OPEN SOURCE ENTERPRISE SERVICE BUS:
ANALYZING IMPACT AND VALUE IN A LARGE ORGANIZATION

Master’s (two years) thesis in Informatics (30 credits)
Sharareh Bagherinia

Autumn 2013 : 2013MASI11
System integration is a challenging task for large organizations since they must manage complex IT infrastructures, and utilize hundreds of applications that cause a myriad of complications as a result of rapidly evolving systems and technologies. Over time customer requirements and the automation of internal business functions forced the gigantic service providers to develop and buy applications for different purposes. Not many years before, the applications in an organization were usually separated silos, specifically designed to answer a particular problem. However, this way of working was not efficient. It is not impossible to design one application that runs the total business, however, in reality it can't respond to all the needs inside an organization. In most cases the applications need to interact with each other and to reuse functionalities or data. At this point the concept of integrating systems first came to existence.

The progress in computer science and information technology has also affected the way that integration between applications was developed and designed over the years. This shows that there's always some new method that should be studied and investigated. In recent years, the Enterprise Service Bus has come to the market and there are many vendors in the market claiming that they have produced the exact solution needed for requirements of organizations.

In this work, the attempt was to construct an understanding of the Enterprise Application Integration practices and methods which have been utilized over the years. Mixed methods research a combination of interviews and questionnaire, and Design Science Research methods were used to generate empirical results. While the first two methods was used to supply the input for challenges of system integration, the DSR build and evaluate method was employed to test the ESB product. The criteria for the test were developed mostly as the outcome of interviews and the questionnaire as the main challenge. The aim was to verify efficiency and effectiveness of the selected and proposed ESB under special designed DSR build and evaluate phases.

As a result of both theoretical and empirical studies in combination with selection of different methods in Information System research, the work presents a knowledge foundation for organizations with the aim to make a change in their current system integration practices and patterns utilizing approaches facilitated by ESB products.

Key words: System integration, Enterprise Service Bus (ESB), large organizations, Mule ESB, SOA, Apache Service Mix, Open ESB, Open Source
Acknowledgements

I would like to take this opportunity to show my appreciation and gratitude to all who helped me in this work especially Dr. Anders Hjalmarsson and Prof. Peter Rittgen for their guidelines and suggestions.

Thanks to Joacim Järkeborn and Calvin Hall for their help and support that contributed tremendously during the writing of the thesis.

…….. I would like to take this opportunity and thank my parents for all they have done to guide me through the right path in my life…..

August 2013
Sharareh Bagherinia
# Table of Contents

1  INTRODUCTION ................................................................................................. 1

   1.1  BACKGROUND .................................................................................................... 1

   1.2  PROBLEM DEFINITION .................................................................................... 3

   1.3  PURPOSE ............................................................................................................. 4

   1.4  RESEARCH QUESTIONS ................................................................................... 4

   1.5  EXPECTED RESULTS ....................................................................................... 4

   1.6  DELIMITATIONS ............................................................................................... 5

   1.7  TARGET GROUP ................................................................................................. 5

   1.8  THESIS DISPOSITION ....................................................................................... 5

2  RESEARCH APPROACH AND PROCESS ............................................................... 7

   2.1  RESEARCH PERSPECTIVE ............................................................................... 7

          2.1.1  The selected perspective............................................................................. 9

   2.2  RESEARCH STRATEGY ...................................................................................... 9

          2.2.1  Sampling .................................................................................................. 10

          2.2.2  Case Study ................................................................................................ 10

   2.3  DATA COLLECTION PROCESS ...................................................................... 11

          2.3.1  Interviews ................................................................................................ 11

          2.3.2  Documents .............................................................................................. 12

          2.3.3  Questionnaire .......................................................................................... 12

   2.4  DATA ANALYSIS PROCESS ......................................................................... 12

          2.4.1  Qualitative Data Analysis ....................................................................... 13

          2.4.2  Quantitative Data analysis ...................................................................... 13

   2.5  STRATEGIES FOR VALIDATING FINDINGS ................................................. 14

   2.6  RESULT PRESENTATION METHOD ............................................................... 15

3  THEORETICAL BACKGROUND ....................................................................... 16

   3.1  KEY CONCEPTS ............................................................................................. 16

   3.2  SUBJECT AREAS RELEVANT TO RESEARCH ........................................... 17

   3.3  PREVIOUS RESEARCH ................................................................................. 19

          3.3.1  Enterprise Application Integration (EAI) ...................................................... 19

          3.3.2  Architectural approaches to EAI ................................................................. 23

          3.3.3  Different types of EAI ............................................................................... 25

          3.3.4  Service Oriented Architecture .................................................................. 27

          3.3.5  Different approaches to integration .......................................................... 31

          3.3.6  Comparison of different approaches in Enterprise Integration ................ 32

   3.4  WHAT IS ENTERPRISE SERVICE BUS? ......................................................... 33

          3.4.1  ESB’s core functionalities ......................................................................... 35

          3.4.2  The Enterprise Service Bus Model ............................................................ 36

   3.5  OPEN SOURCE /CLOSED SOURCE ESBS .................................................... 36

          3.5.1  Source code availability ............................................................................ 37

          3.5.2  Open standards vs. vendor lock-in ............................................................ 37

          3.5.3  Support communities ............................................................................... 37

          3.5.4  Cost ........................................................................................................... 37

          3.5.5  Open source ESBs ................................................................................... 37

   THE PROPOSED ESB ........................................................................................ 38

          3.5.6  Motivation for choosing Mule ESB ........................................................... 39

   3.6  MULE ESB ..................................................................................................... 39

   3.7  ENVIRONMENT SET UP ............................................................................... 40

   3.8  MESSAGE FLOWS .......................................................................................... 40

   3.9  SUMMARY OF THEORETICAL FINDINGS .............................................. 41

   3.10  ARGUMENTS FOR EMPIRICAL STUDY ..................................................... 42

4  EMPIRICAL STUDY ............................................................................................ 43

   4.1  PURPOSE ......................................................................................................... 43

   4.2  SAMPLING ..................................................................................................... 43
4.2.1 The case........................................................................................................................................ 43
4.2.2 Methods for sampling................................................................................................................ 43
4.3 INTERVIEWS .................................................................................................................................. 44
4.3.1 Question Selection and Validation ............................................................................................. 44
4.4 INTERVIEW DATA PRESENTATION .......................................................................................... 45
4.4.1 Integration Developer.................................................................................................................. 45
4.4.2 The integration tester ................................................................................................................ 47
4.4.3 The integration manager ......................................................................................................... 48
4.4.4 The Integration Architect 1 .................................................................................................... 49
4.4.5 The Integration Architect 2 .................................................................................................... 51
4.5 QUESTIONNAIRE ....................................................................................................................... 52
4.5.1 Questions .................................................................................................................................. 53
4.5.2 Results ...................................................................................................................................... 54
4.6 DESIGN AND EVALUATE CYCLE1 .......................................................................................... 60
4.7 DESIGN AND EVALUATE CYCLE2 .......................................................................................... 63
4.8 EMPIRICAL RESEARCH RESULTS .......................................................................................... 64
5 ANALYSIS AND RESULTS ........................................................................................................... 67
5.1 ANALYSIS ...................................................................................................................................... 67
5.2 RESULTS SUMMARY .................................................................................................................. 75
6 DISCUSSION ................................................................................................................................... 79
6.1 CONCLUSION ............................................................................................................................... 79
6.2 IMPLICATIONS FOR THE SUBJECT AREA .................................................................................. 80
6.3 METHOD EVALUATION ............................................................................................................... 81
6.4 RESULT EVALUATION ................................................................................................................ 82
6.4.1 Possibilities to generalize ........................................................................................................ 83
6.4.2 Propositions of future research ............................................................................................... 83
7 BIBLIOGRAPHY .......................................................................................................................... 84
APPENDIX A: ........................................................................................................................................ 90
  Database to ActiveMQ scenario results ........................................................................................... 91
  ActiveMQ to Database Messaging .................................................................................................. 92
  Mule test results ........................................................................................................................... 93
APPENDIX B: ........................................................................................................................................ 94
SAMPLE OF QUESTIONNAIRE ........................................................................................................ 94
List of Figures

FIGURE 3-1 - ENTERPRISE APPLICATION INTEGRATION PROJECTS (LAM & SHANKARARAMAN, 2004) ...................................... 22
FIGURE 3-2 - PROGRESS OF ARCHITECTURAL APPROACHES (SWITHINBANK ET AL., 2007) .............................................. 24
FIGURE 3-3 - DIFFERENT STYLES OF APPLICATION INTEGRATION (PAUTASSO ET AL., 2008) ........................................... 32
FIGURE 3-4 - ESB IN SOA (RICHARDS, 2006) .................................................................................................................. 34
FIGURE 3-5 - ENTERPRISE SERVICE BUS MODEL (MULESOFT, 2013) ............................................................................ 36
FIGURE 4-1 - ADAPTER (INTEGRATION TESTER DESCRIBE) ........................................................................................... 47
FIGURE 4-2 - YEAR OF EXPERIENCE OF QUESTIONNAIRE PARTICIPANTS ................................................................. 54
FIGURE 4-3 - QUESTIONNAIRE RESULT .......................................................................................................................... 55
FIGURE 4-4 - QUESTIONNAIRE RESULT .......................................................................................................................... 56
FIGURE 4-5 - QUESTIONNAIRE RESULT .......................................................................................................................... 56
FIGURE 4-6 - QUESTIONNAIRE RESULT .......................................................................................................................... 57
FIGURE 4-7 - QUESTIONNAIRE RESULT .......................................................................................................................... 57
FIGURE 4-8 - QUESTIONNAIRE RESULT .......................................................................................................................... 58
FIGURE 4-9 - QUESTIONNAIRE RESULT .......................................................................................................................... 58
FIGURE 4-10 - QUESTIONNAIRE RESULT ......................................................................................................................... 59
FIGURE 4-11 - QUESTIONNAIRE RESULT ......................................................................................................................... 59
FIGURE 4-12 - JDBC TO ACTIVEMQ MESSAGE FLOW1 ................................................................................................ 61
FIGURE 4-13 - THE JMS TO JDBC MESSAGE FLOW2 .................................................................................................. 61
FIGURE 4-14 - MULE TEST RESULT (1) .......................................................................................................................... 62
FIGURE 4-15 - MULE TEST RESULT (2) .......................................................................................................................... 63
FIGURE 5-1: MULE ESB'S MESSAGE PROCESSING CYCLE1 .......................................................................................... 73
FIGURE 5-2: MULE ESB'S MESSAGE PROCESSING CYCLE2 .......................................................................................... 73

List of Tables

TABLE 1 - THE HISTORY OF INFORMATION SYSTEM INTEGRATION ........................................................................... 20
TABLE 2 - CRITERIA FOR DIFFERENT APPLICATION INTEGRATION METHODS ......................................................... 32
TABLE 3 - THE ASSESSMENT SUMMARY OF THREE OPEN SOURCE ESBs ................................................................ 38
TABLE 4 - THE SUBJECTIVE ASSESSMENT SUMMARY ............................................................................................. 38
TABLE 5 - MULE ESB SUMMARIZED FEATURES ..................................................................................................... 39
TABLE 6 - SUMMARY OF QUESTIONNAIRE RESULT .................................................................................................. 65
TABLE 7 - SUMMARY OF THE RESULTS .................................................................................................................... 77
1 Introduction
1.1 Background

The concept of system integration is not new and has been employed from the time of connecting different applications in a point to point manner. Integrating different functions in an industry started more than 50 years back when industries tried to integrate functions such as design, production, management, marketing, and finance (Alsene, 1994). According to (Singletary, 2004, p. 1), "integration of separate and isolated islands of systems began with projects involving a few applications". Organizations initiated integration projects following different approaches such as ERP, EAI, Middleware, and Component ware (ibid).

Why different applications should be connected together? Why is there a need to integrate applications? The problem that system integration is intended to address is the ability to have information flow inside the enterprise emerging from different applications built for responding purposes at the right time and right place (Naveen, David C, & T.M., 2003). The aim of using system integration methods in an organization revolves around different requirements which have been revolutionized from the past until now but the most important requirement was that the applications were providing functionalities in an isolated manner (Linthicum, 2000), (Naveen et al., 2003). Moreover it was hard to share data and functionality they offered in a unified and efficient way (Roshen, 2009).

In the past applications were designed in a way that remained fixated and not for changing problems. The developers estimated the software requirements not far in advance because the business environment which used the IT products did not change rapidly (Linthicum, 2000). All the data, technologies, and hardware were defined and there was no need to worry about the next requirement to show up and what strategy to use in order to be ahead of competitors in providing the services. However, with the advent of internet and concepts such as "e-Commerce", companies began to integrate business processes and customers with their internal systems, this way the need for integrating applications were inevitable since utilizing the benefits of e-commerce, companies became faster to respond to market needs and service offerings (Naveen et al., 2003). Usually businesses merge and consolidate to lower the costs of service offerings and cut the redundancies, in this case application integration is also needed, since merged companies have different applications for different purposes running on various types of platforms (ibid).

Moreover, companies started to realize that to be able to respond to employees, suppliers, customers, and business partners requirements with on-demand services and information from different sources of enterprise systems, they need to unite the capabilities and offerings of their applications (Ortiz, 2007). According to (Janssen & Cresswell, 2005), to be able to provide service that involves different applications, integration is needed, in other words some sort of unification of business processes and data between separate applications was needed (ibid). Historically, different approached emerged to facilitate application integration, Enterprise Application Integration (EAI) enabled the isolated applications to share functionality and data since mid-1990s. EAI is not one product or approach; it is combination of different approaches to enable data and business processes to become integrated across the organization (Lee, Siau, & Hong, 2003).

According to (Erasala, Yen, & Rajkumar, 2003), the main focus of any selected EAI approach is to increase the sharing of useful information to all applications involved in system integration and to information islands and bottlenecks. According to (Linthicum, 2000), EAI
promotes integrating of business processes and data without constantly changing the applications.

Before the introduction of services in IT world, the concept has been with us for a long time. Service is the outcome of a process in which one part is the service provider and on the other hand is service consumer (Roshen, 2009). The procedure starts with the service consumer requesting for a specific type of service then the service provider is responsible to offer the needed service back to the consumer considering different aspects negotiated in advance as qualities that the service provider must comply with and finally both parties profited from the outcomes of the targeted service because the service provider has used all the resources to be proficient in providing the service which can be reused by other service consumers with slight differences. So the concept of service in our daily life offers us advantages such as efficient way of doing things and getting the required output in a more money and time saving manner.

According to (Roshen, 2009), the reusability concept originates from the fact that services can be used again with slight changes in configuration and tying up different actors playing their role along the process of service delivery.

The concept of software service is similar to what we have experienced in real life. Software service has long been introduced to solve the problem of computing the information and provide different results needed in various situations. According to (Roshen, 2009), the most important contributor to the concept of service in software and the sub-concepts originated from this notion such as Service Oriented Architecture (SOA) is distributed computing as well as advances in computer programming languages.

Computer programming languages have undergone a huge transformation. The first languages were based on sequential command execution and there were no consideration for code reuse at all because the tasks needed throughout the code were coded in a redundant manner. With the advent of procedural or modular languages the first spark of reusability ignited in software industry. Although the introduction of object oriented languages as the third stage in the development of programming languages affected the concept of service in software industry, all the services, in this case methods, functions, and procedures could be used in one application but the concept of reusability and becoming service oriented is about different applications being able to use each other's services and functionalities. The concept of system integration is mostly affected by the advances in distributed computing which help different applications share functionality and data which is the fundamental objective of system integration and SOA (Roshen, 2009).

Service orientation is not only about sharing data among different applications but also the sharing of functionalities that different applications working based on specific business logic are being developed. During the years different technologies were introduced to solve the integration problems and even different methods of integration have been used until now (Ruh, Maginnis, & Brown, 2002). When enterprises seek to increase time to market with their IT solutions, they alleviate their productivity, and reduce costs by streamlining their business processes, they should focus on their internal and external information system integration (Zhigang & Huiping, 2009). SOA provides an architectural design to connect information systems in a way that business processes and data can be reused as services. Some applications publish the services and others consume the service. The business process or data is not duplicated and it can be reused throughout the enterprise (Valipour, Amirzafari, Niki Maleki, & Daneshpour, 2009). Not long after the introduction of SOA web services emerged to facilitated employing SOA paradigm in organizations. However, web services are prone to
security and reliability issues from long ago (Gulledge, 2006). In recent years, a new solution to easy implementation of SOA and consequently integration solutions was offered by different vendors, ESB, is the new offering to system integration market which facilitates SOA (ibid).

1.2 Problem Definition

Large enterprises tend to gather different applications from custom-built to buying from third party vendors to be able to quickly respond to the market needs of their customers and be able to handle business functions and processes. Every business process in an organization includes many sources of data to interact and help the flow of information run smoothly.

In past years many applications were developed in-house for better quality services and solutions offered to the customers as well as other applications offered by vendors in software market. The applications are usually comprised of functionalities required for the customers to plan their business functions and processes as well as being able to implement them in an expedited manner. This means that it is important for applications with different platforms implemented with different languages had to be able to communicate data and functionalities needed within enterprise software architecture.

From the past, large organizations started manual integration which requires extra code for introducing new applications in the peer to peer application integration model. It is always the case to find the best solution to avoid integration methods which are costly in terms of time and efficiency and the point to point integration model soon was replaced by spoke and wheel paradigm which is called Enterprise Application Integration. However, EAI was not scalable and introduced single point of failure in networks (Gulledge, 2006).

The development in system integration began with the introduction of Service Oriented Architecture (SOA) in IT and business areas which brought more flexible integration methods and tools. The concept of service orientation introduced technologies that helped large enterprises implement communication among different applications via standard methods which made the application integration platform independent and loosely coupled. The concept of usability of services introduced by SOA has helped organizations to accomplish their business goals and enabled them to be more responsive to the market change. However, the initial architecture model of the SOA was not so efficient for large organizations. The main reason is that the architecture which is hub-and-spoke uses a central message broker that can handle the messaging style in integrating different applications. Using a single message broker can be useful if it is not used as an integration solution in an organization which has many applications and services. A single message broker is replaced by a new Bus-like architecture, called Enterprise Service Bus.

For many large organizations, it is important to consider the drawbacks of the integration styles they have implemented during the years as well as utilizing the new ways which will alleviate the challenges of integration. From the past until now combinations of different methods have been utilized to make up a robust infrastructure for integration in many enterprises but the need for a more advanced tool to implement SOA concept seems inevitable. With the introduction of ESBs many software vendors started to implement a solution which runs both on desktop and server computers. Furthermore, there are many alternatives from the large vendors like IBM and Oracle to open source offerings including Sonic, Mule, and others to choose from. The need to specifically match the challenges of integration with the product offerings was identifiable in previous literature. The previous
research focused on the technical perspective of system integration. In this work the main attempt was to investigate the research phenomena from a socio-technical perspective.

1.3 Purpose
The aim in this study is to investigate and generate an understanding of the problems and issues of system integration in an enterprise and create an understanding of what ESB products can offer beneficially to enterprises. The most suitable ESB product based on the introduced criteria from the previous studies on the open source market products was selected and employed. Furthermore the study will implement tests the chosen ESB software in action with a specific criteria. Since data integration is one of the major concerns of large organizations, it is important to consider cases which support the needs of data integration.

By implementing the tests with the chosen ESB platform it is easier to propose a better solution from an architectural point of view for the developers and architects involved in integration projects.

1.4 Research questions
Based on the previous section of this study this section introduces the consequent research questions which will be answered and investigated in the following chapters.

The author in this study will answer the following research questions with a suitable research method to both satisfy the technical needs of the industrial partner of this study, the research case, and be able to establish a basis for further research in the academic field and in addition to contribute to the body of knowledge.

What is the impact of ESB on integration projects in an enterprise?

1. What challenges do system integration present?
2. How suitable is ESB for addressing system integration challenges at a large organization?

1.5 Expected Results
The aim of this study is to give a comprehensive understanding about enterprise application integration using the most recent concepts that supports Service Oriented Architecture which is ESB.

The outcome of this study is to define the role of ESBs in integration projects at large enterprises owning huge number of applications which need to communicate in different scenarios to make the business processes run and be adaptable to quick changes. This helps organizations especially in improving their integration projects which usually involves the applications that are implemented in different environments and also are being managed by different teams around the globe.

There are some criteria to choose a fitting integration pattern as well as a tool to perform the designed practices; one of the outcomes of this thesis is to identify the most suitable open source tool for integrating applications based on the literature and previous works on this area.

Furthermore, different cases were considered to test ESB in order to evaluate the functionalities, efficiency, and simplicity of the selected ESB. The implemented tests focused mainly on integrating databases and a messaging system to send the data to other applications.
via an open source messaging server. The outcome from this part of the thesis helps to
determine if the ESB suits the requirements for integrating applications. In addition, it can be
used as a guideline to implement the integration projects in a more extensive scale in large
organizations.

1.6 Delimitations
The area of system integration is a broad area and consists of different methods and
approaches. In this study, the attempt was to focus on the challenges in integration projects
provided by experts who work in this area for many years and try to make the general area
specific. The main delimitation was that the ESB product could not be tested in a real
environment with real data due to strict security rules for industrial critical data, however, the
attempt was to follow the same data format as it is in real situation.

1.7 Target Group
The results of this study can be used both from a scientific and technical perspective. The
technical part will be useful for the integration architects and developers to get familiar with a
powerful ESB application which is available in open source market; this means that the
guidelines from this study could provide a knowledge base for IT professionals working
mainly on integration projects.

The ESB concept will be explained in great detail which would be a good knowledge source
from an academic and technical perspective. The information presented could be useful for
software architects, system analysts as well as integration architects who are responsible to
handle integration issues in integration projects.

On the other hand, the conceptual findings of this study could be used by researchers as well
as students who wish to investigate on Enterprise Service Bus and its role in integration
projects.

The designed cases to evaluate ESB product could be the basis for more complicated models
to be able to test the selected ESB considering different criteria.

1.8 Thesis disposition
To accomplish the goals of this study this report includes six chapters. Below the content of
each chapter is presented to clarify the overall structure of the thesis.

The first chapter includes the introduction about the concept of system integration as well as a
brief history and the steps of development in this field of study which generate a foundation
for the following chapters. A number of concepts related to the system integration were
presented briefly and at the end the research aim and the expected results for this study were
discussed.

Chapter two presents the research strategy and selected research design to carry out the
research process. The selected methods will then be used to complete the empirical study
which clarifies the data collection procedure as well as the validity of the results.

The third chapter is comprised of presentation of previous related literature in Enterprise
Application Integration, Service Oriented Architecture, and Enterprise Service Bus in current
system integration patterns which makes a firm basis for next chapters.
Chapter four consists of the empirical study based on the selected research methods. The data is gathered to support the relevance of the research done to be able to conduct the study based on some real world results which leads to the next chapter.

The fifth chapter offers analysis and discussion for theoretical study in the third chapter and the empirical findings in chapter four. The analysis in this chapter should provide a solid knowledge base for the research questions which are specified in the first chapter. Based on the data analysis method used in this chapter the author discusses to what extent the research questions introduced in the first chapter were answered and the whole aim of the study was accomplished.

Chapter six finalizes the study presenting reflection, and conclusion of what has been achieved conducting the whole research procedure. Furthermore the general possible ideas to be worked on as a future work in the selected filed is proposed.
2 Research Approach and Process

In this chapter the methods presented are used to be able to give structure to the steps in research in information technology. Selection of a suitable paradigm and based on that the data gathering and data analysis methods helps to give a structure to the whole study. Based on the aim and research questions in previous chapter, the research is designed and then performed utilizing the research strategy, data collection, and data analysis methods accordingly.

2.1 Research Perspective

According to (Hevner, March, & Park, 2004) research in the information systems discipline is mostly portrayed based on two paradigms, namely behavioral science and design science. The former is used to develop and verify theories that explain or predict human or organizational behavior and the latter aims to produce a new artifact which helps to extend boundaries which prohibit human and organizations to perform. The artifact should indeed adhere to the organization's goals thus both of the above paradigms promote organizations to investigate in IS discipline and reach their final goal, development of a product which benefits the organization and its customers by offering better service that is more effective and efficient. (Hjalmarsson & Rudmark, 2012)

The design-science paradigm came to existence as a complement to two other paradigms namely, positivism and interpretivism. The positivist research is based on an epistemological position which applies methods and strategies used in natural sciences to the research and study of a social reality and beyond. In other words, the human behavior is studied using scientific methods. According to (Silverman, 1993) positivism is concerned with testing the correlation between variables and is mostly based on quantitative methods for data analysis. The positivist paradigm has some characteristics that are worth mentioning when trying to choose a suitable method for performing research (Oates, 2006).

- Our world is rule based and regular and is not random which exists independent of humans
- The research is performed objectively and the researcher's believes is detached from the truth
- Research is based on testing hypotheses using empirical data. The hypothesis is either confirmed or rejected based on the data gathered.
- Statistical analysis and mathematical modeling and proof is preferred to make a logical and objective ground for the analysis done (Quantitative data analysis)

On the other hand, interpretivism distinguishes between people and the objects of natural sciences and as a result a researcher in social sciences grabs the subjective meaning of social action. This phenomological view of human behavior is concerned with this fact that social reality is meaningful for people and therefore human action is meaningful.

Here the aim is to understand phenomena by meanings and values people assign to them. Based on the definitions presented there are some characteristics related to this paradigm in IS research that is worth mentioning (Oates, 2006):

- There is no single version of 'the truth'. The right thing is a construction of our minds either individually or in groups and therefore different people from different backgrounds, cultures perceive the world in a distinguished way.
- Researchers are not neutral. Their assumptions, beliefs, and values will affect the process of doing the research.
The researcher tries to understand people in their natural setting from the perspective of its participants not his own.

The data analysis gathering and analysis method preferred is qualitative which are the analysis of mostly text, metaphors or images used by people.

There is no one result for the interpretation done by researchers.

Hence in recent years there is another paradigm which has gained interest among researchers. This emergent paradigm combines qualitative and quantitative research where more than one method is used (Bryman, 2006). According to (Tashakkori & Teddlie, 1998), introduced the notion of pragmatism as the new paradigm that combines the characteristics of positivism and interpretivism after fighting for authority of one on the other. Logically pragmatism is branded as mixed method research which contains elements of both qualitative and quantitative research (ibid).

(Hevner, March, Park, & Ram, 2004) in their works on DSR paradigm claim that when conducting research in Information systems (IS) the DSR (Design Science Research) offers methods and strategies to be utilized in order to achieve the innovative, efficient, and effective artifact which satisfies a number of identified business needs. According to (Hevner et al., 2004) the artifact can be constructs, models, methods, or instantiations or a combination of them. Knowledge is then achieved by the process of iterative building and testing designed artifact until a desired result is achieved to solve the problem. The testing or applying the designed artifact is a joint activity done together with the users which reflects the usability of the artifact in the context it will be used.

Moreover, it is cited that the importance of design is mentioned in IS literature. Referring to previous works the relevance of IS research is in a straight line related to its applicability in design meaning that "the implications empirical IS research should be implementable" to motivate critical way of thinking and prepare the basis for other researchers. Below the characteristics of DSR paradigm is summarized (Hevner, March, & Park, 2004)

- It is both a process and a product, it defines multiple contextually situated world-states
- Knowledge is gained via design and development of constructs, models, methods, and instantiations. The artifacts are then evaluated considering the utility they provide to solve problems.
- It combines quantitative and qualitative methods for problem definition and then evaluation of the artifacts built through an iterative and developmental process depending on the artifact type designed and developed.
- The DSR is different from Routine Design, because the aim in DSR is to add to the knowledge base and then contribute to both business needs and knowledge foundation.
- Artifacts are not full-grown information systems, because the time for development is short and usually the innovative idea generating from the design and evaluation process results in a more effective and efficient system.

Based on the framework presented by (Hevner et al., 2004) the DSR paradigm includes seven steps or guidelines to be used in order to identify a well conducted research. In this study the DSR guidelines and its framework are used to help implement the needed artifact for evaluation of the research purpose of whether an Enterprise Service Bus a better integration solution to be implemented in a large enterprise.
The characteristics of the DSR guidelines are presented below:

1. The research must produce an artifact to address a particular problem.
2. Gain knowledge and understanding that is relevant to the solution and build the artifact to address the problem based on the knowledge gained.
3. The "utility, quality, and efficiency" of the artifact should be rigorously evaluated using appropriate evaluation methods.
4. The research contribution should be identified by artifact’s design, development, and evaluation processes.
5. Rigorous methods should be applied both in the construction and evaluation of designed artifact.
6. Design as a research process is the iterative to find the best solution.
7. Communication of research is the detailed information presented to the research audiences.

2.1.1 The selected perspective
In this study a combination of two perspectives were used, each suitable for a part of study therefore the selected perspective is "mixed method" or pragmatism. The study combines qualitative and quantitative research to answer the research question. Moreover, since the research objective is to investigate a phenomenon in an organization a suitable research strategy is a case study that utilizes some aspects of DSR paradigm as a tool. This is intended to gain more confidence in the generated results.

2.2 Research Strategy
In this session the methods used to apply the selected paradigm are presented in an effort to clarify the way the total research is accomplished. The strategy is chosen based on what is going to be researched and what is the aim and purpose of the research. According to Oates, the research strategy used is directly affected by the research philosophy selected by the researcher.

Moreover, a case study is a research strategy which contains the following characteristics (Oates, 2006):

- Investigate detailed information about one instance of a real life phenomenon.
- The research can contain both qualitative and quantitative data obtained through a large range of data generation methods.

Another general definition of a case study is presented by (Robson, 1993) can illuminate more about the characteristics of a case study as a research strategy. "Case study is a strategy for doing research which involves an empirical investigation of a contemporary phenomenon within its real life context using multiple sources of evidence".

(Hevner, March, & Park, 2004) presented a framework and guidelines to be used within DSR research. The researcher can use the cycles and guidelines to weave his/her own research strategy based on the needs and requirements proposed by research questions.

According to the DSR framework the research is divided in three cycles, each present part of the artifact development and evaluation process. In this study the research process includes (Hjalmarssson & Rudmark, 2010):
1. Identification of the current problem or opportunities addressed by the use of the developed artifact.
2. Gather and combine information from literature review and the real situation in company of case system integration.
3. The previous step will be a knowledge base for the artifact development; therefore it is a combination of requirement gathering and problem identification which is relevant to the current situation.
4. Design of the artifact utilizing available means to reach the final goal while the requirements and needs are satisfied.
5. Apply methods of evolution to investigate "Utility" of the artifact in the designed context.
6. Present contribution of the research to prove the usability of the outcome the research provides to the research target groups.

2.2.1 Sampling
To start with data collection process, a target population should be selected to gather sample empirical data. In both qualitative and quantitative research a sampling scheme should be employed to select participant of data collection methods (Onwuegbuzie & Collins, 2007).

According to (M. N. Marshall, 1996), a significant step in research is to choose a sample for study since it has never been possible from different factors such as practicality, efficiency, or ethicality to study a total population.

Sampling techniques are different since they serve different purposes and actually use different methods to select population. The main category of sampling is probability and non-probability sampling. Under each category some specific techniques are provided by literature to select from(Oates, 2005). (Onwuegbuzie & Collins, 2007), categorized probability sampling as suitable for quantitative research and purposive sampling for qualitative research which can infer that selecting units of a population based on purpose associated with answering a research question(ibid). Mixed method research as it is defined can use sampling methods from both qualitative and quantitative research (Onwuegbuzie & Collins, 2007).

In this study I employed two different data collection method to gather data and based on the requirements of my data gathering activity I used critical case and convenience sampling for interview data gathering. This method is useful for selection of a target population based on factors such as accessibility and being willing to help (Oates, 2005). In addition, together with critical case sampling snowball sampling method for questionnaire data gathering was employed. Snowball sampling helped to find a suitable target population for the questionnaire.

2.2.2 Case Study
According to (Bryman & Bell, 2012), by strategy we mean the general orientation to conduct research. Selection of the research perspective usually leads to a special selection of methods for data collection also strategy to gather required data in the process of research. Furthermore, the process of knowledge creation from the collected data also plays an important role which usually is defined from the style of analyzing data which is either a systematic approach or analytical approach to understand the relationship and interactions in the data collected (Miles & Huberman, 1994)
According to (Oates, 2006), there are different types of research strategies in Information System research such as survey, design and creation, experiment, case study, action research, and ethnography each has its own characteristics and suitable for different types of research.

In this study, the case study is selected based on the fact that the nature of this work is exploring a problem by testing, observing the simulated situation in a smaller scale.

In recent years this research strategy has gotten more attention from researchers especially in Information system and computer software engineering. The fact is that nowadays there is shift away from controlled laboratory experiments in IS field to a phenomenon without controlling the factors affecting its behavior (Runeson & Höst, 2008). The critiques toward case study approach are about the value of the results, not being able to generalize the case, and biased by researchers (Runeson & Höst, 2008). However, the case study researcher can meet the critiques by selecting suitable research methods as well as emphasizing that knowledge is not only related to statistical impact (Runeson & Höst, 2008).

The case study is helpful when combining different methods for generating knowledge. There are two types of case studies, single-case and multiple-case, in which the former (single case) is employed with in this study. According to (Oates, 2006), there are three methods of analyzing case studies: exploratory, descriptive, and explanatory. In this study the case explores several factors and compares those factors with what exists in theory, therefore an exploratory type case study is used. In addition one of the primary factors for selecting a case study is that the author starts to explore a phenomenon in which he or she has no prior knowledge of. Therefore a case study is a suitable strategy for this research.

2.3 Data Collection Process
In this stage the data is gathered to make a solid foundation for the research. Usually two types of methods are used to identify and gather required data. In this stage it is important to know which method of data gathering is suitable for starting the process. When the selected strategy is case study different methods of data generation can be utilized. In qualitative research there are a number of ways to gather information, such as text analysis, interviews, questionnaires, and observations (Oates, 2005).

2.3.1 Interviews
According to (Oates, 2005), an interview is a special type of conversation between people, moreover an interview is a good tool to gather detailed information of a phenomenon. Through the use of interviews a researcher can uncover answers to complex questions by investigating of emotions and occasions that are difficult to be observed through other means of qualitative data collection methods such as questionnaires (ibid). Interviews usually are categorized into two groups of structured and unstructured, but there is a third option namely semi-structured that is a combination of the two main interview techniques (C. Marshall & Rossman, 2010).

According to (Longhurst, 2003), structured interview follows a predefined list of questions and the questions are asked in the same order and the same way it is written. On the other hand the unstructured interview interviews focus on themes and allow the informant to speak freely. A semi-structured interview technique combines elements of the two former interview methods where the interview has some predetermined structure in addition to allowing the interviewee the flexibility to express ideas, issues, events, and behaviors which have been experienced.
In my data collection process, a semi-structured interview technique is utilized which contains specific themes and list of questions needed for generating the necessary data. However, depending on the interviewee and the amount of contribution he or she puts on the interview, I am willing to have open discussions and not follow any specified order based on the feedback I get from the interviewee (Longhurst, 2003; Oates, 2005).

### 2.3.2 Documents
As I explained my primary data collection method above, in addition I employed a secondary enhancement method that utilizes documents to assist the primary method of interviews. According to (Oates, 2005), it is important to consider certain factors when using documents as secondary data collection method. Factors such as purpose, context, and timescale of the written documents should be considered in order to gather suitable and relevant sources (ibid).

According to (Bowen, 2009), organizational and institutional documents have been part of qualitative data gathering processes for many years. Document analysis is usually used as a supplementary method for other qualitative methods as a means for triangulation. Researchers usually use data triangulation, which is uses a combination of methods to research a specific phenomenon. According to (Oates, 2005), special consideration should be given to the evaluation of the document regarding trustworthiness and authenticity. In this research I tried to select documents which were trustworthy and valid based on the criteria such as context, time scale, and authenticity that exist within a related, if not the same, research domain.

### 2.3.3 Questionnaire
Questionnaires are another data collection method which is used as a cost-effective means to collect empirical data. It usually includes pre-defined collection of questions that are provided to respondents in a specific order. This method is used mainly in a survey research strategy, however it is also useful in other strategies including case studies (Oates, 2005).

This data collection method is useful when the aim is collecting data from larger population. It also supports different type of questions; close-ended and open-ended which former provides respondents with range of answers to choose from and the latter where the respondents writes the answer of their own (ibid).

One the factors that should be considered while designing a questionnaire is content validity. This means that the questions should cover the domain which is going to be investigated. Moreover, Construct validity is related to the connection between what questions are measuring and what was the designer intended to measure. Reliability is the last factor that should be taken into consideration while utilizing questionnaire. This factor is difficult to measure since it is related to facts that if the results will be the same if the questions are sent to the same respondents (Oates, 2005).

In this thesis a combination of open and close ended questions is used for designing the questionnaire. To comply with content validity I received help from people working in the research domain to validate the correlation between question contents to that of the selected domain.

### 2.4 Data Analysis Process
According to (Hevner et al., 2004), the aim of design Science methodology is to improve the current situation by designing innovative artifacts. An application’s domain consists of people, organizational systems, and technical systems which interact to reach a goal. The
designed artifact should address an important organizational problem. The second critical activity in the DSR cycles is the evaluation of the designed artifact.

Moreover, artifacts are instances not a fully developed information systems or solutions, alternatively they define ideas, technical capabilities, and products which are further the implementation cycle where more effective and efficient usage will be achieved. The evaluation process plays a critical role in DSR process and can be categorized in terms of completeness, consistency, accuracy, performance, reliability, usability, and fit with the organization (ibid). In this research I used interviews and documents as sources for building two artifacts and then evaluated the artifact based on performance by increasing message loads.

This stage of the research includes an analysis of the findings based on source data collected from various sources and the developed artifact. The process of categorizing and classification of the gathered data helps to draw insightful conclusions. The choice of the selected analysis method is mainly affected by the nature of data gathered. In general there are two types of data in this research namely qualitative and quantitative. Where the research produces qualitative data, qualitative data analysis is used; otherwise quantitative data analysis is suitable.

2.4.1 Qualitative Data Analysis
According to (Caudle, 2004), qualitative data analysis is to gather data from different sources such as interviews, on-site observations, and documents, then present the findings in a responsible manner. Analysis reveals the most important part of the data related to the research that has been performed (ibid). Many researchers use different methods of qualitative analysis such as ground theory, phenomenology, discourse analysis, and narrative analysis. On the other hand there are some methods of qualitative data analysis which are more generic and are not categorized in such traditional methods (Thomas, 2003).

This type of data analysis method is called the "general inductive approach" and is used by a considerable amount of qualitative researchers. The main purpose of this type of approach is to allow the research findings to be extracted from the frequent, dominant or significant themes in raw data (ibid).

In this research I started to gather empirical data from interviews which is textual data, the material from the interviews were recorded by "Audacity 2.0.3" and saved for further review. The next step was to write the transcription of the interviews. As I used a general inductive approach for qualitative data analysis, I read through the transcribed data and tried to find categories, themes and relations which were more close to my research questions and the investigation of the selected domain.

2.4.2 Quantitative Data analysis
According to (Oates, 2005), quantitative data is numerical data which is mostly yielded from surveys, experiments, and other research strategies. Both positivist and interpretive researchers can have quantitative data in their research. However, by generating quantitative data a suitable analysis method should also be utilized to present the pattern of numbers in a readable and clear way to enable the researcher to draw conclusions. Apart from visualization techniques for analyzing quantitative data, descriptive statistical methods can also be helpful to define a pattern such as average value of a range of data (ibid).
In this thesis two different data collection method provided me with quantitative data. At first to answer the first research sub-question, I employed a questionnaire as means of collecting quantitative data. I analyzed the results using visualization methods such as bar and pie charts. This is the simplest way to present the data to readers and enable the researcher to compare different results to give an appropriate discussion and arrive at a relevant conclusion. (Oates, 2005) argues that bar charts are suitable for displaying frequencies. However, in my data the bar charts show the population of respondents to a specific answer provided. Pie charts are used to verify the intervals of years of experience to show the sample population targeted is suitable experts to specific phenomena of System integration and enterprise application integration.

Moreover, in the process of build and evaluate phases I collected numerical data which helped me to verify and analyze the number of messages inserted and processed for each message flow. The line graphs are the special tools utilized to depict the degree of growth in the specific ESB software processing response to increasing increase the message loads. According to (Oates, 2005), line graphs are suitable to show trends, for example a changing state for a specific subject that is under investigation.

2.5 Strategies for validating findings
The evaluation of the research plays an important role to make the work trustable and the findings reliable for other researches. Moreover, it is important that as a researcher the evaluation is accomplished by the researcher to make sure that the thesis can be followed by others. In this section the evaluation methods used to generate more dependable results will be discussed. According to (Oates, 2005), both positivist and interpretivist research communities have different criteria for validating the quality of research. Method triangulation is when there is more than one method to collect research data and approach research questions using multiple modes of research. As mentioned before the selected philosophical stance of this work is that of pragmatism and therefore it contains data collection methods used in both interpretivism and positivism.

Evaluation of the collected data and information is an important matter in a scientific study since it is essential to validate the findings and affirm the quality of research. In this work different approaches were performed to authenticate both qualitative and quantitative results. Techniques such as participatory modes of research in collaboration with professionals who have long term experience in the target domain, also member checking and reconfirmation of the technical findings to certify the correlated table, diagrams and further premises (Creswell, 2009; Hall, 2013).

Despite the fact that “different accounts of data concerning to a single phenomenon may lead to multiple realities” it was tried to coalesce the suppositions through considering the overlapping data results to identify consistencies (Creswell, 2009).

As it is mentioned before this study is based on a practical subject matter, which demands data collection methods that are preferable in applying theoretical facts in a realistic setting.

Researchers designed items to evaluate the results of a study based on principles which consider logic, description, rationality and simulations (Hevner et al., 2004).

The below evaluation criteria (Petter, Khazanchi, & Murphy, 2010) are used to validate whether the proposed conceptual model is:
1. **Plausible** – checks the coherency of an idea and investigates if its concept is rational. This can be realized by making a comparative assessment between theoretical foundation of the thesis and the research area.

2. **Effective** – controls the general concept for being described in a comprehensible way and also the related theories for being presented in the expressed solution.

3. **Feasible** – investigates whether a concept is practical. This can be done by professionals examining the feasibility of theoretical model.

4. **Predictive** – ascertains that a concept produces the intended results. This forms the required consistency between research project elements.

5. **Reliable** – scrutinizes whether a concept can be practically approved. This can be completed through handing over the experimental data to confirm the results.

### 2.6 Result Presentation method

In this study the results are presented as a combination of textual information and explanations in the form of tables and models. The aim of the textual explanations is to clarify how the comparison between theoretical and empirical studies could relate or differ to help the author find the right answers for the research questions. Moreover the diagrams including bar chart and line graphs were utilized to illustrate the results.
3 Theoretical Background

In this chapter the literature related to the selected topic were gathered and reviewed to be able to answer the research sub-questions. The collected literature in this chapter will provide a solid foundation for the selected research in the area of system integration in general.

3.1 Key Concepts

Information System Integration

System integration is comprised of different methods to connect applications in a large enterprise and mainly supports the business processes of a company to function in an agile and effective way (Hohpe & Woolf, 2012). Information system integration is all the activities and projects which initiates and executes in enterprises with huge number of applications from COTS, core legacy systems to in-house built applications on different platforms to share data and functionality and to avoid data and functionality duplication throughout the company (Roshen, 2009).

Enterprise Application Integration (EAI)

According to (Goel, 2006), is the response to the business requirement that diverse applications should communicate regardless of their platforms in order to accomplish required business purposes in an effective and flawless way. EAI is the limitless sharing of data and businesses among different applications and data sources in an enterprise (Linthicum, 2000).

Enterprises require the data and processes are share without being forced to change applications and their data structures dramatically. EAI provides a common way of communication for both business processes and data among applications (ibid).

Message Oriented Middleware (MOM)

Message Oriented Middleware is the mechanism to exchange data between two or more applications. It consists of an intelligent bridge that facilitates communication between two or more applications (Linthicum, 2000). Traditionally MOM was used as a point to point integrating mechanism, transferring data messages between applications (ibid).

According to (Chappell, 2009), MOM is a key part of the next generation architecture, called Enterprise Service Bus, providing the foundation the network of virtual channels that an ESB uses to route messages throughout an extended enterprise and beyond.

Service Oriented Architecture (SOA)

Service Oriented Architecture is an architectural design with the aim of connecting applications in such a way that business functions and data can be reused as services among other applications. In other words, in this type of software architecture services are functions offered by service providers and service consumers can request for the services (Valipour et al., 2009).

In large enterprises applications usually require compatible functionality. SOA solved the problem of code and data duplication in large organizations and introduced the concept of loose coupling and reusability in system integration (ibid).
Web services

Web service is a specification which helps systems to communicate with each other based on open standards. Different platforms have their own specific mediation such as JAX-RPC and JAXM which map java data types to SOAP. Applications with different platforms use this general specification offered by web services to mediate between the web service and their internal specifications (Valipour et al., 2009)

It is important to understand that Web services do not provide an integration solution by themselves. Introducing Web services into an environment does not replace the need for middleware. Web services are not a new form of application integration or EAI; it simply adds new components that can be utilized effectively in a variety of architectures. Three types of open standards are WSDL (description standard), UDDI (discovery standard), and SOAP (messaging) (Goel, 2006).

Message Broker

Message broker is a new version of Message Oriented Middleware which has more features than traditional middleware. Message brokers are added with message transformation, translation and routing which is useful mainly in spoke and hub architecture. Applications are connected to the broker with adapters (Goel, 2006).

Enterprise Service Bus (ESB)

According to (Rademakers & Dirksen, 2008), an ESB can be a pattern or a product used for simplification of the integration and functionality reuse in application integration. Each application, as an end point in connected directly to ESB which plays as a middleware and offers a collection of features that according to researchers are helpful to simplify system integration (ibid).

3.2 Subject Areas relevant to research

In this section the relevant subject areas which can be used to illuminate the research question and its sub-questions will be discussed to motivate the selection of the suitable areas.

Research question:

What is the impact of ESB on integration projects in an enterprise?

- What challenges do system integration present?
- How suitable is ESB for addressing system integration challenges at a large organization?

The related research areas

It is important to break down the research questions to research areas that help to find the answers to them. In this section the author provides the selected areas and tries to motivate the selection. The selected areas of research provide a suitable foundation to show what previously has been done on the subject area. It also assists the researcher to answer sub-research questions.
The selected areas are helpful to base the reliable significance of the research on the body of knowledge presented in selected areas. Based on the research questions I selected two areas to research in previous literature namely Information system integration, Enterprise application integration, and Enterprise Service Bus. However, after starting to select and read the literature, the focus is more on literature that are more inclined to Enterprise Application Integration and Enterprise Service Bus.

**Information system integration:**

Information systems have become the source of information and mainly the knowledge-base for enterprises and their businesses are dependent upon its productivity and competitiveness. Information systems are the fabric of inter-organization and intra-organization collaborative business processes (Irani, Themistocleous, & Love, 2003)

The definition of Information system integration and the history behind it is offered by a variety of articles in this domain. However, the general definition of system integration can be derived from the fact that organizations are nowadays surrounded by a changing business environment. In order to be responsive to the needs of customers and suppliers, enterprises can no longer use disparate architecture for their single applications, such as Enterprise Resource Planning (ERP) or Supply Chain Management Systems, and so forth (Hasselbring, 2000; Irani et al., 2003)

In the move towards a more common shared architecture, there were some driving forces which led enterprises towards changing the disparate systems working in parallel to a more integrated architecture which both supports high business competitiveness and globally generated information and knowledge (ibid). Moreover, rapid change in both internal and external environments is a major cause for enterprises to concentrate all the required business processes and implement them in one information system.

According to (Hasselbring, 2000), as value chains extend over enterprises the previous way of limiting business processes in one single application or a number of separated applications cannot be efficient. As the suppliers-customer value chains extends over enterprises the information architecture used by both of the parties will have more commonality than before.

In order to identify the challenges of developing and utilizing Information system integration in an enterprise, it is important to first identify the umbrella concept of system integration to illuminate the first research question. The main purpose of selecting this research area is to define the system integration in general and break down this area into relevant subjects which can provide a suitable answer to the research question.

**Enterprise Application Integration:**

The definition of the Enterprise Application Integration is almost the same as what authors defined as Information system integration, however, EIA is more specifically the paradigm that provides methods, standards, software, and architectural model to address the problem of system integration (Erasala et al., 2003). "Enterprise application integration is the creation of business solutions by combining applications using a common middleware" a middleware is application software that provides a mediation layer for applications (Ruh et al., 2002)

The EAI has different definitions based on the way the EAI is designed to help solving integration problems. It depends on how the applications are integrated using this paradigm to
define this area. For instance, according to (Erasala et al., 2003) EAI is defined as the integration of disparate and dissimilar application systems to share information using a common interface. Other authors define EAI as the unrestricted sharing of information and business processes in a large enterprise among all the information systems. The authors claim that EAI helps to unify business processes across organizations. (ibid)

Traditionally each department within an organization was responsible for managing one or many specific business processes. Consequently, different standalone applications where running business processes separately for each department. However, this traditional way has changed as a result of the changes in business environment surrounding organizations. Disparate activities throughout departments could not work separately since businesses forced organizations to unify data and business functions. (ibid)

Businesses have had a great impact on the usage of EAI by organizations, there are a number of drivers according to (Laftsidis) EAI supports new channels of distributing information like Mobile phones and new electronic devices. E-commerce is another driving force which leads organizations to consider EAI for more effectively sharing and integrating business processes which play important role in gaining maximum advantage of different businesses in each stage of the value chain. Moreover some authors talk about the mergers and consolidations, expanding customer/supplier relationships, and moving information from traditional limits as drivers of using EAI (Linthicum, 2000).

In addition to the impact and drivers which businesses have for using EAI practices and methods in an organization, the evolution of IT also has had a great impact on the different maturity levels which EIA has transitioned through. Many previous works that had viewed EAI from their specific perspective, talked about the impact that the change from homogeneous data and systems to heterogeneous data and systems had on EAI.

In order to answer the research question the literature in this area should be reviewed. This helps the author to gather information about the EAI as a method or paradigm that helps organizations accomplish their integration projects. To find the answer to the first sub-question, the previous literature on EAI which illuminates challenges of integration in general should be reviewed.

### 3.3 Previous Research

#### 3.3.1 Enterprise Application Integration (EAI)

When starting to investigate mid- or even large size organizations the first thing that is observable is thousands of applications that has been designed, developed, implemented and still maintained. These applications require support to be able to respond the time to market needs that pile up in the huge backlogs stacked on one another to be implemented in an efficient, quick and creative way.

For example, for a specific organization designing different types of cars and the items needed to be assembled together to make a functioning part is main business. From the beginning of design until the end of their business processes, several applications are designed to support a specific need or functionality. Therefore when the scenario of user interacting services offered by different applications is analyzed the results show complex data communication among applications involved in the service offering scenario. Moreover, different applications in a scenario can have an effect or change on the data which has been the input to the graphical user interface.
Enterprise application Integration (EAI) is a domain that has been researched since the 1990s (Alsene, 1994). The same author, claims that in his time of writing his research paper, for ten years researchers in social sciences tend to show more interest on computerized integration of enterprises. At the beginning of evolution of this area, the purpose was to unify different functions in an enterprise and avoid functionality of different divisions in an enterprise change to separated components. At first integration of all enterprise activities were the purpose and later came the idea of application which could feed from a central database across the whole organization. Finally the first MRP (Material Requirement Planning) system, which introduced integration of functions of material planning and inventory control, was set up.

(Alsene, 1994), argues about integration attributes and also refers to the historical evolution of system integration from the past until now which could not be found in many literature, presented in a really organized and consistent way. According to (Alsene, 1994),

"Since the early days of computing, organizations have aspired to integrated, enterprise-wide information system architectures. Throughout the years, these aspirations have been reflected in the quest for integrated MIS, enterprise-wide data models, and integrated databases"

Moreover, the historical evolution process of Information Technology integration during the years is worth mentioning. Below, the summarization of historical events regarding IT integration is presented.

<table>
<thead>
<tr>
<th>Time Frame</th>
<th>Dominant Integration Technology Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before mid-1960s</td>
<td>Programmed interface between autonomous applications and data silos</td>
</tr>
<tr>
<td>Mid 1960s</td>
<td>Shared databases especially relational database</td>
</tr>
<tr>
<td>1970s</td>
<td>EDI—interfaces among separate businesses</td>
</tr>
<tr>
<td>Late 1980s</td>
<td>ERP—A set of integrated applications that use a single database</td>
</tr>
<tr>
<td>1990s</td>
<td>Middleware—allows information sharing among heterogeneous databases &amp; applications</td>
</tr>
<tr>
<td>1990s</td>
<td>XML—Standard language for use in sharing information over the web among heterogeneous databases and applications especially between different firms</td>
</tr>
<tr>
<td>Mid 1990s</td>
<td>EAI—Integration among ERP, Legacy Systems, and web applications</td>
</tr>
<tr>
<td>Late 1990s</td>
<td>Web Services, SOA</td>
</tr>
<tr>
<td>Early 2001</td>
<td>Enterprise Service Bus</td>
</tr>
</tbody>
</table>

Table 1 - The history of information system integration (Lee et al., 2003; Ortiz, 2007; Singletary, 2004)
Some authors tried to find a good explanation for this evolving process and motivate why the above stages in time, changed the way that information system integration was used during the years. According to (Erasala et al., 2003; Hohpe & Woolf, 2012) the change from centralized platforms which contained homogeneous processes and data such as legacy systems, to newer platforms such as UNIX and Windows NT was one of the reasons for developing new ways of facilitating disparate applications. The development in programming languages paradigms such as object oriented or component based programming also played an important role in introducing the need to integrate applications because of the fact that applications became different from each other mainly create the integration requirements (Erasala et al., 2003; Roshen, 2009; Sharif, Elliman, Love, & Badii, 2004).

The above table illustrates a snapshot of the evolving nature of Enterprise integration until the emergence of EAI. However, EAI was the beginning of a new era in enterprise integration. According to (Janssen & Cresswell, 2005), EAI is not a specific technology, the staff, users and stakeholders of an organization don't recognize it directly. In addition, EAI is not one technology and needs a portfolio of multiple integration technologies that support different functionalities to implement EAI framework as a whole in an enterprise (ibid).

According to (Vernadat, 2003) as the enterprise modeling has changed from entity-relationship and activity modeling to object and flow modeling, peer to peer system integration has changed to inter-organizational exchanges of knowledge and services. He presents the evolution of enterprise integration as the drastic change from specific communication protocols such as (MAP, TOP, field-buses), various special standards for data exchange. One of the most important protocols was HTML, and complicated data exchange infrastructure which supported distributed computing environments. (ibid)

In 2000, ten years after the introduction of the EAI paradigm, the most common way to do integration was by using Message Oriented Middleware (MOM), with complex configurations to enable integrating applications on computer networks based on TCP/IP (Vernadat, 2003). However, the middleware should provide some extent of criteria which in his work presents qualities such as scalability, security, reliability and integrity. The message format of that time was to use XML (Extensible Mark-Up Language) to enable separation of data exchange format from the logic of the documents and the data itself (ibid).

Moreover, it is argued that integration in general is the result of generating applications and development processes without considering any strategy or a central vision. The problem arises when enterprises divided in different departments, also offering specific services tended to develop single purpose applications considering a set of users without thinking of integrating all of them into larger systems which contain multiple applications (Linthicum, 2000).

According to (Hohpe & Woolf, 2012) enterprise application integration is not an easy task because integration is not just one style or method to be used. The approach used to accomplish integration in a typical organization is related to their applications, developed or bought from third party vendors, operating on different platforms and using diverse technologies inside or even outside the company. Additionally, some applications are not designed to be integrated with other applications but their data is critical to other applications. All of this makes the process of integrating applications complicated and critical to every enterprise (Roshen, 2009; Hohpe & Woolf, 2012; Xin, Yunlong, & Lin, 2009).
In order to implement EAI a standard architecture was needed to follow a pattern of how EAI should be utilized inside organizations. The literature about EAI usually pointed out different types of EAI and then selects which type is more suitable to their research.

(Linthicum, 2000) stated that organizations must recognize the fact that what are their needs for business processes and data, which should be integrated. Based on the requirements there are different types of EAI to implement and choose from. (Lee et al., 2003) pointed out the importance of communication, cooperation, and coordination of business and IT in order to implement a successful EAI solution for and enterprise. Based on (Ken & Tatsubori, 2006), the integration should not be emphasized on the IT area, therefore the alignment between IT and business functions which are accomplished using the applications and providing services to customers play an important role in application integration. In other words, integration should not be done just on an IT level of an enterprise and it should be aligned with both business and IT.

According to (Lee et al., 2003), since EAI is a business-oriented process, before starting to implement EAI projects, it is highly recommended to investigate business processes which need to be integrated. Analysis of the current and target state of business processes is required in EAI projects (Lam & Shankararaman, 2004). The above diagram illustrates different processes that should be followed in EAI projects. It points out the inner circle as the process that should be followed to solve EI problem, the next circles consequently show the deliveries by following the inner process and the risks that should be taken into account to deliver a fruitful EI project at the end (ibid).

The emphasize on investigating business process identification before commencing EAI projects has been pointed out in previous research on this area, thus neglecting this critical stage of EAI projects can result in defects and failures. As such, implementing EAI, like any other IT project requires understanding of the structure of organization to ensure the
stakeholder commitment, knowledge sharing, considering costs and benefits (Janssen & Cresswell, 2005)

3.3.2 Architectural approaches to EAI

The research on EAI, generally presented three types of architecture to implement EAI, since EAI comprised of different methods, tools, and techniques to implement a specific target solution ((Lam & Shankararaman, 2004)). Two types of architecture has been defined with their specific pros and cons in which the literature motivates the reason why one is more utilized among EAI practitioners. The architectural design of the EAI is built based on the integration requirements gathered in early stages of the project. The architecture is comprised of complex collection of technologies which should be mastered by architectures before design of the integration architecture (Lam & Shankararaman, 2004).

- **Point to point integration**
  In this type of architecture, applications are integrated in the point-to-point approach, a separate integration program connects each source and target application (for managers). In this approach the applications usually have specifically defined data type and format and the traditional architecture supported which directly links one system to the other (Lam & Shankararaman, 2004).

- **Spoke-and-hub integration**
  In this type of architecture, applications are connected to a central hub which is usually called "Message Broker", facilitating of message exchange between integrated applications. Message brokers also are engines that are responsible for the transformation of message formats between the two applications (Chappell, 2009). One of the important characteristics of message brokers is that the message exchange is an asynchronous communication model. This helps the independent operation of applications, this way the source applications sending a request and does not change to idle mode until getting the result of their request (ibid).

- **Service Oriented Architecture**
  According to (Swithinbank et al., 2007), SOA is the architectural approach for integration based on the concept of service. The business functions needed for building distributed systems can be defined as service to end users or other applications which also provide other services. In order to implement business processes a sequence of services that can communicate with each other is needed. It is a type of architectural design that creates association between business and computational resources which are organizations, applications and data. There are different specific implementations of SOA reference model using web services but web services such as .Net, J2EE, CORBA are not SOA (Valipour et al., 2009).
Researchers in the EAI area generally tried to provide common guidelines or frameworks to grant usage of the above methods in practice. Therefore many articles try to compare and contrast the characteristics and values of the three architectural designs and mostly talk about point to point integration as "Traditional" which means it has been used before and is not acceptable in current situation. Moreover, the emphasis is on SOA as the most advantageous approach. Below the summarization of comparison of the most common architectures is presented:

- **Costly Solution** - According to (Lee, Siau, & Hong, 2003; Erasala, Yen, & Rajkumar, 2003), point to point architecture is more costly in time and money due to the fact that for each pair of applications specific code needs to be implemented to connect the other one.

- **Agility and flexibility** - In hub and spoke, on the other hand, applications connect to a middleware through a common interface layer, thus message broker eliminated the need for writing extra code for each application separately, this method is more agile and quicker to change (ibid).

- **Scalability** - The first architecture had scalability problem, if the number of integrated application increases then the effort and time needed to produce integration connections increases as well, as a result "spaghetti integration" is built (Puschmann & Alt, 2001). (Linthicum, 2000) argued that the major problem with the hub-and-spoke model is that it is not scalable because all the information from applications connected to the hub server will be consumed and if the number of applications increase there will be resource-limitations since the huge amount of information ends up in one server and consume the available resources and there will be performance issues as well. Moreover, (Goel, 2006) discussed that a single hub is easy to manage but scalability is the most critical issue with this approach. He argued that if the number of messages increase hardware constraints and performance issues will produce a single point of failure which is caused by the dependence of scalability to the hardware. To overcome
this issue organizations use a federated hub-and-spoke architecture which contains multiple hubs instead of a single hub (ibid).

- **Complexity**- Hub and spoke is logically similar to star topology in networking. This approach tends to minimize the number of connections needed compared to point-to-point integration. For a sample with "n" entities to be integrated, for point-to-point integration, \( n \times (n-1) \) connections needed. While, with significantly lower cost in integration and maintenance, hub-and-spoke, produces \( n \) connections for the same number of applications (Erasala et al., 2003)

### 3.3.3 Different types of EAI
Different authors provided knowledge foundation for types of EAI in various ways. In the process of implementing EAI, in addition to the target architecture as well as the data and business processes which should be identified for integration purposes, the type and level of integration should also be aligned to the purpose of integration project. (Lee et al., 2003; Linthicum, 2000; Puschmann & Alt, 2001) pointed out the following EAI types of integration.

- **Data integration**
  The most common level of integration between applications is the sharing of data, not the application business logic (Lee et al., 2003; Linthicum, 2000). EAI expanded the traditional data level integration, by providing a common framework for integration (ibid).

- **Application interface integration**
  This level of integration enables linkage of business processes and the data at application interface level (Lee et al., 2003; Linthicum, 2000).

- **User interface integration**
  User interface is used as common point of integration for developers. This type of integration is mostly used to access databases and processes of legacy systems.

- **Method integration**
  This method provides share of business logic by implementing components that can be leveraged throughout the enterprise. (Linthicum, 2000) pointed out that method integration was targeted at as the next level of integration solved by EAI. Method integration has a tight connection to the concept of "reuse".

Additionally, other researchers added more categories to this list and tried to compare and contrast different types of integration referring to the relevant literature (Al Mosawi, Zhao, & Macaulay, 2006). In their research also considered the following was also considered:

- Object integration
- Presentation integration
- Process integration
- Internal Process integration
- Cross-enterprise integration

(Puschmann & Alt, 2001), in their analysis of categorized different types of applications which needs to be integrated as:

- **Homogeneous with one instance**
  In this case each process is supported by one database and one application
• **Homogeneous with several instances**  
  Different business units are responsible for running similar business processes that run on similar applications with separated databases.

• **Heterogeneous**  
  Several different business processes in different business units supported by different applications. The problem here to cope with is the various data models that applications are built upon and this requires different semantics of the data that need to be exchanged, therefore to be able to translate these different semantics a common communication model and a common language is required for this level of integration.

**Technical and Behavioral Integration**

(Lee et al., 2003, p. 56), argued that enterprise integration is a critical factor to have a more agile and flexible organization. They cited that “agility and flexibility means to *continuously monitor market demand; quickly respond by providing new products, services and information; quickly introduce new technologies; and quickly modify business methods*”. They argue that to achieve agility both technical and behavioral integration is demanded. Integrating software and hardware is called technical integration which is one side of integration. Behavioral integration is somewhat bigger challenge as it redistributes the roles and positions and if the behavioral integration is not managed correctly this could be detrimental to the organization. Change management and transformational issues are interconnected together.

**Loose integration**

There are different levels or degrees of integration according to the literature namely loose or tightly coupled integration. These two characteristics define the degree of dependency applications have on each other, while the integration business process tends to require different applications to interact with each other to produce data needed for the service consumer.

In recent years the concept of loose coupling has received major attention by enterprises, based on the fact that the main result of loose coupling is more flexible, simplistic, and effective integration among applications (Papazoglou, Traverso, Dustdar, & Leymann, 2007). Usually loose coupled integration in research is tightly connected to the definition of SOA architecture. According to (Jammes & Smit, 2005), based on SOAprociples services are publishes based on a contract that provides information of the service functionality with a schema design defining the rules for both service provider and consumer to follow.

**Tight integration**

Tight coupled systems have many unknown dependencies to other systems, therefore if an application needs a service from another application, it will be in wait mode and subsequently can not continue to run other tasks until it gets a response from the other application which provides data or service(Valipour et al., 2009). Based on arguments by (Jammes & Smit, 2005), tightly coupled interactions between entities or applications create dependencies which are context-related and stateful and result in inflexible and fragile systems.
3.3.4 Service Oriented Architecture

In order to define the role of SOA in system integration the related literature in this area was reviewed to illustrate the characteristics of SOA which are beneficial to EAI as a new architectural approach for EAI. It is noticeable that the literature usually talks about SOA and web services, because web services are one of the suitable methods to implement SOA architectural model in practice.

The first characteristic of SOA is the definition of service in different literature. (Hohpe & Woolf, 2012) identified Service as Shared business functions which are well-defined and universally available and responds to requests from “service consumers”. According to (Valipour, Amirzafari, Niki Maleki, & Daneshpour, 2009), SOA is comprised of services that are modularized. These modularized services can then provide coordinations to support real-time business processes to function correctly throughout the enterprise. The authors pointed out that SOA is the result of evolution in programming languages and paradigms, distributed computing and business technology (Valipour et al., 2009; Krafzig, Banke, & Slama, 2004).

According to (Krafzig et al., 2004), the progress in programming languages contributed to the development of interfaces that provide access techniques to services in SOA based architecture as well as implementation of platforms that are elements of SOA. Moreover, the evolution in distributed computing provided underpinnings for a variety of remote access technologies to choose from and advances in business computing was one of the role playing factors to implementation of ERP, CRM systems which are the basis for data and business logic in service oriented structure (ibid).

Based on (Li et al., 2010) research, SOA is aiming to solve the problem of system integration and interoperability providing new integration patterns and supporting system infrastructure. SOA key concepts such as service orchestration and encapsulation develop application integration through certain mechanisms to accomplish specific business processes, while in their work they point out the challenges and bottlenecks related to service access security across the organization and related service orchestration (ibid).

SOA articles usually point out the concepts such as loosely coupled software services with defined interfaces that are accessible without any knowledge of the way the platform is implemented. Loose coupling in SOA refers to the number of known dependencies that are not so many versus tight coupling which is defined as many unknown dependencies between service provider and service consumer (Krafzig, Banke, & Slama, 2005; Roshen, 2009; Valipour et al., 2009). (Swithinbank et al., 2007), stated that SOA is mainly related to business concepts based on which information systems are designed information processes to support business functions. On the other hand, EAI is based on IT investments on information and infrastructure to provide patterns and products for integration of applications by reusing existing functionalities. It was also noted that, SOA aligns IT services more closely to bring value to fundamental business by using IT (ibid).

According to (Hohpe & Woolf, 2012), in SOA integration of a new application is done by using existing remote services provided by other applications, thus calling a service can be regarded as integration between applications. SOA based integration tools usually provide enough simplicity to call an external service the same as a local method. However, to be able to define the benefits of SOA for integration it is valuable to refer to literature which cited the characteristics of SOA which are beneficial to system integration.
3.3.4.1 Benefits of SOA for system integration

According to (Papazoglou et al., 2007; Valipour et al., 2009) specific characteristics of SOA make it a novel approach compared to previous architectural patterns of integration. Previous literature argued about these characteristics and supported the benefits of SOA for system integration (Rehan & Akyuz, 2010).

- **Stability of service interface** – developing interfaces is expensive and they are one of the main origins of costs in enterprise system implementation process. Stability of interface services and their relation to the percentage of change forced by customers can help lower amount of costs (Gulledge, 2006).

- **Service isolation to consumers** – the development of services is confined to the needs and requirements of service consumers therefore the extent of change and improvements are limited to consumers (Valipour et al., 2009).

- **Reusability and less re-implementation costs** – the concept of reuse is stated in numerous literatures and its relation to SOA is clearly defined based on the fact that SOA is the key enabler of reuse of services and business functions without the need to implement them. All the implemented services are accessible from service repository (Rehan & Akyuz, 2010).

- **Providing a simple scalable paradigm** – scalability is the ability to extend a paradigm, SOA is a scalable paradigm because it minimizes the extent and scope of assumptions each application has on the other applications which is integrated via SOA paradigm (Valipour et al., 2009).

- **Supports agility** – the flexibility and adaptability offered by SOA promotes the business agility and change in contrast to previous integration approaches which support exponential point to point interface connection and interaction (Krafzig et al., 2004).

- **Interoperability** – standard-based communication of applications in SOA architecture underpins the interoperability of application on different platforms, communication protocols and languages (Chappell, 2009; Valipour et al., 2009).

- **Self-healing** – ability to recover from possible errors and return to the previous state (Papazoglou et al., 2007).

- **Location transparency** – According to (Valipour et al., 2009), location transparency is one of the most important characteristics of SOA that supports more availability and performance enhancement of SOA integration.

- **Service orchestration** – each business process executes using one or more service interactions and this character of SOA means that the during business process execution many services are invoked by some events (Papazoglou et al., 2007).
• **Service choreography** – is related to public message exchanges, general rules for interaction between services and the contracts and agreements between applications or services of different parties rather than a single party (Papazoglou et al., 2007).

Many of the above characteristics are repeated over and over in different research projects to emphasize the contribution of SOA to system integration patterns. According to (Papazoglou et al., 2007), the benefit of using SOA approach is loose coupling allows to break down the integration logic into distinct and easily manageable pieces. Moreover, service orchestration and service choreography are two characteristics that mainly define the interaction protocols coordinating and controlling how services are collaborating.

### 3.3.4.2 Challenges related to SOA

Many organizations rushed into migrating to SOA from the time it was introduced as a new paradigm based on many new benefits that could solve integration issues caused by previous approaches, however, in this section the author investigated the literature to bring up the possible challenges of SOA adaptation and migration for organizations.

The challenges of SOA can be categorized into different sub-categories such as migration and adaptation, design, and test. In this section, I tried to identify specific challenges related to each sub-category in previous literature. According to (Umar & Zordan, 2009), the SOA integration can be divided into two types of approaches that each of them can contribute to some issues:

- Migration – change the applications architecture internally to newer versions, usually included legacy systems which don't follow the concept of SOA.

- Integration in-place – applications are not modified to reusable components; instead, they are connected using web services or Enterprise Service Bus.

(Umar & Zordan, 2009), argue that regardless of which type of integration, summarized the pitfalls of SOA integration such as:

  - If the target applications are too inflexible and costly to maintain, the integration in-place approach does not work.

  - Outdated and old functionalities will remain in the system and also the possibility of using a new product more flexible to SOA approach will be ignored.

  - SOA is producing a great deal of confusion due to its specific array of standards and new products.

According to (Krafzig et al., 2004), pointed out a number of key factors to consider as a challenge to introduce SOA in a large organization.

- Introducing SOA is a change and usually there will be some resistance against it.

- SOA will create overhead which needs budget allocation in early stages of the project.

- The technical standardization of SOA-related technologies is in progress.
• Potential service consumers must be involved in the design process of services.
• The designed services must provide reusability which creates overhead for developers and architects.
• To implement services, the business specifications of the services should be clearly defined for developers.
• The current infrastructure should support migration and adoption of SOA.

The authors also argue that, if the environment of the enterprise is right SOA can produce its benefits. Introduction of SOA is usually a long lasting process and the effects of this architecture can be revealed during the process and not all at once (Krafzig et al., 2004).

According to (Mahmood, 2007), SOA-based integration requires a great deal of investments both in technology and development. He argues that the return of investment is not going to reveal quickly. On the other hand, the combat of efficiency versus flexibility always exists and SOA is not an exception.

Moreover, (Feuerlicht, 2006) discussed that the challenge in SOA projects is not only the agreement on the technical standards and adopting technology standards are essential requirements to achieve integration with lower cost between disparate platforms. Issues such as service analysis and agreement on data structures across the organization are also important to be considered. It is also argued that in the process of implementing based on SOA, some standards are mature enough to be used while others are in the maturity process which provide more functionalities, however, the complexities and policies related to technologies make it harder to deliver stable solutions (ibid).

According to (Maurizio, Sager, Corbitt, & Girolami, 2008), SOA implementation is affected by IT governance. It is stated that SOA governance should be enforced as an extension to the IT governance that is mainly dealing with decision rights, processes, and policies that are put together to persuade adopting and implementing of SOA. Therefore, the clear identification of roles and responsibilities based on an effective governance framework contribute to successful SOA projects. Based on (Varadan, Channabasavaiah, Simpson, Holley, & Allam, 2008), the most important challenges in SOA adoption and implementations are “technology, program management, organization and governance”.

• **Technology** – introducing new technology requires new skills at different levels supporting the implementation, development, and infrastructure design. Therefore centers of excellence should be established to deliver needed skills for new technology adoption.

• **Program management** – The service oriented concept is new to organizations that various entities do not cooperate with one another. Program management can clarify responsibilities, deliverable, contracts and the way different entities should negotiate. Moreover, it will have effect on cooperation of domain experts, stakeholders and the ability to manage the challenges of service-based development.

• **Organization and governance** – Service oriented model shifts the IT organization toward a new stage which changes the business process from individual units processes to a more overall business process. Moreover, a good approach selection and specified SOA governance enforces a shared vision of services between business and IT, and defines service life cycle.
3.3.5 Different approaches to integration
(Hohpe & Woolf, 2012) in their concrete reference to application integration patterns they discussed about the technical approaches which are used in projects implementing different integration styles mentioned in the previous section.

In this section, the author tries to discuss and summarize different technical approaches in integration projects with focus on data integration, finally the more emphasis will be on the most recent approach and review different motivating theories and reasons supporting them.

According to (Thullner, Schatten, & Schiefer, 2008), there is more than one integration approach or solution; these approaches can be used to wire the two applications in different enterprise application integration projects. Below the different technical approaches are summarized (Hohpe & Woolf, 2012; Roshen, 2009).

- **File transfer:**
  This type of integration simply is sharing data in a file and sharing the data between applications. It is important to mention that the two applications should implement this mechanism in a way that both parties can read and write the same data type on the file. It is common to use the FTP protocol to send the files between different application servers.

- **Shared Database:**
  Applications write data on a common database in this case. Since all applications can communicate with SQL language to the database, the data update is quicker than in the previous integration method. Based on the fact that all the applications in different platforms can implement SQL, the integration process is simpler than the File Transfer method as well as the point to point integration no longer exists like the previous method.

- **Remote Procedure Call:**
  This method is used to share application functionality among divergent applications. Therefore the procedures of one application can be remotely invoked by other remote applications. In this type of integration one application plays the server role which usually implements the functionality needed by client applications to remotely call and use it.

- **Messaging:**
  In this type of integration applications get connected to a messaging system including different mechanisms and technologies to be able to support asynchronous messaging which helps to transfer data and calls behavior using messages. In this type of integration the client or service users send a message to the server or service provider but does not wait for the reply and continues its process, therefore applications are loosely coupled and messaging is asynchronous. This type of integration is useful when large amount of data and transactions are happening frequently among applications.

According to (Pautasso, Zimmermann, & Leymann, 2008), choosing between different styles to integrate systems is a significant architectural decision which affects the requirements to identify a suitable middleware platform and features that the integrated system will provide. Moreover, they argued that it is important to investigate capabilities offered by different styles to be able to find the right styles for integrating systems.
3.3.6 Comparison of different approaches in Enterprise Integration

In this part, the approaches mentioned in previous sections are compared and contrasted in order to get an overview of which method or technique is better to be chosen in integration projects. Enterprises usually tend to choose different approaches in a combination style which suites the requirements in integration projects (Dossot & D’Emic, 2009; Hohpe & Woolf, 2012).

The table presented below summarizes the most important criteria to be considered to choose the best integration technique among others.

<table>
<thead>
<tr>
<th>Method Criteria</th>
<th>File Transfer</th>
<th>Shared Database</th>
<th>Messaging</th>
<th>RPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loose Coupling</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integration Simplicity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tight coupling</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data format Standardization</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data inconsistency</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Developer involvement</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Sharing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Functionality Sharing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data timeliness</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A-Synchronicity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2 - Criteria for different application integration methods (Dossot & D’Emic, 2009; Hohpe & Woolf, 2012)

The table illustrates various criteria which were selected based on previous works done on Enterprise application integration patterns. In a large enterprise with a high degree of diversity in applications used, it is always important to choose the pattern which works better based on the requirements specified in integration projects.

- **File Transfer** – is a tightly coupled approach, needs applications to be implemented for the file format which is agreed on between two applications. It has data integrity problems as well as involves application developer in the integration process, most importantly there is a lag between each updates in the data shared between applications (Curl & Fertalj, 2009).
**Shared database** – approach tries to omit the data integrity problem by using a unified database for all the applications. Moreover the problem with the data access in a timelier manner has been changed to a faster approach to receive the most recent data. However, it is not usually possible to design a unified schema for database which suits all the applications. Moreover the external packages usually have their own schema and also when the number of applications is significant there will be performance and scalability problems as well (Curl & Fertalj, 2009; Hohpe & Woolf, 2012).

**Remote Procedure Call (RPC)** – introduced another aspect in integrating applications which leads to tight coupling. The issue with RPC is the remote call always will not be successful due to network errors and change in the remote procedure. The tightly coupled applications can hardly be modified and interpretability between programming languages and RPC standards are still not solved (Curl & Fertalj, 2009).

**Messaging** – helps loosely coupled applications with the possibility to transfer data and share functionality. It offers a combination of the previously mentioned integration methods. This mechanism is suitable when numerous applications need to transfer data in an asynchronous way meaning that applications on both ends should not be ready at the same time to communicate (Curl & Fertalj, 2009).

According to the previous section, the discussion about different styles of application integration led to this point that based on the criteria presented messaging is a better solution. It decouples applications like what is happening in file transfer but it uses faster mechanism with smaller data packets. The data is transferred in asynchronous way using mechanisms to notify receiver application about the new data or change which has arrived on its end. Furthermore applications can share functionality using this mechanism, in case there is a need for method invocation in other applications, a message can be sent containing the information of the remote function. The messaging combines all different integration styles in such a way that brings more advantages which match the criteria needed for application integration (Hohpe & Woolf, 2012; Rademakers & Dirksen, 2008).

In addition, the most important part of messaging is the transformation of message content as well as the ability to route messages to the target application. But how all these requirements should be implemented in one single solution? Enterprise Service Bus is the implemented solution which resembles the traditional hardware bus, moving data along a universal channel between applications (Ortiz, 2007).

### 3.4 What is Enterprise Service Bus?

"An ESB is a standards-based integration platform that combines messaging, web services, data transformation, and intelligent routing to reliably connect and coordinate the interaction of significant numbers of diverse applications across extended enterprises with transactional integrity." (Chappell, 2009, p. 6)

According to (Rademakers & Dirksen, 2009), an ESB can be a pattern or a product used for simplification of the integration and functionality reuse in application integration. It is usually defined with Service Oriented Architecture; however it makes the service consumer decoupled from the service provider's implementation and support more scalable backbone
for SOA which helps the point to point interaction between service consumers and service providers be replaced by a centralized bus which routes messages back and forth between applications (Thullner et al., 2008), (Parlanti, Pettenati, Bussotti, & Giuli, 2008). The figures below illustrate the role of ESB in decoupling service consumer from service provider in an SOA implementation. In this figure, towards the use of SOA the attempt is to decouple service provider from service consumer, therefore the client does not need to know how to separately communicate with service providers and if a service provider is changed then the consumer has to change as well. By using an ESB the only thing that a service consumer knows is to request (Richards, 2006).

Based on (Schmidt, Hutchison, Lambros, & Phippen, 2005), ESBs are enablers for SOA providing connectivity layers between services. Their emphasis is on the characteristics of ESB to provide underpinnings for the promise of SOA “Publish-Find-Bind”. A service is the meta-data driven description of a component which can be understood by a program. The published meta-data which provides identification of the service component enables other disparate applications to reuse the service. An ESB provides visualized service registry both for requesters and providers of the service to have updated and visualized relationships and interactions (ibid). It is important to note that ESBs are not aware of the business logic of service providers and service consumers. Moreover, they don't include the containers which host the services.

(Dodani, 2004) describes ESB as the “Service Broker” for SOA which provides mediation capabilities for implementing flexible SOA via below capabilities:

- It is a communication middleware that supports various protocols, APIs, platforms.
- A mechanism for intelligent processing of service requests and responses.
- A system for managing loosely coupled applications and their interaction.
- Providing standard-based tools for facilitation of swift service integration.

According to (Robinson, 2004), it is important to first define what we mean by an ESB, is it a product, technology, standard. ESB is a collection of infrastructure capabilities implemented by middleware technology and acts as an enabler for the concepts of SOA. ESB facilitates service, message, and event-driven interactions in heterogeneous environments which supports suitable service levels and manageability.

It is important to note that, the literature usually compares ESBs as a distributed infrastructure to other solutions such as “message brokering” technologies which provide underpinnings for
Hub-and-Spoke architecture. According to (Robinson, 2004), this is forged dissimilarity because the two technologies focus on two different requirements. It is always important to investigate the needed physical distribution to capabilities of the candidate technology to find a suitable match as a solution (ibid).

### 3.4.1 ESB's core functionalities

An ESB product should provide a number of core functionalities to be utilized in application integration, below is the summarized list of feature according to (Chappell, 2009; Rademakers & Dirksen, 2008; Robinson, 2004):

- **Service location transparency** – Allows to share services across organizations, the communication protocol usage promotes location transparency and interoperability.

- **Transport Protocol conversion** – This capability allows integrating applications with any transport protocols like HTTP(S) to JMS, FTP to a file batch, etc.

- **Message transformation** – Allows the transformation needed to change the format of a message which includes a business service request to the structure and format that a service provider expects using open standards such as XSLT, JAXB, and XPath.

- **Message routing** – This allows to channel a request to a particular target application or the destination based on deterministic or variable routing criteria

- **Message enhancement** – The ability to add missing information to an incoming message which should be contained in the message for service provider.

- **Security** – All services are available via ESB throughout the enterprise and there should be security mechanisms implemented to prevent unauthorized access to services.

- **Message processing** – This allows to be sure that the message sent requesting for a service delivered even if there is some problem preventing this.

- **Transaction management** – This allows providing a framework for the coordination of multiple resources across multiple disparate services.

The above functionalities are the most common ones that an ESB should provide to the integration projects. The combination of the above mentioned functionalities are needed in every ESB product to be able to implement a functioning ESB as an architectural infrastructure supporting SOA (ibid).

According to (Ji-chen & Gao, 2006), with the advent of Service Oriented Architecture many organizations experienced flexible connectivity, and standard-based application integration using Web-services. However, Gartner predicted that during 2003 Enterprise Service Bus can play an important role in application integration projects and the trend will continue to grow starting with smaller organizations in 2003 and then will be utilized by larger organizations (Thullner et al., 2008).

(Ortiz, 2007) shows that according to the investigation done, between years 2006 and 2012 worldwide enterprise service bus application's market share has increased. Yet another research done by Wintergreen research discusses that: "Enterprise Service Bus (ESB) markets at $190.5 million in 2006 are expected to reach $494.4 million by 2013. Market growth
comes because ESB enables the flexible IT architecture that is needed to respond to market
shifts brought by speeded product cycles and competitive challenges’ (ibid).

3.4.2 The Enterprise Service Bus Model
The previous literature in general provides the following conceptual model to introduce ESBs.

![Figure 3-5 - Enterprise Service Bus model (MuleSoft, 2013)](image)

According to the above conceptual model ESBs play a mediator among different applications
based on different platforms and data types. The integration mechanisms provided by an ESB
assists the easier way of adding new applications to the integration landscape. Applications
communicate with each other through their connection to the ESB, while the complications of
implementation the logic behind the integration is mainly dealt by the ESB (Chappell, 2009;
Robinson, 2004).

According to (Menge, 2007), ESB supports transactions among different resources and the
main importance is not to write extra code and be able to transfer messages and integrate to
other applications.

3.5 Open source /Closed source ESBs
The debate between benefits and drawbacks of using open source products or closed source
never ends. Each community claims to be a better solution both from software developments
and design as well as deployment and utilization (Thullner et al., 2008).

According to (Rademakers & Dirksen, 2008), the commercial vendors of ESBs provide the
new functionality which is built upon their previous EAI products. On the other hand, there
are other vendors both in the commercial and open source market which offer products built
from the ground up.

The difference between closed and open source ESBs is the usage-based license fee. The open
source ESBs usually have license such as Apache or GPL but their source code is open and
they don’t have usage license fee (Rademakers & Dirksen, 2008).
3.5.1 Source code availability
Using open source product helps to make the maintenance and enhancement process for the users a quicker one because by the open source product the code is available and the developer can fix the bugs and problems if they occur, otherwise the process of the request for the fixes and the release to the test and finally product environment can take some time as well as it needs contracts between the vendor and the customer for the support and enhancement (Thullner et al., 2008).

3.5.2 Open standards vs. vendor lock-in
According to (Rademakers & Dirksen, 2008), the previous EAI products mainly, IBM's WebSphere message Broker, TIBCO's Business Work, and Sonic XQ implemented the messaging and transformation logic by using proprietary technology, this forces the user to be locked in the technologies used by the vendor, however, the Open source ESBs provide the user with a wide range of open source frameworks such as ESB products are based on open standards, such as Java Message Service (JMS), XML, J2EE Connector Architecture (JCA), and web services standards. These frameworks are helpful to extend the functionalities needed and therefore the customization of the product can be done easier with less timing and less cost.

3.5.3 Support communities
In this case usually the support is not given directly by the company offering the software package and it will be outsourced to other companies, therefore the communication between the users and the developers who were part of the project is not possible. On the other hand, behind an open source product there will be a support community where users can directly communicate directly with lead developers of the products, this will be helpful for both parties, however, if that is not enough then there will be commercial support offered by open source vendors outsourced to other companies as well (MuleSoft, 2013).

3.5.4 Cost
The economical aspect of using open source and closed source packages or frameworks lies in the amount of budget a company can spend on either if the project using the open source framework is successful or not (Thullner et al., 2008). According to (Thullner et al., 2008) if the utilization of open source is successful in a company it will have less impact on the labor costs introduced by a failure project. Moreover, a commercial software vendor earns revenue by renewal in license and support for the software, and update programs and in some cases stops the support for a product because of phase out or stop in further development. On the other hand, when the open source product is used the cost can be controlled by the company not the vendor (Thullner et al., 2008).

3.5.5 Open source ESBs
Open source ESBs are built on open standards and with the increase in utilizing of open standards such as JMS, JBI, SOAP, and XML in integrations the ability to code in Java and do configurations in XML is needed from developer side. Open source standards allow for low entry cost, maximum flexibility, reuse, and investment protection (Ji-chen & Gao, 2006). The open source market offers a rather significant number of open source ESBs which makes it hard to decide which is the best, however, in this study the focus was on a product which attracted more attention in recent years and is mentioned in different journal articles as one of...
the possible options to be used in integration projects (Ji-chen & Ming, 2006), (Sanjay P. Ahuja, 2011), (Parlanti, Pettenati, Bussotti, & Giuli, 2008)

Apache Service Mix\(^1\), Mule ESB\(^2\), and Open ESB\(^3\).

The table below summarizes the needed functionality and features in the above mentioned open source ESBs (Rademakers & Dirksen, 2008)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Mule</th>
<th>Service Mix</th>
<th>Open ESB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support for ESB core functionality: location transparency, transport</td>
<td>Good</td>
<td>Good</td>
<td>Average</td>
</tr>
<tr>
<td>protocol conversion, transformation, routing, message enhancement,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>security, and monitoring and management</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Documentation and user friendly manuals</td>
<td>Good</td>
<td>Average</td>
<td>Good</td>
</tr>
<tr>
<td>Market visibility</td>
<td>Very Good</td>
<td>Good</td>
<td>Average</td>
</tr>
<tr>
<td>Flexibility/customizability with custom logic</td>
<td>Very Good</td>
<td>Good</td>
<td>Average</td>
</tr>
<tr>
<td>Supporting wide range of Transport Protocols and connectivity endpoint</td>
<td>Good</td>
<td>Good</td>
<td>Average</td>
</tr>
<tr>
<td>Integration with other open source projects</td>
<td>Very Good</td>
<td>Very Good</td>
<td>Average</td>
</tr>
<tr>
<td>IDE support</td>
<td>Very Good</td>
<td>Good</td>
<td>Good</td>
</tr>
</tbody>
</table>

Table 3 - The Assessment summary of three Open Source ESBs, (Rademakers & Dirksen, 2009)

(Sanjay P. Ahuja, 2011) provided different performance tests among three open source ESBs that Mule and Service Mix where part of the tests. The table below summarizes what they have shown as subjective assessment.

<table>
<thead>
<tr>
<th>Features</th>
<th>Mule</th>
<th>Service Mix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>Code base/Examples</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>Ease of Development</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>API/Documentation</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>Online help</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>Community/Forums</td>
<td>A</td>
<td>A</td>
</tr>
</tbody>
</table>

Table 4 - The subjective assessment summary (Sanjay P. Ahuja, 2011)

**The Proposed ESB**
The market for selecting a product as an ESB provide companies with a list of different vendors, including the major ones such as IBM, Oracle, Microsoft, TIBCO. Some of the

---

\(^1\) http://servicemix.apache.org
\(^2\) http://www.mulesoft.org/
\(^3\) https://open-esb.net
products offered from the mentioned vendors are built on their previous solutions for enterprise application integration. Furthermore, there are vendors who built their product from scratch. Based on the previous works done on enterprise service bus as a product used for implementing scenarios, Mule ESB\(^4\) from (MuleSoft, 2013) is proposed to be the basis for this work.

*MuleSoft* community which is built to support the users all around the world the most suitable data gathered as well as literature supporting the selection criteria for this product. The combination of all those comments and feedbacks guided this work to be a test basis for using this product as a solution to one of the integration problems which can happen not typically in case under investigation but also in other companies where there exists a combination of applications tied together for the sake of data and functionality integration.

### 3.5.6 Motivation for choosing Mule ESB

According to (Richards, 2006) the first criteria for choosing a typical ESB is the core ESB functionalities that are needed in integration projects. With reference to figure 3.2 as well as combining the requirements from governance guidelines of company of case, it is possible to map and compare key requirements and functionalities offered by MuleSoft's product (Sanjay P. Ahuja, 2011). The table below summarizes the features offered by MuleESB which is worth mentioning in order to map the requirements and the offering compared to other ESB products in open source market.

<table>
<thead>
<tr>
<th>Selection Criteria</th>
<th>Mule ESB</th>
<th>Matched with requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support for ESB core functionalities such as: loose coupling, routing, transformation, connectivity, etc.</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Supporting common standards for data transformation, different transport protocols, common open libraries</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Simplicity and user friendliness</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Performance</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Market visibility</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Authentication/Security</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Support for open source frameworks</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Customizable user defined logic</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

*Table 5 - MuleESB summarized features*

### 3.6 Mule ESB

According to (MuleSoft, 2013), Mule is the most widely used open source ESB in the world with numerous users supported by active community. It has two editions namely, community and enterprise which can be installed on Mac, Windows, and Linux. The most current version which is also used in this work is 3.3. The community edition is free to be used and it works under the open source CPAL license.

The community edition has some restrictions regarding the messaging servers supporting our test scenarios, instead of using Native Websphere MQ which is a message broker from IBM, the Apache ActiveMQ was used which is an open source JMS broker. According to

---

\(^4\) [http://www.mulesoft.org/](http://www.mulesoft.org/)
Mule does not focus on the transport layer in application integration, it provides functionalities such as routing and transformation but does not care over which of the supported transports such as HTTP, JMS, TCP, SMTP, etc. the messages are transported.

### 3.7 Environment set up

The platform installation is not a hard task; however some environment variable setup should be done after unzipping the package that Mule comes with to a suitable folder. Apache ActiveMQ software also installed in a similar way and further setup can be done from Mule's IDE which contains built-in Eclipse for development purposes. This helps to add customized functionalities which are not offered with the community edition.

For testing the scenarios with Mule ESB there were number of choices for database server and the full installation of MySQL was chosen since it is rather easy to be configured with Mule to minimize the configuration burden, however, Mule supports different database vendors and it is possible to use all of them. Full installation for MySQL 5.2 CE was done in this case. In addition, for setting up connectors and be able to connect to a data source, specified JDBC driver which is a .jar file should be installed and referred to Mule's IDE (Ahlberg, 2010).

### 3.8 Message Flows

In Mule Studio it is possible to develop and design different integration patterns using different built-in tools. According to (Rademakers & Dirksen, 2009), Mule offers the message flow as the intermediary layer between service provider and service consumer. Based on the theoretical part of the thesis, ESBs main functionality is to provide the mediation layer offering message routing and transformation. Mule offers the mediation layer needed for the integration case scenarios. It provides message transformation and routing for different types of service consumer and service providers. In addition, the connections to different components are possible if the needed configuration files and drivers be added to an integration message flow (Ahlberg, 2010; Rademakers & Dirksen, 2008).

As a system integrator the task is made easy in the IDE level, therefore the integrator is not involved in the lower level development and all the configurations are done based on XML, Plain Old Java Objects (POJO), and Spring. Below a conceptual example of a message flow is presented
3.9 Summary of Theoretical Findings

In this section the summary of results from the theoretical and literature review is presented to be able to give explanations regarding the results gathered in this stage of the study. The connection between research questions and the investigated areas helps the researcher to provide explanations for the research. Here the author presents the results to be able to answer research questions presented in the previous chapter.

Research Question: What is the impact of ESB on integration projects in an enterprise?
This question is answered by answering the two sub-questions. Below the summarized information gathered is presented.

- The first sub-question: What challenges do system integration present?
  There are different components that can lead to challenging integration projects both from a technical and business point of view.

  Challenge of Integration Projects management
  From the past different methods, approaches and architectural patterns were introduced to provide a suitable solution for system integration. In (3.3.1) a conceptual model was provided illustrating integration project's different stages, deliverables in each stage and risks. There are number of risks in integration projects mentioned that can be summarized as follows:

  - Integration requirement gathering, Business and IT coordination.
  - Business process and data identification.
  - Behavioral integration is more challenging than technical integration section (3.3.3)

  Issues of IT Governance
  Enterprise integration on a large enterprise scale usually involves thousands of applications developed on different platforms, frameworks, standards. An IT organization usually owns IT architecture which provides governance and standards for integration projects. In section (3.3.4.2) it is noted that successful integration projects have clearly identified IT governance which define roles, procedures and responsibilities.

  Challenges of old technologies
  - Costly solutions
  - Agility and flexibility
  - Scalability
  - Complexity
Challenge of new technology
Refer to (3.3.4.2)
- Migration of inflexible system's architecture such as legacy systems to support new technologies
- Confusion due to a new array of standards and tools
- Resistance to change
- Current infrastructure support the change to new technology
- Budget investment and slow return of investment
- Service analysis and agreement on data structures
- New technology needs new skills
- Need for program management
- Organized governance

- The Second sub-question: How suitable is ESB for addressing system integration challenges at a large organization?

In the previous literature ESBs usually described with core functionalities such as transparent service layer, messaging, loose coupling, security, and more. To be able to define a suitable solution for challenges, I summarized what ESB can provide in regards to challenges in previous question (see 3.4, 3.4.1, 3.4.2)
- Promotes SOA implementation by providing mediation capability
- Provides loosely coupled application integration
- Facilitates service, message, and event-driven interaction
- Message transformation, routing
- Protocol conversion
- It supports flexible IT architecture
- Deals with integration logic
- An ESB should provide core functions in section

3.10 Arguments for Empirical study

The theoretical study introduces the key points in the previous works on a specific topic with relation and connection to research questions. The combination of the theoretical findings and empirical results usually leads to more reliable conclusions, therefore empirical studies provide underpinnings to verify concepts, methods, and frameworks in the theoretical study of a thesis.

To be able to answer the research question and the sub-questions in this work, I can find the relationship between the main challenges of system integration and investigate a case study to verify real challenges related to the area of system integration challenges. The collected data can provide a solid foundation to confirm whether the concepts and previous research on system integration have a relation with a real phenomenon which is system integration challenges in real world projects. The investigation of ESB's impact on a large enterprise and its usability to provide a complete solution for challenges investigated in the empirical study provides evidence which can confirm the concepts of the theoretical findings or will totally provide different results.
4 Empirical study

4.1 Purpose

"An empirical study is really just a test that compares what we believe to what we observe. Nevertheless, such tests, when wisely constructed and executed and when used to support the scientific method, play a fundamental role in modern science." (Perry, Porter, & Votta, 2000, p. 347). In order to support progress and improvement in computer science and software engineering both (research and practice), we must design better empirical studies and be able to portray trustworthy conclusions based on the results of our findings (ibid). Empirical studies are used to explore, describe, estimate, and define natural, social, or behavioral occurrences using confirmations received from gathered data based on exploring, observing or experience (Sjoberg, Dyba, & Jorgensen, 2007). According to (Sjoberg et al., 2007), case studies are particularly important to use for “industrial evaluation” of information systems and software engineering methods, tools, and frameworks before being experimented in larger scale projects. In this work I have chosen to use a case study to investigate the challenges of system integration and if ESB can provide a suitable solution to cope with the barriers and challenges of system integration.

4.2 Sampling

4.2.1 The case

The selected case in this research study is a large international company with operational and technical departments around the world. The study was conducted based on the information and data gathered from IT departments of the selected company which by itself is a large international IT company providing IT services and solutions both in Sweden and other countries such as India, France, Brazil, China, and Belgium. The mission and vision of all the IT companies around the world is one thing; to provide support, maintenance, and development of the whole company’s IT projects. Today, the IT Company has 6000 employees in 35 locations around the world providing IT services, consultancy, and telematics services both to the parent company and other related companies in the same area.

4.2.2 Methods for sampling

In order to start with the data collection process, the target population required for gathering sample empirical data should be selected based on a suitable selection and sampling method which can yield proper supporting information during the empirical study of the thesis work. In both qualitative and quantitative research a sampling scheme should be utilized to select a number of participants for gathering empirical data (Onwuegbuzie & Collins, 2007).

According to (M. N. Marshall, 1996), a significant step in any research is to choose a sample for the study because it has never been possible from different factors such as practicality, efficiency or ethicality to study a total population.

In this thesis I used a sampling scheme of critical case and convenience sampling for the selection of both interview and questionnaire respondents. In the selection process of the target population, I focused on the ease of accessibility and communication and the advanced level of knowledge the target population possesses related to the area under study.

Both of the methods used are suitable in mixed method research, according to (Onwuegbuzie & Collins, 2007), critical case sampling is to select settings, groups, or individuals based on
specific characteristics that as a result of being involved in the process of data collection, produces trustworthy and dependable insight to the favored phenomena under investigation. Moreover, convenience sampling is based on choosing a target group or individuals that are appropriately available and willing to participate in the study (ibid).

The first step in the process was to contact the selected respondents and explain about the purpose of the interview. I selected respondents based on the information I got from different project maintenance and support teams. The most important characteristic that made the respondents more suitable as my target group was their related years of work experience in the fields related to system integration also their current responsibility in the projects. Moreover the accessibility and communication factors also affected the selection of my respondents in this study. After getting positive response from the respondents I started to book meetings in the same building of the respondent's workplace to make the process easier for them. It's also important to mention that I selected different people with different positions related to design, development and implementation processes of system integration to avoid biased perspectives on one aspect of system integration challenges.

4.3 Interviews
In this research, interviews were conducted to add more knowledge to the process of system integration. The semi-structured interview was performed for data collection because semi-structure allows the interviewee to be unencumbered by the context of questions and to be able to extend their thoughts and enable gathering the in-depth knowledge and experience from the research area (Oates, 2005).

To prepare for the interviews the author first reviewed the literature and a number of documents on the internal website of the company. The attempt was to keep the questions in the domain of the case under study and produce a complete picture of the processes, tools and methods available in the current situation.

The purpose of performing interviews was to investigate requirements and current existing problems from different points of view in integration projects. The integration developer, Integration Architect, Integration tester, and an integration manager that has worked in this area more than 10 years.

To make the process of the interview more clear to interviewee and to help create an overall picture of the research the interviewer introduced the topic of the research project and some background information about the interviewer. Moreover the aim of the thesis and the expected outcome of the interviews were explained. The interviewee talked briefly about the current projects and his position in the company. The semi-structured interview helped the author to perform open discussions with the interviewee and further explore in depth the requirements, advantages and disadvantages of the current situation and aim for the next step in research. The method of recording the interviews was a software program called "Audacity" which is free to download and possible to export audio files from each conducted interview separately.

4.3.1 Question Selection and Validation
In the process of selection and validation of questions for semi-structured interview, first a set of questions were selected and written down by the author. This was done with regard to the
literature and information review done prior to this section. The author made sure that the questions will cover the domain in which the research is done. While writing the transcripts the process of summarization was done at the same time to be able to later skim and take the gist of what is related to the research question easily for the data analysis process. The list of questions is presented in appendix chapter.

4.4 Interview data Presentation

In this section I have summarized the information and discussions from the interviews performed about the system integration process, methods, problems, and requirements from interview participants who have been working more than 10 years in the company. Results from this part of the empirical study are used to design an integration artifact with selected ESB which will be presented further on in this chapter.

4.4.1 Integration Developer

The first respondent has been working as a developer for five years with specialty in UNIX and Linux platforms. He is programming in C and C++ and has been working with developing integration solutions for different applications within the IT department. Solutions were usually in Windows and UNIX Platforms on the C, C++, and Java architecture. Integration included IBM Web Sphere MQ, File transfer and an old in-house developed Message Oriented middleware. Solutions which he worked with include integration of several systems. Core processing solution based in Linux on the C/C++ architecture and interface channel based in XMS (MQ) and XML.

First we began with a discussion what Integration means from his perspective, I was curious to find out what different people define as integration, he pointed out that integration between applications is sending information (data) from one application to another. What I understood from his explanations was that in today's IT projects application integration plays a fundamental role and he explained in more detail that in order to avoid a whole mess in integration infrastructure of large companies, instead of having each project initiate on their own which is adding to project risk and may build a wrong thing, more focus should be on standardization of methods, components, patterns, and frameworks.

Then we continued to discuss about the common ways of designing integration projects in the specific team that he is working, here I wanted to grasp the main approaches of integration currently utilized by different teams.

He answered, "Present integrations in the organization are based on File transfer, WMQ messaging with adapters, Web services etc. However, it would really depend on the actual needs and requirement on which kind of integration would be most suited. For instance, if for every request there should be a reply within a particular time frame and both applications have to be in complete sync then MQs provides a good solution. However, loosely coupled integrations like a fire and forget request may be based on a File protocol. Therefore in some legacy systems total files are sent based on the fire/forget or FTP protocol to different applications".

Integrated applications play a major role in selecting which method to be developed, in a large enterprise; applications usually are classified as below:
- **Mainframe applications**, producing huge amount of critical data, bulk data and transaction processing

- **Package applications (COTS)**, proprietary applications handling a broad business functionality

- **Custom built applications**, in-house developed applications in different languages

According to what I understood from my first interviewee, in large companies it is the usual case that there are different applications, some of them are masters of data like legacy systems and COTS which are less flexible to be integrated.

The process of designing the integration between applications usually takes time according to him; the applications which are going to be integrated should be investigated further to draw the models which they follow for communication. He pointed out that: "The Schema is a contract that both application owners and the application developers should design how the data sent and received is going to be adapted to the target applications. For example if we have a thick java client that communicates with legacy servers, it's needed to first study the business logic for the applications communicating and then design how the communication is going to happen".

When we discussed the different ways of integration and their challenges and simplicity the outcome where mostly emphasizing on the application platforms. "… Integration of different modules of the same/similar applications, in this instance, it is generally considered that the integrating modules would be developed on the same platform, language, have similar input/output arguments/parameters. These integrations have fewer problems be integrated to each other as they are built on the same platform and therefore requests and reply most of the times are compatible."

The challenges are usually when the applications are from different platforms, languages, and use different communication protocols. I also understood from the discussion about the platforms, methods, tools to implement and design integration is mainly dependent on the IT governance of a large company and usually they will show up in the bugs and problems in frameworks to be used. Some of the frameworks have not been tested in a real situation and when the first time it is used, then the problems will be added to the complexity of designing integration code. In addition, it is easy to make configuration errors for connections due to the number of integrations which may not be point to point infrastructure-wise but for implementing the solutions we take each pair separately for more clarity in way of working.

According to the previous literature, I asked about the complexity of integration projects and how the gathered requirements will be estimated to different tasks, he said: " The process of development starts with the schema design, then the requirements from the business process perspective confirms which integration approach to be used… if it is a new integration project then the first priority is to use messaging and WMQ, moreover the emphasize is on loosely coupled messaging such as fire and forget or publish/Subscribe than request reply, however the main identifier is the business process requirement for such scenarios".

When we discussed about the outcome of the integrations, it was explained that it is usually data and sometimes functionality that is shared between applications in integration projects.
but in the case under study data level integration is more in focus than integration of functionalities. Therefore the applications usually send and receive data through a central broker but different methods of sending and receiving using the broker can be used. Usually if the messaging is based on fire-and-forget it is easier to implement and maintain than request/reply.

4.4.2 The integration tester

The second interview was done with a system analyst in .Net applications who works in one of the teams which support and maintain many applications mainly in .Net. He was involved in many integration projects during the years.

The interview began with open discussions about the regular procedure of developing and designing of integration projects. He mentioned that "It is important to have standard methods to design integration among applications; a common way of designing patterns can reduce the burden on teams which own applications."

We continued to discuss how different applications need a different method of integration. He mentioned that "… One important thing that teams should think about is data transformation procedures which need careful design and coding. In .Net applications usually time consuming to first come to a conclusion for two applications to send and receive in which way as well as the design of adapters which play an important role in data transformation".

He also added that "The challenging matter is that for each enterprise application there is a specific adapter developed for data transformation between two applications and the broker just had the role sender and receiver of the messages on the channel. Therefore it is desirable to implement a standard to separate data transformation from the application, however in our company's case this is not happening now. The transformation layer is developed inside the application implementation; however it depends on how much data formats differ between applications."

We continued to discuss the concept of loose coupling and how much this is implemented in the company, he replied: "Based on the infrastructure we can say that we are performing loosely coupled approach because we have queues and channels, it is easy to get connected to certain channel for specific information, however, information wise we are tightly coupled to the applications instead of being generic type of information, because if we have an application that has SAP format of information, it is tightly connected to that application's information, if we change the application to something else other applications does not know about the format and it has to be changed "

Here a summarization of what it meant by an adapter in general is presented:

![Figure 4-1 - Adapter (Integration tester describe)](image-url)
The discussion about common problems in integration design, development, and maintenance led to the following problems:

- Designing the adapters is tricky and time consuming, if applications change, the adapter should be developed with regard to new requirements, so it is really application specific.

- Use of standard methods is vital to integration projects otherwise the process will become complex in large organizations.

- It is essential to understand the important factors underlying the integration so that it can be built on the factors that would be most influential.

- Designing data transformation for each side of applications is a cumbersome process which is not an efficient way of integration in volatile business environments especially if two applications have specific data format such as legacy systems and SAP.

- SOA as a concept is not implemented and utilized in projects if we define a service we define a data or information that is specific to an application rather than a more general definition.

- The entire integration registry is built upon applications rather than services.

4.4.3 The integration manager

The third interview was done with one of the infrastructure architecture manager which can help to give a better understanding of the different policies, limitations, and challenges in integration projects and some discussions was focused on the future perspective of integration projects.

The infrastructure manager is mainly responsible for decision making and determining which problems are threatening or and determining investigated in the integration department.

The interview started by discussions about the number of applications integrated which there will be possibility to grow. Some packaged applications are going to phase out after some years and the current solutions supporting their integration to other applications must change or be investigated carefully for the future.

When we discussed the current solutions and integration challenges for teams, he mentioned that the current solution for integration contains all the requirements for integration systems such as:

- Connectivity
- Data transformation
- Routing
- Security

However, for each of the features there is a separate solution which together forms the total integration infrastructure and messaging backbone for the large company.

When we discussed the ESB concept and a complete solution which can make it easier to integrate applications, he mentioned that "…using open source products as a complete
solution can be possible however, there are some problems for database and data integration when a data format should be translated to another data format. The priority is on loosely coupled solutions which are easier to implement and maintain, in addition,

When we talked about the requirements for integration from his perspective he mentioned the following points:

- It could replace the sending and receiving batch files in legacy systems with a better solution which automatically manages the protocol transformation.
- Application integration is not just the underlying technologies; it should be possible to understand the business requirements aligned with that as well.
- The main challenge is when the platforms integrated are different.
- Design of data transformation for each individual application should be revised with a new solution.
- How and when to move from a rigid, non-flexible tasks to a more robust and agile one (a batch oriented distribution is harder to maintain than a more loosely coupled scenario).
- How to optimize the application integration in such a way that both development and maintainability of integration solutions become more efficient than today.

4.4.4 The Integration Architect 1

The integration architect has been working in the integration area for eight years and mainly working with the modeling and design of integration solutions. He is part of the integration delivery center and usually in each project especially new projects; integration architects have an important role.

The first question that I asked about his main responsibility in one of the ongoing projects, I asked how they usually start to work with modeling and designing integrations, I wanted to know if there's a process that is followed, any procedures that should get done to reach other steps.

His response to the two first questions was:

"In each new project there should be an integration architect together with other developers to implement the solutions. He said the main thing at the start is to define the business processes and flows because it is possible that a wrong integration planning can cost more than be cautious at the first stages of the work and don't rush to the end… By flows I mean what information is needed from target and what is sent by the sender of information or data… in many cases there are more than 20 applications involved in the integration design process… therefore it should be a pre-study design to draw the whole picture of what mappings we have and what extra information is needed from other integration parties"

I then asked how you keep track of all these mappings, because a simple mistake somewhere can end up as a wrong definition, requirement gathering, estimation and so forth.

He continued that:" All the integrations with different applications are recorded in integration requests to one of the departments, there are number of specialists to review the integration requests… these documents don't provide detailed information of the integration design and more are the mapping that show the two applications connected, also the approach of their connections, and some more detailed information about the flows and channels."
I pointed out this fact and asked for his idea. According to the theories about integration, lack of understanding of business processes involved in an integration solution, can cause solutions that are working in a wrong way and need to be re-engineered, he noted that:

"Yes, I do agree with this especially when in one company there almost 2800 different applications, sometimes this process is easier because we re-engineer an old application to a new version, then we use the same integrations as before and at least it is clear which systems are involved in the business process, but for a new application this process takes longer time".

We continued to discuss in more detail about the integration requests and how the process continues because I wanted to find a pattern in the way they were performing modeling of integrations, He said: "After defining the mappings and getting confirmation from the integration office about the correctness of the mappings, all those information will be saved in a repository which is accessible for all the applications involved in integrations, then we start with more detail design of which channels we should use, which approach to choose, these are more technical because we need to know non-functional requirements such as volume and frequency of information, all these will have effect on which integration approach to choose"

After finding out the process of work, I wanted to see the challenging part of the work from his perspective, I asked about the process, load of work, and its relation to work agile, He pointed out notable information in this regard which is valuable to be mentioned: "To be able to model in right way we had many meetings with other applications because there were no models before and all those flows of information or in other words, integration flows where set up 10 or 20 years before without any central repository or knowledge of the detail information, there were some investigation on validity of the flows for the new application as well"

I found out that there are many discussions from both a business process and IT perspective to design the integration flows. In fact, some business mappings where done even before starting to model the flows of information. According to him, the business mappings accelerated the process; because it helped the two parties of integration define the requirements in a more effective way.

One of the challenges mentioned was that each project has different environments and this adds to the complexity of integration flows. If there are three different environments, the integration should be modeled for each of them using different configurations. From my discussions with him I got this feedback that the process of defining requirements before modeling takes some time for him, a lot of communications between information object consumers and information object providers is needed before starting the main process.

I wanted to dig deeper into the technical issues with regard to his responsibility and possible challenges that he faced, He noted that: "From a technical perspective, the non-functional requirements are important because we should know what is the format of the data sent and how frequent it is sent, then we decide to choose a pattern that is suitable for this exact flow, he added the problem is that we have to do this for as many peers in integration that we have, so a lot of work load exist in this process, we set up the queues, acting more like a police role to see what is allowed from the IT governance point of view, because we can't just choose something that is suitable for the application, that will end in a chaotic situation in a large company, some previous patterns like batch file transfer is not allowed anymore based on technical problems it caused also massive maintenance and support activities".
He also added that: "Standardization of which approach and technology to use is important in large companies, now everything is based on using IBM's web sphere MQ for messaging as a tool of integrating different systems on different platforms. Here data is inserted into a destination database also the sender application’s data is sent on a queue which is read by the destination application.

Moreover, the second choice is a simpler in-house message oriented middleware (MOM), he said if previously our old-developed projects were integrated using file transfer, they still keep that and use it even if it is not allowed and have many drawbacks, but for the new solutions this is changed to new technical approaches such as WMQ messaging."

Finally, the most important challenge from his point of view was: the wait time between processes which on one hand prevent un-organized way of working but the drawback is the lag time to wait for each process to be done completely and start the next process. In addition, if there's a mistake in the process, all the steps should be taken from start and there are no short-cuts to decrease the lag time.

4.4.5 The Integration Architect 2

The integration architect has been working in the integration area for 6 years and mostly working with modeling and design of integration solutions. He is part of the integration delivery center and usually in each project especially new projects, integration architects have important role. He also works in one the projects that recently started, which is not a migration from an old system to new a version with more functionality.

The first question that I asked was about his main responsibility in one of the ongoing projects, I asked how they usually start to work with modeling and designing integrations, I wanted to know if there's a process that is followed, any procedures that should get done to reach other steps.

His response to the two first questions was: "Based on the fact that this company provides a clear governance of what is the process of integration, an architect's responsibility is clear, to first plan for gathering IT and business information regarding the two systems which is going to have a flow of information between. Both systems have representatives from IT and Business… to have a clear definition of the concepts, what is going to be done… should be answered first, what information is going to be transformed from one system to another".

I asked about different patterns of exchanging data, how it is possible to estimate which pattern is suitable for a specific scenario, he said: "The most important thing in designing the patterns is that which application sends the data and how many targets are interested in that data, is the application is the master of that data and many applications are interested in receiving that data... This case is mostly implemented in Publish/Subscribe pattern".

According to the theories about integration, lack of understanding of business processes involved in an integration solution, can cause solutions that are working in a wrong way and need to be re-engineered, I pointed out this fact and asked for his idea, he noted that: "This is totally true, even when we start to select the patterns and go into a deeper analysis of how to implement the patterns, a clear definition of why this is required by business and IT will make it easier to choose between patterns that are more suitable".
In previous literature, in recent years the concepts of SOA has gained attention as a more efficient solution to integration, how much do you think currently the concepts are used in this company, what I understood was that services in this company's case is the data that is provided by applications, so it is mostly a customized definition that what is a service defined. When we talk about the service contracts, it means the way the data will be send and received, the format and the sample message that is sent and received are defined in the contract.

Our discussion about the non-technical part of integration projects directed to the fact that in large companies, there should be governance or a center of control that provides rules, guidelines, and frameworks for the integration projects, He noted that: "This is true and really important because the aim is maintaining what is produced and if we have a cluttered infrastructure without rules then it will impact in long run on maintaining the developed integrations". Moreover, he pointed out the fact that existing infrastructure needs technical competence, the need for different skills in each team for designing, developing and implementing integrations is critical, therefore the whole landscape needs training if a new method of integration is introduced.

According to what we discussed, the most emphasize is now on messaging to send and receive information in the company, the integration backbone is WMQ for message transfer and different adapters to transform the message content. Data is inserted to destination database also the sender application’s data is sent on a queue which is read by the destination application.

Moreover, there are other applications that do the routing. Therefore, the integration backbone is consisting of combination of products to provide the ESB functionality, I asked. He replied: "Yes, this is true that we don't have ESB as a product but we tried to combine different products to produce the same functionalities of ESB".

One of the most important features of integration backbone or the function in integration layer is transformation of messages from one data format to another data format, I asked about the possibility that applications have in the company to decouple the transformation logic from application logic, He pointed out: " It depends on how target applications define boundaries on their data format which is received by the message, he pointed out that if the application used different format then the translation is required... Canonical formats can be a replace to have a more standard message formats from holistic view of the company rather than the format that each application interprets the data, in the case of using a common format, then the transformation of messages should be done in integration layer, not application specific layer, new applications are more flexible to follow a general message format because they can build the canonical format directly, but the old legacy systems which have a specific message format need transformation".

4.5 Questionnaire
As discussed in chapter 2, the empirical study includes a questionnaire as a complementary method to interviews, since it covers a bigger population in a large organization and provides the researcher with more generalizable results. The group of respondents was selected among experts in the area of system integration with more than 5 year experience. The means of communication was email and the sample selection was done with the help of my supervisor from the company’s side, since he himself works as an integration architect for many years in the case company. In order to comply with content validity according to (Oates, 2005) one of
the factors that should be evaluated in the design of the questionnaire, I got help from my supervisor. He reviewed my questions and also I consulted another integration architect to make sure that the questionnaire was correctly designed to enable collection of a relevant sample of the area under research which is system integration challenges, considering the first research question. According to (Dawson, 2009), while using questionnaires as a method of data collection, the researcher should avoid vague questions, with the help from my supervisors I reviewed the questionnaire before starting to send it, therefore I tried to avoid any type of ambiguity in the questionnaire design.

I designed a short questionnaire with 10 questions in total, where the initial question was for gathering general information. I used email as my communication method. The main aim of using a questionnaire was to study a specific target group which I was introduced to and received their information during a seminar in the field related to system integration practices and method development held in the company of case. I sent out the questionnaire to 50 people all with positions related to integration architecture and development. The sampling method in this case also would be critical case and convenience sampling and snowball sampling with a specific target group working in the same area of research.

I utilized a questionnaire because it has many advantages that could help me collect data, according to (Tashakkori & Teddlie, 2003), it helps to measure attitudes and clarify what research participants think, inexpensive method since I used email, I administered the questionnaire in a group that was my sample. It is possible to adhere to ethical issues that should be considered in research (ibid).

It is important to mention that I used the specific surveying design website http://www.surveymonkey.com to design and analyze the online survey results. The survey link is http://www.surveymonkey.com/s/J3JYJ65 was sent to all the participants with a short explanation as reminder of the goal of the survey. The feedback I received was from 18 people who completed the survey. All the questionnaire respondents have years of experience in the field of system and application integration. This will increase the possibility to gather, analyze and present more reliable and acceptable data.

4.5.1 Questions
As I mentioned before, a questionnaire’s design is advantageous and it was close to the criteria which I needed to select as one of my data collection methods. I used a mixture of open and close ended questions. Moreover, different question types are included in the design which includes quantity or information, list or multiple choices, and scaling. The questionnaire includes 10 questions that are presented below, the complete list of questions are in the Appendix chapter.

| Q1 | Please fill in the information below : Name, Email, Position, Years of related experience |
| Q2 | In your opinion, what is the most important challenge in managing integration projects in a large enterprise? |
| Q3 | Traditional Integration approaches (batch file transfer and MOM ) are efficient ways of integrating legacy systems with different platforms such as ..Net or Java? (Efficient: Time, Cost, flexible to change, Testing, Maintenance…) |
Q4 Which of the following approaches is most common to use in integrating applications?

Q5 Do you think the current integration infrastructure successfully separated application logic from integration and process routing logic?

Q6 To what extent do you agree that the current integration approaches support scalability (meaning that a new application can easily be integrated to the integration portfolio of your team)?

Q7 To what degree do existing integration governance policies helped the flow of integration process go smoothly for your project?

Q8 Do you think the new "Service Bus" model is more effective, flexible, testable,… in contrast to previous ways of integration?

Q9 What were the challenges you faced in design, implement, test, and deploying of application integration? (More than one answer can be selected)

Q10 Do you think the current integration approach provides enough tools to facilitate testing strategies in integration projects?

4.5.2 Results

In this section, the results of the online questionnaire are illustrated with the help of an automatic data analysis tools offered by “SurveyMonkey”. Here I present the results of the survey, and further on analyze the data in the next chapter.

In the first question I collected the personal information; the aim was to mostly get the responsibility and years of experience of the respondents. So I present the years of experience as one of the factors of validating the answers to the questionnaire that is presented below:

**Years of experiences**

- Less than 1 year
- 1 to 3 years
- 3 to 5 years
- 5 to 10 years
- More than 10 years

Figure 4-2 - Year of experience of questionnaire participants
In your opinion, what is the most important challenge in managing integration projects in a large enterprise?

<table>
<thead>
<tr>
<th>Answer Choices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaps in integration</td>
</tr>
<tr>
<td>Integration does not reflect business</td>
</tr>
<tr>
<td>Unclear and un-estimated work items</td>
</tr>
<tr>
<td>Business Process and data identification</td>
</tr>
<tr>
<td>Gathering project requirements via business and IT coordination</td>
</tr>
<tr>
<td>Challenges with behavioral integration</td>
</tr>
<tr>
<td>Challenges in following long administrative processes for requirement gathering</td>
</tr>
<tr>
<td>All</td>
</tr>
</tbody>
</table>

Figure 4-3 - Questionnaire result
• Traditional Integration approaches (batch file transfer and VCOM) are efficient way of integrating legacy systems with different platforms such as .Net or Java? (Efficient: Time, Cost, flexible to change, Testing, Maintenance…)

Figure 4-4 - Questionnaire result

• Which of the following approaches is most common to use in integrating applications?

Figure 4-5 - Questionnaire result
Do you think the current integration infrastructure successfully separated application logic from integration and process routing logic?

To what extent do you agree that the current integration approaches support scalability (meaning that a new application can easily be integrated to the integration portfolio of your team)?
- To what degree existing integration governance policies helped the flow of integration process go smoothly for your project?

![Figure 4-8 - Questionnaire result](image)

- Do you think the new "Service Bus" model is more effective, flexible, and testable, in contrast to previous ways of integration?

![Figure 4-9 - Questionnaire result](image)
• What were the challenges you faced in design, implement, test, and deploying of application integration? (More than one answer can be selected)

![Figure 4-10 - Questionnaire result](image)

• Do you think the current integration approach provides enough tools to facilitate testing strategies in integration projects?

![Figure 4-11 - Questionnaire result](image)
4.6 Design and Evaluate Cycle1

According to (Hevner et al., 2004), the aim of design Science methodology is to improve the current situation by designing innovative artifacts. An application’s domain consists of people, organizational systems, and technical systems which interacting to reach a goal. The designed artifact should address an important organizational problem. Moreover, the second critical activity in the DSR cycles is evaluation of designed artifact. Artifacts are in many cases not a fully developed information system or solution, alternatively they define ideas, technical capabilities, and products which the implementation cycle and further on, effective and efficient usage will be achieved. The evaluation process plays critical role in DSR process and can be categorized in terms of completeness, consistency, accuracy, performance, reliability, usability, fit with the organization (ibid).

In this work, I apply the DSR build and evaluate cycle to identify what an open source ESB product can provide in terms of message processing under an increasing load of data. As it is explained above the aim is to see the performance and usability of Mule ESB under increasing load of data which is compatible with DSR paradigm. Based on the interviews with two integration architects and the integration developer, I was able to discover what is the core method used in integration backbone of the company and also the main issue which is message transformation development code and development time for each integration. I started to design the first cycle using Mule ESB. As the integration tool, in my first cycle is Mule ESB enterprise edition version 3.3.0, I tried to implement what it is called "Message Flow" between two endpoints one is the sender of the messages and another is the receiver of the message. The first flow shows a flow which transfers data from database table in the form of XML messages to JMS server. The next flow is the reverse flow, which shows the same endpoints but in reverse direction, as I got feedback from both of the architects it is important to identify and test ESB’s ability to do the reverse processing and insert functionality. In a daily scenario that integration architects are involved with, the data is inserted to the destination database also the sender application’s data is sent on a queue which is read by the destination application.

As the application integration style in the company under research is data integration, generating data tables in a database is part of the integration design so that all the data in the applications can be shared among other applications. The application receiving the data plays the service consumer role and the applications sending the data as a message plays the service provider role in one message flow.

According to the interviews (4.4) the flow between two applications in the integration landscape has a queuing mechanisms based on queue managers such as IBM WMQ. Mule ESB provides the endpoints and components needed to implement the Database to JMS (Java Message Service) message flow. JMS is the Java application API which supports messaging between applications. While the tests were performed in “Mule - ESB Software Product” environment, while two designated frequencies were chosen regarding the corporate standards in the company; Correspondingly the packet volume of functional data had been chosen within a specific applicable range and it is shown as “Rows per Cycle” by means of “Extent of inserted or processed messages”; which specifies the total number of comprised messages during a single cycle in a particular replication (frequency).

The selected flows were more interesting for both integration architects since they use the same concept and styles in a real work environment when the applications are based on Java
enterprise edition. A sample database table called 'Authors' was created in MySql. Since Mule ESB provides support for a wide range of databases there was no limitation of choice for database servers.

MySQL is selected because of its ease of use and the author’s familiarity and convenience with it. It should be mentioned that any database server management tool could be used and there’s no priority on the selection of database servers. The table includes columns first-name, last-name, address, telephone, and id as the primary key. The database here as inbound endpoint, sends data to Mule ESB for further processing and puts the message payload on the Active MQ's queue as an xml message. In order to implement this flow using Mule, some settings and changes to Mule's configuration file should be done to be able to run the flow in Mule's IDE (Appendix A).

The solution presented below is the reversed case of the previous scenario, the message flow from inbound to outbound end points inserts the message payload to the database table connected to the database outbound endpoint. The data contained in message payload which is a XML string then a transformer is needed to act as a bridge which changes the XML string to Java Objects.

As it has been explained before first the endpoints are configured and get referenced to the datasource that in this case will be Mysql adapter as well as ActiveMQ adapter to enable correct connection to these two sources. Here each message on the inbound queue contains information which is going to be inserted in the table for more processing purposes. The figure below shows the same procedure as for the JDBC to ActiveMQ flow to generate the results which are inserted information according to the configuration settings for both JMS endpoint and database endpoint.
Mule ESB client provided easy to configure functions with a minimum amount of code writing and less difficulty to test the message flow. During the first phase of building the integration message flow, I selected the polling frequency that polls the rows of database and set that on 10 ms (milliseconds), then changed the number of rows after each test was finished and tried to measure the message processing of ESB when I increase the load of data which here is number of rows of data-table which is processed as an XML message. For each message which was processed by the “Maps to XML” and the payload was on the JMS Server which here is Active MQ. I recorded the number of received messages and also checked the payload to see if the data is correct.

According to the diagram presented below, for increasing load of data I started the flow with 10 rows and increased the number of rows to 20, 50, 100, 200, and 500 to be able to identify the number of processed and inserted messages. The blue line in the diagram presents the Mule ESB’s processing progress under increasing load of data. The red line presents the second flow which shows the number of inserted messages data in the database table.

![Frequency (10 ms)](image_url)

It is important to mention that after the first flow for message transformation and delivery, I shared the solution and test that with both system architects in separate sections to get feedback from them; I selected the architects according to what they explained about their responsibility, they could give me a clearer answer than others. The feedbacks from the architects were the input for the next build and evaluate cycle where I tested the flows with another polling frequency.

In most of the cases it is advantageous to evaluate the ESBs robustness regarding the insertion of records read from a JMS server or a queue manager and evaluate how it functions with the increase of polling frequency, which is one of the factors to set in Mule ESB to handle the amount of time during which the data is read from the database and JMS server to be processed further in both of the message flows.
4.7 Design and Evaluate Cycle 2

In the next design and evaluate, the main focus was the increase of polling frequency to 100 ms (milliseconds) to test the same flows under the same amount of increasing messages which is polled based on a different frequency. This step of testing, number of rows in database set for the testing was also 20, 50, 100, 200, and 500. In the process of evaluation a number of issues were faced which could be solved based on the Mule's error and exception handling functionalities. Mule ESB offers features to handle different types of problems. In this case for each record in the database which is fetched once a flag is set, then the query for the database endpoint updates the column for the record which is processed once. When the flag is set then the next polling will skip the columns with the flags set to 'Processed' attribute.

During the running and testing of message flows another problem arose regarding that if during the polling of the database the messages do not have the right format to be processed and transformed in MuleESB or if one of the connectors fails during the running of flow Mule provides strategies to be prepared for these types of errors that might occur in real world integration scenarios. Mule provides roll back strategies such as retry policy which in this case was not the main concern of this work and this existing functionality is not tested due to the fact that the main focus was message processing and transformation. According to Mule's online documentations, new exception handling in Mule 3.3.0 enable message flows to avoid facing unhandled exceptions while the flows are running.

The diagram below illustrates the results of the tests with an increase of messages containing rows of database tables. The analysis of the diagrams data presentation is done in chapter 5 of this work, however, the blue line presents the number of processed messages and the red line presents the inserted messages in the database table.

![Figure 4-15 - Mule test result (2)](image-url)
4.8 Empirical research results
The empirical study provided me with the information from five interviews with experts working in different positions in relation to different phases in system integration projects; the aim was to identify the challenges of system integration in different phases. As a result of the interviews I was able to answer the first sub-question and was able to build the two DSR build and evaluate phases also evaluate the ESB software with two very primitive but important integration solutions that are evaluated based on selected criteria. Here the selected criteria are the software's processing performance under increasing message loads.

• What challenges do system integration present?
According to the interview results the following challenges could be identified, since I could find a pattern for challenges from theoretical studies and finalized my theoretical with a categorization for different challenges of system integration projects, I searched to identify the same themes and patterns in my five interviews:

- **Data or message Transformation**- Designing data transformation for each side of applications is a cumbersome process which is not an efficient way of integration in volatile business environments.

- **Business and IT cooperation** - The process of cooperation of business and IT is challenging and also critical for the design of correct integration solution.

- **Existing processes involved** - The processes involved until the planning phase of the integration finalizes is blocking the agility of IT projects.

- **IT Governance**- IT governance rules, processes, frameworks play an important role to let the different phases of system integration projects flow in the right direction.

- **SOA in small scales** - There is no enough evidence that SOA is applied to the integrations in the case under study, the definition of the service is connected tightly to the applications.

- **Project environment Diversity**- Diversity of environments for different application can make the process even more complicated.

- **Inadequate testing**- In the testing process is not efficient because the tools are not showing the detailed information of what was the source of errors.

- **Technical Skills**- Introduction of new technologies needs skilled professionals to maintain and support the solutions.
- **Protocol Transformation** - Protocol transformation are more challenging when integration is between different platforms and Protocol.

- **Standardization** - Use of standard methods throughout the company for integration.

Moreover the summary of the questionnaire is also provided with concentration on the answers with high number of respondents as the components that could be derived from the questionnaire results. In the next chapter a more detailed analysis will be done based on both results from interview and questionnaire to answer the first sub-question of the research.

<table>
<thead>
<tr>
<th>Q1</th>
<th>Years of related experience</th>
<th>-</th>
</tr>
</thead>
</table>
| Q2   | Gathering project requirement via business and IT coordination  
Business Process and data identification | 64.7%  
58.8% |
| Q3   | Disagree  
Neutral | 44.4%  
22.2% |
| Q4   | Hub- and Spoke  
SOA | 52.9%  
35.2% |
| Q5   | Agree  
Disagree | 55.5%  
16% |
| Q6   | Agree  
Disagree | 55.5%  
11% |
| Q7   | High  
Moderate | 47.6%  
11.7% |
| Q8   | Agree  
Disagree | 55.5%  
0% |
| Q9   | Testing and maintenance  
Setting up contracts between two or more applications  
Setting the queues and testing connections | 72.2%  
66.6%  
61.1% |
| Q10  | Neutral  
Disagree | 38.8%  
27.7% |

Table 6 - Summary of Questionnaire Result
• **How suitable is ESB for addressing system integration challenges in a large organization?**

ESBs as products facilitate most of the technical requirements of the integration project. According the inputs from the two architects and the developer I tried to evaluate the ESB software under load of data to see how it handles increase of data load in two different frequencies. In each of the build and evaluate cycles I experienced some of the advantages that an ESB product can offer:

- It provides out of the box message transformation
- Message routing and protocol conversion
- Simplicity of configuration
- Error handling and logging
- Easy installation
- Support for open standards
5 Analysis and Results
In this chapter the comparison between the theoretical and empirical study will be presented to make a solid basis for the conclusions and suggestions for future work using the knowledge generated from this research. The analysis of the results achieved from the empirical section will be summarized to facilitate mapping of the findings in theoretical part with the empirical studies to mainly answer the sub-questions and consequently the main research question "What is the impact of ESB on integration projects in an enterprise?"

5.1 Analysis
In chapter 2.4 the author explained the process of data analysis to provide the solid foundation for further comparison of empirical data with the theoretical findings. In theoretical studies the selected research areas were enterprise application integration and Enterprise Service Bus. The main focus was to find the most important key research areas related to the research areas mentioned above, such as EAI, architectural approaches related to EAI implementation and so forth to gain an overall understanding of the subject and produce enough information to be able to answer research the sub-questions. Consequently the empirical studies helped the author to identify a relationship between the key components found in theoretical studies. Below the author presents a "Comparative analysis" to analyze the outcome of theoretical and empirical studies to answer the research question and sub-questions.

1. What challenges do system integration present?
To be able to answer this question I revisit the components mentioned in the final section of the third chapter (3.9). In previous researches which were mentioned in chapter 3, the challenges could be categorized as below:

- **Challenges presented by integration project management**
  In section 3.3.1 a conceptual model was presented to provide illustrations of different stages and processes involved in integration projects. The investigation in several research articles identified the following challenges explicitly related to integration project management:
  - Gathering integration requirement via Business and IT coordination.
  - Business process and data identification.
  - Behavioral integration is more challenging than technical integration.

The procedures involved in each software project are in tight relation to business requirement gathering processes. The coordination and cooperation of business and IT is an important factor because a mistake in this phase can produce an integration solution that does not work properly. The incorrect business process and data -that play an important role in defining integration requirements- can produce risks such as inappropriate or inadequate integration architecture (3.3.1).

As discussed in section (3.3.3) each integration project could have two sides, technical integration and behavioral integration. As it is mentioned previously, the behavioral integration can be more challenging as this aspect of integration is related to change management and transformational issues that are interrelated. According to empirical studies, I could find almost the same components related to integration project management. I identified that business and IT cooperation is challenging and critical in defining, design and architectural patterns of integration solutions. The IT and
business functions cooperate to identify the requirements in the initial up of integration projects.

Empirical studies also can support the fact that identification of the needs in different business scenarios and what is needed as the data, is critical to integration projects. Both of the integration architects emphasized on their sessions with representatives from both IT and business to start gathering requirements, this enabled them to identify patterns and architecture supporting the requirements. However, I have found an aspect from empirical studies that is important to mention:

- Challenges that are related to long procedures, administration and documentations in integration projects can prevent un-organized way of working and on the other hand, can stop the process from progressing in a really efficient way (4.4.4), (4.4.5).

- As the questionnaire results show 64.7% of the respondents had selected gathering of project requirements via business and IT coordination as the most important challenge in integration project management. However, 58.8% of the responses agreed on Business process and data identification as the challenge.

- **Challenges of old technologies**

  In the theoretical study the evolution of system integration architecture (3.3.2) gives supporting evidence to explain the problem with previous architectures such as point to point integration and hub and spoke integration. According to chapter three, the Enterprise Application Integration concepts and patterns have evolved through the years and many enterprises moved from point to point integration to messaging pattern to avoid heterogenous applications connected together in a point to point manner.

  The literatures reviewed usually focus on the increase of maintainability and support cost as well as slow transition to changes in point to point integrations. Moreover the scalability and complexity of implementation were defined as challenges in literature studies regarding the old integration technologies. The following components were found in literature studies:
  
  - Costly solutions
  - Agility and flexibility
  - Scalability
  - Complexity

  In the empirical study, it could be clearly identified that traditional methods of integration such as point to point integration has been used before and some applications still used file transfer protocol and batch files as means of integration. This also could be recognized from the internal organizational documents, however, in recent years, the new projects are not allowed to use FTP or batch file transfer anymore because point to point integration increases the complexity, cost in maintenance and development of applications (4.4).
The results of questionnaire (4.5.2) also show relation to this category which provides answer to the first sub-question. Here I present the results regarding the challenges that utilizing of old technologies present in the company of the case. According to the questionnaire results, 52.9% of respondents think that Hub-and-Spoke is used more than the two other architectural approaches in application integration with the SOA in the second place and Point-to Point in third place.

The third question in the questionnaire directly evaluates the traditional approaches of integration practiced at company of the case provides us with this fact that old technologies such as batch file transfer and primitive Message Oriented Middleware is not efficient way of integrating old systems in terms of development time, cost, flexibility to change, testing and maintenance. According to the results, 44.4% disagreed that old technologies provide efficient solutions versus 5.5% who strongly agreed.

Moreover, the sixth question targets the aspect of scalability in current integration solutions that are practiced throughout the company to verify correlations between theory and empirical results. According to the results 27.78% of the respondents strongly agreed that the current integration approach is scalable while 55.56% of the respondents agreed that the current approaches used in integration projects support scalable solutions.

- **Challenges of new technologies**

  According to the theoretical study, there are some challenges involved in using new technologies that provide newer features to solve previous problems. On the other hand, they can result in new challenges for system integration teams. The following challenges were identified in the theoretical study regarding the introduction of a new architectural approach in integration, which is SOA with reference to (3.3.4.2).

  - Migration of inflexible systems architecture such as legacy systems to support new technologies
  - Confusion due to a new array of standards and tools
  - Resistance to change
  - Current infrastructure support the change to new technology
  - Budget investment and slow return of investment
  - Service analysis and agreement on data structures
  - New technology needs new skills
  - Need for program management
  - Organized governance

According to the empirical study technical challenges which can be consequence of adapting to new technologies were as follows:

- **Adaptation of inflexible old technologies**

  The empirical study reveals that one of the challenges according to interviews was the data format transformation of legacy systems. To support new integration approaches such as SOA, there should be special projects
supporting this change to transform data format of legacy systems to a generic format that all the applications can use.

- **Project environment diversity**
  The diversity of environments and their versions based on different requirements of users make the change process to new technologies a challenge since the change should be applied to different versions of application.

- **IT governance**
  According to both the theoretical and empirical studies I could find this component of integration projects important in many aspects specially in providing a standardized way of working throughout a large organization. The relationship between effective IT governance and success in integration projects has been highlighted in empirical studies.

  Moreover, section (3.3.4.2) it is clearly identifiable that IT governance reveals to be a success factor in integration projects which defines roles, procedures, and responsibilities in integration projects. It also characterizes frameworks, standards, and platforms that should be used across the organization which prevents chaotic way of working in a large enterprise.

  In both the interviews (4.4) and questionnaire (4.5), it is possible to verify the important role of IT governance in the integration projects and a challenge if it hinders the project implementation flow and progress. As the results of the questionnaire clearly provides 47% of the respondents verified the high degree of IT governance role and its policy selection to help the progress of integration projects.

- **SOA in smaller scales**
  Since the underlying concept of SOA takes time to be implemented and the technology is still new to the company under study, one of the challenges introduced by the experts during empirical studies was the possibility that the current infrastructure supports the adaptation of new technologies.

- **Technical skills**
  According to interviews (4.4), the introduction of technologies needs investment in new competence and skilled professionals.

- **Standardization**
  Standardization of methods and frameworks used by different projects in the company helps the progress of teams in design and development of integration solutions. This important component if not taken into consideration, can lead to chaotic development cycles.
• **Other Challenges**
  The following challenges were also identified, however, according to the interviews there are some challenges that are mainly faced in integration projects that was interesting enough to be presented below:
  
  - **Data or message transformation**
    According to interviews, designing data transformation for each application involved in integration solutions, needs more generic way or method than the current solutions that the data transformation is done by each side of integration separately.
  
  - **Inadequate testing**
    According to the empirical studies one of the challenges is to test integrations. The interview with the integration tester clarifies that the current tools do not provide detailed information of the tests.
  
  - **Protocol Transformation**
    The interviews show that protocol transformation between different applications is also challenging when there are different platforms involved in the integration project.

The results of the questionnaire also provided some aspects related to challenges that the currently practiced or even new integration approaches could be presented at large enterprises.

  - **IT Governance** – 47.06% of the respondents approved the role of proper IT governance in support of integration projects
  
  - **Service Bus model** – As a new way of integrating applications, 55.5% of the respondents agreed that this model provides more effective, flexible, and testable solutions than previous approaches

Moreover, the following challenges had greater number of respondents that can support the popularity of them in integration projects. In addition, it was not possible to categorize them as specific type of challenge in integration projects.

  - **Testing and maintaining the integrations** - 72.2%
  
  - **Setting up the contracts** - 66.6%
  
  - **Setting up the queues and testing connections** – 61.1%
  
  - **Design of transformation logic in applications** – 50%
  
  - **Sending in the integration request and designing the integration approach** 44.4%
  
  - **Tools for testing**
In addition, based on the questionnaire, I tried to identify the important role of facilitated testing in integration projects, therefore I asked if the tools provided for testing is enough to perform test strategies in integration projects, 38.8 % were neutral, while 27.7% disagreed that the tools are enough for carry out test strategies.

2. How suitable is ESB for addressing system integration challenges in a large organization?

The theoretical study showed the following factors as what ESBs can do to address the issues and challenges of system integration with regard to different categories provided in both theoretical studies (3.4), (3.4.1).

- Promotes SOA implementation by providing mediation capability
- Provides loosely coupled application integration
- Facilitates service, message, and event-driven interaction
- Message transformation, routing
- Protocol conversion
- Supports flexible IT architecture
- Deals with integration logic
- An ESB should provide core functions in section (see 3.4.1)
  Service Location Transparency, Transport protocol conversion, message transformation, message routing, message enhancement, Security, message processing, transaction management

As for the empirical studies the author's investigation in the company of the case showed that the ESB software product as Mule ESB has never been used by any of the interviewees. The two architects had some experience with the same software from another smaller company (4.4.4, 4.4.5). However, the internal documentations of the company made a good basis to prove the features offered by the ESB software are beneficial for integration projects of the company. According to internal documentation the aim is to utilize ESB model instead of the current overall integration approach.

The first two data collection methods of empirical studies were carried out to identify the challenges of system integration in integration projects which were in the process of development at time of study. However, it is possible to pinpoint the relation between the challenges and the offerings of ESB product introduced by" Mule Soft" as Mule ESB.
According to section 4.6, the author used the input from the interviews to perform tests that focused on message transformation and the ESBs capability to support increasing load of messages.

The result of the tests with focus on an increased frequency of messages using Mule ESB is presented below:

As it is shown in the diagrams above, the results indicates that in the lower frequency (10 ms), by increasing the numbers of rows in a cycle, the number of processed messages are increasing proportionally together with the inserted messages; also the rate of this increment is growing rather equivalently for both inserted and processed messages and as it is clearly observable from the diagrams, both red and blue lines have quite a similar slope with a parallel inclination which demonstrates a symmetric growth rate for both cases. It must be noted that the diagrams are schemed in an exponential scale.
Alternatively, in the higher frequency (100 ms) we can witness an asymmetric inclination in the aforementioned lines, which indicates that the increasing rate is not identical for the lines; and since the system is functioning in a higher frequency, while the number of rows per cycle (and accordingly the total number of messages) is increasing, the processes are performing more quickly and the comprised data can be processed at a faster level. In the second figure, it can be seen that the slope of the blue line is bigger than that of the red line and consequently the growing rate of processed messages is rising faster than the increment rate of inserted messages.

This illustrates the fact that when dealing with a huge amount of data, applying a higher frequency of replication would help the system to have a faster response and execution rate. Moreover, the blue line illustrates the second message flow in each of the build-evaluate cycles in chapter four. As the number of the messages increases, and since a higher frequency to check the incoming data is selected, the numbers of messages processed and inserted are close to each other, however, with the increase of the number of messages the process of insert to data base table and running insert query takes more processing time than transformation of messages from JDBC endpoint to be put on JMS endpoint.

It is possible to substantiate the performance of Mule ESB from the test result diagrams in Figure1 and Figure2, since the slopes are exponentially growing with incremental load of data. Mule ESB provides high performance with regard to message transformation, processing and transformation in a message flow.

The summary section of chapter four provides the authors experience in performing tests mentioned above. Below the advantages of using Mule in build and evaluate cycles are presented:

- It provides out of the box message transformation
- Message routing and protocol conversion
- Simplicity of configuration
- Error handling and logging
- Easy installation
- Support for open standards
- Easy testing within Java eclipse client
- Provided logging of tests
- Good community support
- Robust solution
- Good performance regarding message transformation
- Good performance regarding message delivery

The comparison between the empirical and theoretical data shows that an ESB can be a robust solution containing all the requirements mentioned in the company's documentations as well as the interviews (see 4.4.4, 4.4.5). Review of internal documentations of the company demonstrates that the change in the application integration is inevitable for large enterprises when the number of applications grow. According to the documents the ESB model is more scalable, efficient, and faster solution than the current infrastructure. Then the ESB model implemented based on the SOA can help to have a more flexible, easy to manage, loosely coupled pattern.
Moreover, ESB as a model for integration is presented in the company's internal documents all the questionnaire respondents were familiar with the concept. The 8th question of the questionnaire verified that 55.5% of respondents agreed with the fact that the ESB model will provide a better solution with regard to efficiency, testability, and flexibility.

5.2 Results Summary
To answer the main research question in this thesis "What is the impact of ESB on integration projects in an enterprise?" the empirical study was performed utilizing different data collection methods through the case study research. This helped to identify some factors regarding the field of research identified in the theoretical study.

1. What challenges do system integration present?
Integration project development is part of any IT solution within the company that was selected as a case to design research and investigate specifically "the impact of Enterprise Service Bus on integration projects in an enterprise". Since different data and functionality throughout a large organization is usually offered by different applications, if a new project is in the development phase, it also will have an integration project supporting the integration part of the IT solution. In this work the identified challenges were presented in three different categories based on the related factors and components in the theoretical study of the research phenomena. To answer the first sub-question, here is a summary of all the findings in three different categories as provided earlier:

- **Challenges presented by integration project management**
  The combination result of interviews and survey results showed that Business and IT cooperation provides requirements for integration projects. One of the challenges faced by practitioners in this area is to receive the right information from both parties to identify business processes, data and other non-functional requirements such as the amount, frequency, and type of data. This process can involve many applications throughout the company. There's a high risk of faulty design and development if the data and business processes involved in integration projects are not verified in early cycles of integration project development.

  Extensive and time consuming administrative process related to registration of integration requests by integration architects in integration projects is another challenge which can be both time consuming and ineffective way of work in the time of agile development and importance of efficiency. Under this category the results generated from both the questionnaire (4.5) and the interviews are matching and supporting the above result. Therefore the main challenges under this category are gathering integration project requirement via business and IT coordination, data and process identification as the second important challenge in managing integration projects.

- **Challenges presented by old technologies**
  According to the interviews the old approaches and technologies which have been used are not allowed anymore in new projects. This is a clear evidence of the challenges of old approaches such as point to point integration. Moreover, to consider the results of the questionnaire, it is possible to specify that experts working in this
area by 52.94% of the respondents agree on the fact that Hub-and-Spoke is nowadays practiced as the main architectural approach, however according to interviews with the two architects I could find out the increasing trend towards SOA and ESB concept.

Both of the data collection methods clearly support the factors as challenges of old technologies, therefore challenges of using old technologies according to empirical studies are:

- Long development time
- Cost
- Low flexibility to changes
- Testing and maintenance

**Challenges presented by new technologies**

New technologies can both be beneficial to use and have some side effects that are usually related to change. According to interviews (4.4), challenges presented by new technologies are related to inflexible infrastructure and diversity of environments related to each IT solution within the company. Moreover standardization of frameworks and methods needs an authority to provide unified way of work throughout the company. IT governance plays authoritative role to provide and supervise standard way of work when new technologies are introduced to a company.

The empirical studies showed that SOA is started to be considered as a new way of integration work, therefore the investigation via interview and questionnaire showed some parameters as challenges presented by new technologies such as technical skills, SOA in smaller scale, and standardization.

Empirical studies provided the following results as general challenges of system integration projects:

- Data or message transformation
- Inadequate testing
- Protocol transformation
- Testing and maintenance
- Setting up contracts
- Setting up queues and testing connection
- Design of transformation logic in applications
- Sending in the integration request and design of integration approach

2. **How suitable is ESB for addressing system integration challenges at a large organization?**

To verify the suitability of Enterprise Service Bus as an integration tool, there are many factors involved. Suitability identification of a tool or product needs investigation and testing aspects of it which shows more value for the research case. In this thesis I could map the use of a new integration product with the requirement related to challenges generally proposed by both the interviews and questionnaire.

What the selected ESB can provide may not solve challenges presented by integration project management; however the ESB product focuses on the technical
perspective of the integration projects. According to both interviews and questionnaire outcomes I could find the problem of message transformation as one of the challenges in integration projects. The build and evaluate cycle with feedback from interviewees produced the results of performance evaluation of Mule ESB which are presented in Figure 5.1 and Figure 5.2. Below I summarize my findings to answer the second research sub-question.

<table>
<thead>
<tr>
<th>Integration Challenge</th>
<th>What ESBs Offers in Theory</th>
<th>Mule ESB's Empirical Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT and Business requirement</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Data and Process Identification</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Long development time</td>
<td>Promotes SOA implementation</td>
<td>Out of the box message transformation</td>
</tr>
<tr>
<td>Cost</td>
<td>Faster development less cost</td>
<td>Open source and Open standards provide less cost</td>
</tr>
<tr>
<td>Low flexibility to changes</td>
<td>It supports flexible IT architecture</td>
<td>Provides scalable integration solution which is flexible</td>
</tr>
<tr>
<td>Testing and maintenance</td>
<td>Deals with integration logic</td>
<td>Easy testing within Java eclipse client</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Error handling and logging facilitates maintenance</td>
</tr>
<tr>
<td>SOA in smaller scales</td>
<td>Promotes SOA implementation</td>
<td>-</td>
</tr>
<tr>
<td>Data or message transformation</td>
<td>Message transformation, routing</td>
<td>Good performance regarding message transformation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Good performance regarding message delivery</td>
</tr>
<tr>
<td>Inadequate testing</td>
<td>One of the core functions of ESB products support monitoring and management</td>
<td>Easy testing within Java eclipse client</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Error handling and logging facilitates maintenance</td>
</tr>
<tr>
<td>Protocol transformation</td>
<td>Core function of an ESB</td>
<td>Message routing and protocol conversion</td>
</tr>
<tr>
<td>Design of transformation logic in applications</td>
<td>Core functionality of ESBs</td>
<td>It provides out of the box message transformation</td>
</tr>
</tbody>
</table>

Table 7 - Summary of the results
The table presented above summarizes the findings regarding the suitability of ESB product Mule ESB. To answer the question of how suitable ESB is to address challenges of integration, I compared the challenges with the findings of both theory and empirical study. As it is clearly identifiable, Mule ESB is not able to solve integration project management issues. Its offerings are aligned with technical issues of integration projects. Moreover, it is possible to verify that some of the issues presented by interviews and questionnaire are specific to the company of the case. Therefore Mule ESB is suitable to solve general issues and challenges rather than specific ones such as:

- Setting up contracts
- Setting up queues and testing connection
- Design of transformation logic in applications

Considering the fact that many of the technical challenges are addressed by Mule ESB, its value and impact in large organization with hundreds of applications and platforms is clear. Since efficiency, flexibility, maintainability and testability are the values which are always the goal to be achieved by any organization.
6 Discussion

6.1 Conclusion

The phenomenon of system integration, like all technological advancements of information system technology has been evolving during the years. Since the 1990s specific products have come to market to facilitate the integration of different applications. For large enterprises comprised of different departments that provide various services, different application are developed or purchased from third-party companies. The heterogeneous business environment of today has become hard to manage and maintain and the possibility of reinventing the wheel exists somewhere else in the enterprises portfolio. Digitalization, E-commerce, E-government, service oriented concepts, and globalization are some of the drivers for integrating enterprise systems since the main purpose is to have a quick response to market changes and provide customers with the right data throughout the enterprise at the right time.

The problem in this area is to choose the best method that suits the organization. Because there is "no one fits all" solution for integrating applications, however, it is possible to investigate the different capabilities of different approaches of system integration and find one more suitable to match our target requirements. In the past, the main approach was manual integration with use of the batch file transfer from legacy systems to other systems. Gradually this has changed by the introduction of Message Oriented Middleware, EAI products, SOA, and Web services. Consequently the concept of EAI or "Enterprise Application Integration" took over the integration market with a wide range of products offered by different vendors. The offering of EAI was a better solution than point to point integration. While it tried to decouple applications and change the previous approach to a more flexible and manageable integration approach.

The main breakthrough in the area of system integration was introduced by Service Oriented Architecture that introduced the concept of service in software architecture. Since it has been in existence for more than one decade, it has provided a new way of designing application architecture has thus, impacted the application integration. As it provides offerings of heterogeneously spread applications as "Service", it has created an easy and manageable way of integrating business processes and data as reusable services. The concept of Enterprise Service Bus was introduced to handle complexity of implementing SOA concept in enterprises.

For many large organizations, it is important to consider the drawbacks of the integration styles they have implemented over the years as well as utilizing the new ways that will help to employ more efficient and effective ways to integrate systems. From the past until now a combination of different methods have been utilized to make a robust infrastructure for integrating applications in many large enterprises thus the need for a more advanced tool to implement SOA methodology seems inevitable. With the introduction of ESBs many software vendors started to implement solutions that run both on regular desktops and servers. Furthermore, there are many alternatives from well-known vendors like IBM and Oracle and other open source offerings including Sonic, Mule, and others to choose from.

In this thesis the aim was to explore and investigate the value and impact of Enterprise Service Bus in the area of system integration; the study was based on an investigation on integration platforms used by many well-known companies around the world and have been evaluated by previous literatures in terms of different functional and non-functional criteria.
However, this work first investigated the general problems and challenges faced by system integration architects, developers, and experts who have contributed to this area of research. Consequently the purpose was to specify the most important challenges in system integration projects and investigate if the proposed ESB product can provide a solution to challenges mostly emphasized by experts working in this area for many years. The ESB product was tested based on the feedback from the interviews. The empirical results prove that Mule ESB performs well under high message load processing. The author also tried to emphasize on the features and facilities offered by Mule ESB.

On the other hand, Mule ESB similar to other products does not cover the challenges introduced from the project management perspective in integration projects. Mule ESB and similar products do not provide integration architects with integration and functionality requirement gathering features. Although Service Oriented Architecture, as an integration approach, provides a robust concept to follow in contrast to older methods yet remains prone to problems such as service transparency, service duplication, proper IT governance, rapidly changing business requirements, and so forth. Since the main focus in this work was to find a solution for the integration challenges, and the case company's infrastructure is slowly changing to support the service bus concept, the theoretical investigation of Mule ESBs features to support SOA can we considerably useful for any organization of similar size and challenges.

The analysis and results provided in the previous chapter show the alignment between the theoretical study and the results gained from the empirical research. Moreover, the empirical study mainly supports the validity of Mule ESB as an integration infrastructure in the research case from a technical perspective. Mule ESB provided all the requirements and features for the development of integration tests and showed high performance during the build and evaluates processes.

### 6.2 Implications for the subject area

The research area investigated within this thesis is related to enterprise application integration, information system integration, and Enterprise Service Bus. The research question in this thesis is formulated in a way that investigates the impact of a new technology presented in the application integration area. Informatics is a spacious discipline encompassing multiple academic disciplines such as computer science, artificial intelligence, and cognitive science (Wikipedia, 2013) . The objective of this scientific discipline is to study the information generation and representation, information communication and processing using information systems (ibid).

Information system integration and its subsidiary phenomena such as EAI, SOA, or ESB focus mainly on supporting the integration of applications involved in the business processes of organizations that nowadays support internal and external value chains consisting of customers and supplier’s information. One business process can involve many applications and information systems; therefore the research area is relevant to informatics since the objectives of both phenomena are information. Moreover, informatics studies the human and behavioral sciences, organizational science and management with regard to information systems.

The result of this thesis may not involve users of information systems directly; however the system integration approaches under investigation aim to provide better IT solutions which
consequently involve users to have better experience of the IT systems. The integration of functionalities and data facilitates users executing tasks using applications integrated together.

Integration architects and integration developers to map their challenges to the features presented by ESB products can use the results of this thesis. The results of the build and evaluate cycles present value and impact the ESB products especially Mule ESB offer to integration projects. It provides flexibility to developer and integrator to implement customized functionalities that support open standards and many other features that were not included in scope of this thesis work. Mule ESB provides values in terms of simplicity of use, flexibility, and ease of customization from developer's perspective. Moreover, the outcome of the tests can be used as a guideline for integration developers to test and utilize Mule ESB's functionalities.

6.3 Method evaluation

Theoretical study
This section provides a general evaluation on this research. The attempt was to bring up the critical points in current integration practices and patterns of the studied case and also generate an understanding of ESB as a concept and also as a service which plays a supporting role to Service Oriented Architecture. The sections below present some evaluation points from the author's point of view. A series of theoretical investigations were used to determine the most appropriate topics which are related (directly or indirectly) to the concept of “Enterprise Service Bus” and its application in system integration and Enterprise Application Integration. These identified subject areas were studied thoroughly for further discussion about the research questions; correspondingly the importance of framing the constituents and structuring the conceptual elements obliged the writer to scheme a further pattern which would be handful for explaining future analysis.

As it was mentioned in prior chapters, the literature review of this study was drawn out of reviewed articles which were found in scientific journals, academic research publications, course textbooks, international conferences and relevant Internet sources. This cross section was the foundation of making a theoretical structure that not only supports my research but also updates my results and facilitate further investigations in this very field.

Alternatively this study indicates that the result of this work by itself cannot be considered as an infallible fact that would embed faultlessly in every system of any large organization, but obviously there is a likelihood of compatibility with numerous large enterprises. Besides, the work helped me to enter a new level of understanding of the concepts which were scrutinized in the study.

Empirical study
The empirical study is based on a case study as the research strategy, because this method is very suitable for investigating specific research phenomena within an organization or a company. The selected organization is a large international organization which owns IT solutions and consulting services provider departments all around the world. I employed both interviews and a questionnaire to collect empirical data to answer the first research sub-question. The interviews are categorized as semi-structured since they provide open
discussions between myself and the research respondents interviewed. Five interviews were organized with domain experts who work with a focus on integration projects. Moreover, to get more data for the challenges I employed a questionnaire with 10 questions. Based on the feedback from the two interviews with integration architects and the summarization of challenges of integration I employed some Design Science Research tenets to utilize the build and evaluate phases for testing the selected ESB product under specified circumstances that were defined as challenges by system integrators. During the empirical study I have not faced many complications; however it is usually hard to find the right candidates when investigating an organizational phenomenon. The build and evaluate process was to run the software, test each case with different inputs and select the frequency of polling information. Here I both received feedback from integration architects and integration developers to design the right flows to be tested.

6.4 Result evaluation
Since the thesis is based on a mixed method paradigm, selecting a validation method is not definitively determined like in the case of a positivism or interpretivism paradigm. Based on the fact that principles of DSR include build and evaluation cycles, the evaluation of the artifact is done for making sure the criteria such as rigor and relevance is met during the investigative process (Hevner et al., 2004).

The DSR clearly does not offer a specific criteria to validate the results, since the main research question here is the impact and value of the selected ESB product and this was mainly answered by the DSR build and evaluate cycles, it is important to evaluate the artifact based on the message flows implemented by Mule ESB. I selected the following criteria discussed in chapter two that are presented below:

**Plausible:** This criterion is concerned with the sensibility of designed artifact and if it is reasonable or it is just a belief (Petter et al., 2010). Since the pattern or in this thesis the message flows were connected to the investigated domain and the software product used for modeling is universally used, the designed artifact is plausible. It was not modeled based on the author's belief, it is grounded on a related knowledge domain to solve a specific real problem, and therefore plausibility is met.

**Effective:** According to (Petter et al., 2010), a pattern, here the modeled message flows by Mule ESB that is the main evidence for showing that evaluation is effective. If the pattern can describe the concept it has been designed to describe. In this work the effectiveness of the model was reviewed by help from the integration architects who are domain experts and had experience in working with similar ESB products.

**Feasible:** This characteristic is concerned with the fact that if the pattern or model is workable. Moreover, a feasible patterns can be implemented and used by other practitioners (Petter et al., 2010). Since the flows generated by Mule ESB are working with real environments and also the results were tested based on the polling frequency and increases on message loads, the model implemented is feasible. It can be implemented again and be tested based on different factors that can be produced by experts of the related domain.

**Predictive:** According to (Petter et al., 2010), a pattern is predictive when there is a possibility for the researcher to predict the result of the pattern or model. To measure this characteristic
the pattern should be used in a real environment by practitioners to determine if it can yield the results that are expected. In this work the author together with two integration architects tested the message flow based on a number of different criteria. However, it was desirable to use the model in a real environment. The security and policies made it hard to examine the modeled flows with real data in a real environment.

**Reliable:** reliability in paradigms older than pragmatism means that if the questions of questionnaire are sent to the same respondents it should get the same results (Petter et al., 2010). For the patterns produced by the DSR paradigm the reliability is measured if the used pattern produces consistent results regardless of the user of the pattern. In this work, since the Mule ESB is used based on the user manuals and instructions, anyone who follows the same instructions is able to produce the same results, therefore the model is reliable.

6.4.1 Possibilities to generalize
According to (Oates, 2005), using a case study as the research strategy is criticized for generating knowledge that is only related to the case under study. However, it is possible to generate broader results and conclusions that can relate to cases that are beyond the specific case itself.

In order to generalize the study, it is important to mention that the system integration, Enterprise Service Bus, and its related concepts are practiced and highly valuable for large organizations. The empirical data, can be generalized for the companies with similar size, as the case under study, however, with small to medium sized companies the integration pattern can take advantage of the test scenarios which were selected to show the most important functionalities in an ESB.

It's worth mentioning that the research questions where designed in a way that make it possible to generalize the case to other companies struggling to find a more efficient way of integrating systems.

6.4.2 Propositions of future research
There are a number of key points that have been mentioned in this research that can be investigated further in future researches. It is desirable for large enterprises that the authorized users have access to selected services from external environments, in that instance it will be mobile applications that request for services remotely.

The security and authorization that is provided by an open source application vendor is always taken for granted and nobody relies on the security offered by these vendors. It is desirable that there could be more in depth investigation of the security and scalability issues when we consider mobile devices connecting to the infrastructure for valuable data.

Another aspect that can be considered is the investigation of cloud computing and Mule's cloud hub. How this can help to integrate in a more efficient way to be able to connect mobile devices to the other applications in a large organization.
7 Bibliography


undertaking a research project: How to books.


Laftsidis, A. Enterprise Application Integration. *IBM Zweden*.


Appendix A: Mule Message Flow Set up

Configuration settings for the data sources and endpoints

Global element setting

In this scenario also a custom class transformer has been developed to do transformations from xml to java objects, then the java objects are inserted into the database table by the insert queries defined on the JDBC outbound endpoint. It is worth mentioning that the parameters in the statement are bound with corresponding entries in the payload. Below the XML configurations for the whole scenario is presented.
Database to ActiveMQ scenario results
After running the first flow in MuleESB the database information is sent to the queue "In" and as long as the connector is working the messages are saved on the queue and are ready for the target application to get connected and read from the queue.

Fig. 4-1 Message queue (In) on messaging server

The messages inserted on the queue by MuleESB

<table>
<thead>
<tr>
<th>Message ID</th>
<th>Correlation ID</th>
<th>Persistence</th>
<th>Priority</th>
<th>Redelivered</th>
<th>Reply To</th>
<th>Timestamp</th>
<th>Type</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID:baby:PC-65027-12906416103235-0:1:1308:12:1</td>
<td></td>
<td>Non Persistent</td>
<td>4</td>
<td>False</td>
<td>2012-10-19 12:12:00:719 CEST</td>
<td>Delete</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID:baby:PC-65027-12906416103235-0:1:1308:12:1</td>
<td></td>
<td>Non Persistent</td>
<td>4</td>
<td>False</td>
<td>2012-10-19 12:12:00:720 CEST</td>
<td>Delete</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID:baby:PC-65027-12906416103335-0:1:1308:12:1</td>
<td></td>
<td>Non Persistent</td>
<td>4</td>
<td>False</td>
<td>2012-10-19 12:12:00:722 CEST</td>
<td>Delete</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID:baby:PC-65027-12906416103335-0:1:1308:12:1</td>
<td></td>
<td>Non Persistent</td>
<td>4</td>
<td>False</td>
<td>2012-10-19 12:12:00:723 CEST</td>
<td>Delete</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID:baby:PC-65027-12906416103335-0:1:1308:12:1</td>
<td></td>
<td>Non Persistent</td>
<td>4</td>
<td>False</td>
<td>2012-10-19 12:12:00:732 CEST</td>
<td>Delete</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 4-2 The detailed view of messages in queue
Example of one of the message payloads which contain one row of the table information

```xml
<?xml version="1.0" encoding="UTF-8"?>
<table>
<record>
    <field name="phone" type="java.lang.String">5665425</field>
    <field name="lname" type="java.lang.String">Red</field>
    <field name="fname" type="java.lang.String">Sue</field>
    <field name="idauthors" type="java.lang.Integer">6</field>
    <field name="Address" type="java.lang.String">BrusselAB</field>
    <field name="Processed" type="java.lang.String">Y</field>
</record>
</table>
```

**tbl. 4-1 The message payload equivalent to table data**

**ActiveMQ to Database Messaging**

The results from the second scenario show success in building the results and insert them in the table in MySQL database. This way the application which uses this database can fetch the records sent by another application on the queue.

The query which inserts the messages to MySql table is presented here to show how message payload information is inserted in the equivalent columns in the table named bookinfo.

```xml
<jdbc-ce:outbound-endpoint exchange-pattern="one-way" queryTimeout="-1" connector-ref="Database" doc:name="Database" queryKey="Insert">
    <jdbc-ce:query key="Insert" value="INSERT INTO bookinfo (idauthors, fname, lname, Address, phone, Processed)
VALUES (#[map-payload:idauthors],#[map-payload:fname],#[map-payload:lname],#[map-payload:Address],#[map-payload:phone],#[map-payload:Processed])"/>
</jdbc-ce:outbound-endpoint>
```

**tbl. 4-2 How to insert message data to bookinfo**
Mule test results

The test results from are presented below which shows the performance of Mule ESB after running each test in 60 seconds. The values are rounded and are mean values based on the number of messages piled on the queue.

a) Mule ESB

<table>
<thead>
<tr>
<th>Frequency (ms)</th>
<th>Rows per cycle</th>
<th>Messages processed</th>
<th>Messages inserted</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10</td>
<td>5560</td>
<td>1660</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>5720</td>
<td>1810</td>
</tr>
<tr>
<td>10</td>
<td>50</td>
<td>6190</td>
<td>2270</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>6980</td>
<td>3020</td>
</tr>
<tr>
<td>10</td>
<td>200</td>
<td>8560</td>
<td>4530</td>
</tr>
<tr>
<td>10</td>
<td>500</td>
<td>13310</td>
<td>9060</td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>1510</td>
<td>1360</td>
</tr>
<tr>
<td>100</td>
<td>20</td>
<td>1880</td>
<td>1540</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>2970</td>
<td>2070</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>4810</td>
<td>2960</td>
</tr>
<tr>
<td>100</td>
<td>200</td>
<td>8450</td>
<td>4730</td>
</tr>
<tr>
<td>100</td>
<td>500</td>
<td>19400</td>
<td>10050</td>
</tr>
</tbody>
</table>

b) Explanation

Polling Frequency
The amount of time in which the ESB platform polls/fetches data from the sending database server.

Rows per cycle
The number of rows of data in database table polled after each frequency cycle is passed.

Messages processed
The number of messages that have been transformed by ESB and delivered to the target correctly.

Messages inserted
The number of messages that have been inserted to the target database server.
Appendix B: Interview Questions
A sample of possible questions

<table>
<thead>
<tr>
<th>Index</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Please explain about your responsibility in the integration projects?</td>
</tr>
<tr>
<td>2</td>
<td>What are the common approaches of integration in your team?</td>
</tr>
<tr>
<td>3</td>
<td>Why do you think the methods that are used are suitable for the current integrations?</td>
</tr>
<tr>
<td>4</td>
<td>How many applications are integrated in your development team?</td>
</tr>
<tr>
<td>5</td>
<td>Explain the process of your work generally</td>
</tr>
<tr>
<td>6</td>
<td>Which methods cause problem in your work?</td>
</tr>
<tr>
<td>7</td>
<td>In integration projects which part of the process is more challenging?</td>
</tr>
</tbody>
</table>

Sample of Questionnaire

Q1. Please fill in the information below:
- Name
- Email
- Position
- Years of related experience

Q2. In your opinion, what is the most important challenge in managing integration projects?
   - Gaps in integration
   - Integration does not reflect business
   - Unclear and un-estimated work items
   - Business Process and data identification
   - Gathering project requirements via business and IT coordination
   - Challenges with behavioral integration
   - Challenges in following long administrative processes for requirement gathering
   - All

Q3. Traditional Integration approaches (batch file transfer and MOM) are efficient way of integrating legacy systems with different platforms such as .Net or Java? (Efficient: Time, Cost, flexible to change, Testing, Maintenance…)
   - Strongly agree
   - Agree
   - Neutral
   - Disagree
   - Strongly Disagree
Q4. Which of the following approaches is most common to use in integrating applications?

- Point to Point
- Hub-and-spoke
- Service Oriented Architecture
- Other (please specify)

Q5. Do you think the current integration infrastructure successfully separated application logic from integration and process routing logic?

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly Disagree
- Other (please specify)

Q6. To what extent do you agree that the current integration approaches support scalability (meaning that a new application can easily be integrated to the integration portfolio of your team)?

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

Q7. To what degree existing integration governance policies helped the flow of integration process go smoothly for your project?

- Very High
- High
- Moderate
- Low
- Very Low
- Other (please specify)

Q8. Do you think the new "Service Bus" model is more effective, flexible, testable, etc. in contrast to previous ways of integration?

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly Disagree
Q9. What were the challenges you faced in design, implement, test, and deploying of application integration? (More than one answer can be selected)

- Design of transformation logic inside applications
- Setting up the queues and testing connections
- Setting up the contracts between two parties (Two or more applications)
- Testing and maintaining the integrations if something goes wrong
- Sending in the integration request and designing the integration approach
- No challenge faced
- Other (please specify)

Q10. Do you think the current integration approach provides enough tools to facilitate testing strategies in integration projects?

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly Disagree
University of Borás is a modern university in the city center. We give courses in business administration and informatics, library and information science, fashion and textiles, behavioral sciences and teacher education, engineering and health sciences.

In the School of Business and IT (HIT), we have focused on the students’ future needs. Therefore we have created programs in which employability is a key word. Subject integration and contextualization are other important concepts. The department has a closeness, both between students and teachers as well as between industry and education.

Our courses in business administration give students the opportunity to learn more about different businesses and governments and how governance and organization of these activities take place. They may also learn about society development and organizations’ adaptation to the outside world. They have the opportunity to improve their ability to analyze, develop and control activities, whether they want to engage in auditing, management or marketing.

Among our IT courses, there’s always something for those who want to design the future of IT-based communications, analyze the needs and demands on organizations’ information to design their content structures, integrating IT and business development, developing their ability to analyze and design business processes or focus on programming and development of good use of IT in enterprises and organizations.

The research in the school is well recognized and oriented towards professionalism as well as design and development. The overall research profile is Business-IT-Services which combine knowledge and skills in informatics as well as in business administration. The research is profession-oriented, which is reflected in the research, in many cases conducted on action research-based grounds, with businesses and government organizations at local, national and international arenas. The research design and professional orientation is manifested also in InnovationLab, which is the department’s and university’s unit for research-supporting system development.