REFERENCE ARCHITECTURES AS MEANS TO AID IN SYSTEM DEVELOPMENT

Maurice Zeito

Degree project in
ICS
Master thesis
Stockholm, Sweden 2013

XR-EE-ICS 2013:002
REFERENCE ARCHITECTURES AS MEANS TO AID IN SYSTEM DEVELOPMENT

Maurice Zeito

A Master Thesis Report written in collaboration with
Department of Industrial Information and Control Systems
Royal Institute of Technology
Stockholm, Sweden

April, 2013
Abstract. In comparison with the ever increasing development of enterprise architectures and frameworks, reference architectures has yet to reach the same maturity level. This brings questions regarding definition, implementation and governance of reference architectures at different levels; Business, System, Technical. In this report, an in-depth analysis of the concept of reference architectures is made, focusing on the abovementioned attributes. The purpose of the report is to provide the reader with an insight on how reference architectures can be implemented in order to aid in the work of system development.

Keywords. Enterprise Architecture, Reference Architecture, Reference Applications, Patterns, Enterprise Architecture Framework, Information System, Design Pattern, Software Architecture
Acknowledgement

I want to thank my family for supporting me and believing in me throughout the whole project. I want to especially thank my sister for listening to my presentation and giving me the constructive criticism I needed to hear.

I also want to thank my supervisors for giving me feedback and giving me that extra push needed to complete the report.

“If you’re not prepared to be wrong, you’ll never come up with anything original”

– Sir Ken Robinson
# Table of Contents

Abstract. 2  
Keywords. 2  
Table of Contents 4  
1 Introduction 4  
1.1 Background 4  
1.2 Scania 5  
1.3 Purpose 5  
1.4 Scope and Delimitations 5  
1.5 Abbreviations 6  
1.6 Outline 6  
1.7 Guidelines on how to read the thesis 7  
2 Preface 8  
2.1 The Authors View on the Subject 8  
2.2 The Definitions 8  
2.3 The Framework 9  
3 Related Work 11  
3.1 Reference Architecture at Scania 11  
3.2 Reference Architecture in Avionic Software Development 15  
3.3 The Holistic View on Enterprise Architecture 16  
3.4 The Zachman Framework 17  
3.5 DoDAF 18  
3.6 TOGAF 19  
3.7 Archimate 20  
3.8 Conclusion 21  
4 Method 22  
4.1 General Methodology 22  
4.2 Case-adapted Method 22  
4.2.1 Project Establishment 23  
4.2.2 Theory 23  
4.2.3 Data Collection (Interviews & Survey) 23  
4.2.4 Analysis of Collected Empirical Data 24  
5 Theory on Reference Architecture 25  
5.1 Reference Architecture and Patterns 25  
5.1.1 A DoD Perspective on Reference Architectures 30  
5.2 Software Architecture 30  
5.3 Architectural Patterns 31  
5.4 Design Patterns 32  
5.5 Reusability 34  
6 Hypothesis 36  
7 Empirical Data 38
1 Introduction

The introductory chapter to the master thesis will provide an insight on the background of enterprise architecture as well as giving the reader the opportunity to see the relation between information technology (IT) and the building blocks that give meaning to IT. This is then followed by the purpose, scope and delimitations of the master thesis.

1.1 Background

With the ever increasing development of organizations technological advancement in today's modern society, regarding complexity and scope, numerous questions considering implementation and management of information technology emerge. This development is a subject of scope of design and complexity that IT systems bring forward when implemented. Because of this, the main challenges are set into managing and communicating the IT organizations objectives and goals throughout the whole organization[2-5]. Furthermore, since an organizations information system (IS) is under constant change, with new technology emerging and being introduced, there is a sought after need for handling this change[6]. In order to comply with these objectives and goals and implicitly derive results throughout the organization, different methods have been proposed. These methods take form in enterprise architecture (EA).

The development of frameworks in relation to EA brings forward the discussions on reference architecture as an own concept. In order to manage and maintain the reference architectures life-cycle and its surrounding applications, frameworks developed by the industry provide the means to maintain the reference architectures sustainability and increasing maturity.

Questions regarding abstraction level of design patterns emerge when learning from previous experiences in the field of architecture. Since reference architecture as a concept hasn’t reached maturity, it is still a field that needs to be explored in order to meet certain requirements demanded from the IT community. The question relies in defining and providing means of deriving different levels on a reference architecture, depending on the solutions and processes that are underlying the objective of constructing or utilizing a reference architecture.

It is known that numerous definitions are provided as a result of the increasing development of IT. Although this brings means of settling for one definition in perspective of ones own organization, it also brings confusion and reluctance to embrace new definitions emerging. The definition on reference architecture has reached a point in which it is bound by other definitions within the same area of research. According to[7], this may be the case in which the term “reference architecture” is bound by the term “architecture” itself, broadening the definition to be applicable to almost anything.

Reference architectures provide the means of communicating proven experience in previous projects. By incorporating pattern styles and working in accordance with different methodologies and frameworks such as the Zachman framework, TOGAF, DoDAF etc., an implementation of reference architecture is imminent. This relies in the fact that, although the term “reference architecture” is not widely used, the concept exists in ways of how different project roles work. It is strongly coupled to reusability, requirements engineering and tool support in the ordinary day work with EA and all the IT methodologies existing today, old and new. Finding the reference architecture of an organization due to the varieties of definition being used is not easy. However, finding the relationship between different levels (Business, System and Technical architecture) is one step into realizing what kind of reference architecture is being implemented or developed.
1.2 Scania

Today, Scania is a leading manufacturer of heavy trucks, busses and industry engines. Furthermore, the company provides services and is a retailer of service based products and financial services. Scania is operative in more than 100 countries in Europe, Latin America, Asia, Africa and Australia and have more than 35 000 employees.

The vision of Scania is to contribute to a sustainable economic growth of the company, customers and to the society. This includes providing energy efficient products and solutions to customers in order to increase efficiency and sustainability.

The growth and development of Scania in all of its visions, goals, business solutions and technical executions requires facing the challenges that are incorporated in today’s IT-development. In order to achieve and persevere this development, different methodologies and know-how experience within the field of IT need to be taken in and adapted for the work within Scania.

1.3 Purpose

The purpose of this thesis is to provide Scania as well as other companies, an insight on how reference architecture can be implemented in order to aid in the work of system development. This implies the understanding of requirements, roles, communication and development of reference architectures. It also gives the Royal Institute of Technology (RIOT) the possibility to review what concerns and questions that are relevant regarding reference architecture. Due to the circumstance that reference architecture is a very abstract field or research, the question that needed to be answered for the thesis was also an abstract one. The challenge was to maintain a certain level of abstraction and still provide a depth within the topic for the reader. The thesis was written in order to assess from other companies the answers to the following questions:

1. What constitutes a reference architecture?
2. How is the reference architecture maintained (framework support)?
3. How are design patterns implemented and packaged for reusability in the future?
4. What type of patterns were developed and/or reused to accommodate the reference architecture in the different organizations?
5. Why are reference architectures implemented to begin with?

The answers provided will help in the analysis process where a comparison will be made and a conclusion on the results can be presented. The answers will also provide the means of formulating different hypotheses that will be tested towards respective organizations that the interviews were conducted with.

1.4 Scope and Delimitations

The scope of the thesis is constrained to only take account for 5 number of organizations during the data collection phase. With that restraint, the empirical data collected might be of qualitative value rather than a quantitative one. The thesis is written in order to shed more light into the internal workings and decisions when deriving a reference architecture. Since the area on reference architecture is vast, the time needed to research on every part is not available. Therefore the main ideas and concepts of reference architectures and correlated subjects are studied and presented in a theory section. The empirical data chapter however, provides the reader with a more meaningful and in depth research on how other companies manage their reference architectures and frameworks. With this in mind, the thesis should be read as related works in the area of reference architectures.
1.5 Abbreviations

**ADM**  Architecture Development Method  
**CIO**  Chief Information Officer  
**DoDAF**  Department of Defense Architecture Framework  
**DoD**  Department of Defense  
**DPE**  Developer Platform Evangelism  
**DTO**  Data Transfer Object  
**EA**  Enterprise Architecture  
**EAM**  Enterprise Architecture Management  
**GERAM**  Generalised Enterprise Reference Architecture and Methodology  
**IS**  Information System  
**IT**  Information Technology  
**ICS**  Industrial Control Systems  
**KTHB**  Kungliga Tekniska Högskolans Bibliotek  
**RIOT**  Royal Institute of Technology  
**SOA**  Service Oriented Architecture  
**TAFIM**  Technical Architecture Framework for Information Management  
**TOGAF**  The Open Group Architecture Framework  
**UML**  Unified Modeling Language  

1.6 Outline

Chapter 1 is the introductory chapter which contains background, purpose and scope of the thesis.  
Chapter 2 provides an insight on the authors viewpoint of different definitions stated in the thesis.  
Chapter 3 discusses the main concepts of architectural frameworks.  
Chapter 4 consists of the methodology of how the thesis was conducted.  
Chapter 5 is theory related to reference architectures in general.  
Chapter 6 consists of hypotheses derived from interviews.  
Chapter 7 details the empirical data.  
Chapter 8 is where conclusions are drawn.  
Chapter 9 brings a discussion on the conclusions.  
Chapter 10 discusses the validity and reliability of the conducted thesis.  
Chapter 11 is about future work within the same area of research.
1.7 Guidelines on how to read the thesis

Chapter 2 is written in order to create a basis for understanding, from a reader's perspective, for the analysis phase. Chapter 3 provides a background on where architectural designs and decisions hold their basis. Furthermore, it offers the possibility to see the relation between the holistic view of EA and how frameworks are developed to be specific for an enterprise. Chapter 5 provides the reader with a theory on the background of pattern design and how it is related to reference architectural views. The theory on reference architecture gives the reader a general understanding on where the questions were derived from and their importance to the data collection phase. Chapter 6 relates the data collection phase to the analysis phase and contributes to the validity and reliability chapters further on in chapter 10. For the reader, if only interested in hypothesis derived and conclusions of the analysis phase, see chapters 6 – 9.
2 Preface

This chapter presents a preluding discussion and a presentation on the bigger picture of reference architectures, frameworks, architecture and other topics discussed in this paper. It is strictly written to provide a common ground and understanding, from the authors view, on the correlation between the different definitions that are discussed further on in the thesis. This is done to let the reader know what the author means when talking about reference architectures and framework. Drawing analogies to the periodic system, it serves as a foundation to understanding what is written in the thesis. In the same way, the periodic system provides a common ground on how to organize and name the different elements. This does however not mean that every company need to convey to the authors definitions, it merely provides a methodology on assessing the information gathered and executing the comparative analysis.

2.1 The Authors View on the Subject

Several definitions have been formulated and worked out to describe different ideas and concepts, new and outdated. On the topic of frameworks such as the Zachman framework, DoDAF, TOGAF etc. there has been numerous additions and alterations to these frameworks that are implemented to an enterprise. Some enterprises follow the definitions of EA, frameworks and reference architecture strictly and some less strict. In order to write the thesis, it seemed like a good idea to settle down for one definition for each topic. This served useful during the analysis phase of the thesis and for the discussion of the analysis.

Since the thesis was written to provide the reader(s) with an insight on how other companies define and implement reference architectures, EA and maturity frameworks, it felt natural to choose a maturity framework that is well known and widely used in the industry. Therefore, the Zachman framework was chosen as a generic framework to accommodate the analysis. Furthermore, since it was planned that the Zachman framework would be used as basis for a comparative analysis, it would be easy to follow the correlation between the companies regarding their scope in the understanding of reference architectures and in the context of EA.

2.2 The Definitions

The thesis was based on several definitions that add up to the meaning of reference architecture. In order for the reader to make an understanding of what is being written in the following chapters, definitions of the main concepts are listed below. This is to provide a common understanding between the reader and the author due to the numerous definitions presented in the IT-community. Each definition will be presented in a numbered order, facilitating and providing an overview for the reader to follow.

1. Reference Architecture

“…is, in essence, a predefined architectural pattern, or set of patterns, possibly partially or completely instantiated, designed, and proven for use in particular business and technical contexts, together with supporting artifacts to enable their use. Often, these artifacts are harvested from previous projects.” – Rational Unified Process, RUP[8]

2. Architectural style

“… a family of systems in terms of a pattern of structural organization. More specifically, an architectural style determines the vocabulary of components and connectors that can be used in instances of that style, together with a set of constraints on how they can be combined. These can include topological constraints on architectural descriptions(e.g., no cycles). Other constraints – say, having to do with execution semantics – might also be part of the style definition.” – Some Patterns for Software Architectures, Mary Shaw[9]
3. Architecture

“Architecture is the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principle guiding its design and evolution.” – IEEE Standard 1471-2000[10, 11]

4. View


5. Viewpoint

“A specification of the conventions for constructing and using a view; a pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis” – IEEE Standard 1471-2000[10, 11]

2.3 The Framework

In order to make a comparative analysis and developing a discussion around the subject of reference architectures, a reference framework is needed for aligning the companies towards a common denominator, the use of a reference architecture. The Zachman framework provides a method of placing the organizations in the study at different levels of the framework. From this, reasonable conclusions can be drawn on where the concept of reference architecture can be seen and how it can expand to be incorporated at different levels. Drawing an analogy to the periodic table of elements, the same idea is being thought of, when structuring and categorizing in according to the matrix elements present in the Zachman framework. The Zachman framework was choosen because of its simplicity and matrix descriptions. It is easy to follow and can include almost any organizational structure.

Figure 1 An example of mapping organizations towards the Zachman framework
Figure 1 shows an example of how different organizations are placed in accordance with the information given from them on where they feel reference architecture might be existent in their organization. The framework above, used in the thesis, is a modified version of the Zachman framework in which not all levels are considered. This has to do with the scope of the project where it would be too extensive to consider all the levels in the Zachman framework.

From an empirical data collection phase, information from surveys and interviews conducted bring means of envisioning and creating a holistic view on the placement of reference architectures for each organization. From this, it is open to draw conclusions on communication needs between the different levels between the different reference architectures. Moreover, key factors such as economical benefits and efficiency in communication might be addressed. It also brings questions on how big a reference architecture might be and its restrictions to be included at higher and lower levels in relation to the Zachman framework. The framework also provides guidance in asking questions regarding time and roles and the resources needed to handle these in perspective of a reference architecture.
3 Related Work

In the following subchapters, the related work within the area of enterprise architectural frameworks is presented. A preluding chapter describes the background of enterprise architecture as a holistic view. Further on, introductory chapters on different general frameworks will be presented. The proposal of this structure is for the reader to get a general perspective for the concept of enterprise architecture (EA) and the frameworks that support the implementation of EA.

3.1 Reference Architecture at Scania

Reference architecture, according to Scania, is “a combined description of chosen architectural- , design- and code patterns to realize some examples of functional and non-functional requirements. The ambition of the description is that chosen solutions can be reused in their entirety or in parts.” This definition provided by Scania does in the same way convey to the main principle of reference architectures, stating that patterns are utilized in order to describe architectural realizations and that these patterns are being reused from previous projects or realizations of a solution. Although many similarities exist between the definition given by Scania and the one stated in RUP[8], it shows that the definition of reference architecture is broad but still manages to convey the true meaning of the concept.

The main idea of implementing reference architecture at Scania is to:

- Create a viewpoint of the link from requirements to application development.
- Managing synergy between patterns and technology.
- Create examples where technology and methods are implemented.

This is done to achieve:

- Faster system development
- Facilitate maintenance
- Facilitate operations
- Separate functional and non-functional code
- Facilitate the understanding of how the architecture is to be implemented
The model shown in Figure 2 is based on a software architectural methodology and portrays a layered architecture. The main idea of this reference application, is to separate functional and non-functional logic. The non-functional logic is shown gathered to the right in the cross-cutting layer in Figure 2.

The model is based on different key principles that, according to the application architecture guide, promotes minimization of costs and maintenance requirements[12]. These principles are set to facilitate the creation of a coherent viewpoint, allowing a separation of concerns in a way that minimizes the number of interactions between different areas. This results in low coupling, but raises the question on interoperability between the different areas. Although this is another topic of research, it raises interest for future works in the area of reference architectures and interoperability.

**Figure 2 Candidate for reference architecture**

One key design principle adopted by software architects utilizing the candidate depicted above, is to minimize complexity by separation of concern areas. The candidate above shows a layered architecture in which means of achieving low complexity is provided. For software architects, this means separation of code from respective area[12].
The demands on the architecture are:

1. Several different presentation layers should be present
2. Data sources should be encapsulated so that reusability is possible
3. Functionality should be implemented in standard components
4. Separation of functional and non-functional logic
5. The need for consequence alterations should be minimized
Figure 3 Organizational aspects of reference architectures
Figure 3, shows the framework for architectural descriptions at Scania. It was derived by Gartner, an information technology research and advisory company. The model is shown here due to the ability it has of portraying the crosscutting relationship of reference architectures and patterns throughout the organizational levels shown above.

Figure 3 provides a structure for enterprise architectural descriptions by dividing the organization at different levels; Business, Information, Technical, System and Operation. Although the figure above does not provide a direct instantiation of a reference architecture, it shows on the important aspects relating to those. These aspects concern communication and pattern relationship through the different levels. As an example, the reference application shown in Figure 2 proves to be a perfect candidate at the level of system architecture in Figure 3. This is due to the fact that the reference application is an abstract architectural description that can be used to design one or more of the applications shown in the figure above.

Furthermore, the architectural structure above shows on the aspects of notation such as the language of UML and the road to implementation. This notation is the combination of syntax and semantics of the different architectural artifacts. In the example of Figure 3, this is depicted by the red boxes related to each other by black doublesided arrows.

3.2 Reference Architecture in Avionic Software Development

A case-study made by [13], show how a reference architecture is created in order to aid in development of avionics software. The reference architecture named ADAGE, is built upon a software factory paradigm as shown in Figure 4. The idea of the paradigm is to benefit from already existing and proven concepts of avionics software products. This is further discussed in chapter 5 “Reference Architecture” in the thesis.

The ADAGE architecture provided the means to, through a layered architectural style, improve productivity and quality for avionics software development[13]. From this software paradigm, it shows how the reference architecture is strongly correlated to the reuse of design. This will be further discussed in chapter 5.6 on reusability.
3.3 The Holistic View on Enterprise Architecture

In order to present the field of enterprise architecture and where it holds its basis, a plausible definition of architecture might be appropriate to be presented. According to IEEE-Std-1471-2000 (Recommended Practice for Architectural Description of Software-Intensive Systems), architecture is defined as “the fundamental organization of system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution” [10, 14]. The architecture of a system serves as basis for further improvements of the systems; expressed in analysis, design, optimization, validation, implementation and construction of the system [6, 14-16].

Viewing EA as a tool, it is an approach in defining and encapsulating the core values that constitutes business and IT and the junction between these, using principles and paradigms that are of necessity when describing the junctions. Furthermore, “what is part of the enterprise architecture, and what is only an implementation within that architecture, is a matter of what that business defines to be the architecture, and what not” [15]. In this sense, EA is viewed and implemented in accordance to what is believed to be the goals and objectives of the enterprise organization. With this in mind, EA is beneficial in the sense that it facilitates the scope and complexity of a system, providing technical oversight and improves decision making throughout the enterprise organization [16, 17].

The idea of EA is to cope with the scope and complexity of today’s increasing information systems, bringing means to provide a logical concept or architecture for defining and describing integration of systems and components within a system. It is not confined within the boundaries of technical concerns, rather reaches throughout the organization in form of roles, business processes and consumer values to mention a few [2, 15]. This implies the fact that defining architecture for an enterprise results in efficiencies concerning communication, innovation, stability and flexibility.

From this holistic view on EA, further insight is provided from well-established maturity frameworks that have been developed over the years. “Frameworks structure architecture description techniques by identifying and relating different architectural viewpoints and the modeling techniques associated with them” [15]. Correlating this to reference architecture, the need for management and governance is of great importance. Developed frameworks are crucial in their way of maintaining reference architectures and secure their functionality. This is shown in Figure 5 in accordance with [18].

![Figure 5 Reference architecture and framework relation][18]
In the following subchapters, a brief introduction to three frameworks is presented. This is to provide the reader an overview of the most common frameworks and maturity models that are used in the industry. A general introduction to the Archimate language supporting one of the frameworks is also presented. The main reason for presenting these frameworks is to show what levels are present considering scope and complexity of architectural frameworks, and what methodology is presented within each framework.

### 3.4 The Zachman Framework

In 1987 John A Zachman presented, in his article “A Framework for Information Systems Architecture”, a classification schema presenting descriptive representations for information systems[6, 19]. The Zachman framework is a logical and apprehensible approach into classifying and managing descriptive representations in an enterprise.

The framework is built upon a 6x5 matrix, presenting in a simple way, the intersection between roles and communication interrogatives. The intersections are represented in Figure 6, also showing at what level of abstraction the intersection are made.

![Figure 6 The Zachman framework][15]

The Zachman framework is easily understood, because of its way “…it addresses the enterprise as a whole, it is defined independently of tools or methodologies, and any issue can be mapped against it to understand where they fit”[15]. This means that it is easy to see roles and the actions coming from the intersections, promoting traceability. Although this framework is simple and concrete, it is this simplicity that gives it drawbacks. The framework does not specify the relation of the different cells, making it difficult to implement in work.

According to Zachman, the framework draws analogies and has similar attributes to classical architecture. The matrix shown above is a representation of the deliverables in classical architecture, specifying how a work product is developed. Because of this abstract formalization of the framework, much like the holistic view on EA it also provides several views to be looked at by different roles in a company. It needs to be pointed out that, these views are not set into stone. A deliverable can be described in different ways and be represented by different roles and have different descriptions. This creates numerous ways in which views are presented.

---

[15]: #
3.5 DoDAF

The Department of Defense Architecture Framework (DoDAF) is a widely known framework which was developed upon the Technical Architecture Framework for Information Management (TAFIM) by the U.S. Department of Defense[6, 20]. The framework provides different views in which definitions are described on how to organize the enterprise architecture specifications. The framework provides the means to help the Chief Information Officer make key decisions. It is also developed so that components in the DoD are strictly conformed to the DoDAF and maintained accordingly. In relation to reference architectures, the “Conformance ensures that reuse of information, architecture artifacts, models, and viewpoints can be shared with common understanding”[21]. As depicted in Figure 7, the framework provides three main views; an operational view, systems view and technical standards view.

![Figure 7 The Department of Defense Architecture Framework (DoDAF)[22]](image)

The main idea of the DoDAF is, in accordance with version 2.0, to establish a guidance for architecture components to meet the purpose of a function. From this, the concept “Fit-for-Purpose” stands as a core approach maintained throughout the frameworks key principles[22]. The framework has been undergoing development, and since version 2.0, the DoDAF has gone from a “product-centric to data-centric focus”, data architecture allowing better analytical results for decision makers[22]. The ‘Fit-for-Purpose’ architectures is in alignment with the data-centric focus and facilitates the collection of data and creation of views, aiding the decision-makers by fulfilling requirements and needs.

A view on the definition on reference architecture from a DoD perspective will be presented in chapter 5.1.1.
3.6 TOGAF

When considering the DoDAF and the Zachman framework, it is natural to extend the study to incorporate The Open Group Architectural Framework, TOGAF. Based on the TAFIM and passed on by the DoD to The Open Group[6, 23], the framework has, like the Zachman and DoDAF reached a certain maturity level due to the wide use of it. The TOGAF started out being a generic framework applicable to technical and development architectures, but evolved further into considering EA frameworks[15]. It is considered to be a high level, holistic approach into design of concepts and methodologies. TOGAF has been revised numerous times and the latest version of the TOGAF is version 9.1 “Enterprise Edition”. Moreover, since version 8 the framework is dedicated to enterprise architectures. In comparison to the Zachman framework, which provides views and no particular guidance, the TOGAF stands out in its way of communicating a methodology and guidance in its architecture as depicted in Figure 8.

The methodology of the TOGAF is based on the architecture development method (ADM) and constitutes of nine steps in an iteration process. Every iteration is followed by new decisions that are based on competence, resource availability and value to the enterprise[23]. The basic principles of the ADM is that “Every phase is validated against and validates the current requirements of the business”[23]. TOGAF is organized into four domain levels; Business, Application, Data and Technology architecture. These levels are, as shown in Figure 8, related through the methodology that TOGAF communicates. In the preliminary step, scoping and organizational issues are set up, followed by the vision and purpose in the next step. Moving further on in the TOGAF, the current and future decisions are set up for the business architecture, information systems architecture and the technological architecture. After modeling these steps, the consideration of migration, governance and change management is needed in order to execute future architectures.[6]
3.7 Archimate

Having discussed the different frameworks that can be used to support EA, it is appropriate to bring forward a discussion on a modeling language that supports a framework. One modeling language that has reached a certain maturity is Archimate. The language “…has been transferred to The Open Group, where it is slated to become the standard for architectural description accompanying the Open Group’s architecture framework TOGAF”[11].

An information system is dependent on different architecture descriptions that in a way support and organize structural and behavioral properties of a system[24]. These descriptions provide means to manage the development and needs of an enterprise and its layers. The Archimate language provides a concrete presentation for the support of these architectural descriptions. According to [24], Archimate is:

1. A lightweight language providing support for the structure and architecture framework in domains, layers and aspects.
2. Created to be incorporated with the architecture framework of TOGAF, developed by The Open Group.
3. Supported by numerous tools that are commercially available.
4. Although similar to the Unified Modeling Language (UML), an approach to focusing on semantics rather than just notion, i.e. syntax. [15, 24]

With this being said, it should be clear that Archimate is merely a language and does not promote a way of working. Furthermore, the Archimate language is more than a set of lines and shapes, it also incorporates a meta model, explains entities and conveys a meaning with the architecture.[25] This is shown in the way Archimate associates the three layers; Business, Application and Technology. The relationship adheres to the same three layers presented in TOGAF, facilitating the mapping of Archimate to TOGAF. The mapping is depicted in Figure 9.

![Figure 9 Mapping Archimate to TOGAF](image)
3.8 Conclusion

There are many frameworks existing today that convey the essence of EA and the way of implementing the idea in an enterprise. From the holistic view on EA, frameworks have been developed to provide further guidance into supporting the implementation of information systems. The common denominator of these frameworks, is that they all aim to provide guidance in how to document and manage EA in an efficient and suitable way. Nevertheless, they all have to cope with some concerns that arise[26].

Following concerns are addressed:

- EA management is introduced from scratch
- Too abstract, expensive or difficult to implement
- EA management fully embraced with no starting point and management initiative to begin with
- Poor documentation leading to not knowing why something has been done.
- Frameworks and EA approaches are often delivered as packages and don’t support development of these.

Since this report strikes its focus on the key principles of reference architectures, it is naturally to incorporate the understanding of patterns. One “solution” to these concerns stated above, is to apply Enterprise Architecture Management Patterns (EAM)[26]. Wandering from the relation between software architectures and reference architectures, EAMs provide the means to address the alignment of business and IT. The subject of EAM will be discussed brief in chapter 5 to acknowledge the fact that reference architectures also can be applied to other critical areas within an enterprise. However, with these frameworks being presented, it is natural to point out that the frameworks are tools in managing systems. Independent of scope and complexity, they all provide the baseline for improvement and development.
4 Method

In this chapter, the method of the proceeding investigation will be presented. The first subchapter describes the general method of how to carry out a case-study. This is later on followed by a case-adapted methodology that is constrained within the scope of the thesis. This section of the thesis is to provide the reader the general idea on how the research study was carried out for the thesis.

4.1 General Methodology

The method used for the thesis regarding the assessment of an analysis and the realization of results from the analysis, is based on a general methodology as described by [27]. Case-studies provide a systematic way of analyzing information and present results. The reason for employing a case-study research is for the researcher to acquire an understanding on how and why something happened. It is also a way to present rich information on occurring events and the roles within the study of research. Although a holistic view is provided, case study results are often tied to specific situations where generalization is not applicable. [27, 28]

The idea of a case-study methodology used for the thesis is based on the aspects of using a step-wise approach. This will be based on a generalized method described by Figure 10.

![Figure 10 General approach](image)

4.2 Case-adapted Method

The figure depicted in Figure 10 shows a narrow description of the progression of the investigation that is underlying the results of the report. In order to have a traceable methodology, further steps need to be accounted for. Regarding the thesis as a case-study, the general method has been improved according to Figure 11 below.

![Figure 11 Case-adapted method](image)
This way of executing the thesis is done in accordance to how it was proposed by the department of ICS at RIOT[29]. As well as introducing additional phases to the method, the case-adapted method utilize a “feedback loop” designed specifically to aid in the process of the three phases; Theory, Data collection and Analysis. This “feedback loop” and the different phases of the model in Figure 11 are described below.

4.2.1. Project Establishment

In order to start the thesis, a meeting with RIOT and Scania was planned. During the meeting, the scope of the thesis was discussed as well as the focus of the research. The project establishment also included setting goals and objectives for the thesis and a time plan in accordance to the thesis timeframe set by RIOT and Scania. This phase was one of the most crucial parts of the project because it served as a basis for the continuation of the thesis and the commencement of the succeeding phases.

4.2.2. Theory

During the theory phase, an extensive study regarding the focus of the research was made. This included reading literature on enterprise architecture, pattern design, reference architecture, system architecture, information systems and reference frameworks. The literature was mainly peer-reviewed reports written by researchers in the area of EA and IT. Furthermore, the reports were found using the databases provided by the library at RIOT, KTHB. The main databases used for the thesis were (In numbered order, nr 1 being the main database used):

1. Inspec
2. Scopus

Others documents concerning EA and reference architecture were provided by Scania. The information gathered during the theory phase was mainly used as a basis for chapter 5, Reference Architecture. It is also shown in Figure 11 how the theory phase contributed to the data collection phase through the design “reference loop”. Relevant questions that were needed for the data collection phase were derived from the literature study during the theory phase in order to assess the empirical data. The theory phase also contributed to refining the questions stated in the purpose chapter of the thesis.

4.2.3. Data Collection (Interviews & Survey)

The empirical data was collected during a meeting with 5 organizations. As shown in Figure 11, once the theory phase was complete, questions relevant for the case study were derived in order to proceed with conducting the interviews during the data collection phase. The questions derived were formulated at an abstract level in order to get a holistic perspective on the field of research. Relating back to the “feedback loop”, the data collected during this phase would serve as basis for the analysis phase, but it would also be incorporated for further use after the analysis phase as described in chapter 4.2.4.

Furthermore, the empirical data was collected by conducting interviews with different respondents from different companies. Although the questions derived from the theory phase were formulated and organized according to a strict template and in numbered order, the interviews were not conducted in a strict manner rather in a fluent conversation where all questions were accounted for. As a result of the feedback loop, after analyzing the empirical data, new questions arised and second interviews were held, this time in a more strict approach and with more depth. The analysis phase also served as a basis for deriving a new set of questions where a survey was sent out to different organizations within the EA community and incorporated into the reference loop. The survey was
designed using Google formmaker (located in Google drive) and shared with different groups on the social network Linkedin. The results from the survey contributed to the understanding on the field of research and facilitated the analysis phase.

Before meeting with the respondents and conducting the interviews, careful planning was necessary in terms of how to document and keep records of what was being said during the meeting. It was decided that in order to obtain all the information given, it was suitable to record the meetings as an audiofile and at the same time take key notes of what was being said. The recordings would later serve useful when analyzing the empirical data and writing the report.

4.2.4. Analysis of Collected Empirical Data

The analysis phase was initiated by listening to recordings and looking at notes collected during the meetings for the respective organization. The natural order of conducting this, was to listen to the earliest recordings first. While listening to the recordings, first draft notes on key topics were made using Microsoft Word. These key notes provided the basis for chapter 7, where an extensive summary of the recordings is presented. The draft also served as basis for the formation of hypotheses which are presented in chapter 6.

Moreover, the survey created and sent out to different discussion groups on LinkedIn, provided a quantitative accumulation of thoughts concerning the topic on reference architecture. This accumulation was then used as basis for adding to the information that was given during the interviews.

Formulating hypotheses would serve as validity for the conducted interviews in means of accurately drawing conclusions in advance.
5 Theory on Reference Architecture

For the following chapter, an extensive theory on reference architectures will be presented. As showed previous in chapter 4, the questions derived for the interviews are correlated to chapter 5 and the theory behind reference architectures. Therefore, the main questions derived in chapter 1.3 will be followed through the next following chapters in order to relate the core understanding of the field of research. First, an introduction to the concept of reference architecture and patterns is given. Further, a chapter on software architecture will be presented in order to bring forward the theory on design patterns. This is to make sure that the reader is provided with the background on how design patterns are developed and designed.

The chapter on design patterns add to the definition on reference architecture in the way design patterns are abstracted and reused. Design patterns are means of implementing reference architectures at lower levels of an organization. The chapter is then followed by a theory part on reusability and requirements that are set for the reference architectures in order to package the concept of reference architecture and provide a full insight on the main parts that constitute a reference architecture.

5.1 Reference Architecture and Patterns

There are many definitions on reference architecture. According to the Rational Unified Process, RUP[8], a reference architecture is “…in essence, a predefined architectural pattern, or set of patterns, possibly partially or completely instantiated, designed, and proven for use in particular business and technical contexts, together with supporting artifacts to enable their use. Often, these artifacts are harvested from previous projects”.

The general idea of what constitutes reference architectures, in the IT community, is still new. Due to this fact, there is no common understanding on what a reference architecture is in a broader definition [7, 18, 30]. From this inconsistency in definition, the term “reference architecture” has never gained maturity[13, 18]. Furthermore, this concern also brings questions regarding the level of abstraction and the ad-hoc implementation of reference architectures. This is mainly due to the fact that the concept holds its basis in software engineering, although research has found that the idea extends the possibility to implement it in areas of business and IT [7, 18]. As discussed further by[7], one of the main reasons of mistaken use of the term “reference architecture” lies within the term itself. It is the wide definition of the word “architecture” that lets “reference architecture” cover almost anything.

In order to implement reference architectures, the need to know when to use it and how to use it is of great interest. Since the level of scope that reference architectures hold is wide, implies that the perception ranges from software design patterns to business and IT. According to[7, 31], this level of abstraction can be visualized from a generic viewpoint to a more specific one. It also brings means to improve effectiveness in an ever increasing complex system. According to[18], reference architectures facilitate the means of managing synergy, providing a common understanding of architecture principles and in its essence provide guidance.

Reference architecture can be viewed as an encyclopedia of stored knowledge and best practices, describing past experiences in the work[7, 18, 31]. The accumulation of experience provides the guidance in applying and implementing architectural frameworks and design patterns. These proven principles minimize costs and maintenance requirements and promote attributes such as:

- Modifiability
- Performance
- Scalability
- Security
- Interoperability
- Reusability
- …
Foremost, it is patterns that are the cornerstones in describing how reference architecture function. According to [26], “Patterns are a general, reusable solution to a common problem and are dependent on their context”. This is due to the fact that patterns also share the same know-how accumulation of experiences. The idea of patterns is further related to design patterns and software architecture as described in subchapter 5.2 and 5.3. Other approaches in applying patterns have been proposed by [26] as EAM patterns. This approach document solutions of recurring problems within EA domains and manage to align business and IT. “The EAM patterns build up a pattern language, which has been documented in the EAM Pattern Catalog” [26]. Three types of patterns are included in the EAM; Methodology, Viewpoint and Information model patterns. The three types are described below:

1. **Methodology patterns** or M-Patterns, address specific methodologies supporting and detailing specific activities. Such activities have been left out in the methodologies addressed in e.g. frameworks such as the TOGAF. “The procedures defined by the M-Pattern can be very different, ranging from e.g., visualizations and group discussions to more formal techniques as e.g. metrics calculations” [26].

2. **Viewpoint patterns** or V-Patterns, help in the guidance of understanding views. From V-Patterns, visualizations in the form of diagrams, reports, etc., address problems in EA management. The main goal of the pattern is to extend viewpoints that are set as examples, often given by industrial users, to include a wider view that can be utilized for a longer period. Furthermore, V-Patterns are shared as a utility function for I and M-Patterns.

3. **Information Model patterns** or I-Patterns, addresses the issues concerning EA management by gathering and supplying the know-how information models. These information models contain definitions and descriptions of used concepts. As mentioned above, V-Patterns can be used to visualize the information in I-Patterns, but the information can also be accessed directly by the M-Patterns. Different languages can be used this pattern, but “…a language adequate to the problem to be addressed should be used”.

Reference architectures are contextual in the sense that they aim to link the as-is scenario of an enterprise to the to-be scenario, guiding in the build-up of present and future missions, visions and strategies [18]. This is done by utilizing the shared knowledge within an area of interest and is the core function when describing a reference architecture [18]. Since reference architectures are based on high levels of abstraction, they propagate and are developed even further to actual systems. Because of this abstraction, the applicability of the term is wide; a coverage is needed [7]. This can be shown by the classification scheme in Figure 12 developed by [7], describing how the abstraction of reference architectures can be classified from an abstract viewpoint (patterns), to generic and further on to conceptual architectures that comprehend fully developed enterprise solutions.
Figure 12 ERA classification scheme[7]

The classification scheme presents a view that shows on the gap between many types of reference architectures based on abstraction and coverage. According to [7], “…when RAs are made close enough to Solution Architectures their value to projects becomes enormous”. Although this is highly desirable, the flexibility of the reference architecture becomes compromised and cost concerns need to be addressed. As shown in the figure, the concept of enterprise reference architecture (ERA)[7] is introduced in order to make high trade-offs between cost efficiency and flexibility of reference architectures. ERAs resemble solution architectures and should be developed by the own business addressing specific business objectives. The implementation of the ERAs should be managed by the architects who developed them and the knowledge database they refer to should come from existing projects and know-how experience within the area of subject. This means utilizing patterns and reusable components as a source of knowledge.

Reference architectures can also be defined from the well known “4+1” view shown in Figure 13. This view is often shown and implemented for software architectures but according to [8], “The RUP suggests that a reference architecture should be defined along different levels of abstraction, or “views,” thereby providing more flexibility in how it can be used.” These views will be discussed further in chapter 5.2.
Relating back to the definition on reference architecture, the know-how experiences mined and the shared knowledge from previous projects are expressed in a broader definition to include architectural patterns. Patterns are inputs to a reference architecture and are contributing to the experiences regarding the implementation of a reference architecture. This is depicted in Figure 14 below and in accordance with[18]. Moreover, patterns are also “…a general, reusable solution to a common problem and are dependent on their context”[26, 32].

Pattern concepts describe an abstraction in how to express relations between the context, problem and solution [18, 33, 34]. The abstraction is a key ingredient in the way of expressing a pattern, and is done in order to ease the understanding of that pattern. As patterns become more complex and abstraction is suitable, the fine line of utilizing the abstraction is crucial. The challenges lie in conveying the meaning of the pattern and still make it easily understandable. Therefore the level of abstraction needs to be inherent in a way that details of patterns can be implemented in numerous of
ways. This is an iterative process that refines the meaning of the pattern and its uses for the whole of the reference architecture, as shown in Figure 14.

One example shown in[1] shows a typical architecture pattern to the left, identifying participants and systems, and a reference architectural pattern to the right. This is shown in Figure 15.

Figure 15 Abstraction difference: Architecture Pattern and Reference Architecture Pattern[1]

Figure 15 is an example of the abstraction being made to an architecture pattern, facilitating the reusability of that pattern. The architecture pattern shows the relation between a named browser interacting with both a customer (customer being defined) and with an Apache Tomcat server. On the right in Figure 15, it is shown that although the structure of the pattern remains, the entities are abstracted and therefore leaving interpretations of what a “person” or a “browser” is, unrestricted. This brings a certain degree of freedom when defining what those entities should consist of. The abstract structure of the pattern also provides the means of communicating relations between architectures on different levels. The use of a pattern facilitates[1]:

- The documentation of processes that share common structure.
- Sharing of a design pattern across different architectures.
- Solving a problem using a preferred approach.

Since reference architectures are based on previous experiences from other architectures, mining patterns are only of usage if they are packaged for usage by others[18]. And this is often the case only if proper documentation is made upon it. It is discussed by [18], that “Patterns can be transferred through examples, such as storytelling, or through informal or formal forms of documentation”. However, working by already established examples can prove to have the disadvantage of not forwarding the know-how experience. As stated by [18], “…patterns are only valuable if they can be used by others…”, and it is this statement that communicates the core concept of reference architectures. Patterns, in their way of being abstracted, communicate the main purposes of reference architectures[18]. Although patterns provide a way to describe reference architectures, there are many
kinds of patterns, described at many different levels of a reference architecture; Business, System, Technical etc.

5.1.1. A DoD Perspective on Reference Architectures

From a DoD perspective, the definition on reference architecture is in line with the aforementioned definitions presented. The reference architectures “primary purpose is to provide guidance and constrain on instantiations of solution architectures”[35]. The positive cause of this is shared by the common language provided for stakeholders, adherence to standards, facilitation of solving problems through implementation and validation of solutions towards a reference architecture. As a result of the abovementioned attributes, the definition provided by the DoD on reference architecture is as follows:

"Reference Architecture is an authoritative source of information about a specific subject area that guides and constrains the instantiations of multiple architectures and solutions."

This definition provided by the DoD diverts itself from the definition stated in the thesis in means of not describing the reference architecture as a way of reusing architectural artifacts or patterns. Although the definition incorporates the means of guidance, it does not convey a deeper meaning as the one stated by RUP.

According to[35] and in alignment of what has been said before, reference architectures can be defined at numerous levels and within different subject areas, providing viewpoints and relations between these. Furthermore, the abstraction of reference architectures can be defined from a specific to a more generic level. For the DoD, the reference architecture[35] constitutes of:

- Strategic Purpose
- Principles
- Technical Positions
- Patterns
- Vocabulary

The two last elements are described further in chapter 5.3 and 5.4.

5.2 Software Architecture

Although the definition on software architecture may be bound to the definition of architecture, as a structure of components and rules, it is extended into comprising[36]:

- A collection of software and system components, connections and constraints.
- A collection of system stakeholders-need statements.
- A rationale which demonstrates that the components, connections and constraints define a system that, if implemented, would satisfy the collection of system stakeholder need statements.

The definition on software architecture can also be that which is described by[9], where the definition propose a more pattern styled approach. The approach suggests that the architecture is built of components together with descriptions of the interactions of these, the result being architectural styles as described further in chapter 5.3. The paradigm of styles brings forward means to build, or design, an architecture in different ways. However, this design paradigm offers only a view on the structure of software and lacks a view of the overall system operation[36].

-30-
With the increasing understanding of how architectures can facilitate the work within system development, the increasing complexity in implementing systems follow[37]. The “4+1” model view describes the different views that concerns different stakeholders in a project. As shown in Figure 13, it is constructed of four views and one use-case view or scenario-view as described by[38]. The relationship between the views is correlated to the connections and constraints of elements and components[37] that one finds in the definition on architecture.

The 4+1 views are described as follows[38, 39]:

- Logical view: This view is the object-oriented model of the design that primarily supports the functional requirements. Abstractions are made in the form of objects or object classes.
- Process view: Takes into account non-functional requirements, as well as addressing issues of concurrency and distribution. Abstractions here are made at different levels, each level with different concerns.
- Physical view: Describes mapping of software to hardware and shows its distributed aspects. This view takes into account non-functional requirements.
- Development view: Describes the organizational aspect of the software in its development environment. Often, this view relates internal requirements to facilitate development, software management, reuse and constrains.

The main idea is to organize a certain architecture around these four views, which then can be showed through a use-case view. The use-case view puts it all together, portraying an abstraction of the important requirements. The 4+1 view shown in Figure 13 is generic allowing other notations, styles and tools to be applied. In the context of software reference architectures, they are abstractions of more concrete architectures at a certain level[30]. These reference architectures provide

5.3 Architectural Patterns

In order to structure and provide means to abstract the architecture, patterns are utilized[9, 12]. In this subchapter, different types of architectural pattern styles will be addressed and presented. These patterns each have functionalities in order to aid in improving and facilitating a reference architecture. Moreover, the patterns can be combined at different levels, “…as both a repository and an interpreter – or by elaborating a component of one pattern using some other pattern…”[9]. According to[40], architectural styles are important in the sense of their way of accumulating decisions made on architectural elements and the constraints they hold between elements and the relationship between the elements.

According to[Shaw] the definition on architectural styles is “…a family of systems in terms of a pattern of structural organization. More specifically, an architectural style determines the vocabulary of components and connectors that can be used in instances of that style, together with a set of constraints on how they can be combined. These can include topological constraints on architectural descriptions(e.g., no cycles). Other constraints – say, having to do with execution semantics – might also be part of the style definition.”

Different pattern styles are described further below. Each pattern is described in accordance with the descriptive formulation stated in[9, 12], which incorporates a problem, context, solution, diagram and a significant variant description.

1. Pipelined pattern

The pipelined pattern style is often utilized when a series of independent computations are made on a data stream and can, in principle, work in parallel and reduce system latency. The pipelined style stream data from one process to another, through pipes or system operations as one could see them. This is done by transforming inputs to outputs incrementally[9].

2. Layered pattern

Layered architecture provides a way of grouping functionality into layers within an application, each functionality being related to a role or responsibility[12]. This is done by stacking the layers vertically as depicted in Figure 2. Promoting reuse, it combines responsibilities and abstractions being made,
directly to the layer beneath. Different characteristics of layers are present. Applying strict layering allows interaction within components in the same layer, while a more relaxed layering allows for interaction with other layers. As discussed in chapter 3.1, the separation of concerns brings means of supporting flexibility and maintainability. Relating to reusability, since no dependencies exist from lower layers to higher, it is possible to reuse them for other scenarios[12].

3. N-Tier/3-Tier Architectural pattern

Resembling layered architecture, the N-tier and 3-tier styles also include a separation of functionality into segments[12], with the difference that each tier can be located on a physically separate computer. The functional decomposition of applications brings means of improving scalability, availability, manageability and resource management. The tiered pattern is independent from other tiers except the ones directly below or above it. Nth tier only handles requests from n+1th tier and forwards that request to the n-1th tier. It consists of three separate logical parts, each with a specific functionality and each located on separate physical servers. The independency provides the means of performing changes and updates without affecting the application as a whole, promoting maintainability. This pattern style is to be considered when sharing business logic between applications and “…you have sufficient hardware to allocate the required number of services to each tier”[12].

4. Service-Oriented Architectural Style (SOA)

SOA architectural styles are to be considered when aiming towards enabling functionality as a set of services. The main idea of utilizing this pattern style is to provide a baseline of interoperable services which are communicated using a range of protocols and data formats[12]. This is facilitated because of how loosely coupled the services are. This means that each service is independent of others and updates or replacements can be performed without breaking applications that use it[12]. Utilizing communication protocols provides the means of accessing and locate services on a network, locally or remotely. The benefits of this architectural style is that it:

- Promotes reuse of common services and thereby reducing costs
- Provides loose coupling and means of abstraction
- Protocols being industry standards provide means of facilitating interoperability, allowing other platforms to be built upon the service application.
- Duplication of functionality in services is minimized due to specific functionalities.

The abovementioned architectural patterns are some of many patterns existing within the industry.

5.4 Design Patterns

Having discussed the concepts of reference architectures, software architectures and their relationship, it facilitates the discussions on concepts of design patterns and their uses in more depth. In relation to what has been said about patterns in general in chapter 5.1 Reference Architecture and Patterns, design patterns are in the same way relying on a know-how design experience. Following the paradigm of [34], design patterns abstractified, named and identified in order to convey the true meaning of the pattern and its uses. By reducing complexity and facilitating the understanding for users, design patterns in their abstraction provide means for reusability and adding to the base of know-how experience[41]. Acting as micro architectures[34], they contribute to the vision and overall guidance of the system architecture.

The main use of design patterns is correlated to communication needs, and their ways of facilitating these. According to[41], pattern-based vocabulary where developers discuss patterns on white-boards and around the office, as means of referring to previous design experiences and knowledge. The reference to previous work shows that the developers “bundle knowledge about design constraints, the contexts in evaluating and deciding on design options to solve the design problems at hand”[41]. The main goal of utilizing design patterns is to increase efficiency for developers in their work.

Although this know-how experience is passed on, design patterns vary in abstraction and complexity. Because of this, there is a need to classify and structure them for future uses by others. Retrieving
design patterns, that may be benificial to a challenge that is faced, has proven to be difficult[34, 42]. For design patterns to serve useful and achieve a high value ad-hoc implementation, information about the pattern need to be documented in a structured way. There are several ways proposed to solve this concern. Figure 16 shows a classification schema proposed by[34].

<table>
<thead>
<tr>
<th>Jurisdiction</th>
<th>Class</th>
<th>Characterization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Creational</td>
<td>Structural</td>
</tr>
<tr>
<td>Object</td>
<td>Class</td>
<td>Adapter (class) Bridge (class)</td>
</tr>
<tr>
<td>Object</td>
<td>Abstract Factor</td>
<td>Bridge (object) Flyweight Glue Proxy</td>
</tr>
<tr>
<td>Object</td>
<td>Prototype Solitaire</td>
<td></td>
</tr>
<tr>
<td>Compound</td>
<td>Builder</td>
<td>Composite Wrapper</td>
</tr>
<tr>
<td>Compound</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 16 Classification scheme for design patterns**[34]

The design pattern should be classified in a way that conveys the true meaning of it. These also aid in means of communicating the different patterns for reuse and packaging. In order to create a library of design patterns, a set of characterizations need to be made. According to[34]:

**Intent** characterization describes what the pattern does and what intent it has. It also addresses design issues and problems.

**Motivation** characterization addresses the scenario in which the design pattern is suitable for use. It facilitates understanding the abstraction being made to the pattern.

**Applicability** addresses situations and examples of factors such as poor design and also includes how such situations are to be recognized.

**Participant** characterization describes the responsibilities and classes and objects addressed in a pattern.

**Collaborations** expresses how collaboration between users are being made in order to accomplish respective responsibilities.

**Diagrams** are graphical representations of design patterns carried out by using the object modeling technique.

**Consequences** describes how patterns support its main goals. It also take into account trade-offs and results of the patterns. It also addresses the ability of a pattern to vary system structures independently.

**Implementation** takes into account the awareness of pitfalls, hints and techniques during implementation.

**Examples** characterization provides examples from different domains and from real systems.

**See Also** describes what related intent characterizations there are between different patterns and how patterns can be correlated to each other in means of combining one pattern with another.
This list of characterizations are guiding users in communicating different design patterns and their uses. They also have the intent of providing a common ground and understanding of what a pattern does and how it can be reused.

5.5 Reusability

The main idea of adding the context of reusability to the concept of reference architecture is to provide means of utilizing architectural artifacts previously used in other projects[31]. This works in the line of promoting reference architectures and the advantages they bring. It facilitates communication, synergy and reduces complexity of an architecture. Often, reusability is addressed in software architectures where software objects are reused in terms of specifications, requirements, architectures etc. Although the concept of reuse is simple[31], reusing an object more than once brings questions regarding context of that object. As discussed in chapter 5.4, the context of a design pattern being characterized imposes questions regarding isolation; the relation of one object to other(s)[31]. However, this does not imply that reusability addresses software objects only, rather the experience contained in the objects and the packaging of this experience in order to make it reusable. The approach is to find the important and key experiences gathered from a project and recognize what to choose for future reuse. Difficulties from this lies in project roles not being able to gather experiences appropriate enough for reuse[31]. In order to improve and implement the idea of reusability, a cultural change in the industry need to be made. The change requires motivation, results and intent to be adopted culturally in the industry/enterprise/company. According to[31], an improvement paradigm has been proposed to focus on improvement within this area of research. The paradigm is depicted in Figure 17.

![Figure 17 The improvement paradigm][31]
The paradigm is based on improvement in software development and the resulting products. It implies a way of executing a methodology in which accumulation of evaluated experience is gathered in a form that can be easily understood and that provides means of accessing integrated models that can be modified to meet current needs[31].

In relation to what is being said on reference architectures, reusability plays a profound role in the subject of research. It is one of the main characteristics that describes a reference architecture, at a certain level. Although the improvement paradigm described and portrayed above is strongly correlated to software architectures, there is no reason why this does not apply to higher levels of architectures. Reference architectures described at generic levels are able to be reused for other instantiations using the same methodology; previous experience that is gathered from other projects. The reuse of a generic architecture is set in the different levels of abstraction being made on different patterns. According to[31], reusability can be performed at different levels in a reference architecture; Reference, Conceptual and Implementational.
6 Hypothesis

In relation to the questions formulated in chapter 1.3, hypotheses were derived from the interviews during the empirical data collection phase. This chapter aims to provide the reader with an understanding of some of the conclusions that were drawn before conducting the analysis phase. The hypotheses are also means of validating the answers provided from the respondents during the interviews. The validity and reliability is increased when meeting with the respondents one more time in order to make sure that the conclusions drawn were accurate with the recordings and notes of what was being said during the meetings.

Below is a compilation of the different hypotheses derived and their relationship to the questions derived in chapter 1.3. Each questions from chapter 1.3 is followed up by a short respective hypothesis. The reason for structuring it in this manner is to show the relation between the main goal of the thesis, the theory part of the thesis and the answers provided from the interviews.

Question 1: What constitutes a reference architecture?

Hypothesis; Reference Architecture is implemented at different levels

Reference architectures can be found at different levels in relation to the Zachman framework. However, the relation connecting the different levels vertically and horizontally seems to be missing. This causes the formation of islands of reference architectures in the framework. These islands are often found floating at different levels in the framework with no relation between them. Moreover, a consistent use of a framework guiding the development and implementation of reference architectures and enterprise architectures is lacking. Although an organization might choose some principles of a framework adapted to their work, they do not implement it to the full extent.

Question 2: How is the reference architecture maintained (framework support)?

Hypothesis; Best practice and know-how knowledge is utilized

The common idea is that there is an uncommonness of utilizing functions for maintenance and governance. Even though tool support for developing and maintaining the structure of a reference architecture exists, it is not widely used. Methods for maintaining an architecture is chopped up and used in parts and in ad-hoc manner. From this, the conclusion one can draw is that there is no common coordination. It is often referred to the response: “this is the way it has always been done”.

Question 3: How are patterns implemented and packaged for reusability in the future?

Hypothesis; Patterns in general are utilized at lower levels in an organization

Patterns are utilized at lower levels in relation to the Zachman framework. Often, at these levels it usually concerns design patterns for code management. There are architectural patterns at higher levels but the common understanding is that patterns for reuse in general tend to be difficult to apply at higher levels, e.g. Business levels. They are useful in the sense of providing proven practices from other projects, but do not necessarily facilitate the work.
Question 4: What is a reference architecture?

Hypothesis; There is no common understanding of the term reference architecture

The definition on reference architecture is wide and used in different cases. The general idea is that the term “reference architecture” is being adapted to an organization’s own needs, processes, architectural solutions and at the specific level of the Zachman framework of which the reference architecture proves to be more useful. A reference architecture can be applied and developed to an entire organization and/or to a specific project.

Question 5: What types of patterns were developed and/or reused to accommodate the reference architecture in the different organizations?

Hypothesis; Patterns are utilized, but to what extent and at which levels, has not been presented

There are patterns and design principles that are internal for the different organizations. A specific list of patterns and their relation to reference architectures has not been presented for the author to view.

Question 6: Why are reference architectures implemented to begin with?

Hypothesis; A desire to govern and maintain common understanding of different systems and solutions in an organization and in projects, internally and externally.

Acquiring and implementing a reference architecture arise when a need for governance and agreement on different solutions exist in an organization. Moreover, reusability has been a key principle when regarding reference architectures. The main idea of utilizing the reusability of different patterns and the know-how experience from previous projects have been desired. This brings means of support for communication within the organization and important key decisions are easy to follow. The development of reference architectures lies also in the organizational culture that has to be taken into account. The culture brings notions on how co-workers are motivated to address new methodologies such as a new framework. This is due to the fact that the framework rely on communicational needs. It is also implied that in order to improve and streamline the concept of reuse, a cultural change has to be made if not existent.
7 Empirical Data

This chapter will present and conclude the data collection phase where interviews have been conducted with different organizations. Moreover, a general survey on the most important aspects regarding reference architecture as a field of research has been sent out in order to provide consistency and validity in the form of a quantitative analysis.

Subchapters 7.1 and 7.2 relate to the questions derived in chapter 1.3, as a part of the methodology presented in chapter 4.2 and relate to the feedback loop where the iterative process of refining the questions is made.

7.1 Interviews

Chapter 7.1 below, present the main ideas that have been formulated from the analysis phase. The method is to present empirical data from the viewpoints of each organization of which the interviews have been conducted with. This implies formulating the groundbreaking thoughts concerning definition, governance, implementation and reusability in the subject of reference architectures.

In order to present the data collected during the interviews, the answers are divided up in different sections concerning the topic of reference architectures and relating to the questions stated in the purpose of the report. This way of presenting the results provides the readers an incentive to skip the theory behind reference architectures if needed and directly jump to the empirical data.

7.1.1. Role and Organization

Role 1, Organization 1
Technical evangelist within Developer Platform Evangelism (DPE). Working with the development of a target platform of which customers use for their products/services. Has not been working much with enterprise architecture, rather in projects that have contributed/delivered to reference architectures and questions regarding these.

Role 1, Organization 2
Business architect targeted towards system integration.

Role 1, Organization 3
Has been working at different levels; Businessanalyst, Enterprise Architect, Systemanalyst. The role is to work on requirements engineering for next-generation structuremanagement tools.

Role 1, Organization 4
Vice President of the company. Works with developing architectures for customers within the insurance and banking sector. This includes developing and communicating different reference architectures at certain levels adapted to the main systems with the customer. When discussing reference architectures, this implies formulating main ideas and longterm strategies for the customer.

Role 1, Organization 5
Vice President of the company. Develops applications based on reference applications for systems that the customers has.

Role 2, Organization 3
General architect for productionsystems. Working with serviceoriented businessmodels and other assignments. For the moment working as an organizer of conceptual models, participating in the formation of client requirements and the visions and support surrounding these.

Role 1, Organization 6
Works as system designer. This means developing architectures within a system or systems.
7.1.2. Interviews

A. Definition

Organization 1
Reference architectures are means of streamlining the workflow in order to make it easier for different roles to switch between projects and in that measure reduce the support costs, meaning it reduces the time spent on learning new things. Moreover, reference architectures bring means of abstractifying workload in order for different teams in projects to get more freedom of choice. It also means setting guidelines in the company of “dos” and “don’ts”.

Organization 2
Reference architectures are looked at as means of “best practice” and as a dictionary of how things are to be done. Relating to the datamodels, reference architectures provide the means of abstractifying different notions that, later in the work, can be referenced to, be it in current projects or future ones. The term reference architecture cannot be found in this organization. The closest one can get to the term is the levels of information management in the form of conceptual models.

Organization 3
Reference architecture is a set of principles that are implemented for guidance in the work performed. The concept can be found at different levels but is not profoundly stated. The view on reference architecture and its definition is strongly correlated to the definition on general architecture and the means of being something one refers to in order to achieve a common idea and doing the same thing throughout different levels. In relation to the definition stated in RUP, the objection is that reference architectures don’t have to be based on previous projects. It is believed that referencing to an architecture also can be made to an existing reference architecture. Reference architectures provide the means of guidance through different principles.

Organization 4
It is an idealized concept of something of which an organization or a company can map themselves towards. The example provided was a reference to how IBM use reference architectures to find out what business functions are used for creating a bank.

Organization 5
In general, it is hard to say what a reference architecture is. It can be a model or a pattern, usually located at lower levels in the Zachman framework, i.e. solutions and application levels. The general idea for this organization is that in 9/10 times, reference architectures concern applications and is not referred to as a reference architecture, rather a reference application. This reference application is a framework used for modeling an idea before implementing it. These reference applications are also considered to be smaller parts in a bigger reference architecture. For example if the reference application is a service application in junction with a service oriented architecture (SOA).

Organization 6
The term “reference architecture” is not familiar, however the term “framework” has been widely used and recognized.
B. Governance

Organization 1
In regards of governance, the main idea with the reference architecture is to provide a certain degree of freedom for teams in projects, and it is the attribute of abstraction in the idea of reference architectures that allows this. It is also the idea of governing the work produced, providing means of knowing what to be done but also being able to freely work around the current idea. This choice of freedom is essential in reference architectures.

Organization 2
Reference architectures are critical and important, especially for larger organizations. A reference architecture serves as a nexus for large networks and in the process of managing these, governance is needed. Governance resolves the issue of exchanging competence and knowledge from different projects. This is crucial because it is related to business processes, patterns, platforms etc. Although there is a coherence in this organization, the means of communication is inconsistent in a bottom-up manner of working, which raises the question regarding organizational culture and forms of governance.

Reference architectures are conveyed in form of directives but are not explicitly referred to as reference architectures. Furthermore, these directives are not communicated to every level in the organization. In most organizations, support is provided from an IT-department giving suggestions of change, which is then forwarded to higher levels in the organization and where frameworks for these suggestions are developed. In this organization however, there is no IT-department and most of the suggestions of change usually have to be solved at the respective levels where change is needed.

The main issue is, while developing a reference architecture, reorganisations might occur in that department; The architecture in process of development will be replaced by someone developing another architecture. Because of this, it is hard to establish a continuity.

Organization 3
In order to reduce the shortage in communication in the work with developing architectural frameworks, reference architectures are seen as tools in aiding with this issue. Reference architectures also reduce suboptimizations when different roles in an organization don’t work with the same things. When it comes to governance, different organizations are being run bottom-up and others top-down. Somewhere in between, there needs to be a common understanding. The solution to bridge the borders concerning common understanding, is to use centralized governance. This brings means of traceability of information from the organization, where it is known where the information comes from and who the owner of that information is.

Organization 4
As an organization working with consultation for other companies, the main use of reference architectures is the ability to map the company towards these, in order to provide a coherent and general idea of where the company holds its, values, strategies, goals and processes. This concerns also communication between these boundaries. If mapping the company towards this framework fails, reference architectures provide the means of filling in that blank. This architecture also provides a way of faster knowing what it is to be done in order to push it forward to implementation.

Organization 5
Governance regarding reference architectures at application level concerns a reuse of best-practice and know-how experience. Visions and different directives are provided from the customers to the organization.

Organization 6
The reference architecture that was placed at system level (in accordance to the Zachman framework) had decision makers for data models at city plan.
C. Implementation

Organisation 1
n/a

Organisation 2
The implementation phase is based on granularity by all means. The closer to an actual system the reference architecture gets, the more usefulness it serves to be. However, increasing granularity minimizes the ability to see the benefits of the reference architecture. The implementation requires delimitations to be existent. This means that the only way of finding out what one part is doing is by looking at what other parts are connected to that part. Designing the architecture in this way reduces suboptimizations to occur, preventing people from working with the same thing. Implementation is also dependent on defining products and protocols and which sets of these one should choose when describing the reference architecture at concern levels.

Organisation 3
When implementing reference architectures as means of principles and directives for guidance, communication of these directives in a top-down manner of working can be hard to achieve. However, it is crucial in the implementation phase of the reference architecture.

Organisation 4
The reference architecture is implemented based on the concerns that the customers are facing. From this, one can work out solutions based on business goals that have been brought forward.

Organisation 5
n/a

Organisation 6
n/a
D. Reusability and Patterns

Organisation 1
Reference architectures are good to have and they need to be placed on the right level, not making the architecture too abstract. This is to ensure a certain degree of freedom for teams using the reference architecture. Every situation is unique, therefore it is not optimal to push people towards a solution based on what they are doing. Of course if some ideas are progressed to a likeminded solution and the solution is reusable, it might work. However, the general idea is making a solution abstract and still manage to maintain a certain granularity.

Organisation 2
Regarding reusability of reference architectures, it is a matter of best practice. For example "Which functionality is placed in what domain?" referencing to the definition stated by RUP on common solutions or patterns. On a higher business level, there is no guidance on what patterns to implement/utilize/substantiate, but it is often driven from the lower levels. On lower levels, realization of the fact that not working together similarly, might cause negative trends in management. In means of helping each other with solutions, it eliminates the dependence on personnel between projects.

Reference architectures are very important when it comes to exchanging competencies and knowledge from different projects. This concerns processes, patterns, platforms etc. In this organization, patterns are present and utilized at an integration level, but communication of these diminishes at higher levels. Considering reusability, developing or buying general systems may create more work. This is the result of systems not being adapted to the processes of the organization because of the abstraction they hold. As an example, the process of change management where “impact” is an important aspect, has very little meaning since it does not define what impact it concerns; Business, System, Costs or Operations.

In regards of utilizing patterns, there needs to be a change and a prevailing consensus on what components to be regarded as patterns and implemented when it concerns a company with business positioned in different countries. In order to achieve reusability and flexibility, governance has to be a central part in the process. It is necessary for an organizational maturity and development.

Organisation 3
Reusability plays a big role for reference architectures and is the means of referencing and carrying out something in order to get similarities. This is done by looking at different solutions to other challenges and use one or more solutions that fits for the given concern. This however, in contrast to the definition stated by RUP, does not necessarily need to be based on previous projects. Rather this can be reused from a current architecture. Patterns do not hold constraints of being based on diagrams describing systems, they can also describe the reasoning of an idea at higher levels, i.e patterns don’t necessarily need to be UML diagrams but can also describe architectural directive principles. The general idea of reusability for this organization is that “We should not invent the wheel again”, i.e. take value in what we do and what we have done previously. Granularity is important when discussing the properties of abstraction. Dimensioning the abstraction of a pattern may cause a “one size fits none” concern.

Organisation 4
In this organization, when consulting a bank or an insurance company, patterns are utilized in terms of positioning what is to be done. For example in a bank, different patterns are used to describe what business functions are needed in order to develop a bank. In an insurance company, the patterns can be used to map the company towards and define at what level it is located, for example at a level defining strategies. Abstraction of patterns is good to have in order to get a holistic view of knowing what is missing, but it mostly concerns the roles utilizing the patterns. Patterns can be described at different levels, it can be a framework such as Zachman and TOGAF but it can also describe levels within these frameworks.

The important thing when utilizing patterns is not making the mistake of following these slavishly, and always remember what the main goal was in first place. Using patterns also required the
competence of being able to handle them. Regarding reusability, the reference architecture is a way of assuring that everything has been accounted for and then moving on to the next stage of implementation. In means of reusing the reference architecture, it provides the means of viewing how it has been done before and reuse that concept to a new architecture in order to make sure nothing has been left out. This however, is very specific towards processes and the reusability in those aspects is not applicable in terms of direct implementation.

Organisation 5

Being positioned in the borderland of system level and technical level, reusability of reference architectures is always important. Developing a new reference architecture is never a concern due to the fact that there are many examples formulated, for example the Microsoft Application Architecture Guide, where many of the issues are solved. Developing something new seems ridiculous when others have taken the time to formulate a good reference architecture. These formulations then provide the means of reusability regarding best-practices and know-how knowledge.

Reference architectures in means of patterns provide more examples at lower levels. Patterns that are utilized are conveyed, not through documentation but rather through a system called Data Transfer Object, DTO. This system provides a way of viewing what patterns that can be utilized for a certain application. The reference architecture simplifies the work being done if the co-workers understand it, otherwise there is no point of having one. When developing an application for the customer, this organization receives the visions and directives in the form of concrete examples of what the application should do. These directives are then taken in and used in the reference architecture in order to convey the true meaning of the system or application being developed.

Organisation 6

Developing a new system that was going to be integrated to an existing one, the new architecture had certain requirements. One of the requirements was that the architecture should take into account the functions/logic present in the system. In order to realize this, a breakdown of the system was needed in order to present the new architecture in a layered style, although not layered entirely. The purpose of the reference architecture was to make sure that the existing functions/logic was being reused together with the new functions in an appealing interface. It also provided means of combining applications more freely. The old architecture was reused for the new system, for example the integration solution in lower layers was reused in order to get a hold of the logic. This proved to be very helpful when a new user interface on the internet was being developed in means of fast deployment. This was because of already having worked with developing the system as a service. Although some adjustments were made, the building blocks (read patterns) were used.
7.2 Survey

In chapter 7.2, answers to the survey on reference architectures are presented. Here, the answers are made up of a quantitative data collection from different roles in the IT-industry. This will be discussed further in chapter 10 on validity and reliability.

This chapter, much like the preluding subchapter 7.1, provides the reader an incentive to examine the answers provided in the survey that was created. A link to the survey is provided in the appendix of the report. The questions derived and answered are, in the same way as the interviews, related to the questions derived for the purpose of the report in chapter 1.3.

7.2.1. Answers to Survey

A.  
Organization: e-man  
Department: System Development  
Role: Work mostly in the system integration space on larger enterprises in various roles which often touch on the EA perspective (either working with an existing EA group, or perform EA-related activities)  
Years working with EA: >5  
What does reference architecture mean/represent for you or your company?  
Depends (doesn't it always?). One viewpoint is to have a few general reference architectures on how to solve a larger group of problems (in my case often system integration related) that can be reviewed when implementing a new strategy/architecture for a new environment/organization. Another viewpoint is at a specific environment/organization where a defined and agreed upon reference architecture serves as the target for projects (both for new solutions that should adhere to the reference architecture and for old solutions that should be migrated to the RA)  
What do you consider to be the advantages/disadvantages of RA in the work in IT?  
In general notes I view the architectural methodology as always beneficial. There are always risks of performing too much analysis up front (compare to the agile methodology that is receiving a lot of traction). For example, trying to set a RA before knowing the problem domain well enough might produce an architecture that hinders development rather than being beneficial. It can also be said that care must be taken to allow changes in the RA when new requirements are identified. Mostly though, I believe having an RA increases flexibility, lowers TCO and increase the value of investments.  
What scope and complexity do you find RA to have?  
Depends (doesn't it always?). One viewpoint is to have a few general reference architectures on how to solve a larger group of problems (in my case often system integration related) that can be reviewed when implementing a new strategy/architecture for a new environment/organization. Another viewpoint is at a specific environment/organization where a defined and agreed upon reference architecture serves as the target for projects (both for new solutions that should adhere to the reference architecture and for old solutions that should be migrated to the RA)  
Since the term RA is widely used in software development, at which levels in an organization/company do you think RA could be found?  
Conceptual level, Business level, System level  
What do you find the requirements on RA to be?  
Depends on the scope and purpose of the RA  
What tools can be used in the work of implementing a RA?  
There are many EA-tools that could be used, but most companies in my experience rely on standard diagramming and document tools such as Visio/Word backed with IT-processes/organizational responsibilities to guide and enforce the RA utilization.  
What is the importance of utilizing patterns in the work with RA?
I'd say that reference architectures should contain standard patterns that should be followed (unless having very specific requirements), and that these patterns ought to be based on current best practices in the domain that the RA applies to.

**What roles do you find in the work with RA?**
Business architects, System architects, Programmers

**What type of methodology/framework is utilized when implementing, developing and communicating a RA?**
Depends on the organization

---

### B.

**Organization:** Aker Solutions  
**Department:** Business Development  
**Role:** Senior Advisor; Enabling EA initiatives at Aker  
**Years working with EA:** >5

**What does RA mean/represent for you or your company?**
Creating building blocks that can be reused during new implementation or as a vision in restructure of the current IT landscape.

**What do you consider to be the advantages/disadvantages of RA in the work in IT?**

*Advantage:* Is a concert way to show a implementation of chosen principles in the architecture. Speed up implementation of new services.

*Disadvantage:* The chosen RA is not optimized for any particulate service which lead not optimized solution.

**What scope and complexity do you find RA to have?**
RA is define as building blocks, these building blocks can be organisation, process, data, service, application or infrastructure. The scope is define most of the time that the building block can stand by itself.

Since the term RA is widely used in software development, at which levels in an organization/company do you think RA could be found?
Conceptual level, Business level, System level, Technical level

**What do you find the requirements on RA to be?**
Reusable building blocks.

**What tools can be used in the work of implementing a RA?**
A proper EA suite tool such as Mega, Aris etc. is helping to find and define these building blocks.

**What is the importance of utilizing patterns in the work with RA?**
It might speed the initial process, but pattern will only take you so far, the details has to be specified for each given situation.

**What roles do you find in the work with RA?**
Business architects, System architects

**Other?**
The EA team.

**What type of methodology/framework is utilized when implementing, developing and communicating a RA?**
TOGAF
C.

Organization: Extenda AB
Department: Research and Development

Role: Chief Architect for the product development. Internally I am responsible for the technical aspects of the products and most important how we develop them. How we shall communicate internal and external, how to ensure quality, work methodologies and what tools do we need. Externally I usually take part of the customer projects to ensure the deployment model will work.

Years working with EA: n/a

What does RA mean/represent for you or your company?
We have a RA for our products, how they can be combined, integrated with other systems, secured and deployed in production. A lot of best practices.

For some of our clients we have done RA templates. Each country have different constraints that they can adapt to the template for the country instance.

What do you consider to be the advantages/disadvantages of RA in the work in IT?

Advantages: Something to start from, a common way of doing things that might make it easier to transfer people and knowledge between units/systems.

Disadvantages: I guess too much micro architecture is placed in the RA that should only contain macro architecture. This makes the organizations slower and less flexibly and not being able to respond to changing demands.

What scope and complexity do you find RA to have?
It slowly evolves so hard notice the changes and there I do not consider it important. However very much of the work is based on the RA or the products and how they work.

Since the term RA is widely used in software development, at which levels in an organization/company do you think RA could be found?
Business level, System level

What do you find the requirements on RA to be?
They should only contain macro architecture.

What tools can be used in the work of implementing a RA?
Tools that visualize and communicate the architecture.

What is the importance of utilizing patterns in the work with RA?
As template to get starting from.

What roles do you find in the work with RA?
Business architects, System architects

What type of methodology/framework is utilized when implementing, developing and communicating a RA?

n/a
D.

Organization: IRM
Department: n/a
Role: Senior Consultant

Years working with EA: >5

What does RA mean/represent for you or your company?
Have not heard about it before but I assume you mean an EA Framework.

What do you consider to be the advantages/disadvantages of RA in the work in IT?
Not too much, it is more to be used in the business.

What scope and complexity do you find RA to have?
Mostly too complex.

Since the term RA is widely used in software development, at which levels in an organization/company do you think RA could be found?
Business level.

What do you find the requirements on RA to be?
To be a link between business and IT.

What tools can be used in the work of implementing a RA?
No tools needed.

What is the importance of utilizing patterns in the work with RA?
May give a start.

What roles do you find in the work with RA?
Business architects.

What type of methodology/framework is utilized when implementing, developing and communicating a RA?
The Zachman framework.
E.  
Organization: H&M  
Department: n/a  
Role: IT Architect, Development Support, Methods & Processes  
Years working with EA: >5  
What does RA mean/represent for you or your company?  
Architectural Pattern on different levels and forms of the Framework from EA to Solution Architecture.  
Other?  
IT Development.  
What do you consider to be the advantages/disadvantages of RA in the work in IT?  
A lot of advantages to communicate and re-use.  
What scope and complexity do you find RA to have?  
It very important tool for the architects.  
Since the term RA is widely used in software development, at which levels in an organization/company do you think RA could be found?  
Conceptual level, Business level, System level, Technical level.  
What do you find the requirements on RA to be?  
TOGAF, Our corporate EA MetaModel.  
What tools can be used in the work of implementing a RA?  
ARIS, Sparx EA.  
What is the importance of utilizing patterns in the work with RA?  
The Pattern and RA are one package.  
What roles do you find in the work with RA?  
Business architects, System architects, Other (If other then please specify in the next question).  
Other?  
IT and Solution Architects.  
What type of methodology/framework is utilized when implementing, developing and communicating a RA?  
TOGAF.  
Other?  
The customized for the company.
8 Analysis

In this chapter, an analysis will be presented. The analysis will be done in regards to the relation between the interviews, hypotheses and the survey. The analysis will provide the reader with some additional views on the correlation between the hypotheses and the empirical data. It also brings forth some key answers that were provided during the interviews as means of validity for the analysis being made in relation to the hypotheses stated in chapter 6.

8.1 Interviews → Hypotheses

Hypothesis 1: Reference architecture is implemented at different levels.
Reference architectures, when developed, can be found at different levels. However, it is not fully clear by what definition one refers to when speaking of reference architecture. At system level, it is referred to as reference applications while on contextual level it is referred to as guidelines. The common idea though, is that reference architectures can be implemented at different levels. This is motivated by fact that the respondents coming from different levels are discussing reference architectures. Each respondent discussed reference architectures in relation to their view on the subject and in perspective to their work at a certain level. Although they did not speak of a specific reference architecture, it is fully evident that the concept resides at that level.

Hypothesis 2: Best practice and know-how knowledge is utilized.
The respondents did not talk about a specific way of utilizing best practices in the work with reference architectures. However, much was said about the main principle of reference architectures, i.e. best practice. As stated by the respondent in organisation 2 –
“Reference architectures are looked at as means of 'best practice' and as a dictionary of how things are to be done.”
The respondent in organisation 5 on the questions regarding governance stated that-
“Governance regarding reference architectures at application level concerns a reuse of best-practice and know-how experience”.
This comes to show that the concept of reference architectures very much takes into consideration the reuse of best practice. It also implicitly states that the best practice is utilized at different levels.
This best practice was more often discussed in regards of reusability of patterns at various levels and what role it plays in terms of developing a reference architecture. Patterns are utilized and developed in order to preserve an underlying solution to a problem. The respondent from organisation 3 stated that-
“This is done by looking at different solutions to other challenges and use one or more solutions that fits for the given concern”.
The best practice takes form in different solutions and patterns to solutions that can be reused in the future in means of guidance and improvement.

Hypothesis 3: Patterns in general are utilized at lower levels in an organization.
According to the interviews held, the concept of patterns can be found at different levels; Conceptual, Business, System, Technical. However, patterns are more evident at lower levels, where they are taken in form of different building blocks for building applications and systems. The higher the level is, patterns are more derived in the form of objectives and visions for an architecture. At lower levels, for instance in System level, patterns are physically attainable in the form of different examples. In relation to the statement of hypothesis 3, the hypothesis cannot be confirmed due to this. The respondent from organisation 5 expressed that “Reference architectures in means of patterns provide more examples at lower levels”. Although the respondent was located at a lower level, it comes to show that it is more evident that patterns are developed at lower levels. Depending on what one perceive a pattern to be, it can be found at every level, but that they are more evident in lower levels.
**Hypothesis 4: There is no common understanding of the term reference architecture.**

This statement is clearly accurate. Based on the answers from respondents, the main consensus is that the definition of reference architecture is broad. The respondent from organization 5 states that –

“In general, it is hard to say what a reference architecture is. It can be a model or a pattern, usually located at lower levels in the Zachman framework, i.e. solutions and application levels”.

Relating to the same question, the respondent from organization 6 thought that -

“The term ‘reference architecture’ is not familiar, however the term ‘framework’ has been widely used and recognized.”

Although the answers from the respondents share some similarities, they all provide different definitions. In some cases, a reference architecture is only found at a certain level and in other cases the reference architecture will be seen as a framework. This shows the diversity of definitions that are stated for the concept of reference architecture. As well as confirming hypothesis 4, the analysis also validates the theory revised in chapter 5. Due to the wide applicability of the concept of reference architectures, it can be derived at different levels and thereby also be interpreted in relation to that level.

**Hypothesis 5: Patterns are utilized, but to what extent and at which levels, has not been presented.**

The statement in hypothesis 5 is confirmed based on the empirical data collected. No one of the respondents provided sufficient data on exact levels where patterns are utilized. This has to do with the respondents not being able to provide such data due to company policies.

**Hypothesis 6: A desire to govern and maintain common understanding of different systems and solutions in an organization and in projects, internally and externally.**

Hypothesis 6 proves to be a challenge to confirm due to its broad statement. The answers provided by the respondents all seem to show that reference architectures very much are needed in order to maintain and govern the work with IT. Reference architectures are often developed in order to provide a common understanding between roles. The common understanding is then extended when developing systems and different solutions. More than this cannot be said to confirm the hypothesis.
8.2 Survey → Hypotheses

**Hypothesis 1: Reference architecture is implemented at different levels.**
Although the survey did not have many respondents, the conclusion drawn for this hypothesis is that the statement seems to be accurate. Reference architecture as a concept can be found at different levels in an organization. Since this conclusion can be interpreted at different ways, a second conclusion will be given; Reference architectures can be found at different levels in an organization, however, it is more evident that the reference architecture would be found at a Business and/or at a System level than the levels below and above these. This conclusion is also drawn in relation to the survey. The analysis of the survey shows that the five respondents, for the question on where reference architecture is to be found; five of them answered ‘Business level’, four answered ‘System level’, three answered ‘Conceptual level’ and three again answered ‘Technical level’. Although the question considered where reference architecture was to be found, the conclusion was made that if it was to be found, it could also be implemented at those levels.

**Hypothesis 2: Best practice and know-how knowledge is utilized.**
This hypothesis is not embedded in any of the questions stated in the survey and therefore a proper analysis cannot be formed. However, the answers to the question “What does RA mean/represent for you or your company?” provides assumptions that can validate this hypothesis. For that question in the survey, the respondents state that reference architectures are “means of best practice”, “architectural patterns at different levels” and “creating building blocks that can be reused…”. These are perceived as implicit answers and formulations to the hypothesis.

**Hypothesis 3: Patterns in general are utilized at lower levels in an organization.**
The answers to the survey did not provide any value for analyzing this statement. The respondents of the survey have pointed out that a reference architecture utilize patterns. However, the same respondents have stated that a reference architecture can be found at different levels. The levels mentioned in the survey are all found in the middle of the framework discussed in chapter 2.3. This implies that no direct correlation can be made between the existence of reference architectures at different levels and the use of patterns at different levels. The answers do not give means of drawing the conclusion that the reference architecture is utilizing patterns, only that patterns are utilized and that reference architectures can be found at different levels.

**Hypothesis 4: There is no common understanding of the term reference architecture.**
This statement is to broad be confirmed by a survey, and therefore no direct analysis can be applied. Although different answers were provided from the question “What does RA mean/represent for you or your company”, they do not provide a single conclusion to be made. However, when reading the respondents answers, it can be perceived as if the common understanding of the term is wide and derived from the respondents own conception of the term. This means that each respondent seem to have formulated the answer based on its own experience within the field, and does not provide a common or general definition of the term.

**Hypothesis 5: Patterns are utilized, but to what extent and at which levels, has not been presented.**
As discussed in hypothesis 3, this can be confirmed by the explicit answers to the survey. At what levels patterns are utilized is not presented or discussed at all in the survey. In regards of the reliability of the questions formulated in the survey, refer to chapter 11.
Hypothesis 6: A desire to govern and maintain common understanding of different systems and solutions in an organization and in projects, internally and externally.

There is an underlying challenge of confirming this hypothesis due to its broad statement. The respondents of the survey had all answers that can be interpreted to confirm this statement. However, it has been thought of in retrospect, that each word in the hypothesis can be relatively interpreted differently by each respondent. Each word can have different meaning, and therefore it is hard to make a good analysis of the answers to the survey and to confirm this statement.

8.3 Interviews → Survey

It is evident that the answers provided from the interviews and the answers to the survey are very much alike, but differ in the number of questions asked. The interviews were based on more reflective questions on the subject of reference architectures than the survey did. All of the respondents (survey and interviews), had more than 5 years of working experience in the field of IT and they all had various roles. For the interviews, 2 out of the 7 respondents had the role of CEO and 2 were coming from a business level in the organization. The rest of the respondents came from various roles such as system designer, technical evangelist. One of the 7 respondents had worked at a business level and system level. The respondents for the survey had also various roles coming from R&D and system level. One of the respondents for the survey was a senior consultant. It is evident from the respondents during the interviews, that the area of IT-architecture is well dominated by people with a lot of expertise and several years of experience.

In general, the broad definition on reference architecture is hold by all respondents. They all provided different answers to what they believed a reference architecture to be. However, it was clear that each definition shared common properties with other definitions. Some properties were that a reference architecture consisted of building blocks/patterns that are utilized for further use, while other merely saw the architecture as a methodology in the work with EA and governance. Others perceived reference architectures to be target architectures that are pointed towards when seeking a solution to a problem. The respondents all thought that the reference architecture could be found at different levels but did not provide examples of such architectures. They also believed that patterns were played a big role in the development of a reference architecture but could not define how a pattern should look like.

In conclusion, the respondents provided enough data to create a holistic view on the properties of the reference architecture and its main goals. Both the interviews and the survey provided means of assessing the general thoughts of how a reference architecture can aid in the work with system development and what other underlying architectures and methodologies that are present in order to govern the reference architecture. The main difference between the interviews and the survey was that the respondents of the survey confirmed the theory chapter by explicitly stating that a reference architecture utilizes the different frameworks discussed in chapter 3.4 to 3.6.
9 Conclusions

From the interviews conducted and the survey created, in order to assess the thoughts and ideas relating to reference architecture, different conclusions can be drawn. This chapter will conclude the empirical data chapter, providing the reader with the general conclusions as basis for discussion in chapter 10. It is also a way of relating to the questions derived in chapter 1.3, purpose of the thesis.

The first subchapter, 9.1 will adress the conclusions drawn from the answers provided during the empirical data collection phase as shown in Figure 11 in subchapter 4.2. It will also provide answers to the questions stated in chapter 1.3. Furthermore, subchapter 9.2 will provide a mapping of the different organizations towards the Zachman framework as mentioned earlier in subchapter 2.3. This will bring the conclusions on definition and communication of reference architecture throughout the organization. It will also serve as basis for further discussion in chapter 9.

9.1 Conclusion drawn from empirical data

The answers provided from the interviews and the survey that was sent out, clearly shows that reference architecture as a concept has yet to reach a certain maturity level as other concepts such as different frameworks (Zachman, DoDAF, TOGAF e.t.c). The maturity level is in terms of how well understood the definition, implementation, governance and scope is. Although the concept as a whole is commonly referred to and thought of when discussing EA, the common understanding on implementing the architecture is very vague. Since the term “architecture” itself is vague and abstract, reference architecture becomes very abstract in means of definition.

The main conclusion drawn from the respondents is that there is no common understanding on what reference architecture is meant to do in terms of collaboration, or joint working, with EA. In contrast to the holistic view of EA, discussed in subchapter 3.3, reference architecture does not extend to propose a common methodology. Although the concept shows that there are a lot of benefits such as reusability and desires in higher quality products, as discussed in subchapter 5.5, some questions still remain unanswered. Although the respondents were familiar with the concept, they had different methodologies to work with reference architecture at different levels in the own organization.

From the theory chapter and the empirical data, it is clear that a reference architecture is often referred to in terms of software development, although there are suggestions that the concept can be utilized in other aspects of an organization. Reference architecture constitutes of different architectural patterns as described by the definition. However, it is well perceived that it also includes other core attributes found in EA, such as means of managing applications, roles, operational entities and visionary guidelines.

The patterns are not only strict applied to software development but can also, as discussed with the respondents, be used in a more contextual scope. When regarding the implementation and reusability of design patterns, there are a lot of paradigms describing the work. Often these paradigms show a more holistic approach and usually for a specific level in the organization. However, from the empirical data collection, it was not clear on how such reusability of patterns is utilized. Challenges of bringing this forward lies in the various roles and type of organization that were a part of the study. At a system level as described by the framework in chapter 2.3, reusability was very much present in the form of repositories in which the patterns could be found and reused when facing a similar problem. At the other levels, the repositories would be found as forms of best practices.

Discussing frameworks, the concept of reference architecture does not abide by a particular framework. The frameworks are simply a way of structuring the work of deriving a reference architecture, if needed. This means that the reference architecture itself has no dependencies in the form of architectural guidelines implicitly stated in the framework. It does, however, provide a contingency of showing a common thread between roles, systems, requirements etc. As stated in the first hypothesis, when developing a reference architecture, some principles of framework(s) can be adapted to the work being done and not fully implemented.

The common theme between the respondents was that reference architecture, as a concept, is implemented and developed in order to achieve a common way of working. It provides means of best
practices and know-how experience within a general area in which a challenge is re-faced. Although this was agreed upon, the simple definition of what a reference architecture is, proved to be more challenging to agree on.

9.2 The development of a Reference Architecture at Scania

Based on chapter 5, Theory on Reference Architecture and the empirical data collected, developing a reference architecture at Scania could bring means of improving workflow at different levels. Before developing a reference architecture, it should be clear on the granularity of the architecture and the purpose of that architecture. Since the purpose of this report was to provide Scania with insights of reference architectures and their means of aiding in system development, the conclusion drawn is that reference architectures has yet to reach a certain maturity level as discussed in chapter 1, Introduction. In order to start the work with bringing forward a reference architecture, different aspects should be considered. These aspects are:

1. Defining the purpose of the reference architecture in regards of the work in the organization.
2. Scoping the extent of work which the reference architecture should address.
3. Maintaining reusability requires a change in cultural aspects concerning work in the organisation.
4. Reference architectures should be developed with the help of different tools and frameworks in order to achieve consistency and continuation of using the reference architecture.

The abovementioned points are, based on the theory chapter, a crucial part of the development of a reference architecture. Although tool support was not discussed in this thesis, a computer program such as Sparx Enterprise Architecture (Sparx EA) lets the user build repositories, which could serve as the backbone of the reusability aspects concerning reference architectures. This means developing patterns as a part of the reference architecture and stored in the repository for further use. Sparx EA can also be used to show a comprehensive correlation of different patterns at different levels. Modeling patterns for different business functions in Sparx EA can provide measures of showing their relation to business applications at a system level.

The main conclusion for this formulation of what the repositorial function does, is that it can aid in the development of a reference architecture. It also provides means of locating where a demand of such architecture is needed and actions can be taken into developing one in that area.

As a final conclusion, in terms of developing a reference architecture at Scania, it is evident that it can bring advantages in the form of utilizing patterns that are proven and based on previous experiences. Governing an architecture is in itself a great challenge, but incorporating a maturity framework (TOGAF, DoDAF, Zachman etc.) can provide guidelines in the working with reference architectures.
9.3 Mapped Towards the Zachman Framework

From the interviews conducted, different conclusions could be drawn considering the positioning of the different organizations in relation to the concept on reference architectures. The basis of this chapter is relating to chapter 2, Preface. It is the possibility of further analysis of what has been said in the empirical data chapter. It also serves as basis for conclusions on the connections of reference architectures at different levels.

Since sufficient information was not provided by organization 3, it was preferred to not include the organization in the mapping towards the Zachman framework. The reason is that it was not enough empirical data to evaluate. The following pictures below depict the mapping towards the framework, as discussed in chapter 2.

In terms of organization 1, the respondent could not provide the general idea on reference architectures according to what was being said internally in the organization. The information provided was more likely based on assumptions of what was known on reference architectures and as a result, this meant that the idea of reference architectures was based on the local part of the organization that the respondent was active within. The explanation for this is that organization 1 only was a particular department of the company located in another country. However, from what was being said during the interview, the general idea of reference architecture seemed to be in a Systems level. Not much could be said about the lack of communication to the other levels, neither on whether or not such a reference architecture was existent.

In organization 2 however, there was more information given on reference architectures since the respondent was very familiar with the concept. Figure 19 shows that the concept of reference architecture very much was present in the organization. However, although reference architectures is present at a contextual, business and technological level, there was no direct communication between the levels. The respondent talked about the general idea of wanting to have reference architecture and its very many benefits, but it was also pointed out that there are challenges in making people accept the idea and working coherent.

The respondent pointed out that, although there might be a reference architecture at a technical level, this is not communicated to the top levels. Reference architectures developed at certain levels are often locally active. It is also clear that although reference architectures were discussed at all columns...
at a contextual and business level, the technical side only reflected reference architectures in means of where and how it was to be done.

Conclusively, this might have to do with the fact that reference architectures in organization 2 in general are not correlated. What distinguishes this organization from the others is that a reference architecture at system level never was mentioned.

Figure 19 Organization 2

The main challenge when drawing conclusions from the respondent from organization 4 was the interpretation of what was being said. Here, the respondent drew an alternative framework from which reference architectures could be positioned. Organization 4 is a consulting company helping others in the field of EA. From this, the respondent did not have anything to say relating reference architectures to the own organization.

The conclusion drawn in Figure 20 is simply what the respondent pointed out to be in relation to the field within EA in which it helps other organizations. Therefore, the picture above only shows where the respondent finds itself working in the field of EA and reference architectures. Although it was pointed out that organizations in general often think in terms of reference architectures at almost all levels, the main idea is rooted in the system and technical levels because it is more evident in terms of implementation there. It is more challenging to prove the results of the concept on reference architectures in higher levels due to it only being present in means of documentation and different guidelines.
Organization 5 depicted in Figure 21 was the least challenging of organizations to draw conclusions from. The respondent, being the vice president of the company, was well informed on reference architectures. It was clear to the respondent at what level they worked with reference architectures. There were conclusions drawn on lack of communication since it was clear that they only worked at a Systems level where the concept of reference architectures was very much present, although perceived according to a different definition. The organization was not dependent on visions or guidelines from other levels since it only delivered application solutions to the company it was working for. Internally in organization 5, there was different guidelines on how to build the applications in accordance with the Microsoft architectural guidelines[12].

![Figure 20 Organization 4](image)

![Figure 21 Organization 5](image)
The underlying reason for portraying organization 6 in accordance with Figure 22 was that the development and implementation of a reference architecture was locally formed. This means that there were no means of pursuing a reference architecture in accordance of any vision or guidance stated in a contextual level. The communication from other levels was not present at all, and in fact it was not even desired. Although, any project instantiation at this organization demanded a project instantiation approval, there were no means of stating that a reference architecture was developed. The respondent made it seem as if the reference architecture developed with time, and was in fact not called a reference architecture.

Only by discussing the concept did the respondent become more aware of that it in fact was a reference architecture. It was also clear that, what was developed only was existent at a system and technical level. However, these levels were constrained to only defining the reference architecture in means of “How” and “Where” in accordance to the columns defined in the picture below.

![Figure 22 Organization 6](image-url)
10 Discussion

Although there is much to be said about reference architecture, many challenges are present. The use of a reference architecture provides the means of organizing the work with enterprise architecture at different levels. Challenges reside in organizing the workflow between teams and at different levels in accordance with the framework discussed in subchapter 2.3. The extensive theory provided by different researchers within the area, all show that reference architecture provide the means of utilizing different frameworks in order to attain a certain entirety in the work with EA.

Although reference architecture itself is discussed as a separate entity, it is clear that it shares strong properties with EA, and should be seen as a complementing function in the work with EA. The main issues regarding this is to maintain a rigid concept in which it is applicable throughout different levels in system development. Since it is abstract in the way it is defined, reference architecture is dependent on scoping, whether this is in team projects or coherent at different levels. Only by deciding on a common idea on what the purpose of the reference architecture is, will an organization draw potentials and win returns on the invested idea. These returns will be in the form of speed, quality, coherence and reusability, although it will take time to see these results and the effects of the reference architecture.

In means of governance, there are many ways to go about in governing the reference architecture. Due to the fact that the architecture itself can be implemented at various levels and to different purposes, it is evident that governance is applied ad-hoc in relation to the scope of the reference architecture. Governance is very much dependent on how the organization itself is structured, hence an organization utilizing a centralized governance might differ from another organization in means of pushing a rate of change and conveying directives from visions to implementation. Of course, seemingly it raises more questions regarding when and where to actually develop a reference architecture. With the continuous growing organizational aspects and systems being developed, underlying challenges in managing these aspects are present. Therefore, when developing a reference architecture and dealing with these challenges, frameworks such as Zachman, TOGAF etc. can be utilized as tools in the progress of development of a reference architecture.
11 Validity & Reliability

To ensure the validity and reliability of the results, the interviews were held independently. The answers recorded during the interviews were written down exactly as the respondents answered the questions. The survey was created in Google Docs as a survey form that was only accessible to the writer of the report. This means that the respondents of the survey that was sent out were not influenced by answers provided by other respondents.

Furthermore, it was considered that few interviews that were representative in the research area of the report was better suited; This provided the report with a qualitative empirical data rather than a quantitative one. Although a quantitative data collection can be performed, it was assessed that the time needed for completing one was out of the scope in this thesis. The survey however was planned to complement the interviews with quantitative data but since no more than five organizations answered it can only be considered as qualitative.

Also, in order to ensure that as much information on reference architectures was collected, the organizations were choosen at different levels; Business, System, Technical. From this, it would be easy to assess what the common conception is on reference architecture at the different levels.

12 Future Work

In regards of continuing the research on the concept of reference architectures, means of evaluating these is of great importance. Future work could reflect on the challenges which reside in evaluating reference architectures considering reusability, validity and implementation. Future work could also reflect on the challenges that lie in developing a generic reference architecture, and how it would come about to actually implement it at a desired level; Business, System, Technical… Challenges reside in actually observing reference architectures in their whole. This implies finding reference architectures that serve as blueprints in an organization and how well they are being used. Conducting a case-study in an organization could help in determining how well a reference architecture is used as an underlying asset. This means researching on the usability of reference architectures as means of communication and using it as a tool. What repositorial functions are supporting it? What kind of life-cycle management does the reference architecture have?

Complementing information about reference architectures with these challenges will provide an approach into the implementation of reference architectures. Means of fulfilling future work could be made by using this thesis as an underlying work to derive further intricate questions to be answered. The questions for the survey and the interviews could be revised and altered in order to attain a higher confidence in the hypotheses stated.
13 Appendices

13.1 Questionnaire

“Reference Architecture(s)” in this questionnaire is shortened to RA for convenience. This survey will serve as basis for further questions for in-depth analysis of the roles of design patterns in RA and system architecture as a whole.

The survey consists of two parts. The first part provides questions on a more general level on reference architecture. The second part consists of questions for a more in-depth analysis on the subject.

General information

1. Name of company you work for?
2. In what department do you work?
3. What is your role in the work with enterprise architecture (EA)?
4. How long have you been working with EA?

Reference Architecture PART I

1. Is your company implementing a RA/reference framework in the work with enterprise architecture? What is the name of the RA?
2. What does RA mean/represent for your company?
3. Did your company develop its own RA? If no, then what RA is implemented and who developed it? If yes, what is the role of this person in your company that developed it?
4. How is your RA structured/organized i.e. what architectural style/pattern does your RA adapt? (By architectural style/pattern, I’m referring to the Microsoft application architecture guide (layered style, SOA…)
5. What are you hoping to achieve with implementing RA in your company? What is the scope and complexity of the RA?
6. Since the term RA is widely used in software development, does your company implement the core principles of RA in other functions of the business? (Business, Technical, …)
7. Have you witnessed any changes in minimizing costs and maintenance requirements due to the implementation of RA?
8. How is the RA maintained in regards to support from frameworks? (Maturity frameworks such as Zachman framework, DoDAF, TOGAF etc.)
9. How is the RA validated?
10. At what level is the RA being utilized? Technical/Business/ …
Reference Architecture PARTII

1. Does the RA have any requirements?

2. What requirements are there on the structure of the RA?

3. Does the RA point to any repository function?

4. How often is the RA updated/revised?

5. At what level are patterns utilized? (Business, Application, Technical…)

6. How are architectural patterns being packaged for reuse?

7. What requirements are there on the patterns that are utilized?

8. What tools are being used for the development and communication of the RA? (Sparx EA, Visio…)

9. What roles are included in the work of the RA?

10. If question above is answered, at what levels do you find these roles?

11. What type of methodology is used for communicating the RA?

12. Since abstraction of patterns is important for reuse, at what level would you say the abstraction is made: 1, 2, 3, 4, 5 (1 being not so abstract and 5 being highly abstract)
13.2 Survey

Link to survey:
https://docs.google.com/spreadsheet/viewform?rm=full&formkey=dFk0b0d4S1VxYVlk4c2xZeDQ0eVM2a3c6MQ#gid=0
13.3 Zachman framework
13.4 Project Model
14 References

[8] P. Reed, Reference Architecture: The best of practices, in, Jackson-Reed, Inc, IBM.
[26] b.w.i.t.d.w.b.m. Fakultät für Informatik Technische Universität München, Enterprise Architecture Patterns, in.
Keywords for Literature Search

- Reference architecture* AND Enterprise
- Software architecture AND Reference architecture
- Enterprise architecture* AND Reference architecture
- Reference model*
- Reference AND (Pattern* or model)
- Reference application* AND model*
- Software reusability AND Pattern*
- Reference architecture* AND Reusability
- Reference architecture* AND (Pattern* OR model) AND Reusability