Projeto: Pattern Based Enterprise Architecture

Mafalda Maria Ferreira de Vilar Gaspar

Thesis to obtain the Master of Science Degree in

Computer Science and Engineering

Supervisor: Prof. Pedro Manuel Moreira Vaz Antunes de Sousa

Examination Committee

Chairperson: Prof. José Carlos Martins Delgado
Supervisor: Prof. Pedro Manuel Moreira Vaz Antunes de Sousa
Member of the Committee: Prof. André Ferreira Ferrão Couto e Vasconcelos

January 2021
Acknowledgments

I want to thank all the family, friends, and colleagues who were part of this journey and for believing in me, even when I didn’t believe it myself. Everyone made it possible and easier to end this chapter of my life. I would like to express my affection and appreciation to all of you.

I want to express my gratitude to my supervisor, Professor Pedro Sousa, who knew how to push me to go further towards the goal, discussing and clarifying ideas about this project.

I also thank my dear friend Rafael Ribeiro that Instituto Superior Técnico gave me. Who over the years, has been a huge support for me in my personal and academic life.
Abstract

Nowadays, there is no fixed practice or method for evaluation of Enterprise Architecture (EA) implementation. However, enterprises are looking for evaluable EA implementation to cope with their current problems and answering future needs.

In the context of EA and to accomplish these needs: this thesis will propose three different architectural views. Such as Application Integration Technology - which covers well-known patterns such as Message Brokers; Enterprise Service Buses; Legacy Integration; Portals; Mobile Integration; Business-to-Business (B2B) Integration. Application Structure - that represents the way applications can share data and processes. And Enterprise Architecture - That represents an aggregated view of all the integration patterns mentioned above. Depending on the enterprise's maturity level, these different patterns are presented, using ArchiMate and Google Forms, according to the Open Group SOA Integration Maturity Model (OSIMM) Matrix.

Keywords

Enterprise Architecture, Enterprise Integration, Patterns.
Resumo

Hoje em dia, não existe um método fixo para avaliação da implementação da EA. No entanto, as empresas estão à procura de uma avaliação para a sua implementação de EA para lidar com seus problemas atuais e atender às necessidades futuras.

No contexto da EA, e para atender a essas necessidades: esta tese irá propor três diferentes visões de Arquitetura: Como a Tecnologia de Integração de Aplicações - que abrange padrões bem conhecidos, como Message Brokers; Enterprise Service Bus; Legacy; Portais; Integração Móvel; Integração B2B. Estrutura das Aplicações - que representa a maneira como os aplicativos podem partilhar dados e processos. E Arquitetura Empresarial - que representa uma visão agregada de todos os padrões de integração mencionados acima. Dependendo do nível de maturidade da empresa, esses diferentes padrões são apresentados, utilizando ArchiMate e Google Forms, de acordo com a Matriz OSIMM.

Palavras Chave

Arquitetura Empresarial, Integração de Aplicações, Padrões.
Contents

1 Introduction 2
  1.1 Introduction: Overview ................................................. 3
  1.2 Motivation ................................................................. 3
  1.3 Problem ................................................................. 3
  1.4 Objectives ................................................................. 4
  1.5 Document Structure ..................................................... 4

2 Related Work 5
  2.1 Related Work: Overview ................................................. 6
  2.2 Enterprise Application Integration ..................................... 6
  2.3 The Open Group Service Integration Maturity Model (OSIMM) ..... 9
      2.3.1 Architecture Dimension: Base Model ......................... 11
  2.4 Frameworks .............................................................. 14
  2.5 Communication .......................................................... 15
      2.5.1 Types of message exchange ..................................... 15
      2.5.2 Types of Communication ....................................... 16
  2.6 Enterprise Integration Patterns ....................................... 16
      2.6.1 Application Integration Implementation Patterns and Services .... 17
      2.6.2 Application Structure Implementation Patterns and Services .... 23
      2.6.3 Enterprise Integration Implementation Patterns and Services .... 26

3 Proposed Solution 28
  3.1 Proposed Solution: Overview ......................................... 29
  3.2 Evaluation of The Maturity Level ..................................... 29
  3.3 Patterns conversion to ArchiMate ................................. 35
      3.3.1 Application Integration .......................................... 35
      3.3.2 Application Structure ......................................... 40
      3.3.3 Enterprise Integration Architecture ........................... 41
  3.4 Forms ................................................................. 42
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4.1 Pattern Attribution</td>
<td>43</td>
</tr>
<tr>
<td>3.4.2 Pattern filling</td>
<td>44</td>
</tr>
<tr>
<td>3.5 Blueprints</td>
<td>50</td>
</tr>
<tr>
<td>3.5.1 Application Integration</td>
<td>50</td>
</tr>
<tr>
<td>3.5.2 Application Structure</td>
<td>51</td>
</tr>
<tr>
<td>3.5.3 Enterprise Integration Architecture</td>
<td>53</td>
</tr>
<tr>
<td>4 Evaluation</td>
<td>55</td>
</tr>
<tr>
<td>4.1 Evaluation</td>
<td>56</td>
</tr>
<tr>
<td>5 Conclusions</td>
<td>57</td>
</tr>
<tr>
<td>5.1 Conclusion</td>
<td>58</td>
</tr>
<tr>
<td>5.2 Future Work</td>
<td>58</td>
</tr>
<tr>
<td>6 Attachments</td>
<td>61</td>
</tr>
</tbody>
</table>
# List of Figures

2.1 Point-To-Point Architecture [1] ............................. 7  
2.2 Hub-and-Spoke Architecture [1] ............................. 8  
2.3 Pipeline Architecture [1] ................................. 8  
2.4 Service-Oriented Architecture [1] ......................... 8  
2.5 OSIMM Maturity Matrix - (Group, Service Integration Maturity Model (OSIMM)) ................................. 10  
2.7 The ArchiMate Generic Metamodel [3] ................................. 15  
2.8 Simple Message Broker Reference Architecture [4] ................................. 18  
2.15 Shared Database Diagram [5] ................................. 23  
2.18 Application Structure Reference Architecture ................................. 26  
3.1 Proposed solution - Diagram ................................. 29  
3.2 Application Architecture ................................. 51  
3.3 Application Structure - ArchiMate ................................. 52  
3.4 Service Structure ................................. 52  
3.5 Enterprise Integration Architecture - ArchiMate ................................. 53  
3.6 Enterprise Integration Architecture - Full View ................................. 54
6.1 Application Integration - Google Forms .................................................. 62
6.2 Message Broker - Google Forms ............................................................... 63
6.3 Message Broker (Messaging) - Google Forms ......................................... 63
6.4 Message Broker (Application Interface) - Google Forms ...................... 64
6.5 Message Broker (Security) - Google Forms .......................................... 65
6.6 Enterprise Service BUS - Google Forms .................................................. 65
6.7 Enterprise Service BUS (Translation/Transformation) - Google Forms .... 66
6.8 Enterprise Service BUS (Application Interface) - Google Forms ............ 66
6.9 Enterprise Service BUS Security - Google Forms ................................... 67
6.10 Legacy (Data Interface) - Google Forms .................................................. 67
6.11 Legacy (Message Interface) - Google Forms .......................................... 68
6.12 Legacy (Screen/Report Interface) - Google Forms ............................... 69
6.13 Legacy (Service Interface) - Google Forms ............................................. 70
6.14 B2B Connectivity- Google Forms ........................................................... 70
6.15 B2B Partner Connectivity- Google Forms .............................................. 71
6.16 B2B Partner Management- Google Forms ............................................. 71
6.18 B2B Broker/ESB- Google Forms .......................................................... 72
6.19 B2B Translation - Google Forms ............................................................ 73
6.20 B2B Routing- Google Forms ................................................................. 73
6.21 B2B Application Interface - Google Forms ............................................. 74
6.22 B2B Legacy Integration - Google Forms ................................................ 74
6.24 Portal- Google Forms ............................................................................ 75
6.25 Portal (Broker/ESB) - Google Forms ...................................................... 76
6.26 Portal (Application Server) - Google Forms .......................................... 76
6.27 Portal (EII) - Google Forms ................................................................. 77
6.28 Portal (Application Interface) - Google Forms ...................................... 77
6.29 Portal (Legacy Integration) - Google Forms ......................................... 78
6.30 Mobile Integration - Google Forms ....................................................... 78
6.31 Mobile (Security) - Google Forms ....................................................... 79
6.32 Mobile (Broker/ESB) - Google Forms .................................................. 79
6.33 Mobile (Translation) - Google Forms .................................................... 80
6.34 Mobile (Routing) - Google Forms ....................................................... 80
6.35 Mobile (Application Interface) - Google Forms .................................... 81
6.36 Mobile (Legacy Integration) - Google Forms .................................................. 81
6.37 Mobile (A2A Security) - Google Forms .......................................................... 82
6.38 Application Structure - Google Forms .............................................................. 82
6.39 Application Structure (Messaging) - Google Forms ........................................ 83
6.40 Application Structure (File Transfer) - Google Forms ..................................... 83
6.41 Application Structure (Shared Database) - Google Forms .............................. 84
6.42 Application Structure (RPI)- Google Forms ....................................................... 84
6.43 Integrated Enterprise (Data Cleansing)- Google Forms ................................. 85
6.44 Integrated Enterprise (Metadata Repository)- Google Forms ............................ 85
6.45 Integrated Enterprise (Content Repository)- Google Forms ............................. 86
6.46 Integrated Enterprise (Translation/Transformation)- Google Forms .................. 86
6.47 Integrated Enterprise (Routing)- Google Forms ............................................... 87
6.48 Integrated Enterprise (Service BUS)- Google Forms ...................................... 87
6.49 Integrated Enterprise (Application Server)- Google Forms ............................... 88
6.50 Integrated Enterprise (Process Management)- Google Forms .......................... 88
6.51 Integrated Enterprise (Orchestration)- Google Forms ....................................... 89
6.52 Integrated Enterprise (Portal)- Google Forms .................................................... 89
6.53 Integrated Enterprise (Web Server)- Google Forms .......................................... 90
6.54 Integrated Enterprise (Collaboration Platform)- Google Forms ....................... 90
6.55 Integrated Enterprise (B2B Server)- Google Forms .......................................... 91
6.56 Integrated Enterprise (Mobile Server)- Google Forms ..................................... 91
6.57 Integrated Enterprise (Portal)- Google Forms .................................................... 92
List of Tables

2.1 Maturity Indicators for the Architecture Dimension ........................................ 12
2.2 Simple Object Access Protocol (SOAP) vs. REST ........................................... 25
3.1 Table - Question: How would you characterize your architectural topologies? ........ 30
3.2 Table - Question: What type(s) of data repositories does your organization utilize? .... 30
3.3 Table - Question: What is the standard communication style in your architecture? ...... 31
3.4 Table - Question: How is integration achieved in your architecture? ...................... 31
3.5 Table - Question: What methods do you use to develop your architecture? .............. 31
3.6 Table 2 - Question: What methods do you use to develop your architecture? .......... 32
3.7 Table - Question: How mature are your services implementations? ..................... 32
3.8 Table - Question: How extensive is your SOA? .............................................. 33
3.9 Table - Question: What architectural principles define your approach? .................. 33
3.10 Table - Question: How extensive and sophisticated is your organization's use of frame-  
               works in your architecture? .......................................................... 34
3.11 Table - Question: How are architectural decisions made in your organization? .......... 34
3.12 Table - Question: Does your organization use reference architectures? ............... 35
3.13 Description of concepts - Message broker .................................................. 36
3.14 Description of concepts - Legacy .............................................................. 37
3.15 Description of concepts - B2B ..................................................................... 38
3.16 Description of concepts - Portal .................................................................... 39
3.17 Description of concepts - Mobile Integration .................................................... 40
3.18 Description of concepts - Application Structure .............................................. 40
3.19 Description of concepts - Enterprise Integration Architecture 1 ....................... 41
3.20 Description of concepts - Enterprise Integration Architecture 2 ....................... 42
3.21 Application Integration Form - Attribution of Pattern ....................................... 43
3.22 Application Structure Form - Attribution of Pattern ........................................ 44
3.23 Message Broker Implementation Table [4] .................................................... 45
<table>
<thead>
<tr>
<th>Table Reference</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.24</td>
<td>ESB Implementation Table [4]</td>
<td>45</td>
</tr>
<tr>
<td>3.25</td>
<td>Legacy Integration Implementation Table [4]</td>
<td>46</td>
</tr>
<tr>
<td>3.26</td>
<td>B2B Integration Implementation Table 1 [4]</td>
<td>46</td>
</tr>
<tr>
<td>3.27</td>
<td>B2B Integration Implementation Table 2 [4]</td>
<td>47</td>
</tr>
<tr>
<td>3.28</td>
<td>Portal Integration Implementation Table [4]</td>
<td>47</td>
</tr>
<tr>
<td>3.29</td>
<td>Mobile Integration Implementation Table [4]</td>
<td>48</td>
</tr>
<tr>
<td>3.30</td>
<td>Application Structure Implementation Table</td>
<td>48</td>
</tr>
<tr>
<td>3.31</td>
<td>Integrated Enterprise Implementation Table [4]</td>
<td>49</td>
</tr>
</tbody>
</table>
## Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EAI</td>
<td>Enterprise Architecture Integration</td>
</tr>
<tr>
<td>EA</td>
<td>Enterprise Architecture</td>
</tr>
<tr>
<td>B2B</td>
<td>Business-to-Business</td>
</tr>
<tr>
<td>ESB</td>
<td>Enterprise Service Bus</td>
</tr>
<tr>
<td>MOM</td>
<td>Message Oriented Middleware</td>
</tr>
<tr>
<td>OSIMM</td>
<td>Open Group SOA Integration Maturity Model</td>
</tr>
<tr>
<td>SOA</td>
<td>Service-Oriented Architecture</td>
</tr>
<tr>
<td>TOGAF</td>
<td>The Open Group Architecture Framework</td>
</tr>
<tr>
<td>ADM</td>
<td>Architecture Development Method</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>DB</td>
<td>Database</td>
</tr>
<tr>
<td>RPI</td>
<td>Remote Procedure Invocation</td>
</tr>
<tr>
<td>REST</td>
<td>Representational State Transfer</td>
</tr>
<tr>
<td>SOAP</td>
<td>Simple Object Access Protocol</td>
</tr>
<tr>
<td>RMI</td>
<td>Remote Method Invocation</td>
</tr>
<tr>
<td>CORBA</td>
<td>Common Object Request Broker Architecture</td>
</tr>
<tr>
<td>WSDL</td>
<td>Web Services Description Language</td>
</tr>
<tr>
<td>UDDI</td>
<td>Universal Description, Discovery, and Integration</td>
</tr>
<tr>
<td>ODBC</td>
<td>Open Database Connectivity</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>
1 Introduction

Contents

1.1 Introduction: Overview ................................................. 3
1.2 Motivation ............................................................ 3
1.3 Problem .............................................................. 3
1.4 Objectives ............................................................ 4
1.5 Document Structure .................................................. 4
1.1 Introduction: Overview

This chapter serves as an introduction to this dissertation, starting by presenting a motivation for the study of well-known patterns for the analysis and definition of Applications’ Integration Architecture in Section 1.2. Section 1.3 a brief description of the problem this thesis intended to solve is provided. An overview of the goals of this project is provided in Section 1.4. Lastly, the remainder’s structure of the document is presented in Section 1.5.

1.2 Motivation

Nowadays, due to the evolution of technology, business systems need to be redesigned to reply to market changes. Increasingly, business processes require more information to be shared among many applications. Within this area, organizational silos refer to business divisions that operate independently and avoid sharing information. Silos occur because of how an organization is structured. The same priorities, objectives, or even tools are not shared. Leading to departments operate as individual business units or entities within the company. These behaviors lead to disorganization, duplication of efforts, lack of synergy, and missed opportunities. One way to prevent silos from occurring is through mapping, modeling, and process automation.

According to the Federation of Enterprise Architecture Professional Organizations (FEAPO), Enterprise Architecture (EA) is “a well-defined practice for conducting enterprise analysis, design, planning, and implementation, using a comprehensive approach at all times, for the successful development and execution of strategy.” Regarding EA, Enterprise Architecture Integration (EAI) is a field within this area and an emerging approach to architecture design that ensures data or a function moves from one application to another.

A fully integrated view of architecture: provides a consolidated view of the company’s data; integrates software from diverse sources; offers access to software to those inside and those outside the company; automatically connects the tasks related to a business function; reduces the reaction time of their information system; rapidly develops new applications. [6]

1.3 Problem

This thesis intends to represent the integration architecture of an enterprise and its applications. For this, the biggest problem is, for an enterprise, how to create and represent the views. In a way that it is clear, for an enterprise, how they can define their integration architecture for the applications and the integration architecture for the enterprise.
1.4 Objectives

For this, we are going to use the application integration patterns from the book, The Essential Guide to Integrated Solutions [4], adjusted according to the organization maturity level.

Accordingly, we are going to present three different architecture views: Application Integration Technology that covers well-known patterns such as Message Brokers; Enterprise Service Buses; Legacy Integration; Portals; Mobile Integration; B2B Integration; Application Structure that represents the way applications can share data and processes; and Enterprise Architecture that represents an aggregated view of all the integration patterns mentioned above.

The Solution Proposal (Section 3) is divide into two parts:

• Theoretical - To present well-known patterns of Application Integration Architecture.

• Practical - To use ArchiMate and Google Forms to present these patterns discussed in the theoretical part, in the format of Blueprints (for visualization) and Forms (for gathering the information).

1.5 Document Structure

The remainder of the document is structured as follows. Section 2 presents the background necessary on the problem at hand and discusses the developed work in the area. Section 3 describes the architecture of proposed solution. Section 4 defines the evaluation plan for the developed solution. Finally, Section 5 presents the conclusions to this document.
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Related Work: Overview</td>
<td>6</td>
</tr>
<tr>
<td>2.2 Enterprise Application Integration</td>
<td>6</td>
</tr>
<tr>
<td>2.3 The Open Group Service Integration Maturity Model (OSIMM)</td>
<td>9</td>
</tr>
<tr>
<td>2.4 Frameworks</td>
<td>14</td>
</tr>
<tr>
<td>2.5 Communication</td>
<td>15</td>
</tr>
<tr>
<td>2.6 Enterprise Integration Patterns</td>
<td>16</td>
</tr>
</tbody>
</table>
2.1 Related Work: Overview

In this chapter, it is presented a literature review of the topics related and most relevant to this work.

The remaining chapter is divided as follows: Section 2.2 serves as a brief explanation of the content of the thesis: Enterprise Application Integration. Section 2.3 specifies how to measure the service integration levels of an organization. Section 2.4 does a brief overview of the most common frameworks within the subject of Enterprise Architecture Integration. Section 2.5 covers the topics of types of message exchange and types of communication. At last, Section 2.6 covers the three main architectures patterns in this thesis: Application Integration, Application Structure, and Enterprise Integration;

2.2 Enterprise Application Integration

Applications integration (or enterprise application integration or EAI) is the process of different applications in an enterprise to share data enabling them to work together.

This thesis will identify the different application integration scenarios, and provide reference architecture models for each. With the purpose that given an organization, with a vast number of project teams, several vendors, languages, and visions, - a unification of the vision of the organization's problems between all the distinct elements.

The complexity of the EA can be represented based on a collection of architectural layers: each layer supports the needs of the one above it, with the top one precisely supporting the capabilities required by the business strategy. The layers mentioned above are described in more detail as it follows:

The layers mentioned above are described in more detail as it follows [7]:

• Enterprise Strategy Layer: this layer mainly describes an enterprise strategy regarding product portfolio and appropriate customer segments, appropriate delivery and distribution channels in the given market environment, competitors and core competencies, and capabilities of the company;

• Business Layer: to develop an EA, it is necessary to provide the process and integrated tools to identify the as-is state of the organization (the business and IT) ecosystem) and the desired, to-be state. EA promotes the formation of enterprise blueprints that show how business processes are in the present state and how they can be implemented, exploiting the full range of capability of underlying IT architectural building blocks;

• Application Layer: This layer supports the business and describes the required business functions in the underlying IT application systems, which is very important since the rapid increase of applications, systems, and platforms, and their inter-dependencies turns the procedure of adding and improving IT capabilities a possible risk for the business without a EA strategy;
• Information Layer: information is the fuel that drives business artifacts; its flow achieves value to the user. One key objective of the EA is to convert raw data into relevant information that gives additional insight and value to the business;
• Infrastructure Layer: the infrastructure layer consists of the server, network, and storage infrastructure which supports higher-level functions, for example, a Database (DB), applications, or e-mail servers. There is a high demand for more flexibility and agility also from this layer, due to cost pressure, leading to higher degrees of virtualization and systems consuming less energy, reducing electricity costs.

The fundamental Integration Architecture Variants are:

**Point-to-point** Architecture is used normally in a 1:1 relationship when a sender has to send a message to a single receiver. The sequence in the point-to-point model is as follows: The sender places a message in the queue. The integration broker forwards after processing the message to the appropriate receiver. The receiver receives the message and processes it as appropriate. As the applications universe grows and the Point-to-Point architecture is no longer a viable option. While adequate for simple integration, this model is quickly unmanageable for larger integration requirements [1].

![Figure 2.1: Point-To-Point Architecture](image)

**Hub-and-Spoke** Architecture is a Message Oriented Middleware (MOM) that uses a central Message Broker. The communication is not made between pairs of applications, but between each application (spoke) and the central hub. The broker functionalities include routing and messages transformation to the receiver spoke. This architecture allows content based routing, which performs based on information in the message header or in some element defined in the message body. The hub can apply rules to the content of the message and determine the receiver spoke [1].
In a **Pipeline** Architecture, independent systems along the value-added chain are integrated using a message-bus. The bus capability results in the distribution of the interfaces to the central bus throughout the communication network, which gives applications a local access to a bus interface [1].

In a **Service-Oriented** architecture, the integration of different applications to form a functioning whole by means of distributed and independent services calls, which are orchestrated through an Enterprise Service Bus (ESB) and, if necessary a Process Engine [1].
2.3 The Open Group Service Integration Maturity Model (OSIMM)

The Open Group Service Integration Maturity Model (OSIMM) specifies how to measure the service integration levels of an organization and its IT systems and business applications. In addition, it provides guidance on how to achieve certain levels of service maturity necessary to realize related business benefits. The OSIMM has seven dimensions across seven maturity levels (Group, Service Integration Maturity Model (OSIMM)).

**OSIMM** defines a set of dimensions, representing different views of an organization, as follows:

- **The Business dimension** focuses on business architecture. i.e., the organization’s business practices and policies; how business processes are: designed, structured, implemented, and executed.

- **Organization and Governance** - The Organization and Governance dimension is: Focused on the structure and design of the organization itself and the required standards of organizational effectiveness in the context of an Service-Oriented Architecture (SOA) and SOA governance.

- **Method** - The Method dimension is: Focused on the techniques and processes used by the organization for its IT and business transformation, and the organization’s maturity around the Software Development Lifecycle.

- **Application** - The Application dimension is focused on application style, structuring of the application and functional decomposition, re-usability, flexibility, reliability, and extensibility of the applications.

- **Architecture** - The Architecture dimension is focused on the structure of the architecture.

- **Information** - The Information dimension is focused on how information is structured and how information is modeled.

- **Infrastructure and Management** - The Infrastructure and Management dimension is focused on the organization’s infrastructure capability, service management, IT operations, IT management and IT administration.

The seven **SOA maturity levels** are:

- **Level 1: Silo** - Individual parts of the organization are developing their own software independently, with no integration of data, processes, standards, or technologies.
• Level 2: Integrated - Technologies have been put in place to communicate between the silos, and to integrate the data and interconnections.

• Level 3: Componentized - The IT systems in the silos have been analyzed and broken down into component parts, with a framework in which they can be developed into new configurations and systems.

• Level 4: Service - Composite applications are built from loosely-coupled services.

• Level 5: Composite Services - At this level of service maturity it is now possible to construct a business process for a set of interacting services, not just by bespoke development, but by the use of a composition or business process modeling language, such as BPEL of information and control through the individual services.

• Level 6: Virtualized Services - The business and IT services are now provided through a façade – a level of indirection.

• Level 7: Dynamically Re-Configurable Services - Prior to this level, the business process assembly, although agile, is performed at design time by developers using suitable tooling.

The OSIMM maturity matrix which defines the maturity dimensions and levels is shown in OSIMM Maturity Matrix. Each one of the maturity levels are going to be addressed later in Chapter 2.3.1.

Figure 2.5: OSIMM Maturity Matrix - (Group, Service Integration Maturity Model (OSIMM))

Given the seven different dimensions OSIMM, this thesis will focus on the Architectural Dimension. And given the seven different levels of maturity, this thesis will focus on, through well-known patterns
for the analysis and definition of applications’ integration, given a silo organization (Level 1) growth to maturity level 3 (Componentized). Therefore, in the following chapter, we will analyze in greater detail both the architectural model and the maturity levels 1, 2 and 3.

2.3.1 Architecture Dimension: Base Model

This chapter defines the base model for the OSiMM Architecture dimension base model, since its the area of focus of this thesis. The base model defines a set of generic maturity indicators and attributes that can be used to assess an organization’s SOA maturity level against the OSiMM maturity matrix. Additional maturity indicators, assessment questions, and attribute mappings can be added by vendors or user organizations to extend the base OSiMM model.

The following assessment questions provided by the Open Group that can help determining the Architecture dimension maturity level (Group, Service Integration Maturity Model (OSiMM)).

**How would you characterize your architectural topologies?**

1. What type(s) of data repositories does your organization utilize?
2. What is the standard communication style in your architecture?
3. What methods do you use to develop your architecture?
4. How mature are your services implementations?
5. How extensive is your SOA?
6. What architectural principles define your approach?
7. How extensive and sophisticated is your organization’s use of frameworks in your architecture?
8. How are architectural decisions made in your organization?
9. Does your organization use reference architectures?

By gathering information using these assessment questions, it’s possible to map a maturity indicator to the associated maturity attributes, thereby determining the Architecture dimension maturity level. OSiMM helps an organization to build a roadmap for the transformation to a higher level of service integration. Our goal is to, given a silo enterprise, improve the level of maturity to at least a maturity level 3 (Componentized). Therefore, table 2.1 provided by The Open Group Service Integration Maturity Model OSiMM Version 2 – Architecture Dimension: Base Model, will help to see the distinctions between the levels of maturity and to address the assessment questions to the proper maturity attributes.
<table>
<thead>
<tr>
<th>Maturity Level</th>
<th>Maturity Indicator</th>
<th>Maturity Attributes and Assessment Question Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Silo (Level 1)</td>
<td>- Service components are designed using formal SOA methods, principles, patterns, frameworks, or techniques.</td>
<td>- Low or nonexistent. - No SOA methods or practices are apparent. (1, 7)</td>
</tr>
<tr>
<td>Integrated (Level 2)</td>
<td>- Service components are designed using formal SOA methods, principles, patterns, frameworks, or techniques.</td>
<td>- Limited - Limited use of formal SOA methods and practices can be observed. (1, 2, 5, 6, 7) - Methods and practices are limited to integration between applications or systems. (4, 8, 9)</td>
</tr>
<tr>
<td>Componentized (Level 3)</td>
<td>- Service components are designed using formal SOA methods, principles, patterns, frameworks, or techniques.</td>
<td>- Cross-organizational - Formal SOA methods and practices are employed by multiple groups within the enterprise. (4, 5, 6, 7, 8) - The organization has a loosely defined enterprise architecture supported by limited tooling and governance practices. (9, 10, 11)</td>
</tr>
<tr>
<td>Services (Level 4)</td>
<td>- Service components are designed using formal SOA methods, principles, patterns, frameworks, or techniques.</td>
<td>- Enterprise-wide - Formal SOA methods and practices are employed across the enterprise supported by a formal governance process. (4, 5, 6) - Applications and services are designed using formal SOA principles and patterns. (1, 7, 8, 11)</td>
</tr>
<tr>
<td>Composite Services (Level 5)</td>
<td>- Service components are designed using formal SOA methods, principles, patterns, frameworks, or techniques.</td>
<td>- Cross-organizational - Formal SOA methods and practices are employed by multiple groups within the enterprise. (7, 8, 9, 11) - The organization has a loosely defined enterprise architecture supported by limited tooling and governance practices. (2, 10)</td>
</tr>
<tr>
<td>Virtualized Services (Level 6)</td>
<td>- Service components are designed using formal SOA methods, principles, patterns, frameworks, or techniques.</td>
<td>- Integrated across the enterprise and externally between business partners. - Service components are designed using formal methods, practices, and frameworks that promote the re-use of assets. (1, 3, 4, 5, 6, 9) - Formal enterprise-wide business information services have been developed and deployed. (2, 8, 10, 11)</td>
</tr>
<tr>
<td>Dynamically Re-Configurable Services (Level 7)</td>
<td>- Service components are designed using formal SOA methods, principles, patterns, frameworks, or techniques.</td>
<td>- Adaptive Enterprise - Service components are designed using formal SOA methods, principles, patterns, frameworks, or techniques. (1, 3, 4, 5, 6, 9) - Formal enterprise business information services have been designed and implemented that include both enterprise and external relationship entities.. (2, 8, 10, 11)</td>
</tr>
</tbody>
</table>

Table 2.1: Maturity Indicators for the Architecture Dimension
Benefits of Moving to Higher Maturity Levels

From Silo to Integrated
Organizations transforming from a Silo maturity level to an Integrated maturity level will significantly reduce operational and maintenance costs. From this transition, the data is available on a real-time basis, with reliable delivery of data; and automated data format conversion for the integrating systems. The transformation from structured programming to object-oriented programming would also leverage the re-usability of the code.

From Integrated to Componentized
Organizations transforming from an Integrated maturity level to a Componentized maturity level would benefit in preparing themselves to expose the business functionality at a more granular level; such exposure is required at more advanced maturity levels. The re-usability also matures to be at a business function level as compared to the application level, and enhancements and new functionality are achieved through refactoring of the existing applications into smaller, re-usable components. This componentization also helps the organization in reducing the time-to-market and increases IT response to business changes.

For improving the level of maturity and SOA transformation journey, accordingly to The Open Group this process consists of the following steps:

- Prepare the OSIMM assessment framework.
- Determine the initial level of maturity.
- Determine the target level of maturity.
- Identify the transformation path necessary for the organization to achieve the desired level of maturity.

In this case, the initial level of maturity is a silo organization (Level 1). And throughout well-known patterns such as Messaging and Connectivity; Data Transformation; Message Brokers; Enterprise Service Buses; Application Interfaces; Legacy Integration; Portals; Mobile Integration; B2B Integration; Application Integration Platforms reach the target level of maturity 3 - Componentized.
2.4 Frameworks

Enterprise architecture serves as the blueprint for the system and the project that develops it. An enterprise architecture framework can describe the underlying infrastructure, thus providing the groundwork for the hardware, software, and networks to work together. Many of the enterprise architecture frameworks differ in terms of their approach and level of detail.

There are two main frameworks and standards managed by The Open Group: The Open Group Architecture Framework (TOGAF) and ArchiMate.

• The Open Group Architecture Framework (TOGAF) (Group, The Open Group Architecture Framework (TOGAF))

  • Is used for developing enterprise architectures. It provides a method for designing, planning, implementing, and governing an enterprise information technology architecture. This framework is constituted of seven parts including - Architecture Development Method (Architecture Development Method (ADM)). The ADM specifies a process for developing enterprise architecture and delineates a cyclic process of continuous architecture definition and realization.

![Figure 2.6: Structure of the TOGAF Architecture Development Method (ADM) [2]](image)

• ArchiMate

  • Is an open and independent enterprise architecture modeling language to support the description, analysis, and visualization of architecture. ArchiMate describes its applicability concerning the TOGAF ADM.
2.5 Communication

Still in the topic of EAI, Applications Integration is the process of different applications in an enterprise to share data enabling them to work together. Therefore, it’s essential to understand the following questions to help develop a pattern based integration architecture: What are the Applications in a given organization? For each one, what are the artifacts that enable communication? What kind of technology does this entail? What applications interact? And who calls them?

2.5.1 Types of message exchange

For a successful EAI implementation, the EAI has to be able to communicate with each of the participating applications. There are various ways to do this, including:

- End-to-End - The sender knows the destination of the destination;
- Broker - The sender sends a message to a broker and only this one knows the destination;
- Service invocation - Its functions include transmitting commands from the service consumer to the service provider and receiving results;
- Application Programming Interface (API) - The sender sends a message to a common interface; Direct API calls can be used for synchronous calls to local applications in the same technology and platforms.

In a message exchange what are the senders/receivers:

- End-to-End - The sender calls the receiver;
- Broker - The sender calls the broker and then the broker calls the receiver;
- Invocation of a service - The sender calls a service;
- API - The senders calls an interface;
2.5.2 Types of Communication

Communication can be:

• Synchronous communications - The sender and receiver must be in a state of sync before and during transmission.

• Asynchronous communications - The sender and receiver don’t have to be in sync. The communication may not occur at the same time. The sender can exchange a message and doesn’t have to wait for a response.

Typically, Computer processing is synchronous meaning that the sender waits while the receiver executes. However, in integrated applications accordingly with Gregor Hohpe and Bobby Woolf [5] asynchronous communication is the most appropriate type due to the high probability that not all applications are available at the same type.

2.6 Enterprise Integration Patterns

Enterprises have hundreds, if not thousands different applications, which makes enterprise integration too complex. Instead of starting an Enterprise Integration by stretch, patterns are commonly used as a starting point.

Patterns:

Patterns are abstract enough to apply to most integration technologies but specific enough to provide hands-on guidance to designers and architects. Patterns also provide a vocabulary for developers to efficiently describe their solution. Patterns are solutions to recurring problems within a given context. As stated before, a pattern is a good starting point but accordingly with the enterprise context adjustments will have to be made.

Patterns have become accepted in many different areas and described as very useful because they reuse knowledge already acquired from experienced users and they capture and document proven practices. The current patterns focus on Messaging, which forms the basis of most other integration patterns.

Patterns provide the following benefits [5]:

• Give you guidance for the implementation of solutions;
• Increase development efficiency, because resources are generated from a set of predefined templates;
• Result in higher-quality solutions, through reuse of assets and common implementation of programming approaches, such as error handling and logging;
This patterns are going to be explored in each subsection:

- Application Integration 2.6.1 - which covers well-known patterns such as Message Brokers; Enterprise Service Buses; Legacy Integration; Portals; Mobile Integration; B2B Integration.
- Application Structure 2.6.2 - that represents the way applications can share data and processes.
- Enterprise Integration 2.6.3 - That represents an aggregated view of all the integration patterns mentioned above.

2.6.1 Application Integration Implementation Patterns and Services

- **Message Broker**

  A message broker is an intermediary program. Programs interact by exchanging messages and these may or may not be in the same message protocol. Hence the role of a message broker is to translate a message from the sender's formal messaging protocol to the receiver's formal messaging protocol. The source application (producer) sends a message to a server process that can provide data marshaling, routing, message translation, persistence and delivery to all the proper destinations (consumers). The link between applications and broker are done with an application adapter, which adapts the communication protocols of both applications and broker. This adapter is accountable for: Extracting data from a source node; Transforming the data; Converting the schema; Routing the data to a target node. A message broker is also known as an integration broker or middleware. It uses a Hub/Spoke topology that provides a central message broker with an engine and adapters as spoke.

  In EAI, a Message Broker is required to provide information transmission services, which may be many types of different methods, such as Request/response; Publish/subscribe; One-on-one; One-to-many; Many-to-many communication;

  The broker can be developed by Common Object Request Broker Architecture (CORBA) Notification Service; JMS; Web Service; and other technology standards [8];

  The traditional solution for Enterprise Application Integration is to use MOM. This implies that asynchronous messaging is used to decouple the applications from each other. MOM products are typically built around a central message queue system often called message broker. All applications are connected to the message broker using a centralized interface for sending and receiving messages [9].

  Accordingly to the book, The essential Guide to Integrated Solutions [4], the Simple Message Broker Integration Reference Architecture is as shown in the Figure 2.8:
Description of Contents of Figure 2.8:

**Application Adapters** receive data from the source application and publish messages to the message broker, which, in turn, does **Translation/Transformation/Routing** and passes messages to subscribing adapters which sends them to destination application(s) [10].

**Web services** provide the capacity for organizations to expose specific functionality as services to other authorized users and systems. By using open, platform-independent standards, Web service-based solutions can support integrating distributed systems (e.g., J2EE, .NET, and legacy). In practice, a Web service usually provides an object-oriented web-based interface to a database server that provides a user interface to the end-user. Web services solutions are typically based on the following technologies:

- **Simple Object Access Protocol (SOAP)** is the standard protocol used to transmit XML-based messages in the implementation of web services in computer networks.
- **Universal Description, Discovery, and Integration (UDDI)** is a standard Extensible Markup Language (XML)-based registry used to advertise or publish available Web services.
- **Web Services Description Language (WSDL)** is an XML format of a common language that can be used to describe a Web service [11].

**Enterprise Service Bus**

The need for a new form of infrastructure that combines MOM, web services, transformation and routing intelligence in SOA becomes more clear with a large number of web services usually, where message brokers are replaced by ESB [12].

An **ESB** is a centralized software component that offers integrations to back-end systems and makes them available as service interfaces for reuse by new applications. The main difference between a message broker and an ESB is the topology. ESB uses a bus topology. Bus architecture requires an application adapter on the same platform as the applications and it’s the integration engine that performs message transformation and routing [10].
ESB features: Invocation; Routing; Mediation; Adapters; Security; Management; Process Orchestration; Complex Event Processing; Integration Tooling; Nowadays, ESB solutions are typically built with JMS based middleware systems which guarantee message delivery [9].

Accordingly to the book, The essential Guide to Integrated Solutions [4], the Enterprise Service Bus Integration Reference Architecture is as shown in the Figure 2.9:

![Figure 2.9: ESB Reference Architecture [4]](image)

Description of Contents of Figure 2.9 are already explained in Section Message Broker.

Since the architecture is the same what differs is that the message broker architecture has scalability issues and offers a single point of failure in the network, consequently with a large number of web services commonly, message brokers are replaced by Enterprise Service Buses (ESBs) [13].

For the following patterns the description of the contents of Web services, Database Management System, Application Adapters and Translation/Transformation/Routing are already explained in Section 4.1.1 Message Broker. Broker/ESB it’s explained in Section 4.1.1 Section Message Broker and in Section 4.1.2 ESB.

- Legacy Integration

A legacy system is an existing computer system or application program that the organization wishes to maintain.

Typically, an enterprise has existing legacy applications and databases they want to maintain, therefore the enterprise has to arrange a way to reuse what already exists while adding new applications and data. Determining how existing applications fit into the new view is one of the steps of the integration of legacies [14].

Legacy systems can: Adapt themselves to the changing business; Use the same functionality without re-writing the code; Expose information with a standard approach; [15]

Accordingly to the book, The essential Guide to Integrated Solutions [4], the Legacy Integration Reference Architecture is as shown in the Figure 2.10:
Other description of contents of Figure 2.10: A **Open Database Connectivity (ODBC)** is a standard application programming interface (API) for accessing database management systems (DBMS) [16]. **Java Database Connectivity (JDBC)** is a Java API to connect and execute the query with the database [17]. A **database API** allows processes to make calls to a database to invoke database services. Modern database software such as MICROSOFT SQL SERVER, IBM DB2, ORACLE DATABASE, and SYBASE IQ, all provide APIs. **MQSeries** is an IBM software whose components are used to join other software applications so that they can work together. **TIBCO** - provides integration, analytics, and event-processing software for companies to use on-premise or as part of cloud computing environments [18]. A **Message Bus** is a messaging infrastructure to allow different systems to communicate through a shared set of interfaces. **Screen Interface** - That can be done with: **Screen Scrapping** is the process of collecting screen display data from one application and translating it so that another application can display it [19]. **Service Interface** - That can be done with: **Web Services** - Explained in section 4.1.1 Message Broker; **EJB** - Java APIs for modular construction of enterprise software. **.NET** - Software framework developed by Microsoft. **CORBA** - The Common Object Request Broker Architecture is a standard defined by the Object Management Group designed to help the communication of systems that are used on different platforms. With **CORBA**, users obtain access to information transparently and allow applications to communicate with each other [20].

**B2B Integration**

**Business-to-Business (B2B)** integration is the automation of business processes and communication between two or more organizations. The need for a B2B Integration technology arises because the back-end-application systems are designed and built to operate in isolation. When installed side by side the only way to exchange data between them is by manually re-entering them into the different back-end application systems. However, since this approach is error prone, software technology is necessary in order to transfer data automatically without manual intervention. This software is B2B integration technology. [21]

Accordingly to the book, The essential Guide to Integrated Solutions [4], the B2B Integration Reference Architecture is as shown in the Figure 2.11:
Description of Contents of Figure 2.11: There are three phases of Integration Technology - **A2A Integration** to integrate Back-end applications; **B2B Integration** that provides connectivity to trading partners over networks; and **ASP Integration** that gives a subscriber access to it’s data [21]:

- **Portals**

  Accordingly to Fred A. Cummins, [22] “Portals are windows on the enterprise from the public internet”. The main purpose of a portal is to create contact points where the enterprise can manage who can access the information. This directly benefits the enterprise in two ways: first by routing the accesses - this reduces the computing workload, and second, using a user linking patterns - the enterprise can identify a target audience for marketing.

  There are four main categories: [22]

  - **Enterprise Portal** - This portal is the first *business card* of an enterprise in which, among several web pages, shows the content that the enterprise wants the target audience to see;

  - **Employee Portal** - This portal is a reverse firewall that allows certain people to pass through the intranet and blocks others;

  - **Retail Portal** - This portal is for the conducting business with the public, the first interaction people are not directly allowed to pass throughout the portal but after this step they are most likely to be directly allowed;
• Customer Service Portal - When the customer is known to the enterprise there can be special offers to draw customers’ attention to interesting opportunities;

Accordingly to the book, The essential Guide to Integrated Solutions [4], the Portal Integration Reference Architecture is as shown in the Figure 2.12:

![Figure 2.12: Portal Integration Reference Architecture [4]](image)

• Mobile Integration

Mobile Integration involves the integration into an enterprise infrastructure of mobile devices. Mobile applications need infrastructures for wireless data communication. These can be two main groups: Technologies for data traffic optimized mobile radio networks (3rd generation networks etc), and standards for local wireless networks (Wireless LAN (WLAN), Wireless Personal Area Networks (WPAN)). Based on these, different scenarios of an ad hoc-integration of mobile applications can be identified [23].

Accordingly to the book, The essential Guide to Integrated Solutions [4], the Mobile Integration Reference Architecture is as shown in the Figure 2.13:

![Figure 2.13: Mobile Integration Reference Architecture [4]](image)
2.6.2 Application Structure Implementation Patterns and Services

As stated earlier, an enterprise has multiple applications that are being built independently, with different languages and platforms. The enterprise needs to share data and processes in a responsive way.

There are many ways to integrate multiple applications so that they work together and exchange information such as: File Transfer; Shared Database; Remote Procedure Invocation; Messaging; SOAP and Representational State Transfer (REST) [5].

• File Transfer — Have each application produce files of shared data for others to consume and consume files that others have produced.

![Figure 2.14: File Transfer Diagram [5]](image)

Application A and B are both sharing data. Application A sends data to a shared file location where application B picks it up.

To integrate between two applications or two services from the protocol viewpoint: FTP/SFTP (File Transfer Protocol); HDFS (Hadoop Distributed File System); SCP (Secure Copy Protocol); SMB (Server Message Block); CIFS (Common Internet File System).

File transfer is a common way of integration style because it is: Universal Integration style (Near every platform can read and write files); Integration simplicity; and has the highest level of system abstraction. But on the other way, it has the following disadvantages: Error processing; Data synchronization time-less; data-only transfer.

• Shared Database — Have the applications store the data they wish to share in a common database.

![Figure 2.15: Shared Database Diagram [5]](image)
Application A, B, and C are all sharing the same data context and know nothing about each other.

Shared Database integration style uses the following Protocols: SQL (Structured Query Language); ODBC (Open Database Connectivity); JDBC (Java Database Connectivity); OLE-DB (Object Linking and Embedding Database); RDA (Remote Database Access).

The main advantages are Near-University Integration.; System Abstraction (Application A, B, and C can be in every platform); System Decoupling; Ease of Integration; On the other hand: Doesn’t work well with ORMs; Performance bottleneck issues; Schema change issues; Data owner-ships issues.

- **Remote Procedure Invocation** — Have each application expose some of its procedures so that they can be invoked remotely, and have applications invoke those to initiate behavior and exchange data.

![Remote Procedure Invocation Diagram](image)

**Figure 2.16:** Remote Procedure Invocation Diagram [5]

One of the main differences between Remote Procedure Invocation (RPI) instead of the other integration styles seen above is that Application A and B can not only share data but also invoke functionalities to be able to return the result.

Some Protocols used in RPI Integration style are REST; Remote Method Invocation (RMI); CORBA; WebSockets (Hixie, Hybu, RFC/Request for Comments); CGI (Common Gateway Interface).

The main advantages are: Data encapsulation and Ownership; External Systems Integration; Mature Frameworks and tools; on the other hand, the main disadvantages are: Tight System Coupling; Async communications; Broadcast capabilities.

- **Messaging** — Have each application connect to a common messaging system, and exchange data and invoke behavior using messages.
Applications A, B, and C are all communicating with one another through events using messaging. It doesn’t have to be a Message BUS. It can be a Message Broker, in both ways, they are communicating through events and messaging.

The main difference between integration style through messaging instead of the other integration styles is that throughout messaging applications are communicating to a broker to a queue, so we have a very high level of system abstraction and: Highly Decoupled Systems; Guaranteed Delivery; Async Communications; Broadcast Capabilities; Ease of Scalability.

The main disadvantages are: Integration beyond Firewall; Implementation Complexity; Testing Complexity; Cross-Platform Standards; Async Error Handling.

- **There are mainly two types of web services**: SOAP web services and RESTful web services. Simple Object Access Protocol (SOAP) is a protocol. SOAP was designed with a specification. It includes a WSDL file which has the required information on what the web service does in addition to the location of the web service. On the other hand, REST is an Architectural style in which a web service can only be treated as a RESTful service if it follows the constraints of being: Client Server; Stateless; Cacheable; Layered System; and Uniform Interface.

<table>
<thead>
<tr>
<th></th>
<th>SOAP</th>
<th>REST</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Architecture</strong></td>
<td>SOAP is a SOA based architecture that can be used for middleware interoperability.</td>
<td>REST is an architecture designed for web communication assuming point-to-point communication.</td>
</tr>
<tr>
<td><strong>Reliability</strong></td>
<td>Reliable.</td>
<td>Not Reliable.</td>
</tr>
<tr>
<td><strong>Requirements</strong></td>
<td>Requires standard tools with middleware support.</td>
<td>Doesn’t require any standard tools for requires a standard URI for information exchange.</td>
</tr>
<tr>
<td><strong>Communication Language</strong></td>
<td>XML based on XML formats.</td>
<td>JSON, XML, Microformats...</td>
</tr>
<tr>
<td><strong>Payload</strong></td>
<td>Must support SOAP schema.</td>
<td>Any format.</td>
</tr>
<tr>
<td><strong>Error Handling</strong></td>
<td>Doesn’t support.</td>
<td>Has built-in error handling.</td>
</tr>
</tbody>
</table>

**Table 2.2: SOAP vs. REST**

Accordingly to the book, Enterprise integration patterns: Designing, building, and deploying messaging solutions [5], the Application Structure Reference Architecture is shown in the Figure 2.18:
2.6.3 Enterprise Integration Implementation Patterns and Services

This subchapter represents a Reference Architecture for the fully Integrated Enterprise. It consists of an aggregated view of all the integration patterns, into a fully integrated infrastructure. Although each integration pattern has its unique technology requirement, there is also a significant overlap across patterns, including translation, transformation, and routing services, repositories, adapters, development environment, management, and security. Companies should strive to minimize redundancy at an enterprise level as much as possible to decrease long term maintenance costs. Therefore, and accordingly to the Essential Guide to Integration Solutions book [4], a fully Integrated Enterprise should have the following artefacts:
Figure 2.19: Integrated Enterprise Reference Architecture [4]
3 Proposed Solution

Contents

3.1 Proposed Solution: Overview ..................................................... 29
3.2 Evaluation of The Maturity Level ............................................. 29
3.3 Patterns conversion to ArchiMate .......................................... 35
3.4 Forms ................................................................................. 42
3.5 Blueprints ............................................................................ 50
3.1 Proposed Solution: Overview

This chapter will, through well-known patterns, given a silo organization (Level 1) growth to maturity level 3 (Componentized) accordingly to as mentioned in chapter 2 to the OSIMM model. For achieving this level of maturity, we will use patterns for designing an application integration technology using the ArchiMate for developing the blueprints and Google Form for developing the forms.

The following Image 3.1, presents a diagram of our solution:

![Figure 3.1: Proposed solution - Diagram](image)

The remaining chapter is divided as follows:
- Section 3.2 for the assessment of the level of maturity of the enterprise.
- Section 3.3 for the conversion of elements of the three different architectural views to ArchiMate.
- Section 3.4 that is divided into two parts: Pattern Attribution - with the purpose of through questions, collect important information to do the attribution of the pattern to a given application. And Pattern filling - that gathers information such as descriptions and characteristics to integrate the application within the given pattern.
- At last, Section 3.5 will present the Blueprints of the main patterns described in this thesis: Application Integration; Application Structure; and Enterprise Architecture Architecture.

3.2 Evaluation of The Maturity Level

As discussed in Chapter 2.3 the OSIMM matrix specifies how to measure the service integration levels of an organization. Ideally, depending on the maturity model, the models provided for gathering the information could be more refined. However, in this case, our forms are provided for enterprises with an architecture model base, maturity level from 1 (Silo) to 3 (Componentized). It’s important to throughout questions can assess the maturity level of an enterprise in a way that the elements we want to represent make sense. Meaning, that the enterprise has a level of understanding of not above or below the
components we need to assess.

Ideally, the enterprise provided has a level of maturity 1 - A silo. However, it's necessary to confirm this maturity level, since the forms developed in this thesis are more directed towards an integration level below 3 (Componentized). If the level of maturity is higher than 3 (Componentized), adjustments will have to be made to the questionnaires. With the Open Group assessment questions, we can easily determine if the enterprise has a level of maturity below or above 3 (Componentized).

There are several assessment questions provided by the Open Group that can help determining the Architecture dimension maturity level (Group, Service Integration Maturity Model (OSIMM)), as follows:

How would you characterize your architectural topologies?

Question 1: How would you characterize your architectural topologies?

<table>
<thead>
<tr>
<th>Possible Answer</th>
<th>Attribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monolithic Architecture</td>
<td>Silo (Level 1)</td>
</tr>
<tr>
<td>Layered Architecture</td>
<td>Integrated (Level 2)</td>
</tr>
<tr>
<td>Component Architecture</td>
<td>Componentized (Level 3)</td>
</tr>
<tr>
<td>Emerging SOA</td>
<td>Services (Level 4)</td>
</tr>
<tr>
<td>SOA,</td>
<td>Composite Services (level 5)</td>
</tr>
<tr>
<td>Grid Enabled SOA,</td>
<td>Virtualized services (level 6)</td>
</tr>
<tr>
<td>Dynamically Re-Configurable Architecture</td>
<td>Dynamically Re-Configurable Services (level 7)</td>
</tr>
</tbody>
</table>

Table 3.1: Table - Question: How would you characterize your architectural topologies?

Question 2: What type(s) of data repositories does your organization utilize?

<table>
<thead>
<tr>
<th>Possible Answer</th>
<th>Attribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Don’t utilize any</td>
<td>Silo (Level 1)</td>
</tr>
<tr>
<td>Local repository</td>
<td>Integrated (Level 2)</td>
</tr>
<tr>
<td>Information repository</td>
<td>Componentized (Level 3)</td>
</tr>
<tr>
<td>Multiples centrals repositories</td>
<td>Services (Level 4)</td>
</tr>
<tr>
<td>Relational Database Management System</td>
<td>Composite Services (level 5)</td>
</tr>
<tr>
<td>Workgroup repository</td>
<td>Virtualized services (level 6)</td>
</tr>
<tr>
<td>Electronic Document Management System</td>
<td>Dynamically Re-Configurable Services (level 7)</td>
</tr>
</tbody>
</table>

Table 3.2: Table - Question: What type(s) of data repositories does your organization utilize?
Question 3: What is the standard communication style in your architecture?

<table>
<thead>
<tr>
<th>Possible Answer</th>
<th>Attribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Don’t have</td>
<td>Silo (Level 1)</td>
</tr>
<tr>
<td>Layered architectural style</td>
<td>Integrated (Level 2)</td>
</tr>
<tr>
<td>Component-based Architecture</td>
<td>Componentized (Level 3)</td>
</tr>
<tr>
<td>Client-server architectural style</td>
<td>Services (Level 4)</td>
</tr>
<tr>
<td>Object-oriented Architectural Style</td>
<td>Composite Services (level 5)</td>
</tr>
<tr>
<td>Message bus architectural style</td>
<td>Virtualized services (level 6)</td>
</tr>
<tr>
<td>Service-Oriented Architecture</td>
<td>Dynamically Re-Configurable Services (level 7)</td>
</tr>
</tbody>
</table>

Table 3.3: Table - Question: What is the standard communication style in your architecture?

Question 4: How is integration achieved in your architecture?

<table>
<thead>
<tr>
<th>Possible Answer</th>
<th>Attribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nonexistent</td>
<td>Silo (Level 1)</td>
</tr>
<tr>
<td>Limited</td>
<td>Integrated (Level 2)</td>
</tr>
<tr>
<td>Cross-organizational</td>
<td>Componentized (Level 3)</td>
</tr>
<tr>
<td>Enterprise-wide</td>
<td>Services (Level 4)</td>
</tr>
<tr>
<td>Integrated Enterprise-wide</td>
<td>Composite Services (level 5)</td>
</tr>
<tr>
<td>Integrated across the enterprise and externally between business partners.</td>
<td>Virtualized services (level 6)</td>
</tr>
<tr>
<td>Adaptive Enterprise</td>
<td>Dynamically Re-Configurable Services (level 7)</td>
</tr>
</tbody>
</table>

Table 3.4: Table - Question: How is integration achieved in your architecture?

Question 5: What methods do you use to develop your architecture?

<table>
<thead>
<tr>
<th>Possible Answer</th>
<th>Attribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Silo (Level 1)</td>
</tr>
<tr>
<td>Limited use of formal SOA methods and practices can be observed.</td>
<td>Integrated (Level 2)</td>
</tr>
<tr>
<td>Formal SOA methods and practices are employed by multiple groups within the enterprise.</td>
<td>Componentized (Level 3)</td>
</tr>
</tbody>
</table>

Table 3.5: Table - Question: What methods do you use to develop your architecture?
<table>
<thead>
<tr>
<th>Possible Answer</th>
<th>Attribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formal SOA methods and practices are employed across the enterprise supported by a formal governance process.</td>
<td>Services (Level 4)</td>
</tr>
<tr>
<td>Enterprise frameworks and practices supported by the use of a formal SOA method and reference architectures across the enterprise.</td>
<td>Composite Services (level 5)</td>
</tr>
<tr>
<td>Service components are designed using formal methods, practices, and frameworks that promote the re-use of assets.</td>
<td>Virtualized services (level 6)</td>
</tr>
<tr>
<td>Service components are designed using formal SOA methods, principles, patterns, frameworks, or techniques.</td>
<td>Dynamically Re-Configurable Services (level 7)</td>
</tr>
</tbody>
</table>

**Table 3.6:** Table 2 - Question: What methods do you use to develop your architecture?

Question 6: How mature are your services implementations?

<table>
<thead>
<tr>
<th>Possible Answer</th>
<th>Attribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>We don’t have metrics to verify how mature are our services.</td>
<td>Silo (Level 1)</td>
</tr>
<tr>
<td>Limited use of formal SOA methods and practices can be observed.</td>
<td>Integrated (Level 2)</td>
</tr>
<tr>
<td>Formal SOA methods and practices are employed by multiple groups within the enterprise.</td>
<td>Componentized (Level 3)</td>
</tr>
<tr>
<td>Formal SOA methods and practices are employed across the enterprise supported by a formal governance process.</td>
<td>Services (Level 4)</td>
</tr>
<tr>
<td>Enterprise frameworks and practices supported by the use of a formal SOA method and reference architectures across the enterprise.</td>
<td>Composite Services (level 5)</td>
</tr>
<tr>
<td>Service components are designed using formal methods, practices, and frameworks that promote the re-use of assets.</td>
<td>Virtualized services (level 6)</td>
</tr>
<tr>
<td>Service components are designed using formal SOA methods, principles, patterns, frameworks, or techniques.</td>
<td>Dynamically Re-Configurable Services (level 7)</td>
</tr>
</tbody>
</table>

**Table 3.7:** Table - Question: How mature are your services implementations?
Question 7: How extensive is your SOA?

<table>
<thead>
<tr>
<th>Possible Answer</th>
<th>Attribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>No SOA methods or practices are apparent.</td>
<td>Silo (Level 1)</td>
</tr>
<tr>
<td>Limited use of formal SOA methods and practices can be observed.</td>
<td>Integrated (Level 2)</td>
</tr>
<tr>
<td>Formal SOA methods and practices are employed by multiple groups within the enterprise.</td>
<td>Componentized (Level 3)</td>
</tr>
<tr>
<td>Applications and services are designed using formal SOA principles and patterns.</td>
<td>Services (Level 4)</td>
</tr>
<tr>
<td>Enterprise frameworks and practices supported by the use of a formal SOA method and reference architectures across the enterprise.</td>
<td>Composite Services (level 5)</td>
</tr>
<tr>
<td>Internal services follow an organizational SOA reference architecture for their development.</td>
<td>Virtualized services (level 6)</td>
</tr>
<tr>
<td>Internal and external services follow an inter-organizational SOA reference architecture for their development.</td>
<td>Dynamically Re-Configurable Services (level 7)</td>
</tr>
</tbody>
</table>

**Table 3.8:** Table - Question: How extensive is your SOA?

Question 8: What architectural principles define your approach?

<table>
<thead>
<tr>
<th>Possible Answer</th>
<th>Attribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>No SOA reference architecture</td>
<td>Silo (Level 1)</td>
</tr>
<tr>
<td>Methods and practices are limited to integration between applications or systems.</td>
<td>Integrated (Level 2)</td>
</tr>
<tr>
<td>Formal SOA methods and practices are employed by multiple groups within the enterprise.</td>
<td>Componentized (Level 3)</td>
</tr>
<tr>
<td>A SOA reference architecture emerges at enterprise level</td>
<td>Services (Level 4)</td>
</tr>
<tr>
<td>Central application services throughout the organization follow one standard SOA reference architecture.</td>
<td>Composite Services (level 5)</td>
</tr>
<tr>
<td>Internal services follow an organizational SOA reference architecture for their development.</td>
<td>Virtualized services (level 6)</td>
</tr>
<tr>
<td>Internal and external services follow an inter-organizational SOA reference architecture for their development.</td>
<td>Dynamically Re-Configurable Services (level 7)</td>
</tr>
</tbody>
</table>

**Table 3.9:** Table - Question: What architectural principles define your approach?
Question 9: How extensive and sophisticated is your organization’s use of frameworks in your architecture?

<table>
<thead>
<tr>
<th>Possible Answer</th>
<th>Attribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>There is no framework</td>
<td>Silo (Level 1)</td>
</tr>
<tr>
<td>Methods and practices are limited to integration between applications or systems.</td>
<td>Integrated (Level 2)</td>
</tr>
<tr>
<td>The organization has a loosely defined enterprise architecture supported by limited tooling and governance practices.</td>
<td>Componentized (Level 3)</td>
</tr>
<tr>
<td>Service framework that is centralized at enterprise level</td>
<td>Services (Level 4)</td>
</tr>
<tr>
<td>Enterprise frameworks and practices supported by the use of a formal SOA method and reference architectures across the enterprise.</td>
<td>Composite Services (level 5)</td>
</tr>
<tr>
<td>Service components are designed using formal methods, practices, and frameworks that promote the re-use of assets.</td>
<td>Virtualized services (level 6)</td>
</tr>
<tr>
<td>Service components are designed using formal SOA methods, principles, patterns, frameworks, or techniques.</td>
<td>Dynamically Re-Configurable Services (level 7)</td>
</tr>
</tbody>
</table>

Table 3.10: Table - Question: How extensive and sophisticated is your organization’s use of frameworks in your architecture?

Question 10: How are architectural decisions made in your organization?

<table>
<thead>
<tr>
<th>Possible Answer</th>
<th>Attribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>There is not architectural decisions</td>
<td>Silo (Level 1)</td>
</tr>
<tr>
<td>Made ad hoc</td>
<td>Integrated (Level 2)</td>
</tr>
<tr>
<td>The organization has a loosely defined enterprise architecture supported by limited tooling and governance practices.</td>
<td>Componentized (Level 3)</td>
</tr>
<tr>
<td>Enterprise level based on policies</td>
<td>Services (Level 4)</td>
</tr>
<tr>
<td>A formal enterprise business information model is evolving.</td>
<td>Composite Services (level 5)</td>
</tr>
<tr>
<td>Formal enterprise-wide business information services have been developed and deployed.</td>
<td>Virtualized services (level 6)</td>
</tr>
<tr>
<td>Formal enterprise business information services have been designed and implemented that include both enterprise and external relationship entities.</td>
<td>Dynamically Re-Configurable Services (level 7)</td>
</tr>
</tbody>
</table>

Table 3.11: Table - Question: How are architectural decisions made in your organization?
Question 11: Does your organization use reference architectures?

<table>
<thead>
<tr>
<th>Possible Answer</th>
<th>Attribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Silo (Level 1)</td>
</tr>
<tr>
<td>Under development</td>
<td>Integrated (Level 2)</td>
</tr>
</tbody>
</table>

The organization has a loosely defined enterprise architecture supported by limited tooling and governance practices.

Componentized (Level 3)

Applications and services are designed using formal SOA principles and patterns.

Services (Level 4)

Enterprise frameworks and practices supported by the use of a formal SOA method and reference architectures across the enterprise.

Composite Services (level 5)

Formal enterprise-wide business information services have been developed and deployed.

Virtualized services (level 6)

Formal enterprise business information services have been designed and implemented that include both enterprise and external relationship entities.

Dynamically Re-Configurable Services (level 7)

Table 3.12: Table - Question: Does your organization use reference architectures?

3.3 Patterns conversion to ArchiMate

The following subsections are the conversion of elements of the three different architectural views to ArchiMate. Such as Application Integration Technology - which covers well-known patterns such as Message Brokers; Enterprise Service Buses; Legacy Integration; Portals; Mobile Integration; B2B Integration. Application Structure - that represents the way applications can share data and processes. And Enterprise Architecture - That represents an aggregated view of all the integration patterns mentioned above.

3.3.1 Application Integration

Applications Integration (or Enterprise Application Integration or EAI) is the process of different applications in an enterprise to share data enabling them to work together.

This section does a brief description of the concept and the ArchiMate elements for each element in each application integration scenario. These scenarios are: taken from The Essential Guide to Integration Solutions book [4] such as Message Brokers; Enterprise Service Buses; Legacy Integration; Portals; Mobile Integration; B2B Integration.
Message Brokers:

As mentioned in the previous section 2.6.1, a Message Broker is an intermediary program, and one of the main patterns described for an Application Integration. Accordingly to the book, The Essential Guide to Integrated Solutions [4], the Simple Message Broker Integration Reference Architecture is as shown in Figure 2.8. A brief description of the concept and the ArchiMate elements that these concepts represent are provided in Table 3.13 as follows:

<table>
<thead>
<tr>
<th>Concept</th>
<th>Description</th>
<th>ArchiMate</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Adapter</td>
<td>A layer of software that converts the data from the application into a common form acceptable for integration with other applications</td>
<td><img src="image" alt="Application interface" /></td>
<td>A point of access where application services are made available.</td>
</tr>
<tr>
<td>Translation / Transformation</td>
<td>A database of digital content with an associated set of data management, search and access methods.</td>
<td><img src="image" alt="Application interaction" /></td>
<td>A unit of collective application behavior performed by (a collaboration of) two or more application components.</td>
</tr>
<tr>
<td>Routing</td>
<td>A database of digital content with an associated set of data management, search and access methods.</td>
<td><img src="image" alt="Application interaction" /></td>
<td>A unit of collective application behavior performed by (a collaboration of) two or more application components.</td>
</tr>
<tr>
<td>Database Interface</td>
<td>User interface that allows the user to input queries to a database without using the query language.</td>
<td><img src="image" alt="Application interface" /></td>
<td>A point of access where application services are made available.</td>
</tr>
<tr>
<td>Web Services</td>
<td>Term for a World Wide Web site for users when they get connected to the Web.</td>
<td><img src="image" alt="Application interface" /></td>
<td>A point of access where application services are made available.</td>
</tr>
</tbody>
</table>

Table 3.13: Description of concepts - Message broker

ESB:

Description of Contents of Figure 2.9 are already explained in Section Message Broker.

Since the architecture is the same what differs is that the message broker architecture has scalability issues and offers a single point of failure in the network, consequently with a large number of web services commonly, message brokers are replaced by Enterprise Service Buses (ESBs) [13].

For the following patterns the description of the contents of Application Adapters, Translation-/ Transformation/Routing, Database Interface, and Web services are already explained in Section
Message Broker.

**Legacy Integration:**

As mentioned in the previous section 2.6.1, a Legacy is an existing computer system or application program that the organization wishes to maintain, and one of the main patterns described for an Application Integration. Accordingly to the book, The Essential Guide to Integrated Solutions [4], the Legacy Integration Reference Architecture is as shown in Figure 2.10. A brief description of the concept and the ArchiMate elements that these concepts represent are provided in Table 3.14 as follows:

<table>
<thead>
<tr>
<th>Concept</th>
<th>Description</th>
<th>ArchiMate</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Interface</td>
<td>Set of attributes describing a given entity, used to create processes that read from, or write to, interfaces rather than directly from sources of data.</td>
<td><img src="image" alt="Application Interface" /></td>
<td>A point of access where application services are made available.</td>
</tr>
<tr>
<td>Message Interface</td>
<td>Used to describe a platform which is used to exchange messages.</td>
<td><img src="image" alt="Application Interface" /></td>
<td>A point of access where application services are made available.</td>
</tr>
<tr>
<td>Screen Interface</td>
<td>Shared boundary across which two or more separate components of a computer system exchange information.</td>
<td><img src="image" alt="Application Interface" /></td>
<td>A point of access where application services are made available.</td>
</tr>
<tr>
<td>Service Interface</td>
<td>Published interface used to invoke a service.</td>
<td><img src="image" alt="Application Interface" /></td>
<td>A point of access where application services are made available.</td>
</tr>
</tbody>
</table>

**Table 3.14:** Description of concepts - Legacy

**B2B:**

As mentioned in the previous section 2.6.1, B2B integration is the automation of business processes and communication between two or more organizations, and one of the main patterns described for an Application Integration. Accordingly to the book, The Essential Guide to Integrated Solutions [4], the B2B Integration Reference Architecture is as shown in Figure 2.11. A brief description of the concept and the ArchiMate elements that these concepts represent are provided in Table 3.15 as follows:
<table>
<thead>
<tr>
<th>Concept</th>
<th>Description</th>
<th>ArchiMate</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legacy Integration</td>
<td>Include the integration and Web extension of existing (legacy) systems.</td>
<td><img src="image" alt="Device" /></td>
<td>A physical IT resource upon which system software and artifacts may be stored or deployed for execution.</td>
</tr>
<tr>
<td>Integration Broker</td>
<td>Capability to maintain a unified view of data and information for an organization</td>
<td><img src="image" alt="Device" /></td>
<td>A physical IT resource upon which system software and artifacts may be stored or deployed for execution.</td>
</tr>
<tr>
<td>Enterprise Service Bus</td>
<td>Software architecture model.</td>
<td><img src="image" alt="Device" /></td>
<td>A physical IT resource upon which system software and artifacts may be stored or deployed for execution.</td>
</tr>
<tr>
<td>Firewall</td>
<td>Control point for access to Web Servers, the enterprise intranet, and enterprise systems.</td>
<td><img src="image" alt="Device" /></td>
<td>A physical IT resource upon which system software and artifacts may be stored or deployed for execution.</td>
</tr>
<tr>
<td>LAN</td>
<td>Local area network.</td>
<td><img src="image" alt="Communication network" /></td>
<td>A set of structures and behaviors that connects computer systems for transmission, routing, and reception of data.</td>
</tr>
<tr>
<td>B2B Exchange</td>
<td>Online marketplaces.</td>
<td><img src="image" alt="Application component" /></td>
<td>An encapsulation of application functionality aligned to implementation structure, which is modular and replaceable.</td>
</tr>
<tr>
<td>B2B Server</td>
<td>Automated configuration, management, and coordination of computer systems, apps, and services.</td>
<td><img src="image" alt="Device" /></td>
<td>A physical IT resource upon which system software and artifacts may be stored or deployed for execution.</td>
</tr>
<tr>
<td>B2B Connectivity</td>
<td>Enables collaboration and allows exchange data.</td>
<td><img src="image" alt="Communication network" /></td>
<td>A set of structures and behaviors that connects computer systems for transmission, routing, and reception of data.</td>
</tr>
</tbody>
</table>

Table 3.15: Description of concepts - B2B
Portals:

As mentioned in the previous section 2.6.1, Portal integration purpose is to create contact points where the enterprise can manage who can access the information, and one of the main patterns described for an Application Integration. Accordingly to the book, The Essential Guide to Integrated Solutions [4], the Portal Integration Reference Architecture is as shown in Figure 2.12. A brief description of the concept and the ArchiMate elements that these concepts represent are provided in Table 3.16 as follows:

<table>
<thead>
<tr>
<th>Concept</th>
<th>Description</th>
<th>ArchiMate</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application server</td>
<td>Program that manages all application operations between users and an organization's back-end business applications or databases.</td>
<td><img src="image" alt="Device" /></td>
<td>A physical IT resource upon which system software and artifacts may be stored or deployed for execution.</td>
</tr>
<tr>
<td>Enterprise information integration (EII)</td>
<td>Capability to maintain a unified view of data and information for an organization</td>
<td><img src="image" alt="System software" /></td>
<td>Software that provides or contributes to an environment for storing, executing, and using software or data deployed within it.</td>
</tr>
<tr>
<td>Portal</td>
<td>Term for a World Wide Web site for users when they get connected to the Web.</td>
<td><img src="image" alt="Device" /></td>
<td>A physical IT resource upon which system software and artifacts may be stored or deployed for execution.</td>
</tr>
</tbody>
</table>

Table 3.16: Description of concepts - Portal

Mobile Integration:

As mentioned in the previous section 2.6.1, Mobile integration involves the integration into an enterprise infrastructure of mobile devices, and one of the main patterns described for an Application Integration. Accordingly to the book, The Essential Guide to Integrated Solutions [4], the Mobile Integration Reference Architecture is as shown in Figure 2.13. A brief description of the concept and the ArchiMate elements that these concepts represent are provided in Table 3.17 as follows:
3.3.2 Application Structure

As mentioned in the previous section 2.6.2, there are many ways to integrate multiple applications so that they work together and exchange information such as: File Transfer; Shared Database; Remote Procedure Invocation; Messaging; SOAP and REST. Accordingly to the book, Enterprise integration patterns: Designing, building, and deploying messaging solutions [5], the Application Structure Reference Architecture is shown in the Figure 2.18. A brief description of the concept and the ArchiMate elements that these concepts represent are provided in Table 3.18.

<table>
<thead>
<tr>
<th>Concept</th>
<th>Description</th>
<th>ArchiMate</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile Server</td>
<td>Computer system that responds to requests over a computer network to provide network service and it's easily transportable in a laptop form factor.</td>
<td><img src="image" alt="Device" /></td>
<td>A physical IT resource upon which system software and artifacts may be stored or deployed for execution.</td>
</tr>
<tr>
<td>Stub</td>
<td>Hides the serialization of parameters and the network-level communication in order to present a simple invocation mechanism to the caller.</td>
<td><img src="image" alt="Technology collaboration" /></td>
<td>An aggregate of two or more nodes that work together to perform collective technology behavior.</td>
</tr>
<tr>
<td>Skeleton</td>
<td>Is responsible for dispatching the call to the actual remote object implementation.</td>
<td><img src="image" alt="Technology collaboration" /></td>
<td>An aggregate of two or more nodes that work together to perform collective technology behavior.</td>
</tr>
<tr>
<td>Service Requester</td>
<td>Publishes its interface in the registry.</td>
<td><img src="image" alt="Application collaboration" /></td>
<td>An aggregate of two or more application components that work together to perform collective application behavior.</td>
</tr>
<tr>
<td>Service Provider</td>
<td>Uses the registry to find a service provider and invokes it.</td>
<td><img src="image" alt="Application collaboration" /></td>
<td>An aggregate of two or more application components that work together to perform collective application behavior.</td>
</tr>
</tbody>
</table>

Table 3.17: Description of concepts - Mobile Integration

Table 3.18: Description of concepts - Application Structure
### 3.3.3 Enterprise Integration Architecture

As mentioned in the previous section 2.6.3, an Integrated Enterprise consists of an aggregated view of all the integration patterns, into a fully integrated infrastructure. Accordingly to the Essential Guide to Integration Solutions book [4], the Enterprise Integration Reference Architecture is shown in the Figure 2.19 a brief description of the concept and the ArchiMate elements that these concepts represent are provided in Table 3.19 and in Table 3.20.

<table>
<thead>
<tr>
<th>Concept</th>
<th>Description</th>
<th>ArchiMate</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legacy Interface</td>
<td>Defines several classes and interfaces that provide method for storing objects.</td>
<td><img src="image" alt="Application interface" /></td>
<td>A point of access where application services are made available.</td>
</tr>
<tr>
<td>Data cleaning</td>
<td>Process of detecting and correcting corrupt or inaccurate records from a record set, table, or database.</td>
<td><img src="image" alt="Application process" /></td>
<td>A sequence of application behaviors that achieves a specific outcome.</td>
</tr>
<tr>
<td>Metadata Repository</td>
<td>Database created to store metadata.</td>
<td><img src="image" alt="System software" /></td>
<td>Software that provides or contributes to an environment for storing, executing, and using software or data deployed within it.</td>
</tr>
<tr>
<td>Content Repository</td>
<td>Database of digital content.</td>
<td><img src="image" alt="System software" /></td>
<td>Software that provides or contributes to an environment for storing, executing, and using software or data deployed within it.</td>
</tr>
<tr>
<td>Aggregated Query</td>
<td>Method of deriving group and subgroup data by analysis of a set of individual data entries.</td>
<td><img src="image" alt="Application process" /></td>
<td>A sequence of application behaviors that achieves a specific outcome.</td>
</tr>
<tr>
<td>Rules engine</td>
<td>Software system that executes one or more business rules in a runtime production environment.</td>
<td><img src="image" alt="System software" /></td>
<td>Software that provides or contributes to an environment for storing, executing, and using software or data deployed within it.</td>
</tr>
<tr>
<td>Process Management / Orchestration</td>
<td>Is the means by which a process is executed, and monitored and managed throughout its life-cycle.</td>
<td><img src="image" alt="Application process" /></td>
<td>A sequence of application behaviors that achieves a specific outcome.</td>
</tr>
</tbody>
</table>

**Table 3.19: Description of concepts - Enterprise Integration Architecture 1**
3.4 Forms

Still on the subject of integration of architecture patterns we are going to use Google Forms - that allow us, throughout questions, to gather the information needed to define the three different architectural views. Such as Application Integration Technology - which covers well-known patterns such as Message Brokers; Enterprise Service Buses; Legacy Integration; Portals; Mobile Integration; B2B Integration. Application Structure - that represents the way applications can share data and processes. And Enterprise Architecture - That represents an aggregated view of all the integration patterns mentioned above.
Within this step, two different Forms are attributed to each architectural view such as: 3.4.1 and 3.4.2. In the next sub-chapters, we will see in more detail these two solutions created.

### 3.4.1 Pattern Attribution

First, we created a form, using Google forms, with multiple-choice questions. The purpose of this first form is to, through questions, collect important information to do the attribution of the pattern to a given application.

#### 3.4.1.A Template of an Application Integration Implementation Specification

For the architectural view of Application Integration. And after the study of each well-known patterns such as Message Brokers; Enterprise Service Buses; Legacy Integration; Portals; Mobile Integration; B2B Integration shown in 2.6.1, a group of questions was developed with the purpose to be able to do the attribution of the pattern to a given application.

The following Table 3.21 corresponds to the questions in the Google Forms created and shown in chapter 6. This Form assigns for each question on the form, the proper attribution of pattern to which the application corresponds.

<table>
<thead>
<tr>
<th>Question: Does your application?</th>
<th>Attribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translate a message from the sender's formal messaging protocol to the receiver's formal messaging protocol?</td>
<td>Message Broker (Figure: 2.8)</td>
</tr>
<tr>
<td>Does the source application (producer) sends a message to a server?</td>
<td>Message Broker (Figure: 2.8)</td>
</tr>
<tr>
<td>Use a centralised software component?</td>
<td>ESB (Figure: 2.9)</td>
</tr>
<tr>
<td>Does the application adapter stay on the same platform as the applications?</td>
<td>ESB (Figure: 2.9)</td>
</tr>
<tr>
<td>Have any existing computer system or application program that the organisation wishes to maintain?</td>
<td>Legacy (Figure: 2.10)</td>
</tr>
<tr>
<td>Reuse what already exists while adding new applications and data?</td>
<td>Legacy (Figure: 2.10)</td>
</tr>
<tr>
<td>Communicate between two or more organisations?</td>
<td>B2B (Figure: 2.11)</td>
</tr>
<tr>
<td>Create contact points where the enterprise can manage who can access the information?</td>
<td>Portal (Figure: 2.12)</td>
</tr>
<tr>
<td>Involve the integration into an enterprise infrastructure of mobile devices?</td>
<td>Mobile (Figure: 2.13)</td>
</tr>
</tbody>
</table>

| Table 3.21: Application Integration Form - Attribution of Pattern |

#### 3.4.1.B Template of an Application Structure Implementation Specification

For the architectural view of Application Structure. And after the study of how applications can share data and processes throughout File transfer, Shared Database; Remote Procedure Invocation; Messag-
ing. shown in 2.6.2, a group of questions was developed with the purpose to be able to do the attribution of the pattern to a given application.

<table>
<thead>
<tr>
<th>Question: Does your application?</th>
<th>Attribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Produce files of shared data for others to consume?</td>
<td>File Transfer (Figure: 2.14)</td>
</tr>
<tr>
<td>Consume files that others have produced?</td>
<td>File Transfer (Figure: 2.14)</td>
</tr>
<tr>
<td>Store the data in a common database?</td>
<td>Shared Database (Figure: 2.15)</td>
</tr>
<tr>
<td>Expose some of its procedures so that they can be invoked remotely?</td>
<td>Remote Procedure Invocation (Figure: 2.16)</td>
</tr>
<tr>
<td>Invoke procedures to run behavior and exchange data?</td>
<td>Remote Procedure Invocation (Figure: 2.16)</td>
</tr>
<tr>
<td>Connect to a common messaging system?</td>
<td>Messaging (Figure: 2.17)</td>
</tr>
<tr>
<td>Exchange data and invoke behavior using messages?</td>
<td>Messaging (Figure: 2.17)</td>
</tr>
</tbody>
</table>

Table 3.22: Application Structure Form - Attribution of Pattern

3.4.2 Pattern filling

After the attribution of the proper pattern, gathering information such as descriptions and characteristics are important to integrate the application within the given pattern. For this, a form was created with Google forms, with a group of open questions, where these descriptions of content can be registered.

3.4.2.A Template of an Application Integration Implementation Specification

After the attribution of the proper pattern provided in Table 3.21, Tables: 3.23; 3.24; 3.25; 3.26; 3.27; 3.28; 3.29 correspond to the questions in the Google Forms created and shown in chapter 6. This Form gathers information such as descriptions and characteristics that are important to integrate the application within the given pattern.

• Message Brokers

The Message Broker Integration form template is described in Table 3.23 that corresponds to the questions in the Google Forms created and shown in chapter 6.
### Table 3.23: Message Broker Implementation Table [4]

<table>
<thead>
<tr>
<th>Integration Service</th>
<th>Vendor/Product</th>
<th>Implementation Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Broker</td>
<td>Vendor name/product name</td>
<td>Technology platform</td>
</tr>
<tr>
<td>Messaging</td>
<td>Vendor name/product name if different from broker vendor</td>
<td>Technology platform</td>
</tr>
<tr>
<td>Application Interface</td>
<td>Vendor name/product name or deployment technology</td>
<td>Specify: Integration broker adapter; Web Service; JCA interface; data interface (ODBC; JDBC; OLE; DB; ADO)</td>
</tr>
<tr>
<td>Security</td>
<td>Vendor name/product name or deployment technology</td>
<td>Specify how security is implemented, including integration with LDAP or other repository or product</td>
</tr>
</tbody>
</table>

### ESB

The ESB Integration form template is described in Table 3.24 that corresponds to the questions in the Google Forms created and shown in chapter 6.

### Table 3.24: ESB Implementation Table [4]

<table>
<thead>
<tr>
<th>Integration Service</th>
<th>Vendor/Product</th>
<th>Implementation Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enterprise Service Bus (ESB)</td>
<td>Vendor name/product name</td>
<td>Technology platform</td>
</tr>
<tr>
<td>Translation / Transformation</td>
<td>Vendor name/product name</td>
<td>Technology at hub or target</td>
</tr>
<tr>
<td>Application Interface</td>
<td>Vendor name/product name or deployment technology</td>
<td>Specify: Integration broker adapter; Web Service; JCA interface; data interface (ODBC; JDBC; OLE; DB; ADO)</td>
</tr>
<tr>
<td>Security</td>
<td>Vendor name/product name or deployment technology</td>
<td>Specify how security is implemented, including integration with LDAP or other repository or product</td>
</tr>
</tbody>
</table>

### Legacy Integration

The Legacy Integration form template is described in Table 3.25 that corresponds to the questions in the Google Forms created and shown in chapter 6.
### Legacy Integration - data interface

<table>
<thead>
<tr>
<th>Integration Service</th>
<th>Vendor/Product or Custom</th>
<th>Implementation Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legacy Integration - data interface</td>
<td>Vendor/product name</td>
<td>JDBC adapter to specific mainframe data source</td>
</tr>
</tbody>
</table>

### Legacy Integration - message interface

<table>
<thead>
<tr>
<th>Integration Service</th>
<th>Vendor/Product or Custom</th>
<th>Implementation Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legacy Integration - message interface</td>
<td>Vendor/product name</td>
<td>MQ Series on mainframe provides success to CICS transactions</td>
</tr>
</tbody>
</table>

### Legacy Integration - screen/report interface

<table>
<thead>
<tr>
<th>Integration Service</th>
<th>Vendor/Product or Custom</th>
<th>Implementation Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legacy Integration - screen/report interface</td>
<td>Vendor /product name</td>
<td>Provides all customer order information</td>
</tr>
</tbody>
</table>

### Legacy Integration - service interface

<table>
<thead>
<tr>
<th>Integration Service</th>
<th>Vendor/Product or Custom</th>
<th>Implementation Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legacy Integration - service interface</td>
<td>Vendor/product name or custom code</td>
<td>Order processing service, customer maintenance service, order, tracking service</td>
</tr>
</tbody>
</table>

---

#### Table 3.25: Legacy Integration Implementation Table [4]

---

**B2B**

The B2B Integration form template is described in Table 3.26 and in Table 3.27 that correspond to the questions in the Google Forms created and shown in chapter 6.

---

### B2B Connectivity

<table>
<thead>
<tr>
<th>Integration Service</th>
<th>Vendor/Product</th>
<th>Implementation Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>B2B Connectivity</td>
<td>Vendor/product name</td>
<td>B2B server or central exchange</td>
</tr>
</tbody>
</table>

### Partner connectivity

<table>
<thead>
<tr>
<th>Integration Service</th>
<th>Vendor/Product</th>
<th>Implementation Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partner connectivity</td>
<td>Vendor/product name</td>
<td>List all connectivity options including XML messaging, EDI, FTP, browser interfaces, exchange hub, or other</td>
</tr>
</tbody>
</table>

### Partner management

<table>
<thead>
<tr>
<th>Integration Service</th>
<th>Vendor/Product</th>
<th>Implementation Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partner management</td>
<td>Vendor /product name</td>
<td>Define services provided including process management, collaboration services, service level agreements, or others</td>
</tr>
</tbody>
</table>

### B2B security

<table>
<thead>
<tr>
<th>Integration Service</th>
<th>Vendor/Product</th>
<th>Implementation Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>B2B security</td>
<td>Vendor/product name</td>
<td>Define all security services provided including encryption, authentication through digital certificates or other authentication, nonrepudiation services</td>
</tr>
</tbody>
</table>

### Back-End Application Integration

<table>
<thead>
<tr>
<th>Integration Service</th>
<th>Vendor/Product</th>
<th>Implementation Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Back-End Application Integration</td>
<td></td>
<td>Include all application integration services included in the B2B solution</td>
</tr>
</tbody>
</table>

### Broker/enterprise service bus

<table>
<thead>
<tr>
<th>Integration Service</th>
<th>Vendor/Product</th>
<th>Implementation Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Broker/enterprise service bus</td>
<td>Product/protocol</td>
<td>Technology platform</td>
</tr>
</tbody>
</table>

### Translation and transformation

<table>
<thead>
<tr>
<th>Integration Service</th>
<th>Vendor/Product</th>
<th>Implementation Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translation and transformation</td>
<td>Vendor name/product name</td>
<td>Transformation at hub or target</td>
</tr>
</tbody>
</table>

---

#### Table 3.26: B2B Integration Implementation Table 1 [4]
<table>
<thead>
<tr>
<th>Integration Service</th>
<th>Vendor/Product</th>
<th>Implementation Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routing</td>
<td>Vendor name/product name</td>
<td>Level of routing supported; content based, business rules, header only info</td>
</tr>
<tr>
<td>Application Interface</td>
<td>Vendor name/product name or deployment technology</td>
<td>Specify: Integration broker adapter; Web Service; JCA interface; data interface (ODBC; JDBC; OLE; DB; ADO)</td>
</tr>
<tr>
<td>Legacy integration</td>
<td>Vendor/product name</td>
<td>Define type of integration</td>
</tr>
<tr>
<td>A2A security</td>
<td>Vendor name/product name or deployment technology</td>
<td>Specify how security is implemented, including integration with LDAP or other repository or product</td>
</tr>
</tbody>
</table>

Table 3.27: B2B Integration Implementation Table 2 [4]

**· Portals**

The Portal form template is described in Table 3.28 that corresponds to the questions in the Google Forms created and shown in chapter 6.

<table>
<thead>
<tr>
<th>Integration Service</th>
<th>Vendor/Product</th>
<th>Implementation Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Portal</td>
<td>Vendor/product name</td>
<td>Stand alone or part of application server or message broker platform</td>
</tr>
<tr>
<td>Back-End Application Integration</td>
<td></td>
<td>Include all application integration services included in the B2B solution</td>
</tr>
<tr>
<td>Message broker/enterprise service bus</td>
<td>Product/protocol</td>
<td>Technology platform</td>
</tr>
<tr>
<td>Application server</td>
<td>Vendor/product name</td>
<td>Technology platform</td>
</tr>
<tr>
<td>EII</td>
<td>Vendor, product name or custom name</td>
<td>Level of routing supported; content based, business rules, header only info</td>
</tr>
<tr>
<td>Application Interface</td>
<td>Vendor name/product name or deployment technology</td>
<td>Specify: Integration broker adapter; Web Service; JCA interface; data interface (ODBC; JDBC; OLE; DB; ADO)</td>
</tr>
<tr>
<td>Legacy integration</td>
<td>Vendor/product name</td>
<td>Define type of integration</td>
</tr>
</tbody>
</table>

Table 3.28: Portal Integration Implementation Table [4]

**· Mobile Integration**

The Mobile Integration form template is described in Table 3.29 that corresponds to the questions in the Google Forms created and shown in chapter 6.
<table>
<thead>
<tr>
<th><strong>Integration Service</strong></th>
<th><strong>Vendor/Product</strong></th>
<th><strong>Implementation Notes</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile integration server</td>
<td>Vendor, product name</td>
<td>Define all services provided, including process management, as well as mobile interfaces supported. Also define reliability features.</td>
</tr>
<tr>
<td>Mobile security</td>
<td>Vendor, product name</td>
<td>Define all security services provided including encryption and authentication.</td>
</tr>
<tr>
<td><strong>Back-End Application Integration</strong></td>
<td></td>
<td>Include all application integration services included in the B2B solution.</td>
</tr>
<tr>
<td>Broker/enterprise service bus</td>
<td>Product/protocol</td>
<td>Technology platform.</td>
</tr>
<tr>
<td>Translation and transformation</td>
<td>Vendor name/product name</td>
<td>Technology platform. Transformation at hub or target.</td>
</tr>
<tr>
<td>Routing</td>
<td>Vendor, product name or custom code</td>
<td>Level of routing supported; content based, business rules, header only info.</td>
</tr>
<tr>
<td>Application Interface</td>
<td>Vendor name/product name or deployment technology</td>
<td>Specify: Integration broker adapter; Web Service; JCA interface; data interface (ODBC; JDBC; OLE; DB; ADO).</td>
</tr>
<tr>
<td>Legacy Integration</td>
<td>Vendor, product name</td>
<td>Define type of integration.</td>
</tr>
<tr>
<td>A2A Security</td>
<td>Vendor name/product name or deployment technology</td>
<td>Specify how security is implemented, including integration with LDAP or other repository or product.</td>
</tr>
</tbody>
</table>

Table 3.29: Mobile Integration Implementation Table [4]

### 3.4.2.B Template of an Application Structure Implementation Specification

<table>
<thead>
<tr>
<th><strong>Integration Service</strong></th>
<th><strong>Vendor/Product</strong></th>
<th><strong>Implementation Notes</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Messaging</td>
<td>Vendor name/product name</td>
<td>Technology platform. Messaging services provided such as: publish/subscribe, guaranteed delivery, etc.</td>
</tr>
<tr>
<td>File Transfer</td>
<td>Vendor name/product name</td>
<td>Technology platform. Specify: Format Used.</td>
</tr>
<tr>
<td>Shared Database</td>
<td>Vendor name/product name</td>
<td>Technology platform.</td>
</tr>
<tr>
<td>RPI</td>
<td>Vendor name/product name</td>
<td>Specify: Technologies (CORBA, COM, .NET Remoting, and Java RMI).</td>
</tr>
</tbody>
</table>

Table 3.30: Application Structure Implementation Table
### 3.4.2.C Template of an Integrated Enterprise Implementation Specification

<table>
<thead>
<tr>
<th>Integration Service</th>
<th>Vendor/Product</th>
<th>Implementation Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Cleansing</td>
<td>Vendor name/product name</td>
<td>Done by the tool or scripts?</td>
</tr>
<tr>
<td>Metadata Repository</td>
<td>Vendor name/product name</td>
<td>DBMS technology</td>
</tr>
<tr>
<td>Content Repository</td>
<td>Vendor name/product name</td>
<td>Types of content supported including documents, images, photos, audio and video.</td>
</tr>
<tr>
<td>Translation and transformation</td>
<td>Vendor name/product name</td>
<td>Formats supported</td>
</tr>
<tr>
<td>Routing</td>
<td>Vendor name/product name</td>
<td>Level of routing supported; content based, business rules, header only info</td>
</tr>
<tr>
<td>Service BUS</td>
<td>Vendor name/product name</td>
<td>Technology platform</td>
</tr>
<tr>
<td>Application server</td>
<td>Vendor/product name</td>
<td>Technology platform</td>
</tr>
<tr>
<td>Process Management</td>
<td>Vendor/product name</td>
<td>Modules deployed, including modeling, management dashboard, etc.</td>
</tr>
<tr>
<td>Orchestration</td>
<td>Vendor/product name</td>
<td>Web Service Orchestration, workflow, process engine</td>
</tr>
<tr>
<td>Portal (Workflow, BAM…)</td>
<td>Vendor name/product name</td>
<td>Portal Services provided such as: transactions, workflow</td>
</tr>
<tr>
<td>Web Server</td>
<td>Vendor name/product name</td>
<td>Technology platform</td>
</tr>
<tr>
<td>Collaboration Platform</td>
<td>Vendor name/product name</td>
<td>Hosted within organization or by vendor?</td>
</tr>
<tr>
<td>B2B Server</td>
<td>Vendor name/product name</td>
<td>Technology platform</td>
</tr>
<tr>
<td>Collaboration Server</td>
<td>Vendor name/product name</td>
<td>Technology platform</td>
</tr>
<tr>
<td>Mobile Server</td>
<td>Vendor name/product name</td>
<td>Technology platform</td>
</tr>
<tr>
<td>Portal (Partner, Customer…)</td>
<td>Vendor name/product name</td>
<td>Technology platform</td>
</tr>
</tbody>
</table>

| **Table 3.31**: Integrated Enterprise Implementation Table [4] |

Within this thesis, the original plan was to use Atlas (Atlas is a framework developed by Link consulting that simplifies data gathering because it can collect information from many different sources, regarding multiple points in time and within different levels of detail. Atlas generates views of organization architecture based on information gathered from internal and external sources. All generated views have a time bar that shows past, present, and future (expected) architecture states). The main purpose was for when a customer fills out these forms the blueprints associated with the given pattern were generated automatically. By doing the forms in google forms, there is no approach to generate these blueprints automatically or to save this information. Therefore, these forms are representative of starting points. For starters, these forms would have to be converted to Atlas. And after, adjustments would have to be made to these forms and blueprints within a specific enterprise.
3.5 Blueprints

The following subsections are the blueprints of the three different architectural views. Such as Application Integration Technology - which covers well-known patterns such as Message Brokers; Enterprise Service Buses; Legacy Integration; Portals; Mobile Integration; B2B Integration. Application Structure - that represents the way applications can share data and processes. And Enterprise Architecture - That represents an aggregated view of all the integration patterns mentioned above. A full view of these concepts was modeled using the standard ArchiMate 2.0 architecture description language. ArchiMate is as stated in chapter 2.4, a standard modeling language and framework from The Open Group that covers the domain of Enterprise Architecture. The language includes a minimum set of concepts and relationships and the framework includes a minimum set of layers and aspects required to enable modeling of the majority of cases.

3.5.1 Application Integration

As stated in Section 2.6.1 and accordingly to the book, The essential Guide to Integrated Solutions [4], the elements of the images: 2.8, 2.9, 2.10, 2.11, 2.12, and 2.13 can be covered using ArchiMate as seen in section: 3.3. The following image 3.2 is the blueprint of these five patterns discussed.
3.5.2 Application Structure

As mentioned in the previous section 2.6.2, there are many ways to integrate multiple applications so that they work together and exchange information such as: File Transfer; Shared Database; Remote Procedure Invocation; Messaging; SOAP and REST [5]. A full view of these concepts that create an Ap-
Application Structure map, can be covered using ArchiMate, the conversion of the elements to ArchiMate was already given in Table 3.18.

**Figure 3.3:** Application Structure - ArchiMate

**Figure 3.4:** Service Structure
3.5.3 Enterprise Integration Architecture

As mentioned in previous section 2.6.3 and converted to ArchiMate in Tables: 3.19 and 3.20. A full view of these concepts that create an Enterprise Integration Architecture map, can be covered using ArchiMate.

![Enterprise Integration Architecture - ArchiMate](image)

**Figure 3.5: Enterprise Integration Architecture - ArchiMate**

Another possible view of an Enterprise Integration Architecture, simply for a matter of organization. Can be presented as the following Image:
Figure 3.6: Enterprise Integration Architecture - Full View
Evaluation

Contents

4.1 Evaluation .................................................. 56
4.1 Evaluation

In an organization with a vast number of project teams, several vendors, languages, and visions, - a unification of the vision of the organization's problems between all the distinct elements is going to be done with this solution for application integration architecture. There is no fixed practice or method for evaluation of EA implementation, however, enterprises are looking for evaluable EA implementation in order cope with their current problems and answering future needs.

Within this thesis, what we need to evaluate is: If the forms created can collect all the necessary information necessary for the attribution and fulfillment of the integration pattern; If this information collected is adequate and sufficient to generate the different blueprints; and finally, assess whether the views generated are clear to the organization.

This assessment would go through given an enterprise's data, in this case from SPMS, to collect and generate these different views. Accordingly to the need of the enterprise, the best practice to evaluate the work done in this dissertation would be to test these models with a practical scenario. According to the requirements of a specific client, we would be able to test whether the models proposed in this dissertation were useful to create a corporate integration solution for a company depending on the maturity level. The fieldwork, couldn't be done due to the current situation therefore, the most valuable part of this dissertation is unable to be tested.
5

Conclusions

Contents

5.1 Conclusion ................................................................. 58
5.2 Future Work .............................................................. 58
5.1 Conclusion

Enterprise application integration is a new solution to build enterprises' flexible information systems, where applications, data, and business processes are integrated. In this context, this thesis intends to represent the integration architecture of an enterprise and its applications. The biggest problem is, for an enterprise, how to create and represent the views. In a way, that it’s clear, for an enterprise, how they can define their integration architecture for the applications and the integration architecture for the enterprise.

To solve this, we have created forms for the assessment of the level of maturity of the enterprise and for Pattern Attribution - with the purpose of through questions, collect important information to do the attribution of the pattern to a given application. And Pattern filling - that gathers information such as descriptions and characteristics to integrate the application within the given pattern. After this for generating these different views - Blueprints are provided for the main patterns described in this thesis: Application Integration; Application Structure; and Enterprise Architecture. This solution is provided but not evaluated within a given Enterprise.

5.2 Future Work

After the solution proposed in the dissertation, the future steps are:

• Use real case study scenarios with a greater degree of complexity.

• Evaluation documentation: Since EA involves heterogeneous stakeholder groups such as application owners, business developers, software developer, system analyzer, enterprise architect, and the other may create complexity requirements in an enterprise, appropriate Evaluation documentation of the enterprise artifacts are vital.

• Evaluate if the forms created can collect all the necessary information necessary for the attribution and fulfillment of the integration pattern; If this information collected is adequate and sufficient to generate the different blueprints; and finally, assess whether the views generated are clear to the organization.
Bibliography


Attachments
## Template of an Application Integration Implementation Specification

Select the question that describes best your application:

- [ ] Translate a message from the sender's formal messaging protocol to the receiver's formal messaging protocol?
- [ ] Does the source application (producer) sends a message to a server?
- [ ] Use a centralised software component?
- [ ] Does the application adapter stay on the same platform as the applications?
- [ ] Have any existing computer system or application program that the organisation wishes to maintain?
- [ ] Reuse what already exists while adding new applications and data?
- [ ] Communicate between two or more organisations?
- [ ] Create contact points where the enterprise can manage who can access the information?
- [ ] Involve the integration into an enterprise infrastructure of mobile devices?

**Figure 6.1: Application Integration - Google Forms**
### Integration Service - Message Broker Form

Describe the following contents:

<table>
<thead>
<tr>
<th>Vendor name/product name</th>
<th>A sua resposta</th>
</tr>
</thead>
</table>

| Technology platform      | A sua resposta |

---

**Figure 6.2: Message Broker - Google Forms**

---

### Integration Service - Message Broker Form

Describe the following contents:

<table>
<thead>
<tr>
<th>Vendor name/product name if different from broker vendor</th>
<th>A sua resposta</th>
</tr>
</thead>
</table>

| Technology platform | A sua resposta |

---

**Figure 6.3: Message Broker (Messaging) - Google Forms**
Figure 6.4: Message Broker (Application Interface) - Google Forms
Figure 6.5: Message Broker (Security) - Google Forms

Figure 6.6: Enterprise Service BUS (ESB) Form - Google Forms
Figure 6.7: Enterprise Service BUS (Translation/Transformation) - Google Forms

Figure 6.8: Enterprise Service BUS (Application Interface) - Google Forms
Figure 6.9: Enterprise Service BUS Security - Google Forms

Figure 6.10: Legacy (Data Interface) - Google Forms
Figure 6.11: Legacy (Message Interface) - Google Forms
Figure 6.12: Legacy (Screen/Report Interface) - Google Forms
Figure 6.13: Legacy (Service Interface) - Google Forms

**Integration Service - Legacy Integration - Form**

**Integration Service - Service interface Form**

Describe the following contents:

**Vendor /product name or custom code:**

A sua resposta

**Order processing service, customer maintenance service, order, tracking service:**

A sua resposta

Figure 6.14: B2B Connectivity- Google Forms

**Integration Service - B2B Integration - Form**

**B2B Connectivity:**

**Vendor/product name:**

A sua resposta

**B2B server or central exchange:**

A sua resposta
### Integration Service - B2B Integration - Form

**Partner connectivity:**

**Vendor/product name:**

A sua resposta

**List all connectivity options including XML messaging, EDI, FTP, browser interfaces, exchange hub, or other:**

A sua resposta

---

**Integration Service - B2B Integration - Form**

**Partner management:**

**Vendor /product name:**

A sua resposta

**Define services provided including process management, collaboration services, service level agreements, or others:**

A sua resposta

---

**Figure 6.15:** B2B Partner Connectivity- Google Forms

**Figure 6.16:** B2B Partner Management- Google Forms
Figure 6.17: B2B Security - Google Forms

Figure 6.18: B2B Broker/ESB - Google Forms
### Figure 6.19: B2B Translation - Google Forms

**Translation and transformation:**

- **Vendor name/product name:**
  - A sua resposta

- **Transformation at hub or target:**
  - A sua resposta

### Figure 6.20: B2B Routing - Google Forms

**Routing:**

- **Vendor name/product name:**
  - A sua resposta

- **Level of routing supported; content based, business rules, header only info:**
  - A sua resposta
Integration Service - B2B Integration - Form

Application Interface:

Vendor name/product name or deployment technology:
A sua resposta

Specify: Integration broker adapter; Web Service; JCA interface; data interface (ODBC; JDBC; OLE; DB; ADO):
A sua resposta

Figure 6.21: B2B Application Interface - Google Forms

Integration Service - B2B Integration - Form

Legacy Integration:

Vendor/product name:
A sua resposta

Define the type of integration:
A sua resposta

Figure 6.22: B2B Legacy Integration - Google Forms
Integration Service - B2B Integration - Form

A2A Security:

Vendor name/product name or deployment technology:

A sua resposta

Specify how security is implemented, including integration with LDAP or other repository or product:

A sua resposta

Figure 6.23: B2B A2A Security - Google Forms

Integration Service - Portal Integration - Form

Portal:

Vendor/product name:

A sua resposta

Stand alone or part of application server or message broker platform:

A sua resposta

Figure 6.24: Portal - Google Forms
**Figure 6.25:** Portal (Broker(ESB)) - Google Forms

**Figure 6.26:** Portal (Application Server) - Google Forms
Figure 6.27: Portal (EII) - Google Forms

Figure 6.28: Portal (Application Interface) - Google Forms
Figure 6.29: Portal (Legacy Integration) - Google Forms

Figure 6.30: Mobile Integration - Google Forms
Integration Service - Mobile Integration - Form

Mobile Security:

Vendor, product name:

A sua resposta

Define all security services provided including encryption and authentication:

A sua resposta

Figure 6.31: Mobile (Security) - Google Forms

Integration Service - Mobile Integration - Form

Broker/Enterprise service bus:

Product/protocol:

A sua resposta

Technology platform:

A sua resposta

Figure 6.32: Mobile (Broker/ESB) - Google Forms

79
Integration Service - Mobile Integration - Form

**Translation and transformation:**

Vendor name/product name:

A sua resposta

Transformation at hub or target:

A sua resposta

Figure 6.33: Mobile (Translation) - Google Forms

Integration Service - Mobile Integration - Form

**Routing**

Vendor, product name or custom code:

A sua resposta

Level of routing supported; content based, business rules, header only info:

A sua resposta

Figure 6.34: Mobile (Routing) - Google Forms
Figure 6.35: Mobile (Application Interface) - Google Forms

Figure 6.36: Mobile (Legacy Integration) - Google Forms
Integration Service - Mobile Integration - Form

A2A Security

Vendor name/product name or deployment technology:
A sua resposta

Specify how security is implemented, including integration with LDAP or other repository or product:
A sua resposta

Figure 6.37: Mobile (A2A Security) - Google Forms

Template of an Application Structure Implementation Specification

Select the question that describes best your application:

- Does your application produce files of shared data for others to consume?
- Does your application consume files that others have produced?
- Does your application store the data in a common database?
- Does your application expose some of its procedures so that they can be invoked remotely?
- Does your application invoke procedures to run behavior and exchange data?
- Does your application connect to a common messaging system?
- Does your application exchange data and invoke behavior using messages?

Figure 6.38: Application Structure - Google Forms

82
Figure 6.39: Application Structure (Messaging) - Google Forms

Figure 6.40: Application Structure (File Transfer) - Google Forms
Figure 6.41: Application Structure (Shared Database) - Google Forms

Figure 6.42: Application Structure (RPI) - Google Forms
Template of an Integrated Enterprise Implementation Specification

**Data Cleansing**

Vendor name/product name

A sua resposta

Done by?

- [ ] Tools
- [ ] Scripts

[Anterior] [Seguinte]

*Figure 6.43: Integrated Enterprise (Data Cleansing)- Google Forms*

Template of an Integrated Enterprise Implementation Specification

**Metadata Repository**

Vendor name/product name

A sua resposta

DBMS Technology

A sua resposta

[Anterior] [Seguinte]

*Figure 6.44: Integrated Enterprise (Metadata Repository)- Google Forms*
Figure 6.45: Integrated Enterprise (Content Repository)- Google Forms

Figure 6.46: Integrated Enterprise (Translation/Transformation)- Google Forms
### Template of an Integrated Enterprise Implementation Specification

**Routing**

**Vendor name/product name**

A sua resposta

**Level of routing supported: content-based, business rules, header-only info**

A sua resposta

---

**Figure 6.47: Integrated Enterprise (Routing)· Google Forms**

---

**Template of an Integrated Enterprise Implementation Specification**

**Service BUS**

**Vendor name/product name**

A sua resposta

**Technology platform**

A sua resposta

---

**Figure 6.48: Integrated Enterprise (Service BUS)· Google Forms**

87
### Template of an Integrated Enterprise Implementation Specification

**Application server**

Vendor name/product name

A sua resposta

**Technology platform**

A sua resposta

---

**Figure 6.49:** Integrated Enterprise (Application Server) - Google Forms

---

**Template of an Integrated Enterprise Implementation Specification**

**Process Management**

Vendor name/product name

A sua resposta

Modules deployed, including modeling, management dashboard, etc.

A sua resposta

---

**Figure 6.50:** Integrated Enterprise (Process Management) - Google Forms

---

88
### Template of an Integrated Enterprise Implementation Specification

#### Orchestration

**Vendor name/product name**

A sua resposta

**Web Service Orchestration, workflow, the process engine**

A sua resposta

#### Figure 6.51: Integrated Enterprise (Orchestration)- Google Forms

#### Portal (Workflow, BAM...)

**Vendor name/product name**

A sua resposta

**Portal Services provided such as transactions, workflow**

A sua resposta

#### Figure 6.52: Integrated Enterprise (Portal)- Google Forms
## Template of an Integrated Enterprise Implementation Specification

### Web Server

**Vendor name/product name**

A sua resposta

### Technology platform

A sua resposta

---

**Figure 6.53:** Integrated Enterprise (Web Server)- Google Forms

### Template of an Integrated Enterprise Implementation Specification

### Collaboration Platform

**Vendor name/product name**

A sua resposta

**Hosted within?**

- [ ] Organisation
- [ ] By vendor

---

**Figure 6.54:** Integrated Enterprise (Collaboration Platform)- Google Forms
Template of an Integrated Enterprise Implementation Specification

**B2B Server**

Vendor name/product name

A sua resposta

Technology platform

A sua resposta

**Figure 6.55:** Integrated Enterprise (B2B Server)- Google Forms

Template of an Integrated Enterprise Implementation Specification

**Mobile Server**

Vendor name/product name

A sua resposta

Technology platform

A sua resposta

**Figure 6.56:** Integrated Enterprise (Mobile Server)- Google Forms
**Template of an Integrated Enterprise Implementation Specification**

**Portal (Partner, Customer...)**

<table>
<thead>
<tr>
<th>Vendor name/product name</th>
</tr>
</thead>
<tbody>
<tr>
<td>A sua resposta</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Technology platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>A sua resposta</td>
</tr>
</tbody>
</table>

*Figure 6.57: Integrated Enterprise (Portal)- Google Forms*