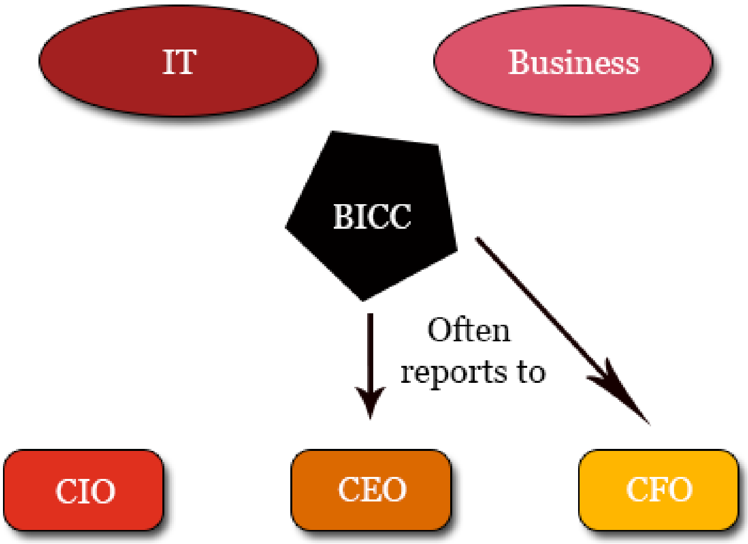



**Figure 31. Frequent Reporting Structure within an Organisation.**

We also discussed the questionnaire results; especially the high number of respondents that indicated that ETL was used within their organisation and that scalability and maintainability were often named as requirements that played a role in the selection of the current BI architecture. The interviewee answered that it is important to keep in mind who is filling in the questionnaire. For example, when an IT manager is asked to indicate what requirements were most important when selecting a particular BI architecture, more technical elements like scalability and maintainability may be named. If the same question is asked to top management (not the IT manager) requirements that are closer to the business may be given. Since most respondents that we asked are from the IT department due to the technical nature of the questions, this may be a reason why these aspects were so often named.

When we asked why ETL was the integration service at so many of the organisations who participated in the interview the interviewee said the following: *'I have the same perception in that almost all of my customers use ETL. I was under the impression that ETL was the technique for BI, but what I understand from your research is that the literature also discusses other techniques'*. We asked whether it might be the case that large BI solution providers may only offer ETL as integration service and that this may be the reason why so many organisations use ETL. The interviewee answered that this may be the case, but that he was not sure. He never saw large BI software packages with a custom integration service.

ETL particularly stands out on two areas, namely its ability to deal with very large amounts of data and next to that its ability to transform and cleanse data, which allows it to integrate heterogeneous sources of data. We asked the interviewee whether the respondents may take this for granted when thinking about a BI solution and do not specify this as a requirement that is important. When they do, it is not surprising that tools like EDR stand out.

The interviewee answered that this may be the case, since data requirements to be in a similar format in order to perform analysis over it. When two heterogeneous sources of data are integrated, there would be only very few cases in which EDR would be applicable – when the data is already stored in the same format, or when only very minor changes need to be made in order to make the data homogeneous.

# 8. Limitations

There are some limitations to the research we have performed and it is important to keep these limitations in mind when using the results of this research:

1. Some subjectivity is involved in assigning scores to the different architectures.
2. The systematic literature review does not cover all literature and there is always some individual bias involved.
3. We may have missed some important requirements.
4. We assumed that all source systems can be used in combination with every integration service, but in reality this is not always the case.
5. We did not discuss the different BI tools in the fourth layer.
6. We identified possible reasons that explain why all respondents indicated that ETL is used in their organisation, but we cannot say with certainty that these are indeed the causes.
7. The number of responses to the questionnaire is not enough to generalise and more responses are required to come with convincing evidence that our framework is indeed correct.
8. The way the validation is done does not give 100% certainty that our framework is indeed correct.

First of all, as we said before the scores we assigned in the framework to the different architectures are based on the literature, but some subjectivity is involved. If a particular architecture is good at providing a particular requirement, one could argue whether a score of 4 or 5 should be assigned. The same is the case for an architecture that is not as well suited at providing a particular requirement. One could argue whether a score of 1 or 2 should be assigned. We could have reduced this subjectivity by assigning scores on a 3-point scale, but that would not allow us to differentiate some architectures from each other, while there are certainly differences to be found. We asked some experts to take a look at the scores that we assigned to the different architectures, but still the scores are not entirely objective.

Secondly, the systematic literature review that we performed does not cover all literature. Due to the fact that we used a select combination of search engines and keywords, we may have missed some relevant results. Next to that, by reading and evaluating literature, individual bias can never be completely excluded.

The third limitation is that we may have missed certain requirements that are important but that were not included in our framework. By allowing respondents to indicate additional requirements we mitigate this for the most part.

The fourth limitation is that we assumed that every source system can be used in combination with every integration service, but in reality, there may exist some systems which are unable to cooperate with certain integration services. However, it would have been impossible to discuss every combination of source system and integration technology in the time available.

The fifth limitation is that we did not discuss the different BI tools in the fourth layer. Instead, we only categorised these for scoping reasons. The choice for a particular BI tool could, however, influence the choice in other layers and thus indirectly the architecture itself. We do have the opinion that we covered the most important aspects that could influence the architecture in our list of requirements.

The sixth limitation is that the results of the questionnaire and interviews indicate that almost all (if not all) organisations use ETL as integration tool. We identified a couple of possible reasons why this may be the case, but we cannot say with certainty that these reasons indeed identify the cause. What we can say is that, given the identified requirements, we think that our framework does recommend the right integration technology according to the literature.

The seventh limitation is that the number of responses to the questionnaire was not high enough to generalise to the entire population of organisations with a BI solution. The results do give an indication of the validity of our framework, but more research should be done in order to come with convincing evidence.

The final limitation is that the best way to validate our framework would be by being involved in the selection process of a BI solution in a number of organisations and then implement all architectures. If the users are then satisfied the most with the architecture recommended by the framework, it would indicate that our framework is correct. This is an unrealistic approach, however.

Another more realistic way of validating the framework is by being involved in the selection process of a BI solution in a number of organisations and selecting the recommended BI architecture indicated by the framework after the requirements are identified and filled in. When a large part of these organisations is satisfied with the chosen architecture, it would provide strong evidence that our framework is valid.

However, the selection process and implementation of a BI solution could span multiple years and a large number of organisations would be required to be willing to participate and trust the framework. This would require a lot of time, so therefore we decided to use the approach we have taken, which is actually exactly the other way around. We asked organisations that already went through the decision process and asked what their requirements were, what their current architecture is and whether they are satisfied. Although not optimal, this approach is executable and provides some indication about the validity of the framework.

# 9. *Conclusion, Contribution and Further Research*

This chapter forms the conclusion of the performed research, explains the contribution that we made to both literature and practise and suggests topics for further research. The main research question of this thesis was: *Which BI architecture fits best to organisation's requirements?*

In order to answer this question we formulated five sub questions:

1. What is business intelligence?
2. Which BI architectures exist?
3. What are the differences between these architectures?
4. What are the requirements that influence the choice for a particular BI architecture?
5. What does the framework that maps the different BI architectures to the requirements look like?

We now discuss each of these sub questions in turn, after which we draw our conclusion on the main research question. After that, we explain our contribution to the literature and to practise and suggest a number of topics for future research.

## 9.1. *Conclusion*

**1. What is business intelligence?**

In the literature, a lot of different definitions of BI exist. Where some authors approach the term from a technical perspective, others take a more organisational view. We analysed the different definitions and chose one that best fits to our perception of BI:

*"BI is a general term for applications, platforms, tools, and technologies that support the process of exploring business data, data relationships, and trends. BI applications provide companies with the means to gather and analyse data that facilitates reporting, querying, and decision making."*

We also looked at the relevance, benefits, costs and risks associated with BI. Both the literature and the responses to the questionnaire indicate that BI will only become more important in the future. Business environments change in an increasingly faster pace and more and more data becomes available to use to your advantage. BI can help organisations in these fast changing environments by providing first-rate business information and knowledge. Next to that, BI also helps in dealing with these large amounts of data.

Benefits of BI include better quality information, better observation of threats and opportunities, growth of the knowledge-base, increased information sharing, improved efficiency, easier information acquisition and analysis, faster decision-making, time- and cost-savings. Costs include hardware costs, software costs, implementation costs and personnel costs.

The most important risk associated with BI is the data quality in source systems. The output of the BI tool can only be as good as the data that is used to generate this output data. In case the data quality is bad and managers base their decisions entirely on the information they get from the BI tool, it can be very harmful for an organisation.

**2. Which BI architectures exist?**

Using the literature we identified a general structure of a BI solution, consisting of four layers: 1) the data source systems, 2) the integration services, 3) the data warehouse architectures and 4) the analytical facilities. We chose to focus on the second and third layers. The reason for this is that the source systems are often already in place, so not much can be changed there and the fourth layer could be the subject of an entire master thesis itself, so we decided to scope this out. Layers 2 and 3 are the layers that affect the architecture the most.

We listed the different options that are available for each layer. In the integration services layer we distinguish approaches from technologies. The approaches we identified are: 1) data consolidation, 2) data federation, 3) data propagation, 4) data access, 5) data virtualisation, 6) data mash ups and 7) a hybrid approach. The integration technologies are 1) EAI, 2) ETL, 3) RT-ETL, 4) EII, 5) EDR, 6) ECM, 7) SOA and 8) ESB which we categorised in the integration approaches.

We also identified five data warehouse architectures: 1) independent data mart architecture, 2) data mart architecture with linked data marts, 3) hub-and-spoke architecture, 4) central data warehouse architecture and 5) federated architecture.

We made realistic/likely combinations of these integration technologies and data warehouse architectures and these combinations form what we consider BI architectures.

### 3. What are the differences between these architectures?

The next step was to identify the differences between these architectures. To do this, we used the literature to identify a list of characteristics on which BI architectures can differ from each other. We made a mapping of these characteristics to the architectures. We assigned scores on how well each architecture performs on each characteristic using the information from the literature.

### 4. What are the requirements that influence the choice for a particular BI architecture?

In order to develop a framework which maps the different architectures to requirements, we needed to develop a list with requirements that are relevant for the choice of a particular architecture. We did this by using the literature. Below, the list of requirements is shown.

1. High information exchange between departments
2. Low development time
3. Few resources required
4. Low competence required of in-house staff
5. Ability to deal with low data quality in source systems
6. Need for real-time data
7. Ability to deal with large amounts of data
8. Ability to deal with frequent changes and updates
9. High scalability
10. High maintainability
11. Low difficulty/complexity BI solution
12. Low impact on performance source systems
13. High performance of BI solution
14. High level of integration
15. High customisability

### 5. What does the framework that maps the different BI architectures to the requirements look like?

The last step was to make the connection between the requirements and the BI architectures. We developed a framework that maps these together. The framework recommends an architecture given the requirements of a particular organisation. Although the framework does not claim that the other architectures cannot satisfy the users, the recommended architecture should be the best option given the requirements.

We validated the developed framework by distributing a questionnaire to organisations that have a BI solution in place and by performing a number of interviews. The results support our framework, but additional research is required for convincing evidence.

**Which BI architecture fits best to organisation's requirements?**

So to answer the main research question, *Which BI architecture fits best to organisation's requirements?* we refer to the developed framework. In this framework, organisations can indicate which requirements are important to them, but also how important on a 5-point scale. The framework then recommends the architecture that is best – given those requirements.

With the introduction of in-memory data warehouses, we expect that architectures will be able to perform better, but at a higher cost. We did not include this in our framework since the influence of the in-memory data warehouse is still not entirely clear in both practise and in the literature, but it is important to realise that this is an emerging technology that may change the landscape in a couple of years.

Still, it is important to realise that there may be organisational specific, unique requirements that cannot be included in the framework (for example, a particular requirement may be that important that an architecture scoring 5 on that requirement has to be selected, so even though the total score of another architecture may be higher – which will be recommended by the framework – a different architecture should be chosen). Therefore, the recommended architecture – given an organisation's requirements – should be adjusted to possible organisational specific, unique requirements that are not included in the framework.

## 9.2. Contribution

### 9.2.1. Contribution to Literature

In this thesis, we made several contributions to the literature. These contributions are discussed below:

1. An analysis of the many definitions of business intelligence in the literature.
2. A summary of the literature on the relevance, benefits, costs and risks of BI.
3. An overview of the different integration approaches, integration technologies and data warehouse architectures.
4. The development of what we call BI architectures that exist of combinations of integration services and data warehouse architectures.
5. The development of a list of characteristics to indicate the differences between the BI architectures.
6. The development of a mapping that scores the different BI architectures on the different characteristics.
7. The development of a list of requirements that can influence the choice for a particular BI architecture.
8. The development of a framework that maps requirements to the BI architectures and that is able to recommend an architecture.

We analysed the many different definitions of business intelligence that are used in the literature and explain why we think the definition of Raisinghani [60] is a very good one. Next to that, we provided a summary of the literature on BI by pointing out the relevance, benefits, costs and risks associated with BI.

We then used the already existing literature to identify a common 4-layer structure that is seen in most BI solutions, after which we gave a comprehensive overview and explanation of the different integration services (split up in integration approaches and integration technologies) and data warehouse architectures available. We then made combinations that are likely to be seen in practise, which form what we call BI architectures. In existing literature, there is written about integration services and also about data warehouse architectures, but not about a combination of the two. In practise these two layers are always used in conjunction, so it could be very beneficial to look at the combination. In this thesis we have done this.

We developed a list of characteristics using the literature which we then used to point out the differences between the different BI architectures using a mapping. After that, we developed a list of requirements that may influence the choice for a particular architecture, again using the literature.

Last, a framework was developed which maps the different BI architectures to the different requirements and which recommends a BI architecture that – given an organisation's requirements – would suit best.

### 9.2.2. Contribution to Practise

For practise, this thesis also made a number of contributions that are discussed below:

1. A comprehensive overview of what BI is, its relevance, benefits, costs and risks.
2. A comprehensive overview of the different integration approaches, integration technologies and data warehouse architectures that are available.
3. A comprehensive overview of the BI architectures and its differences.
4. A framework that helps organisations selecting a BI architecture that best fits their requirements.
5. An overview of the results of the performed interviews and questionnaire which allows organisations to compare their situation with that of others.

The main goal of this thesis is to help practitioners in the selection process of a BI architecture by providing them a framework which recommends an architecture that – given their requirements – would suit best. However, practitioners can also use this thesis to quickly see what BI exactly is, its relevance, benefits, costs and risks. Next to that, it helps them understand what different integration services and data warehouse architectures exist and how they differ from each other. Last, they can also use the results of the interviews and the questionnaire to see what their relative position is on the topic of BI.

## 9.3. Further Research

The past few months we have had the opportunity to perform this research on the topic of business intelligence and architecture. During our research we came across several subjects for further research. First of all, the developed framework should be tested more thoroughly. This, to determine whether organisations can really rely on it or not.

Secondly, as we stated before, with the introduction of in-memory data warehouses, the environment of BI architecture may change in the future. Although it comes at a higher cost, the performance is expected to be a lot better. Since this area is still in its infancy, it would be interesting to perform research in this area. Case studies of implementations of in-memory BI solutions would be very interesting to see. One could also compare in-memory implementations with other implementations.

Another area for future research is what we called the 'fourth layer', the analytical facility. There are a lot of different providers and tools available and when is a particular tool better to select than another? This would require a listing of the different tools and the differences between them. One could, for example, set the scope to the tools that are listed in the Gartner Magic Quadrant. Although Gartner describes all the tools in short, an overview of the exact differences would be valuable.

Lastly, Knabke and Olbrich [44] describe that due to the turbulent environments of today, it is crucial for organisations to draw increasingly faster conclusions out of changing circumstances. This requires information systems to be more agile, including the BI system. They define criteria of agility for a BI system. It would be interesting to see if these criteria are different for the different BI architectures.

# A. Appendix – Lists

## A.1. List of Abbreviations

| | |
|---|---|
| BI | Business Intelligence |
| BICC | Business Intelligence Competence Centre |
| BO | BusinessObjects |
| BODS | BusinessObjects Data Services |
| CEO | Chief Executive Officer |
| CFO | Chief Financial Officer |
| CIF | Corporate Information Factory |
| CIO | Chief Information Officer |
| CRM | Customer Relationship Management |
| DB | Database |
| DBMS | Database Management System |
| DM | Data Mart |
| DQ | Data Quality |
| DW | Data Warehouse |
| DSS | Decision Support System |
| EAI | Enterprise Application Integration |
| ECM | Enterprise Content Management |
| EDR | Enterprise Data Replication |
| EDW | Enterprise Data Warehouse |
| ESQL | Embedded Structured Query Language |
| EII | Enterprise Information Integration |
| EIS | Executive Information System |
| ELT | Extract, Load, Transform |
| ETLT | Extract, Transform, Load, Transform |
| ES | Expert System |
| ESB | Enterprise Service Bus |
| ETL | Extract, Transform, Load |
| ERP | Enterprise Resource Planning |
| FDBS | Federated Database System |
| FTP | File Transfer Protocol |
| HOLAP | Hybrid Online Analytical Processing |
| HRM | Human Resource Management |
| KPI | Key Performance Indicator |
| MIS | Management Information System |
| MOLAP | Multidimensional Online Analytical Processing |
| ODS | Operational Data Store |
| OLAP | Online Analytical Processing |
| OLTP | Online Transaction Processing |
| PC/PS | Private Companies/Public Sector |
| RDBMS | Relational Database Management System |
| ROLAP | Relational Online Analytical Processing |
| RT-ETL | Right-time Extract, Transform, Load |
| SCM | Supply Chain Management |
| SLR | Systematic Literature Review |
| SOA | Service-oriented Architecture |
| SQL | Structured Query Language |
| SSIS | SQL Server Integration Services |
| XML | Extensible Mark-up Language |
| XSLT | Extensible Stylesheet Language Transformations |

## A.2. List of Tables



## A.3. List of Figures

# B. Appendix – Systematic Literature Review

| | | Global Knowledge Gateway | | | Google Scholar | | | Scopus | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | |
| **Keywords** | | **Hits** | **Date** | **Relevant** | **Hits** | **Date** | **Relevant** | **Hits** | **Date** | **Relevant** |
| "business intelligence" and "data quality" | | 423 | 10-3-2014 | 7 | 147 | 10-3-2014 | 1 | 78 | 9-5-2014 | 7 |
| "business intelligence architecture" | | 11 | 19-3-2014 | 2 | 361 | 19-3-2014 | 7 | 21 | 9-5-2014 | 0 |
| "role of bi capabilities" | | 0 | 19-3-2014 | 0 | 23 | 17-3-2014 | 5(3) | 2 | 9-5-2014 | 2 |
| advantages "business intelligence" | | 331 | 19-3-2014 | 0 | 31100 | 17-3-2014 | 3 | 258 | 9-5-2014 | 0 |
| "business intelligence framework" | | 6 | 19-3-2014 | 0 | 342 | 19-3-2014 | 5 | 18 | 9-5-2014 | 1 |
| "data warehouse architecture" | | 11 | 20-3-2014 | 1 | 1960 | 20-3-2014 | 8 | 97 | 9-5-2014 | 7 |
| "business intelligence" AND "data consolidation" AND integration | | 42 | 21-3-2014 | 1 | 354 | 21-3-2014 | 4 | 0 | 9-5-2014 | 0 |
| "enterprise application integration" | | 120 | 24-3-2014 | 1 | 15700 | 24-3-2014 | 4 | 770 | 9-5-2014 | 1 |
| "extract transform load" | | 73 | 24-3-2014 | 3 | 3490 | 24-3-2014 | 4 | 116 | 9-5-2014 | 3 |
| "enterprise information integration" | | 11 | 24-3-2014 | 1 | 2080 | 24-3-2014 | 4 | 115 | 9-5-2014 | 0 |
| "enterprise data replication" | | 1 | 24-3-2014 | 1 | 52 | 24-3-2014 | 1 | 1 | 9-5-2014 | 0 |
| "enterprise content management" | | 394 | 24-3-2014 | 1 | 4600 | 24-3-2014 | 0 | 161 | 9-5-2014 | 3 |
| "service-oriented architecture" | | 180 | 24-3-2014 | 1 | 69300 | 24-3-2014 | 9 | 10783 | 9-5-2014 | 0 |
| "enterprise service bus" | | 44 | 24-3-2014 | 1 | 8360 | 24-3-2014 | 5 | 456 | 9-5-2014 | 2 |
| "independent data marts" | | 1 | 25-3-2014 | 0 | 227 | 25-3-2014 | 7 | 5 | 9-5-2014 | 3 |
| "data mart bus architecture with linked dimensional data marts" | | 0 | 26-3-2014 | 0 | 14 | 26-3-2014 | 8 | 0 | 9-5-2014 | 0 |
| "centralised data warehouse architecture" | | 0 | 26-3-2014 | 0 | 25 | 26-3-2014 | 8 | 0 | 9-5-2014 | 0 |
| "federated data warehouse architecture" | | 3 | 1-4-2014 | 0 | 39 | 1-4-2014 | 7 | 1 | 9-5-2014 | 1 |
| relational database vs multidimensional database | | 24 | 7-5-2014 | 1 | 76000 | 7-5-2014 | 6 | 2 | 9-5-2014 | 0 |
| "in-memory database" | | 3 | 7-5-2014 | 0 | 2270 | 7-5-2014 | 1 | 100 | 9-5-2014 | 3 |
| (fit OR needs OR requirements) AND "business intelligence" | | 1193 | 15-5-2014 | 5 | 48600 | 15-5-2014 | 20 | 885 | 23-5-2014 | 0 |

# C. Appendix – Difference EAI Architectures

| **Evaluation Parameter** | **Hub-and-Spoke Architecture** | **Bus Architecture** | |
|---|---|---|---|
| | | **Proprietary bus based product suit** | **ESB** |
| Installation effort | Less installation effort compared to solutions with bus architecture. | Moderate effort | Moderate effort |
| Administration | Easy to maintain and administrate because of central hub. | Administration may be complex depending upon the integrated systems. | Administration may be complex depending upon the integrated systems |
| Cost | High | High | Low cost because it does not use proprietary formats to enhance performance. Also it does not provide all the services usually provided by proprietary product suits. |
| Scalability | High if federated architecture is used otherwise limited by the hardware of box used to host hub | Highly scalable | Highly scalable |
| Standards | Mostly standard based but may use proprietary internal formats | Mostly standard based but may use proprietary internal formats | Standard based |
| SOA | Can be implemented as service oriented | Can be implemented as service oriented | Service oriented |

***Source: Goel [28]***

# D. Appendix – Comparison Data Integration Technologies

| | ETL | EDR | EAI | ECM | EII | ESB |
|---|---|---|---|---|---|---|
| **When to use each technology** | • Data consolidation<br>• Complex transformations | • One-to-one transfer from one source machine to one target machine<br>• To replicate entire dataset or just the • bytes that have changed | • Integration of transactions and not large data sets<br>• Questions can be answered by joining small amounts of data<br>• Data sources repositories cannot be • directly accessed | • Management of structured as well as unstructured content from different sources<br>• Dynamically accumulating, arranging and refreshing information as the needs and requirements of the enterprise change. | • Usually connecting a large repository with selected data from other sources<br>• Selectively as a tool to extend existing well designed Enterprise Data Warehouse (EDW)<br>• May be indicated when source data:<br>o Volatility is high<br>o Selectivity is granular<br>o Connectivity is reliable<br>o Service levels are compatible<br>o Transformations are minimal and can be expressed as SQL | • Basic Infrastructure to facilitate SOA<br>• Acts as a shared messaging layer for connecting applications and other services throughout an enterprise computing infrastructure<br>• Complex data transformation |
| **When not to use each technology** | • Real-time data<br>• Movement of every data element from operations<br>• Highly dynamic reports<br>• Accessing or publishing XML data<br>• Reason: Designed to build data • warehouses / marts, operate in batch | • Real-time data<br>• Highly dynamic reports<br>• Reason: Designed to replicate data • between source and target machines | • Reporting and analytics<br>• Working with large data sets or aggregated data<br>• Reason: Single-record, push-based • architecture is not designed for pulling result sets from queries but for sending messages | • Building a DW or ODS<br>• Volume data cleansing<br>• Reason: Designed to manage content from • different sources so as to use it meaningfully | • Building a DW or ODS<br>• Volume data cleansing<br>• Time series or data mining analysis<br>• Reason: Designed to deliver requested data • on-demand; does not keep long-term copies of data | • Building a DW or ODS<br>• Massive volumes of data<br>• Time series or data mining analysis<br>• Reason: Designed to provide a service-oriented, high-performance, standards-based business integration infrastructure |
| **Data flow** | Unidirectional –from source to target | Unidirectional –from source to target | Bidirectional | Unidirectional | Bidirectional | Bidirectional |
| **Data movement** | Scheduled –batch<br>Process managed | Scheduled or Transaction triggered | Transaction triggered—asynchronous<br>Transaction managed | Scheduled | Query time<br>Query (SQL) managed | Message-driven |
| **Latency** | Daily—Weekly—Monthly | Near real-time | Near real-time | Daily-Weekly-Monthly | Real-time | Real-time |
| **Transformation, cleansing/enrichment Metadata process reuse** | **Best**<br>Generally high reusability of objects and processes | **Best**<br>Generally high reusability of objects and processes | **Low**<br>Transformations are done with ESQL. Metadata import limited with DB catalogue information | **Medium**<br>Limited reusability of file systems, database objects etc. | **Medium**<br>Transformations embedded in views and other database objects. | **Best**<br>Generally high reusability of objects and processes |
| **Data volume processing** | Very large (millions, billions of records) | Very large (millions, billions of records) | Small (few records)—can handle several parallel pipes of few records | Medium—access to hundreds of thousands or few millions of remote records | Medium—access to hundreds of thousands or few millions of remote records | Small (few records)—can handle several parallel pipes consisting of few records in each. |
| **Support for event monitoring** | Very limited with high latency | Depended on trigger capability of data sources | Best—logic can be added to support true event propagation and not only data transaction movement. | Very limited with high latency | Limited to data events and depended on trigger capability of data sources | Best—logic can be added to support true event propagation and not only data transaction movement. |
| **Transport** | FTP, direct database connection | FTP, direct database connection | Messaging | FTP, direct database connection | Direct database connection | FTP, Messaging |

| | ETL | EDR | EAI | ECM | EII | ESB |
|---|---|---|---|---|---|---|
| **Versioning** | Full support | Limited support | Limited support—custom build | Limited support | Limited support - custom build | Limited support |
| **Complexity of transformation** | Any complexity | Any complexity | Simple syntax transformations | Any complexity | Transformations that can be expressed with SQL | Any complexity |
| **Workflow control** | Scheduling, dependencies and error or exception handling | Scheduling, trigger based | Extensive—rules based | Scheduling | None | Rules based |
| **Major strengths** | • Optimised for data structures<br>• Periodic, batch-oriented (not intended for real-time)<br>• Can move large volumes of data in one step<br>• Enables complex data transformations requiring calculations, aggregations or multiple stages<br>• Scheduling controlled by the administrator<br>• Several GUI based tools available to • increase productivity<br>• High level of reuse of objects and transformations | • Capture only the changes<br>• Lower latency between source and target systems<br>• Can move large volumes of data in one step<br>• Provides convenient data storage, data back-up between two locations<br>• Quick access to data in case of disaster | • Optimised for API-based applications<br>• Real-time (or near)<br>• Move/send individual events or transactions<br>• Some capability for simple and basic transformation and rules<br>• Workflow controlled<br>• Broker capabilities (subscriptions) | • Reduction of paper handling and error-prone manual processes<br>• Reduction of lost documents<br>• Online access to information that was formerly available only on paper, microfilm, or microfiche<br>• Improved control over documents and document-oriented processes<br>• Streamlining of time-consuming business processes<br>• Security over document access and modification | • Relational access to non-relational sources<br>• Ability to explore data before a formal data model and metadata are created<br>• Quicker deployment<br>• Can be reused by ETL and/or EAI further developments<br>• Access in place data, meaning it avoids unnecessary movement of data.<br>• Optimised for global access to remote sources | • Optimised for XML-oriented data transformation<br>• Faster and cheaper accommodation of existing systems.<br>• Increased flexibility; easier to change as requirements change.<br>• Standards-based.<br>• Scales from point solutions to enterprise-wide deployment (distributed bus).<br>• More configuration rather than integration coding. |
| **Main challenges** | • Time to market<br>• Change management<br>• Data moved regardless of real need<br>• Consumes storage systems<br>• Data out-of-synch with the original source when it arrives Large requirements for staging areas<br>• Unidirectional<br>• Lack of multi-site update support (2 phase commit) | • Consumes storage systems<br>• May consume network bandwidth during peak hours<br>• Unidirectional<br>• Possibly high resource utilisation on the source system | • Limited transformation capability<br>• Limited support for data aggregation<br>• Limited to tens of records per transaction<br>• Complexity for development<br>• Longer Time to market<br>• Limited reuse for transformations<br>• Limited support for metadata (use, import and export)<br>• Semantic integrity<br>• May consume network bandwidth during peak hours | • Change management<br>• Data moved regardless of real need<br>• Consumes storage systems<br>• May consume network bandwidth during peak hours | • Need Matching keys across sources<br>• Data types mismatch<br>• Data reconciliation<br>• Possibly high resource utilisation on the source system<br>• Limited to hundreds of thousands of rows for remote result sets<br>• Performance degradation when query pushdown is not used<br>• Limited transformation—bounded by SQL capability and system capacity Multi-site updates require transactional control (2PC) | • Enterprise Message Model is usually required, resulting in additional management overhead.<br>• It normally requires more hardware than simple point to point messaging.<br>• Extra overhead and increased latency caused by messages traversing the extra ESB layer, especially as compared to point to point communications. |

***Source: Sahi et al. [63]***

# E. Appendix – Questionnaire

Bedankt dat u de tijd neemt om deze enquête in te vullen. Deze is opgesteld door Robert Spruijt, student aan de Universiteit Twente. De enquête vormt een onderdeel van mijn afstudeeronderzoek voor de Master Business Information Technology dat ik uitvoer bij PwC. Het onderwerp van mijn onderzoek betreft de selectie van een business intelligence (BI) architectuur. Het doel van deze enquête is het, in het kader van het onderzoek ontwikkeld, framework dat ondersteunt bij de selectie van een BI oplossing te valideren.

De enquête zal ongeveer 10 min. van uw tijd vragen. De gegevens van deze enquête worden vertrouwelijk behandeld en zullen enkel gebruikt worden voor dit onderzoek. De gegevens zullen anoniem worden verwerkt en de resultaten zullen dus ook niet naar u of uw organisatie herleid kunnen worden. Het zou zo kunnen zijn dat ik naar aanleiding van de door u gegeven antwoorden graag contact met u op zou willen nemen. U kunt aan het eind van de enquête aangeven of u dit een probleem vindt of niet. Indien u dit geen probleem vindt vragen wij u om uw contactgegevens.

Daarnaast kunt u na afloop van de enquête aangeven of u geïnteresseerd bent in mijn masterthesis. Daarin vindt u niet alleen het framework en de resultaten van de enquête, maar ook een overzicht van wat BI nu precies is, de verschillende BI architecturen en de verschillen daartussen. Op het moment dat ik mijn thesis heb afgerond zal ik deze dan opsturen naar het door u opgegeven mailadres.

Voordat u gevraagd wordt naar de BI oplossing binnen uw organisatie, wordt een aantal generieke vragen gesteld over uw organisatie zoals industriesector, omvang en omzet. Dit in verband met het herkennen van mogelijke patronen tussen de resultaten van verschillende typen organisaties.

# *Algemene Informatie betreft uw Organisatie*

**1. In welke sector is uw organisatie actief?**

- Landbouw, bosbouw en visserij
- Winning van delfstoffen
- Industrie
- Productie en distributie van en handel in elektriciteit, aardgas, stoom en gekoelde lucht
- Winning en distributie van water; afval- en afvalwaterbeheer en sanering
- Bouwnijverheid
- Groot- en detailhandel; reparatie van auto's
- Vervoer en opslag
- Logies-, maaltijd- en drankverstrekking
- Informatie en communicatie
- Financiële instellingen
- Verhuur en handel in onroerend goed
- Advisering, onderzoek en overige specialistische zakelijke dienstverlening
- Verhuur van roerende goederen en overige zakelijke dienstverlening
- Openbaar bestuur, overheidsdiensten en verplichte sociale verzekeringen
- Onderwijs
- Gezondheids- en welzijnszorg
- Cultuur, sport en recreatie
- Overige dienstverlening
- Huishouden als werkgever; niet-gedifferentieerde productie van goederen en diensten voor eigen gebruik
- Extraterritoriale organisaties en lichamen
- Anders, namelijk...

***2. In welk marktsegment is uw organisatie actief?***

- Beursgenoteerde onderneming
- Niet-beursgenoteerde onderneming
- Publieke sector

***3. Wat is de omvang van uw organisatie?***

- <100 werknemers
- 100-250 werknemers
- 250-500 werknemers
- >500 werknemers

**4. Wat *is* de jaaromzet van uw organisatie?**

- <50 miljoen
- 50-100 miljoen
- 100-200 miljoen
- >200 miljoen

# *De BI Architectuur binnen uw Organisatie*

Voordat we u een aantal vragen stellen over de *BI architectuur*, beschrijven we kort wat wij daaronder verstaan om misverstanden te voorkomen.

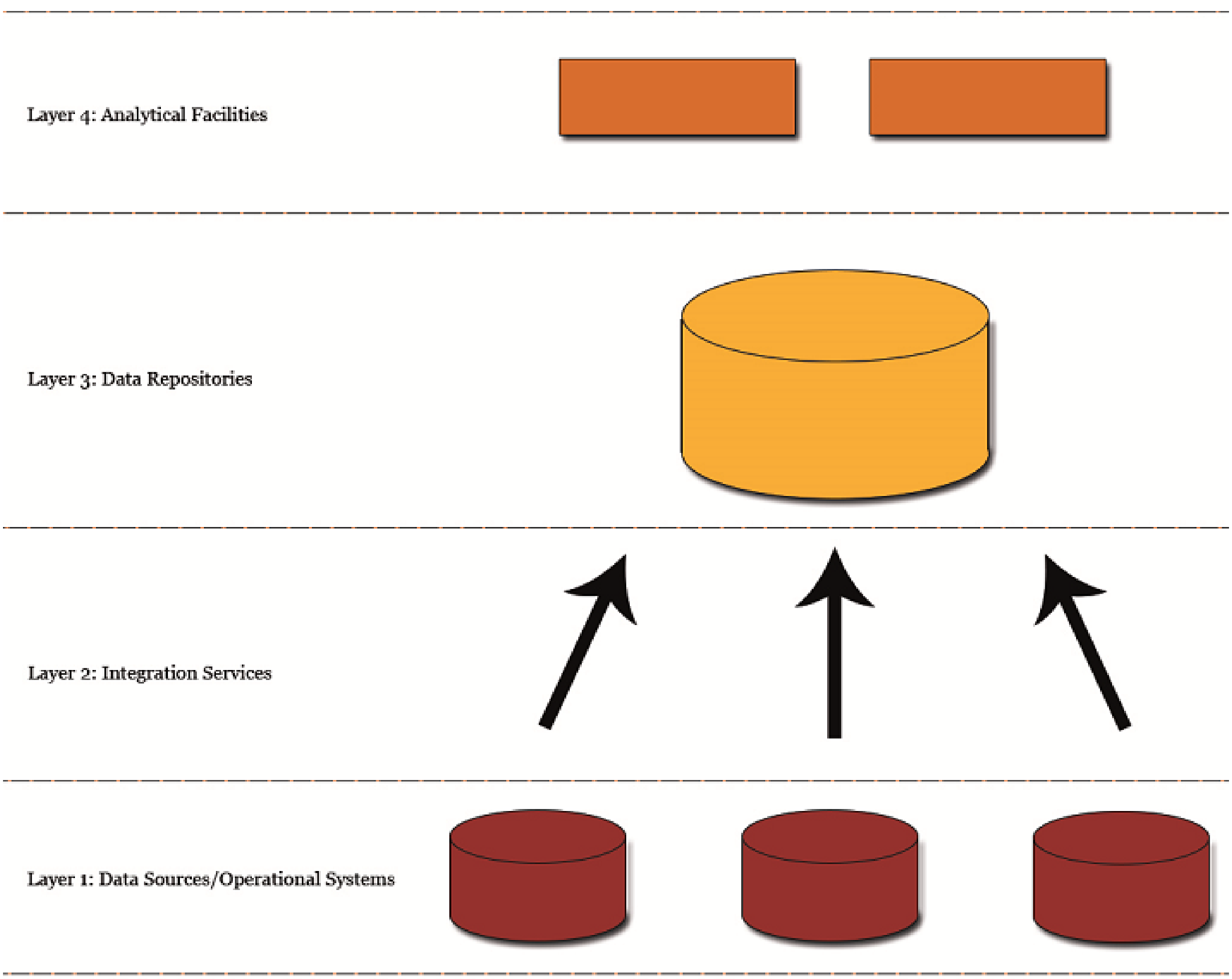
Onder BI verstaan wij het volgende:

*"BI is een algemene term voor applicaties, platforms, tools en technologieën die het proces van verkenning van zakelijke data, data relaties en trends ondersteunt. BI applicaties bieden organisaties de middelen om data te verzamelen en te analyseren met als doel het faciliteren van rapporteren, querying en decision-making."*

Een BI architectuur kan normaal gesproken opgesplitst worden in vier 'lagen':

*Bronsystemen* - Dit zijn de operationele systemen waar de data vandaan komt, denk bijvoorbeeld aan CRM, ERP en HRM systemen.
*Integratielaag* - Voordat analyses uitgevoerd kunnen worden dient de data eerst uit de bronsystemen gehaald te worden. De integratielaag zorgt hiervoor. De informatie vanuit één of meerdere bronsystemen wordt samengevoegd en eventuele aanpassingen kunnen worden gemaakt.
*Data warehouse* - Deze laag bestaat uit een database waarin de geïntegreerde data wordt opgeslagen. Deze database wordt gebruikt voor de analyses.
*Analytische faciliteiten* - Vaak verwijzen mensen naar deze laatste laag als ze het hebben over een BI oplossing. Dit is de laag die bestaat uit de daadwerkelijke BI tool die bijv. de rapportages genereert en helpt met het maken van beslissingen.

In onderstaand plaatje is deze 'lagen' structuur grafisch weergegeven.




### *5. Wat was/waren de voornaamste reden(en) voor uw organisatie om een BI oplossing te implementeren?*

6. ***Wat zijn de behoeftes die een rol hebben gespeeld bij de selectie van uw <u>BI architectuur</u>?(We zijn bij deze vraag op zoek naar de behoeftes die van invloed zijn geweest op de keus voor een <u>BI architectuur</u>, dus niet zozeer generieke behoeftes voor BI zelf)***

- ☐ Ondersteuning voor een hoge informatie uitwisseling tussen afdelingen
- ☐ Korte ontwikkelingstijd
- ☐ Lage hoeveelheid resources benodigd
- ☐ Lage hoeveelheid kennis van eigen IT medewerkers benodigd
- ☐ Goed om kunnen gaan met lage data kwaliteit in bronsystemen
- ☐ Ondersteuning voor real-time data
- ☐ Vermogen van de oplossing om met hoge volumes data om te gaan
- ☐ Vermogen van de oplossing om met frequente updates/veranderingen om te gaan (Aanpasbaarheid)
- ☐ Goede schaalbaarheid
- ☐ Makkelijk te onderhouden
- ☐ Lage moeilijkheidsgraad/Complexiteit van de oplossing
- ☐ Lage impact van de oplossing op de performance van bronsystemen
- ☐ Hoge performance van de BI oplossing
- ☐ Hoge mate van integratie
- ☐ Hoge mate van customisability
- ☐ Anders, namelijk... ____
- ☐ Anders, namelijk... ____
- ☐ Anders, namelijk... ____
- ☐ Anders, namelijk... ____
- ☐ Anders, namelijk... ____

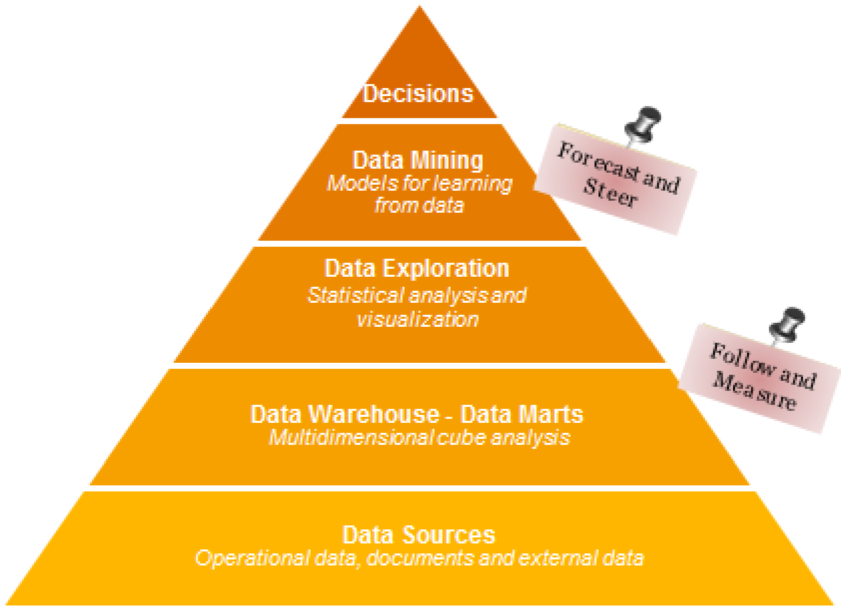
***7. Hoe belangrijk waren de behoeftes die u in de vorige vraag geselecteerd heeft? (Schaal van 1 tot 5; hoe meer sterren hoe groter de rol die deze behoefte heeft gespeeld tijdens het selectieproces)***

| | |
|---|---|
| Ondersteuning voor een hoge informatie uitwisseling tussen afdelingen | |
| Korte ontwikkelingstijd | |
| Lage hoeveelheid resources benodigd | |
| Lage hoeveelheid kennis van eigen IT medewerkers benodigd | |
| Goed om kunnen gaan met lage data kwaliteit in bronsystemen | |
| Ondersteuning voor real-time data | |
| Vermogen van de oplossing om met hoge volumes data om te gaan | |
| Vermogen van de oplossing om met frequente updates/veranderingen om te gaan (Aanpasbaarheid) | |
| Goede schaalbaarheid | |
| Makkelijk te onderhouden | |
| Lage moeilijkheidsgraad/Complexiteit van de oplossing | |
| Lage impact van de oplossing op de performance van bronsystemen | |
| Hoge performance van de BI oplossing | |
| Hoge mate van integratie | |
| Hoge mate van customisability | |
| ${q://QID13/ChoiceTextEntryValue/16} | |
| ${q://QID13/ChoiceTextEntryValue/17} | |
| ${q://QID13/ChoiceTextEntryValue/18} | |
| ${q://QID13/ChoiceTextEntryValue/19} | |
| ${q://QID13/ChoiceTextEntryValue/20} | |

**8. *In hoeverre sluit uw BI oplossing aan bij de behoeftes die u in de voorgaande vraag geselecteerd heeft? (Schaal van 1 tot 5; hoe meer sterren hoe beter uw BI oplossing aansluit bij die behoefte)***

| | |
|---|---|
| Ondersteuning voor een hoge informatie uitwisseling tussen afdelingen | |
| Korte ontwikkelingstijd | |
| Lage hoeveelheid resources benodigd | |
| Lage hoeveelheid kennis van eigen IT medewerkers benodigd | |
| Goed om kunnen gaan met lage data kwaliteit in bronsystemen | |
| Ondersteuning voor real-time data | |
| Vermogen van de oplossing om met hoge volumes data om te gaan | |
| Vermogen van de oplossing om met frequente updates/veranderingen om te gaan (Aanpasbaarheid) | |
| Goede schaalbaarheid | |
| Makkelijk te onderhouden | |
| Lage moeilijkheidsgraad/Complexiteit van de oplossing | |
| Lage impact van de oplossing op de performance van bronsystemen | |
| Hoge performance van de BI oplossing | |
| Hoge mate van integratie | |
| Hoge mate van customisability | |
| ${q://QID13/ChoiceTextEntryValue/16} | |
| ${q://QID13/ChoiceTextEntryValue/17} | |
| ${q://QID13/ChoiceTextEntryValue/18} | |
| ${q://QID13/ChoiceTextEntryValue/19} | |
| ${q://QID13/ChoiceTextEntryValue/20} | |

Hieronder ziet u een piramide die verschillende niveaus van BI weergeeft. Hierbij geldt dat in de onderste niveaus de nadruk van BI vooral ligt op het meten en monitoren van activiteiten, waarbij geldt dat hoe hoger je komt in de piramide, hoe meer de nadruk van BI komt te liggen op het sturen en maken van beslissingen.




### 9. *Welk van de lagen in bovenstaande piramide ziet u terug in uw organisatie?*

☐ Decisions

☐ Data Mining

☐ Data Exploration

☐ Data Warehouse - Data Marts

☐ Data Sources

### 10. *Wat voor type BI oplossing(en) heeft uw organisatie in gebruik?*

☐ Enterprise reporting – Denk aan operationele en bedrijfsrapportages

☐ Dashboards en scorecards – Denk aan grafische rapportages die het monitoren van de organisatie als doel hebben

☐ Data mining en geavanceerde analyses – Denk aan het uitvoeren van geavanceerde query's, statistische analyses, trend analyses, data mining

☐ Visuele en OLAP analyses – Denk aan slice-and-dice analyses met visualisaties, drill-downs, pivots en andere onderzoekende functies

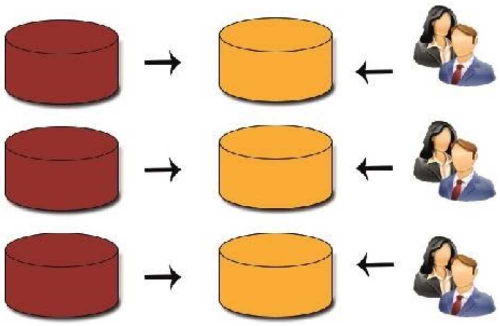
☐ Mobile apps en alerting – Denk aan bedrijfsapplicaties op mobiele devices en het doorzoeken van de data voor mogelijke afwijkingen

☐ Anders, namelijk...

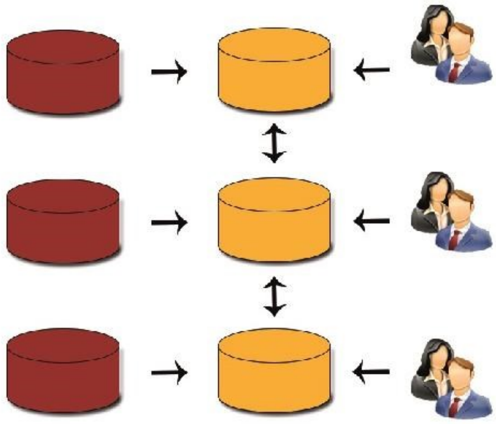
**11. Wat is de naam van uw huidige BI tool? (bijv. Oracle BI, Infor, ...) U kunt meerdere tools benoemen.**

**12. Hoe ziet de data warehouse architectuur van uw organisatie eruit?**

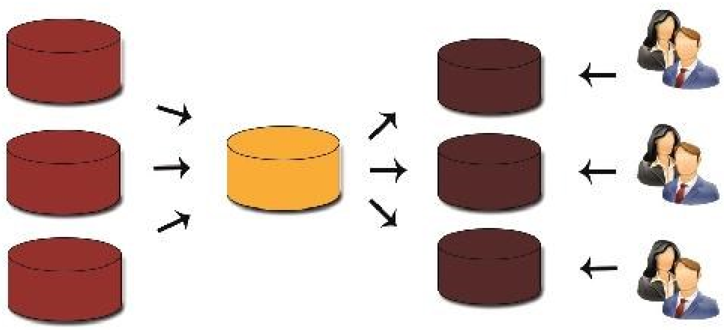
○ Onafhankelijke data warehouse(s) die enkel data bevat van een specifieke functie/regio (Independent Data Mart Architecture)



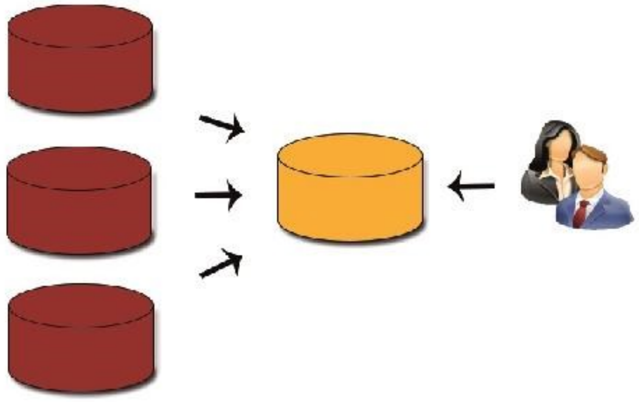
○ Een architectuur bestaande uit verschillende data bases die aan elkaar gelinkt zijn door dezelfde dimensies te gebruiken (Data Mart Bus Architecture with Linked Dimensional Data Marts)



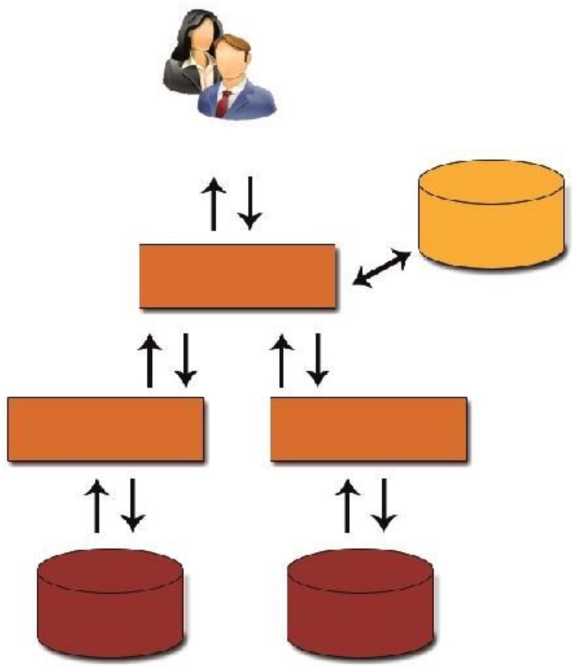
○ Een centrale data warehouse met data van meerdere bronsystemen, gelinkt aan individuele data marts die door de gebruikers gebruikt worden voor analyses (Hub-and-Spoke Architecture)



Een centrale data warehouse met data van meerdere bronsystemen, waarop alle analyses uitgevoerd worden (Central Data Warehouse Architecture)



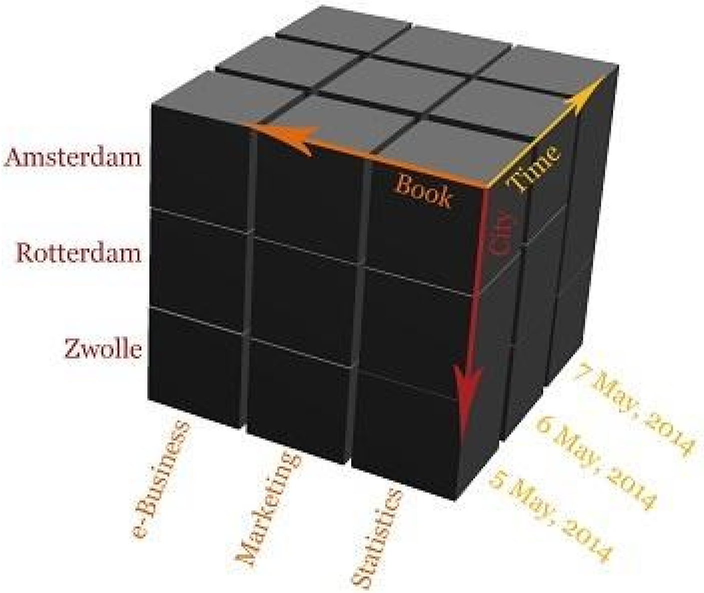
Een architectuur die de query, op het moment dat de gebruiker deze ingeeft, split in verschillende delen voor de verschillende bronsystemen. De gevraagde informatie wordt verkregen van de bronsystemen, samengevoegd en vervolgens teruggegeven aan de gebruiker (Federated Architecture)



Anders, namelijk...

Online Analytical Processing (OLAP) is een technologie die gebruikt wordt om grote databases binnen een organisatie vorm te geven en is een technologie die BI ondersteunt. Het kenmerk van OLAP is dat het vaak dezelfde informatie bevat als de bronsystemen, maar dan is de data op zo'n manier gestructureerd (multi dimensioneel) dat op een snelle en efficiënte manier queries en analyses uitgevoerd kunnen worden.

Data in OLAP databases is gestructureerd door middel van cubes. Cubes combineren verschillende dimensies zoals tijd, geografie en product en bevatten geaggregeerde data zoals verkoop of voorraad cijfers. Een voorbeeld van zo'n cube kunt u in het plaatje hieronder zien. De verkoopcijfers van boeken worden hierin weergegeven voor de dimensies dag, stad en boek.




Er wordt bij OLAP vaak onderscheid gemaakt tussen MOLAP, ROLAP en HOLAP.

***13. Welke vorm van Online Analytical Processing (OLAP) gebruikt u binnen uw organisatie?***

- ○ Wij gebruiken geen OLAP binnen onze organisatie.
- ○ Multidimensional Online Analytical Processing (MOLAP):
  Bij deze vorm van OLAP zijn cubes vooraf berekend en opgeslagen in een eigen format
  Voordelen: Snelle responstijd op queries en kan omgaan met complexe queries
  Nadelen: Data duplicatie, restrictie aan grootte en investering in extra technologie benodigd
- ○ Relational Online Analytical Processing (ROLAP):
  Bij deze vorm van OLAP wordt gebruik gemaakt van relationele tabellen om cube aggregaties op te slaan.
  Voordelen: Geen duplicatie en schaalbaar
  Nadeel: Langzamer
- ○ Hybrid Online Analytical Processing (HOLAP):
  Bij deze vorm van OLAP wordt getracht een combinatie te maken van de voordelen van MOLAP en ROLAP. Alleen geaggregeerde data is opgeslagen in MOLAP cubes; facts worden opgeslagen in een relationele tabel.
- ○ Ik weet/begrijp het niet.
- ○ Anders, namelijk... ____

### *14. Welke integratie techniek(en) wordt/worden er binnen uw organisatie gebruikt om de data uit de bronsystemen te halen?*

☐ *Enterprise Service Bus* (ESB), een platform waar applicaties en services gebruik van kunnen maken om berichten te sturen.

☐ *Service-Oriented Architecture* (SOA), een techniek die applicaties, processen, data activiteiten en operaties beschikbaar maakt als services. Deze services kunnen vervolgens met elkaar communiceren door gebruik te maken van een standaard format.

☐ *Extract, Transform, Load* (ETL), een techniek die meestal op een 'pull' basis werkt. Dat betekent dat updates pas plaatsvinden op het moment dat er hier naar gevraagd wordt. Deze techniek werkt erg goed bij grote hoeveelheden data of batches met data. Vaak vind deze actie plaats op geplande tijdstippen. Deze techniek werkt minder goed met kleine hoeveelheden data zoals individuele transacties of berichten.

☐ *Right-time Extract, Transform, Load* (RT-ETL), een verbeterde versie van ETL. Naast de eigenschappen van ETL is deze techniek tevens in staat om transacties en berichten te verwerken en kan daardoor informatie op het juiste moment aanleveren.

☐ *Enterprise Application Integration* (EAI), een techniek die meestal op een ‘push basis’ werkt. Dat betekent dat op het moment dat een update of transactie plaatsvindt, een notificatie wordt gegeven aan de andere applicaties. EAI maakt het tevens mogelijk voor applicaties om data te benaderen, zonder dat deze de locatie of het formaat hoeven te weten.

☐ *Enterprise Information Integration* (EII), een techniek die als grote voordeel heeft dat er geen extra data warehouse nodig is. Op het moment dat de gebruiker een query ingeeft, wordt de benodigde data uit de verschillende bronsystemen gehaald en teruggegeven aan de gebruiker.

☐ *Enterprise Data Replication* (EDR), een techniek die data van punt A naar punt B kopieert, zonder hier enige wijzigingen aan te brengen.

☐ Anders, namelijk... ______

### *15. Hoeveel bronsystemen worden geïntegreerd door uw BI oplossing?*

______

### *16. In hoeverre is uw organisatie tevreden over de huidige BI oplossing?*

| | Zeer ontevreden | Ontevreden | Neutraal | Tevreden | Zeer tevreden |
|---|---|---|---|---|---|
| Mate van tevredenheid gebruikers | ○ | ○ | ○ | ○ | ○ |
| Mate van tevredenheid IT afdeling | ○ | ○ | ○ | ○ | ○ |

***17. Waarom is uw organisatie in deze mate (on)tevreden over de huidige BI oplossing?***

***18. Wat zijn de grootste uitdagingen voor uw organisatie rondom de BI oplossing?***

***19. Welke investeringen en veranderingen zijn de komende maanden/jaren gepland rondom de BI oplossing?***

***20. Wat voor rol denkt u dat BI in de toekomst gaat spelen? (groter/kleiner en waarom denkt u dat?)***

***21. Bent u geïnteresseerd naar de resultaten van dit onderzoek? Indien dat het geval is zal ik u mijn masterthesis opsturen op het moment dat ik deze heb afgerond.***

Ja, ik ontvang graag de resultaten na afloop van het onderzoek. Nee, bedankt.

***a. Het mailadres waar u mijn masterthesis op wilt ontvangen:***

***22. Het zou zo kunnen zijn dat ik naar aanleiding van uw antwoorden graag enkele aanvullende vragen zou willen stellen. Vindt u dit een probleem?***

| Ik vind het geen probleem als ik gecontacteerd wordt om verdere toelichting te geven op mijn antwoorden. | Ik wil liever niet gecontacteerd worden. |
|---|---|
| ○ | ○ |

***a. Uw contactgegevens:***

| | |
|---|---|
| Organisatie | |
| Naam | |
| Mailadres | |
| Telefoon | |

# Bibliography


[1] 1keydata.com (2014). *MOLAP, ROLAP and HOLAP*. 1keydata.com Retrieved May 7, 2014 from http://www.1keydata.com/datawarehousing/molap-rolap.html

[2] Al-Debei, M. M. (2011). Data Warehouse as a Backbone for Business Intelligence: Issues and Challenges. *European Journal of Economics, Finance and Administrative Sciences*, 33, 153-166.

[3] Ali, N. (2004). *Service Oriented Architecture (SOA)*. University of Limerick. Retrieved March 24, 2014 from https://dspace.ndlr.ie/bitstream/10633/2338/1/ServiceOrientedArchitecture.pdf

[4] Alsqour, M., Matouk, K., & Owoc, M. L. (2012). A survey of data warehouse architectures—Preliminary results. *Proceedings of the Federated Conference on Computer Science and Information Systems*. 1121-1126.

[5] Ariyachandra, T., & Watson, H. (2010). Key organizational factors in data warehouse architecture selection. *Decision Support Systems*, 49(2), 200-212.

[6] Ariyachandra, T., & Watson, H. J. (2006). Which data warehouse architecture is most successful? *Business Intelligence Journal*, 11(1), 4.

[7] Azvine, B., Cui, Z., & Nauck, D. D. (2005). Towards real-time Business Intelligence. *BT Technology Journal,* 23(3), 214-225.

[8] Baars, H., & Kemper, H. (2008). Management Support with Structured and Unstructured Data – An Integrated Busines Intelligence Framework. *Information Systems Management*, 25, 132-148. DOI: 10.1080/10580530801941058

[9] Ballou, D. P., & Tayi, G. K. (1999). Enhancing Data Quality in Data Warehouse Environments. *Communications of the ACM,* 42(1), 73-78.

[10] Bhiri, S., Gaaloul, W., Rouached, M., & Hauswirth, M. (2009). *Advances in Web Semantics I: Ontologies, Web Services and Applied Semantic Web*. Springer.

[11] Bontempo, C., & Zagelow, G. (1998). The IBM data warehouse architecture. *Communications of the ACM*, 41(9), 38-48.

[12] Boyina, N. (2012). *ER Modeling vs. Dimensional Modeling*. Advisory Services PwC.

[13] Cabbibo, L., & Torlone, R. (1998). Querying Multidimensional Databases. *Database Programming Languages*, Springer Berlin Heidelberg, 319-335.

[14] Cameron, S. A. (2011). Enterprise Content Management: A Business and Technical Guide. *BCS, The Chartered Institute for IT.*

[15] Centraal Bureau voor de Statistiek (2014). *Standaard Bedrijfsindeling 2008*. Centraal Bureau voor de Statistiek. Retrieved 18 June, 2014 from http://www.cbs.nl/nl-NL/menu/methoden/classificaties/overzicht/sbi/sbi-2008/default.htm

[16] Chaudhuri, S., Dayal, U., & Narasayya, V. (2011). An overview of business intelligence technology. *Communications of the ACM*, 54(8), 88-98.

[17] Choudhary, R. K. Key organizational factors in data warehouse architecture selection. *Vivekanda Journal of Research*, 24-32.

[18] Colliat, G. (1996). OLAP, Relational, and Multidimensional Database Systems. *ACM Sigmod Record*, 25(3), 64-69.

[19] Convey, C., Karpenko, O., Tatbul, N., & Yan, J. (2001). Data integration services. Chapter1. *Brown University*.

[20] Dahl, O. (2002). Enterprise Application Integration. *School of Mathematics and Systems Engineering*, Växjö University, Sweden.

[21] Davenport, T. H., & Harris, J. G. (2007). *The Architecture of Business Intelligence*. Accenture. Retrieved 13 May, 2014 from http://www.accentureoutsourcing.ie/SiteCollectionDocuments/PDF/ArchBIAIMS.pdf

[22] Erl, T. (2005). *Service-oriented architecture: concepts, technology, and design*. Prentice Hall.

[23] Falagas, ME; Pitsouni, EI; Malietzis, GA; Pappas, G (2008). Comparison of PubMed, Scopus, Web of Science, and Google Scholar: Strengths and weaknesses. *FASEB Journal* 22 (2): 338–42. doi:10.1096/fj.07-9492LSF

[24] Foote, R. & Iyer, H. H. (2013). *Enterprise Service Oriented Architecture Presentation*. PwC.

[25] Friedman, T., & Strange, K. H. (2004). *Architecture: The Foundations of Business Intelligence,* Gartner.

[26] Garcia-Molina, H., & Salem, K. (1992). Main memory database systems: An overview. Knowledge and Data Engineering, *IEEE Transactions* on, 3(6), 509-516.

[27] Gartner. (2014). *Magic Quadrant for Business Intelligence and Analytics Platforms*. Gartner. Retrieved March 14, 2014 from http://www.gartner.com/technology/reprints.do?id=1-1QLGACN&ct=140210&st=sb#

[28] Goel, A. (2006). Enterprise Integration – EAI vs. SOA vs. ESB. Infosys Technologies White Paper. Retrieved April 25, 2014 from http://ggatz.com/images/Enterprise_20Integration_20-_20SOA_20vs_20EAI_20vs_20ESB.pdf

[29] Gray, P., & Watson, H. (1998). *Decision Support in the Data Warehouse*. Prentice Hall.

[30] Güner, S. (2005). *Service Oriented Architecture*. University of Hamburg. Retrieved March 24, 2014 from http://www.sts.tu-harburg.de/pw-and-m-theses/2005/gune05.pdf

[31] Halevy, A. Y., Ashish, N., Bitton, D., Carey, M., Draper, D., Pollock, J., ... & Sikka, V. (2005). Enterprise information integration: successes, challenges and controversies. *Proceedings of ACM* (pp. 778-787). ACM.

[32] Halevy, A., Rajaraman, A., & Ordille, J. (2006). Data integration: the teenage years. *Proceedings of the 32nd international conference on very large data bases* (pp. 9-16). VLDB Endowment.

[33] Hannula, M., & Pirttimäki, V. (2003). Business intelligence empirical study on the top 50 Finnish companies. *Journal of American Academy of Business*, 2(2), 593-599.

[34] Hashimi, S. *Service Oriented Architecture Explained*. ONDotNet.com. Retrieved March 24, 2014 from http://www.cin.ufpe.br/~ajsc2/SOA%20explained.pdf

[35] Hellerstein, J. M., Stonebraker, M., & Caccia, R. (1999). Independent, open enterprise data integration. *IEEE Data Eng. Bull.*, 22(1), 43-49.

[36] Herschel, R. T., & Jones, N. E. (2005). Knowledge management and business intelligence: the importance of integration. *Journal of Knowledge Management*, 9(4), 45-55.

[37] Hu, X. (2010). *Data Warehouse Technology and Application in Data Centre Design for E-Government*. InTech.

[38] Inmon, W. H. (1996). The data warehouse and data mining. *Communications of the ACM*, 39(11), 49-50.

[39] Inmon, W. H. (2000). What is a Data Warehouse? Billinmon.com. Retrieved at March 25, 2014 from http://repository.binus.ac.id/2009-2/content/M0584/M058459913.pdf

[40] Isik, O., Jones, M. C., & Sidorova, A. (2011). Business intelligence (BI) success and the role of BI capabilities. *Intelligent systems in accounting, finance and management*, 18(4), 161-176.

[41] Isik, O., Jones, M. C., & Sidorova, A. (2013). Business intelligence success: The roles of BI capabilities and decision making. *Information & Management,* 50(1), 13-23.

[42] Jindal, R., & Acharya, A. (2004). Federated data warehouse architecture. *Wipro Technologies white paper*.

[43] Jukic, N. (2006). Modeling strategies and alternatives for data warehousing projects. *Communications of the ACM*, 49(4), 83-88.

[44] Knabke, T., & Olbrich, S. (2013). Understanding Information System Agility – The Example of Business Intelligence, *Hawaii International Conference on System Sciences*. 3817-3826. IEEE.

[45] Laumer, S., Beimborn, P. D. D., Maier, D. W. I. C., & Weinert, C. (2013). Enterprise Content Management. *Business & Information Systems Engineering,* 5(6), 449-452.

[46] Loos, P., Lechtenbörger, J., Vossen, G., Zeier, A., Krüger, J., Müller, J., ... & Winter, R. (2011). In-memory databases in business information systems.*Business & Information Systems Engineering*, *3*(6), 389-395.

[47] Maier, M. (2013) *Towards a Big Data Reference Architecture*. University of Eindhoven.

[48] Mansour, R. (2012). *Business Intelligence Strategie & Implementatie*. TopBI. Retrieved May 7, 2014 from http://www.management-in-motion.nl/userdata/download/19Kennissessie%20BI.pdf

[49] Menge, F. (2007). Enterprise service bus. *Free and Open Source Software Conference*.

[50] MicroStrategy (2012). Architecture for Enterprise Business Intelligence, Whitepaper. Retrieved April 7, 2014 from https://www2.microstrategy.com/download/files/whitepapers/open/Architecture-for-Enterprise-BI.pdf

[51] Negash, S. (2004). Business Intelligence. *Communications of the Association for Information Systems*, 13.

[52] Olszak, C. M., & Ziemba, E. (2003). Business Intelligence as a Key to Management of an Enterprise. *Proceedings of Informing Science and IT Education Conference*. Retrieved May 15, 2014 from http://proceedings.informingscience.org/IS2003Proceedings/docs/109Olsza.pdf

[53] Olszak, C. M. (2012). Competing with business intelligence. *Proceedings on IT for Practice*, VSB-TU, Ostrava, 98-108.

[54] Olszak, C. M., & Ziemba, E. (2007). Approach to Building and Implementing Business Intelligence Systems. *Interdisciplinary Journal of Information, Knowledge and Management, 2*.

[55] Olszak, C. M., & Ziemba, E. (2004). Business Intelligence Systems as a new generation of Decision Support Systems. *Proceedings PISTA 2004, International Conference on Politics and Information Systems: Technologies and Applications*. Orlando: The International Institute of Informatics and Systematics.

[56] Öykü, I. (2010). Business intelligence success: an empirical evaluation of the role of BI capabilities and the decision environment *(Doctoral dissertation, University of North Texas).*

[57] Papazoglou, M. P. & Georgakopoulus*, D. (2003).* Service-oriented computing. *Communications of the ACM*. 46(10), 25-28.

[58] Pedersen, T. B., & Jensen, C. S. (2001). Multidimensional Database Technology. *Computer*, 34(12), 40-46.

[59] Power, D. (2001). Support decision-makers: An expanded Framework. *Proceedings of the Informing Science and IT Education Conference*. Retrieved May 15, 2014 from http://proceedings.informingscience.org/IS2001Proceedings/pdf/PowerEBKSupp.pdf

[60] Raisinghani, M. S. (Ed.). (2004). Business Intelligence in the Digital Economy: Opportunities, Limitations, and Risks. *IGI Global*.

[61] Reinschmidt, J., & Francoise, A. (2000). Business intelligence certification guide. *IBM International Technical Support Organisation*.

[62] Rouse, M. & Wigmore, I. *Enterprise Data Replication (EDR)*. Retrieved at April 1, 2014 from http://whatis.techtarget.com/definition/enterprise-data-replication-EDR

[63] Sahi, D., Bandyopadhyay, S., Pal, S., & Abdullahi, I. (2009). *Data Integration Technologies*. Advisory Services PwC.

[64] Sapperstein, J. (2013). *What is a BusinessObjects Universe?* Retrieved at May 28, 2014 from http://alteksolutions.com/wp/index.php/2009/03/what-is-a-business-objects-universe/

[65] Saunders, T. (2013). *How your Business works – Information Management – How to Align your New Solution with Business Needs*. CBIG Consulting. Retrieved April 17, 2014 from http://www.cbigconsulting.com/wp-content/uploads/2014/03/how-to-align-your-bi-solution.pdf

[66] Schiff, M. & Michaels, J. (2007). *Making a Strategic Business Intelligence Choice – A checklist*. Business Objects & MAS Strategies Whitepaper. Retrieved April 17, 2014 from http://download.101com.com/tdwi/ww24/BusinessObjects_MakingaStrategicBusinessIntelligenceChoice.pdf

[67] Schmidt, M. T., Hutchison, B., Lambros, P., & Phippen, R. (2005). The enterprise service bus: making service-oriented architecture real. *IBM Systems Journal*, 44(4), 781-797.

[68] Sen, A., & Sinha, A. P. (2005). A comparison of data warehousing methodologies. *Communications of the ACM*, 48(3), 79-84.

[69] Shariat, M., & Hightower Jr, R. (2007). Conceptualizing Business Intelligence Architecture. *Marketing Management Journal*, 17(2).

[70] Simon, P. (2013). *Too Big to Ignore: The Business Case for Big Data*. Wiley.

[71] Singh, S. (2011). Data warehouse and its methods. *Journal of Global Research in Computer Science,* 2(5), 113-115.

[72] SINTEF. (2013, May 22). *Big Data, for better or worse: 90% of world's data generated over last two years*. ScienceDaily. Retrieved March 12, 2014 from www.sciencedaily.com/releases/2013/05/130522085217.htm

[73] Stang, J., Hartvigsen, T., & Reitan, J. (2010). The Effect of Data Quality on Data Mining – Improving Prediction Accuracy by Generic Data Cleansing. *Proceedings of the 2010 International Conference on Information Quality ICIQ 2010*.

[74] Strong, D. M., Lee, Y. W., & Wang, R. Y. (1997). Data quality in context. *Communications of the ACM,* 40(5), 103-110.

[75] Taylor, H. (2007) Service-Oriented Architecture (SOA) 101 "What's Hype, What's Real?", *Juniper Networks*.

[76] Turban, E., Sharda, R., Delen, D. & King, D. (2010) *Business Intelligence: A Managerial Approach*. 2nd Edition, Prentice Hall.

[77] Wand, Y., & Wang, R. Y. (1996). Anchoring data quality dimensions in ontological foundations. *Communications of the ACM,* 39(11), 86-95.

[78] Watson, H. J., & Ariyachandra, T. (2005). *Data Warehouse Architectures: Factors in the Selection Decision and the Success of the Architectures*. University of Georgia. Retrieved March 25, 2014 from http://www.terry.uga.edu/~hwatson/DW_Architecture_Report.pdf

[79] Watson, H. J., & Wixom, B. H. (2007). The current state of business intelligence. *Computer*, 40(9), 96-99.

[80] Watson, H. J., Wixom, B. H., Hoffer, J. A., Anderson-Lehman, R., & Reynolds, A. M. (2006). Real-time Business Intelligence: Best Practises at Continental Airlines. *Information Systems Management,* 23(1), 7-23.

[81] Weir, R. (2008). *A Configuration Approach for Selecting a Data Warehouse Architecture* (Doctoral dissertation, Napier University).

[82] White, C. (2005). The next generation of business intelligence: operational BI. *DM Review Magazine*.