Comparing Security Architectures
Defining and Testing a Model for Evaluating and categorising security architecture frameworks

Rob van Os
2014

Master of Arts (60 credits)
Master of Science in Information Security

Luleå University of Technology
Department of Computer Science, Electrical and Space Engineering
Comparing security architectures

Defining and testing a model for evaluating and categorising security architecture frameworks

Magister thesis, A7007N

Name: Rob van Os
Student ID: robvan-2
Date: 03.05.2014
Table of Contents

I. LIST OF ABBREVIATIONS .......................................................................................................................... III

II. ACKNOWLEDGEMENTS ........................................................................................................................ IV

1 MANAGEMENT SUMMARY ......................................................................................................................... 1

2 INTRODUCTION ........................................................................................................................................... 2

2.1 ZACHMAN INFORMATION SYSTEMS ARCHITECTURE ........................................................................... 2

2.2 ENTERPRISE ARCHITECTURE ............................................................................................................... 2

2.3 SECURITY ARCHITECTURE ...................................................................................................................... 3

2.4 INFORMATION SECURITY CONCEPTS .................................................................................................... 3

2.5 ARCHITECTURAL CONCEPTS ................................................................................................................. 3

3 RESEARCH QUESTION ............................................................................................................................... 7

3.1 THESIS STEPS .......................................................................................................................................... 7

4 DEFINING A MODEL ................................................................................................................................. 10

4.1 INVENTORY OF SECURITY ARCHITECTURE FRAMEWORKS ................................................................ 10

4.2 ARCHITECTURES IN THE INVENTORY ..................................................................................................... 14

4.3 A MODEL FOR COMPARISON .................................................................................................................. 17

5 ANALYSING THE ARCHITECTURE FRAMEWORKS ...................................................................................... 32

5.1 SABSA .......................................................................................................................................................... 32

5.2 O-ESA ..................................................................................................................................................... 39

5.3 OSA .......................................................................................................................................................... 44

6 ASSESSING THE ARCHITECTURES USING THE MODEL ............................................................................. 52

6.1 SABSA .......................................................................................................................................................... 52

6.2 O-ESA ..................................................................................................................................................... 55

6.3 OSA .......................................................................................................................................................... 60

7 SUMMARY AND CONCLUSIONS .................................................................................................................. 65

7.1 SUMMARY ................................................................................................................................................ 65

7.2 OVERVIEW .............................................................................................................................................. 65

7.3 CONCLUSIONS ....................................................................................................................................... 67

7.4 CRITICAL NOTES ABOUT THE COMPARISON ....................................................................................... 68

7.5 RESEARCH CONTINUATION .................................................................................................................... 69

8 REFLECTIONS AND A FINAL WORD ......................................................................................................... 73

ANNEX A: OVERVIEW OF TABLES AND FIGURES ....................................................................................... 74

A.1 TABLES .................................................................................................................................................... 74

A.1 FIGURE .................................................................................................................................................... 74

ANNEX B: LITERATURE AND REFERENCES ..................................................................................................... 75

B.1 LITERATURE ........................................................................................................................................... 75

B.1 REFERENCES .......................................................................................................................................... 75
## I. List of abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5W1H</td>
<td>What, where, who, when, why, how</td>
</tr>
<tr>
<td>ADM</td>
<td>Architecture Development Methodology</td>
</tr>
<tr>
<td>BIT</td>
<td>Business, Information, Technology</td>
</tr>
<tr>
<td>BITS</td>
<td>Business, Information, Technology, Security</td>
</tr>
<tr>
<td>BP</td>
<td>Border Protection</td>
</tr>
<tr>
<td>CDSA</td>
<td>Common Data Security Architecture</td>
</tr>
<tr>
<td>CIA</td>
<td>Confidentiality, Integrity, Availability</td>
</tr>
<tr>
<td>CG IAF</td>
<td>Cap Gemini Integrated Architecture Framework</td>
</tr>
<tr>
<td>CMMI</td>
<td>Capability Maturity Model Index</td>
</tr>
<tr>
<td>COBIT</td>
<td>Control Objectives for Information and Related Technology</td>
</tr>
<tr>
<td>DODAF</td>
<td>Department of Defense Architecture Framework</td>
</tr>
<tr>
<td>E2AMM</td>
<td>Extended Enterprise Architecture Maturity Model</td>
</tr>
<tr>
<td>EA</td>
<td>Enterprise Architecture</td>
</tr>
<tr>
<td>EISA</td>
<td>Enterprise Information Security Architecture</td>
</tr>
<tr>
<td>ESA</td>
<td>Enterprise Security Architecture</td>
</tr>
<tr>
<td>FEAF</td>
<td>Federal Enterprise Architecture Framework</td>
</tr>
<tr>
<td>HTTP(S)</td>
<td>HyperText Transfer Protocol (Secure)</td>
</tr>
<tr>
<td>IdM</td>
<td>Identity Management</td>
</tr>
<tr>
<td>IEC</td>
<td>International Electrotechnical Commission</td>
</tr>
<tr>
<td>IFEAD</td>
<td>Institute for Enterprise Architecture Developments</td>
</tr>
<tr>
<td>ISO</td>
<td>International Organization for Standardization</td>
</tr>
<tr>
<td>ISRM</td>
<td>Information Security and Risk Management</td>
</tr>
<tr>
<td>ISSS</td>
<td>Information Security Society Switzerland</td>
</tr>
<tr>
<td>IT</td>
<td>Information Technology</td>
</tr>
<tr>
<td>ITIL</td>
<td>Information Technology Infrastructure Library</td>
</tr>
<tr>
<td>MoSCoW</td>
<td>Must Have, Should Have, Could Have, Won’t have</td>
</tr>
<tr>
<td>NAC</td>
<td>Network Applications Consortium</td>
</tr>
<tr>
<td>NIST</td>
<td>National Institute of Standards and Technology</td>
</tr>
<tr>
<td>O-ESA</td>
<td>Open Enterprise Security Architecture</td>
</tr>
<tr>
<td>OSA</td>
<td>Open Security Architecture</td>
</tr>
<tr>
<td>PDCA</td>
<td>Plan-Do-Check-Act</td>
</tr>
<tr>
<td>PDP</td>
<td>Policy Decision Point</td>
</tr>
<tr>
<td>PEP</td>
<td>Policy Enforcement Point</td>
</tr>
<tr>
<td>PKI</td>
<td>Public Key Infrastructure</td>
</tr>
<tr>
<td>PMA</td>
<td>Policy Management Authority</td>
</tr>
<tr>
<td>RBAC</td>
<td>Role Based Access Control</td>
</tr>
<tr>
<td>SA</td>
<td>Security Architecture</td>
</tr>
<tr>
<td>SABSA</td>
<td>Sherwood Applied Business Security Architecture</td>
</tr>
<tr>
<td>SAML</td>
<td>Security Assertion Markup Language</td>
</tr>
<tr>
<td>SQuaRE</td>
<td>Software product Quality Requirements and Evaluation</td>
</tr>
<tr>
<td>SSE-CMM</td>
<td>System Security Engineering Capability Maturity Model</td>
</tr>
<tr>
<td>TOGAF</td>
<td>The Open Group Architecture Framework</td>
</tr>
<tr>
<td>TSA</td>
<td>Target Security Architecture</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modelling Language</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>
II. Acknowledgements

I would like to take the opportunity to thank a number of people who have helped in the process of writing this thesis.

Dr. Abbas Shahim, partner Information Security and Risk Management (ISRM) at Atos Consulting. Thank you for your critical insights into the definition of research question, the proper scientific founding of the contents of this thesis and providing direction to the content and results of the thesis.

Dr. Dan Harnesk, associate professor, division of Information Systems Science. Thank you for your critique on quality model and pointing out the need for clearly stating the reason why a comparison of security architectures is required. Also, thank you for enabling me to submit the research proposal long after the initial deadline.

Hans van Drunen, senior enterprise architect at Atos. Thank you for sharing your expertise on enterprise architecture and the quality evaluation of IT solutions in general.

Pascal de Koning, MSc., senior business consultant at KPN. Thank you for sharing your expertise on the SABSA enterprise security architecture framework and the integration project between the TOGAF enterprise architecture and the SABSA enterprise security architecture. Also, thank you for taking the time to comment on the quality attributes.
1 Management Summary

Security Architecture is a term that is applied to a wide variety of activities, each different in the level of detail and the organisational level at which it acts. This makes communication about Security Architecture between different stakeholders difficult. The same issue also permeates to security architecture frameworks. Because the activities are different, frameworks for those activities can differ greatly as well, both functionally and in quality.

In order for organisations to make an informed decision on the right security architecture framework that matches the organisational needs, a means for comparison is required. In this thesis, a model is proposed for comparing security architectures. The model is then applied to a selection of security architectures. The result provides insight into the security architectures, the purpose that they serve and the quality of the architecture. This provides the organisation with sufficient information to make an informed decision.

The thesis follows a 3-step approach to obtaining the required results. In the first step, a model is created by examining different architecture frameworks and establishing the common elements that they have. Additionally, qualitative features are defined for qualitative evaluation of the architecture frameworks. In the second step, the frameworks are assessed to determine the methods and mechanisms that are used. In the third and last step, the results from the assessment performed in the previous step are mapped to the model. These results, in their common format, can then be compared to determine the differences between the architecture frameworks.

The model created in this thesis can be used for evaluating and comparing security architectures as it provides a common format for describing the features and quality attributes of the architecture frameworks. Because the model acts on two different layers (a high-level taxonomy and a low-level attribute list), the model can be used for both quickly establishing the right selection of candidates and performing a more in-depth comparison of architecture frameworks. The main problem with the model is the amount of time it takes to perform an extensive evaluation. When using the model in practice, it is therefore recommended to first perform the high-level comparison based on the taxonomy and creating an appropriate selection of candidates. Subsequently, this selection of candidates can then be subjected to the more time-consuming extensive evaluation.

This thesis starts out with an introduction to the topic of information architecture and security architecture. The introduction also outlines a number of concepts that are relevant to the thesis. In the following chapter, the research question is outlined and the detailed 3-step approach to answering the question is provided. Chapters 4, 5 and 6 each deal with one of the steps as described in chapter 3. Chapter 7 provides a brief summary and describes the main conclusions from this thesis. The chapter also contains some critical notes and areas of continued research. The thesis concludes in chapter 8 with reflections and a final word.
2 Introduction

Architecture deals with design. When designing a physical structure, architecture deals with sizes, layouts and functions for each individual area and the building as a whole, while taking into considerations the use of such areas and structural requirements that will ensure the continued existence of the structure. The goal is to create a structure that serves the purpose for which it will be used. To reach this goal, the architect must consider regulations that apply to the building and must integrate the wishes of the owner as well as the requirements and prerequisites of the builders that must execute the work. Each of these stakeholders views the building in a different way; it is the architect’s job to create designs that represent the viewpoints of all stakeholders and unify the complex building blocks that make up the structure.

Just to ‘real-world’ architecture, information architecture deals with the same concepts. In essence, it has similar goals, but an entirely different context. The need for information architecture arose from the increasing complexity of the IT landscape. With increasing IT complexity, it became more difficult to obtain insight into all components that make up the IT landscape and the interfaces between those components. It also made projects that deal with integration into the infrastructure more costly as unforeseen situation could arise due to lack of overview. Especially for security, this lack of overview and formalization leads to a high potential for gaps in the infrastructure.

2.1 Zachman information systems architecture

John A. Zachman recognised the need for architecture in the information systems domain and decided to analyse the regular architecture to create an equivalent for information systems. This resulted in the publication “A framework for information systems architecture” [1]. In this publication, Zachman explores several viewpoints (5 in total) that represent different layers of the architecture framework. Within each layer, Zachman addresses several key elements called descriptions. These initial descriptions are material (what), function (how) and location (where). Zachman also argues that there are three other descriptions: people (who), time (when) and motivation (why). These descriptions are further described in the subsequent publication “Extending and formalizing the framework for information systems architecture” [2]. With these two publications, Zachman provides a taxonomy that provides for 30 unique cells. Each cell is a combination of a column (a description that has its own model) and a row (a viewpoint). This information architecture was later described by Zachman to be an enterprise architecture framework rather than an information systems framework [3].

2.2 Enterprise architecture

The Zachman framework was an early attempt at formalizing an Enterprise Architecture (EA) framework. The term EA is used for initiatives that strive to support company decision making by building a bridge between organisational goals and the steps required to achieve those goals. Additionally, EA attempts to describe the relationship between business, applications and infrastructure.

The Zachman framework or taxonomy itself has been subjected to different versions that have added new features and improved on other features through the years. Besides the Zachman framework, many other methodologies and framework have also been created to support the EA process. All these methodologies and frameworks serve the same purpose, but differ in their approach and their maturity level. While some of these frameworks are well described and provide concrete methodologies for development, others merely provide a framework and briefly describe its implementation. This makes a comparison between Enterprise Architectures difficult.
2.3 Security architecture

Similar to Enterprise Architecture, Security Architecture (SA) is also a term applied to different kinds of activities. The difference between comparing EA and SA is that EA, at the least, deals with architecture at the same level: the enterprise level. Security architectures may be applicable at the enterprise level, but also at the very detailed level of security considerations for solutions (which is usually called ‘design’, rather than architecture). This makes SA even harder to compare. This thesis provides a means for assessing and classifying security architectures. The research question will be further addressed in chapter 3. First, the concepts and terms used in this thesis will be explained.

2.4 Information security concepts

This thesis presumes the reader has a basic understanding of information security. Therefore, security related concepts will not be explained in detail. However, to make sure the goals of information security architecture are well understood, a brief definition of basic concepts is provided.

Information security is usually described using three basic attributes [4]:

- **Integrity.** FIPS PUB 199 describes integrity as: “Guarding against improper information modification or destruction, and includes ensuring information non-repudiation and authenticity”.
- **Confidentiality.** FIPS PUB 199 describes confidentiality as: “Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information”.
- **Availability.** FIPS PUB 199 describes availability as: “Ensuring timely and reliable access to and use of information”.

These concepts are often called the CIA-triad. Other concepts are often included, such as authenticity (verification of genuineness), accountability (traceability of actions to unique persons) and the related concept non-repudiation (no deniability of having performed actions).

Security architecture aims at ensuring these basic concepts and principles of information security are accurately addressed in the context of the enterprise at different layers.

2.5 Architectural concepts

To provide a frame of reference for the thesis, the concepts relevant to architecture will be explained in this paragraph. This explanation serves as a glossary for reference in the thesis.

2.5.1 Architecture

Architecture in itself is not exclusive to the field of information technology. For this reason, the term information architecture was coined for this field of expertise. In this thesis, the term ‘architecture’ refers to information architecture. Similarly, other terms that amend the term architecture (enterprise architecture and security architecture) limit themselves to information systems.

In the broader sense of the term, architecture the complex or carefully designed structure of something. It is also “the art or practice of designing and constructing [buildings]” [5]. This means architecture is both a noun (structure) and a verb (practice). In this thesis, the emphasis will be on architecture as a noun (meaning: the result); although the process of architecture is also subject to discussion.

2.5.2 Enterprise architecture

Enterprise Architectures have briefly been described in this chapter. The full definition provided by Gartner for EA is: “EA is a discipline for proactively and holistically leading enterprise responses to disruptive forces by identifying and analysing the execution of change toward desired business vision and outcomes. EA delivers value by presenting business and IT leaders with signature-ready
recommendations for adjusting policies and projects to achieve target business outcomes that capitalize on relevant business disruptions. EA is used to steer decision making toward the evolution of the future state architecture\(^6\). The primary goals of enterprise architecture are effectiveness, efficiency, agility and durability \(^7\).

EA is a topic that has seen much attention since the beginning and middle of the 1990’s. This attention has accumulated to a considerable amount of frameworks. The list of frameworks consists of architecture frameworks that include:

- The Open Group Architecture Framework (TOGAF).
- The Zachman framework for information architecture.

Besides generic frameworks, organisational and localised specific frameworks exist that include:

- Department of Defense (DoD) Architecture Framework (DoDAF)
- Federal Enterprise Architecture Framework (FEAF)

Some of these frameworks have yielded best-practice approaches that are described as methodologies for using the framework.

Traditional architectures can be extended to address security requirements, but in most cases, information security is not formally addressed as part of the framework. This means information security is built on top of the architecture rather than integrated into the architecture. In more recent years, a demand has risen for incorporating information security into enterprise architecture frameworks. This has led to several initiatives that have yielded security architecture frameworks that aim to integrate security requirements into architecture frameworks. These initiatives vary in background, maturity, completeness of vision and integration levels. Additionally, some of these frameworks include methodologies for guidance towards implementation while other frameworks do not address implementation. Instead, these frameworks only provide attributes or patterns for security architecture.

### 2.5.3 Information architecture

Information Architecture has no formal definition. In some Enterprise Architectures, such as TOGAF, Information Architecture is a domain in the architecture framework \(^8\). In general, it can be stated that Information Architecture describes the relationships and coherence between different information systems. This way, information exchange between systems can be mapped and the information being exchanged can be classified.

### 2.5.4 Security architecture

Security Architecture has also been briefly described in the previous part of this chapter. As indicated, Security Architecture can describe activities at different levels of abstraction. This makes it difficult to provide a holistic definition. In general, Security Architecture describes and provides guidance for the protection of information. This guidance could be provided at the business level, at the information systems level or at the technology level. This thesis will go into Security Architecture in more detail. Different kinds of security architecture and the level at which they act will be examined and classified.

### 2.5.5 Reference architecture

Reference Architecture suffers from similar problems as security and enterprise architecture for its definition. The Department of Defense has found similarities in the definitions and goals for Reference Architectures. After comparing several definitions and derived definitions they state that “the primary purpose of a Reference Architecture is to guide and constrain the instantiations of solution architectures” \(^9\). In other words, the Reference Architecture provides a template for solutions. These templates do not provide a complete description of the solution but creates a frame of reference that outlines the basics and boundaries for the solution. Reference architectures will not
be a primary subject of this thesis, but it is important to note that every kind of architecture has their own Reference Architecture.

2.5.6 Architectural patterns
Architectural patterns, also referred to as architectural styles “improve[s] partitioning and promote[s] design reuse by providing solutions to frequently recurring problems” [10]. These patterns are point solutions that can be implemented in a standardised way. Like Reference Architecture, architectural patterns provide a mechanism for standardisation while retaining flexibility. The main difference is that Reference Architectures provide a full architectural template, while architectural patterns are specific for a single solution and thus have more detail.

2.5.7 Architecture domains
EA deals with architecture at different levels of the organisation. Within these levels, the architecture effort is further subdivided into specialised focus points or views. These views are considered to be architecture domains. For example, in the TOGAF 9.1 enterprise architecture framework, the following architecture domains are differentiated between [8]:
- Business
- Data
- Application
- Technology

The exact domains differ between enterprise architectures, but there is a strong overlap of architectural domains between enterprise architecture frameworks. The ‘data’ and ‘application’ architectures are often combined into a single term: Information systems architecture.

2.5.8 Artefacts
Architectural artefacts are products resulting from the architecture process. These artefacts are created to “describe a system, solution, or state of the enterprise” [11]. Examples of artefacts that are created in the architecture process are:
- Matrices
- Models
- Catalogues
- Diagrams
- Policies

Similar documents can also be artefacts.

2.5.9 Architecture views and viewpoints
Architecture views aim to provide a representation of a system. This representation is usually provided to a particular stakeholder. The stakeholder has concerns or crucial interests for the system itself and the viewpoints should provide the stakeholder with information on those concerns and interests. The difference between views and viewpoints is that a view provides the representation, while the viewpoint defines the way in which the view is outlined to the stakeholder. In other words, a single view can be created that has multiple viewpoints that allow the correct representation for each stakeholder.

2.5.10 Other concepts
In the field of architecture, other concepts such as frameworks, taxonomies, models and methodologies are commonly used to describe a part of the architecture or the architecture in itself.

Framework
The ISO/IEC/IEEE 42010-2011 Systems and software engineering Architecture description provides a number of definitions for terms related to architecture. This standard defines an architecture
framework as “conventions, principles and practices for the description of architectures established within a specific domain of application and/or community of stakeholders” \cite{12}. This means that a framework must:

- Describe an architecture
- Provide conventions, principles and practices
- Provide viewpoints and describe stakeholders for each viewpoint

The architecture itself must be specific in its use. The standard further provides a list of requirements that must be met for the framework to be accepted as a framework. These requirements will be used to determine briefly if the architecture framework has been classified correctly.

In this thesis, the terms architecture and architecture framework are used interchangeably. This is because architecture frameworks rarely describe themselves as being a framework. Rather, the generic term ‘architecture’ is applied. In chapters 4, 5, 6 and 7, the term ‘architecture’ can be replaced with the term ‘architecture framework’ when the text refers to one of the architecture frameworks that is being analysed and evaluated.

**Taxonomy**

A taxonomy is used for classification of entities \cite{13}. The taxonomy defines rules to this classification process. The process of classifying itself using the defined rules is also called ‘taxonomy’. In architecture frameworks, taxonomies can be used to classify artefacts or activities.

**Model**

In the context of this thesis, the term ‘model’ actually applies to information models. Information models provide a schematic description of a system or design in which all actors or stakeholders are outlined and the relationships between those actors is explained. Models are commonly defined using a standardised modelling language (such as Unified Modelling Language (UML), or more specifically for EA: ArchiMate). Lee \cite{14} provides definition of information models that is in line with this explanation: “An information model is a representation of concepts, relationships, constraints, rules, and operations to specify data semantics for a chosen domain of discourse.”

**Methodology**

A methodology is a collection of rules, guidelines, methods and principles that provide a means towards reaching the desired end state \cite{15}. The methodology outlines all the tools that are available to the architect and details the inputs and outputs required. In more complex methodologies, multiple steps can be defined and the products that need to be delivered in each of those steps. These products accumulate to the desired end result.

**Process**

Similar to methodologies, a process uses inputs and provides outputs. Where a methodology provides a mere description of rules and guidelines, a process is very strict in defining the exact steps that need to be taken. In essence, a process is more similar to an instruction guide. A process can be part of a methodology.

**Practice**

A practice is a tool that can be implemented to resolve a particular problem. Usually, the term best-practice is used to describe a group of such tools that have been verified to work as intended. The Information Technology Infrastructure Library (ITIL) is an example of such a (best) practice: a set of tools (being service management processes) that can be implemented as a whole or in part to standardise service delivery. Best-practices add to efficiency and effectiveness \cite{16}.
3 Research question

Security architecture suffers from the same issue that faces enterprise and information architecture: the term is used for a wide variety of activities, each different in the level of detail and the organisational level at which it acts. This issue makes it difficult to make an informed decision for a specific architecture or architecture framework. An organisation may choose a framework based on what information may be available to them; for example, past experiences from architects. However, to maximize added value for the organisation and to ensure that the chosen framework can be implemented in the organisation and provides a solution to the organisations issues, they should be able to make an informed decision on the most appropriate architecture framework. It is vital that the security architecture either simplifies or complements existing architectures and does not make them more complex.

Security architectures have a strong focus on requirements engineering. This requirements engineering can act at several levels in the organisation, from the business level (dealing with business requirements) to the technical level (the technical requirements). If a choice is made for an architecture that does not match the organisational requirements and organisational capabilities, negative effect can come from the architectural effort at both the short and the long term. Short term negative effects include situations where an implemented solution does not match the organisational needs because the security architecture was not capable of driving the implementation process effectively. Long term negative effects include situations where the organisation is not capable of maintaining the security architecture, either through lack of expertise and support or through lack of integration with the existing architecture process. This situation can lead to misalignment of enterprise and security architecture efforts and could lead to a high level of IT complexity. Both short term and long term negative effects cost time and money to the organisation and can lead to damaging of the reputation of the IT department. Information architecture aims to add value to the organisation and additionally aims to decrease complexity. If the opposite effect is obtained, the reputational damage to the IT department can be significant.

To prevent negative effects occurring from an implementation of a security architecture that does not match the organisation, it is vital for an organisation to first determine the requirement it has for security architecture. This enables the organisation to make a founded choice for a security architecture that matches those requirements. However, making founded choices requires the possibility of comparing security architectures using some sort of standardized model, which is currently not available.

The research question that this thesis will try to answer is the following: “How can security architecture frameworks be compared to one another?”

This thesis provides a means for such a comparison by defining a model. The model will be used to evaluate three different security architectures, so that the model’s ability to provide specific results for each architecture under evaluation can be confirmed.

3.1 Thesis steps

In order to answer this question, it is important to create a model for evaluating security architecture and to assess whether this model is capable of providing the appropriate results. This thesis uses three steps to answer the research question:

- Step 1: defining a model. A model will be created for evaluating security architecture frameworks.
- Step 2: analysing the architecture frameworks. The next step is to analyse security architectures to determine how they work and what features they have.
3.1.1 Step 1: Defining a model

As indicated, the very first step is establishing the different kinds of security architectures that can be differentiated. This step does not involve detailed analysis of security architectures, but uses the description provided by the architecture. This description should clearly state that the architecture considers itself to be a security architecture.

After the inventory is completed, a set of features will be defined that will be used for the comparison. These features represent qualitative and descriptive features of security architectures. An example of such a feature is the ability to integrate into existing frameworks. This list of features will be created before the details of the security architectures are investigated, because they need to be independent of any existing security architecture. The features can be derived from the basic models used in architecture frameworks and from quality assurance methodologies.

3.1.2 Step 2: analysing the architecture frameworks

To limit the scope of the thesis, a limited number of security architectures frameworks will be analysed and evaluated using the model created in step 1. A selection has been made that includes architectures that bear the name ‘enterprise security architectures’, but are not always perceived as such. The following security architectures will be analysed in the thesis:

- **SABSA.** The Sherwood Applied Business Security Architecture (SABSA) is an architecture framework and methodology for creating a security architecture that is built on business requirements and a risk profile of the organisation. The methodology describes the use of the risk analysis method and describes how to use the framework to create a security architecture that guides security principles from business requirements to IT components. SABSA follows closely the Zachman framework \[17\]. This framework will not be separately analysed, but reviewed only as integrated part of SABSA. A whitepaper has been created that describes how SABSA and a commonly used enterprise architecture (TOGAF) can be integrated. This paper will be considered as integration with existing architectures is an important driver for decision making;

- **O-ESA.** The Open Enterprise Security Architecture (O-ESA) is developed by The Open Group and provides, according to the subtitle of the publication: “a framework and template for policy-driven security” \[18\]. The architecture does not provide a specific enterprise security architecture and also does not describe full a methodology for implementations. Instead, it addresses areas of focus for security architects and provides guidance for integration;

- **OSA.** The Open Security Architecture (OSA) provides a free-to-use framework that can be applied to create a security architecture. According to their site, the OSA vision is to “distill[s] the know-how of the security architecture community and provide[s] readily usable patterns for your application” \[19\].

To analyse the above mentioned architectures, the following main sources will be used:


Other supporting sources will include whitepapers and articles. However, the above mentioned sources are considered leading for analysis. If information on attributes is not described in leading documentation, the supporting sources will be used. The fact that these are supporting sources will be considered in the analysis.
3.1.3 Step 3: Assessing the architecture frameworks using the model

The last step for the thesis is using the analysis from the previous step to fill in the requirements and qualitative attributes in the model. This will provide an overview of strengths and weaknesses of each of the security architecture frameworks, and provide detailed insight into differences between security architecture frameworks for specific features.

The assessment carried out in this step is a qualitative analysis where a set of quality attributes is used to determine the quality of the architecture. The analysis will also comment on the perceived completeness of the quality feature. However, there will not be a detailed qualitative analysis in which each attribute is provided with a ‘grade’ that indicates its quality. The list of features and the set of attributes to perform a qualitative analysis are combined to create a model for evaluation and classification. This model is then used in step 2 for analysis of security architectures.

With the completion of this step, the comparison is complete. The resulting matrix will contain firmly founded statements. The thesis will end in a conclusion and discussion in which the model and the contents of the model are discussed, as well as the applicability of the model and potential next steps for further research.
4 Defining a model
Defining a model is the first step that needs to be taken in order to answer the research question outlined in the previous chapter. As described, this step is comprised of two activities:

- Creating an inventory of security architectures
- Creating a model

These activities are described in detail in this chapter.

4.1 Inventory of security architecture frameworks
When examining architectural frameworks such as TOGAF, several architectural domains are distinguished. These domains are often represented as separate architectures. This means that the following architectures can be differentiated based on TOGAF and similar frameworks:

- Business architecture
- Information-systems architecture, subdivided into:
  - Application architecture
  - Information (or data) architecture
- Technology architecture

These layers are often called the ‘BIT’ layers of EA, after the first letter of the domains. When examining technology architecture, further subdivision is possible. This subdivision is based on the common differentiation made between applications, systems and network. Since TOGAF itself can be considered an enterprise architecture, that can be added to the list of architectures as well.

4.1.1 Security architecture addressed in other viewpoints
All regular architectures have a security component in which architects deal with the security aspects of the architecture domain. A security architect or a group of security architects deal(s) with each of the mentioned architectural layers. In essence, this means that security architecture is addressed in each of those layers and can be performed separately by different architects at each layer. This also means that those activities can be considered separate architectures, thus creating the following list of security architectures:

- Enterprise security architecture, consisting of:
  - Business security architecture
  - Information Systems security architecture, subdivided into:
    - Application security architecture
    - Information (or data) security architecture
  - Technology security architecture, subdivided into:
    - System security architecture
    - Network security architecture
    - Application security architecture

The list above represents a hierarchical list, with enterprise security architecture as the highest level security architecture. Figure 1 shows the hierarchy of security architectures based on the basic EA hierarchy.
4.1.2 Security architecture as a separate domain

Originally, SA was not considered to be a separate part in any architecture framework. Instead, each architecture domain contained a security element. The security architect would address security requirements as a separate activity for each domain. For example, for TOGAF 9, a whitepaper was issued that described the activities that a security architect needed to carry out in each part of the Architecture Development Model (ADM) lifecycle. This lifecycle covers requirements management and business alignment as well as all previously mentioned architecture domains. Additionally, the implementation and change management phases were also addressed to provide a full overview of the tasks that a security architect was burdened with within the scope of the TOGAF ADM. Figure 2 shows the addressing of security architecture at each of the BIT domains (Source: https://en.wikipedia.org/wiki/File:Huxham_Security_Framework.jpg, retrieved 16-7-2013).
This view on security architecture changed over the years, as it became clear that this approach to security architecture failed to provide a clear overview for the security architects; especially in more complex architecture. According to Gartner [21], three possibilities exist for addressing security architecture in the enterprise architecture effort:

- Security permeates EA. This is the traditional approach as just described where security is address in each of the (BIT) layers of enterprise architecture.
- Security as a domain in technology architecture. This solution limits security architecture in its scope and makes security an exclusive ‘IT-issue’. This approach can have advantages in certain situations, especially when technical architecture is the main architecture being developed.
- Create a separate security viewpoint. Essentially, this approach entails the creation of a new domain in addition to the traditional BIT domains to form an EA with 4 layers: business, information systems, technology and security (BITS). This security domain is called the Enterprise Information Security Architecture (EISA). The main advantage is that this provides security architects with their own view and more certainty that they have a complete overview of security artefacts. The potential problem with this approach is the integration aspect. Security was generally seen as a specialist task and carried out separately from the rest of the enterprise architecture effort.

Figure 3 shows the ‘create a separate security viewpoint’ approach with security as a separate domain within EA (source: Gartner, “integrating security into the enterprise architecture framework”, 2006).

![Figure 3: security as a separate viewpoint in EA](image)

The introduction of a new domain is based on a previous whitepaper by Gartner. The main focus of that paper was the incorporation of security architecture into the EA architecture process, thereby identifying the need for a separate domain for optimum integration. This involves both the architecture development process as well as the architecture maintenance process [22]. When considering this setup with a separate domain, enterprise security architecture should not be positioned in the top level of the hierarchy. Instead, the highest level should remain enterprise architecture, and (enterprise) security architecture should be in the domain layer of the model. Figure 4 shows the hierarchy with security architecture as a separate domain.
Figure 4: security as a separate domain in EA

The positioning of enterprise security architecture as a separate domain is not widely adopted. The latest released version of TOGAF, published in 2011, still uses the three traditional BIT layers. The basic outline of the next release of TOGAF (TOGAF next), does not show security architecture as a separate domain or architecture either. In contrary to the post-release outlining of security architecture activities performed for TOGAF 9, there is a strong focus on integrating security into the TOGAF next framework. This should provide for a more unified approach. In addition to the next version of TOGAF not having a separate security domain, the integration effort of SABSA and TOGAF also has not lead to the adoption of such a domain. This integration effort extensively describes how TOGAF and SABSA can complement each other, based on the requirements engineering process that is both central to the TOGAF ADM and well defined in the SABSA methodology through the business attribute profiling process. The integration focuses on mapping SABSA artefacts and architectural efforts to the TOGAF ADM framework and leads to full integration without the need for a separate domain. In other words: the integration of an enterprise security architecture methodology with an EA framework does not require a separate security domain. Therefore, in this thesis, figure 1 will be used as the proposed hierarchy, rather than figure 4.

When examining the SABSA methodology book, this list of architectures is complemented with additional architectures: risk management architecture, management & governance architecture and infrastructure architecture. Similarly, many other kinds of additional architectures have been identified in the whitepapers and other documentation. These architectures are not considered as they are either too specific and single-purpose or do not add anything specific to the hierarchy. For example, infrastructure architecture is part of the technology architecture and comprises mostly network and system architecture. Another example; the risk management architecture is part of the business architecture. While the business architecture could have been further subdivided, this is not common practice and also not relevant within the scope of this thesis.

Other architectures, such as reference architectures, service architectures and solution architectures, are not considered an architecture type in the context of this inventory. These architectures are generic in nature and are therefore not useful for inventory. For example, a solution architecture can be technical in nature but can also be a business solution, acting at a different level of the hierarchy.
4.2 Architectures in the inventory

Figure 1 provides an overview of all identified security architecture types. All architectures outlined in the hierarchy of architectures will be examined in detail to determine their characteristics. Additionally, for each of the architecture types, an example will be provided of a description that bears the same name. Note that this is not an evaluation; the architectures mentioned as examples are in this stage not evaluated to determine if their classification is correct.

4.2.1 Enterprise security architecture

Enterprise security architecture aims to provide a holistic approach to security on the enterprise level, effectively providing guidance to all security incentives within the organisation. This guidance is a form of standardisation in which different security solutions are implemented in a similar matter and move.

An enterprise security architecture [26]:

- Must provide structure, coherence, and cohesiveness.
- Must enable enterprise-to-IT security alignment.
- Must provide abstraction.
- Must establish a common "language" for information security within the organisation.

Further investigation into descriptive texts for enterprise architectures provides additional features for enterprise security architectures:

- Acts at all levels of the organisation, from the business level to the technology level.
- Uses different levels of abstraction to provide insight for a variety of stakeholders.
- Provides a single methodology for architecture and requirements management throughout the enterprise.

An example of an architecture that uses the classification “enterprise security architecture” is SABSA. The SABSA methodology book bears the title “enterprise security architecture” and claims that it provides “a business-driven approach” [27].

4.2.2 Business security architecture

Business architecture is a domain within EA. There have been no architectures identified that name themselves business security architectures. SABSA is an abbreviation for Sherwood Applied Business Security Architecture, but considers itself to be an “enterprise security architecture”.

Business architecture in the context of EA describes how business goals are to be met using EA. It basically outlines the business case for the EA effort and is therefore considered to a “prerequisite for architecture work in any other domain” by the TOGAF EA framework [28]. With that description, business security architecture is a part of the enterprise security architecture effort.

A business security architect [29]:

- Provides a means of establishing a baseline for the current security architecture.
- Provides a means of establishing the target security architecture.
- Must enable business-to-IT security alignment
- Outlines all actors that are relevant to the architecture process.
- Focuses on conflicts between business goals and security policies.
- Deals with information risk at a business level.

As indicated, no examples could be found, although SABSA has a primary focus on business security architecture and positioning information security management as a business enabling process.
4.2.3 Information systems security architecture
Similar to business architecture, information systems architecture is also a domain in EA. For information systems security architecture, no examples could be found.

An information systems security architect[^30]:
- Determines current security-specific architectural elements.
- Identifies guidelines and standards.
- Classifies information.
- Identify criticality of availability and correct operation of each function.
- Determine attack vectors for the target systems.

Additional tasks are described, but the list above represents the most vital tasks.

This is somewhat different from the view of SANS. According to the SANS information systems security architecture[^31], an information system security architect:
- Conducts a security assessment
- Formulates Target Security Architecture (TSA) designs
- Constructs policies and procedures
- Implements (or guides the implementation of) the TSA design
- Integrates security practices to maintain secure status

The architecture briefly covers the need to create a physical and a logical architecture and focusses mainly on technology related application and data security architecture, thus matching the hierarchy as outlined in figure 1.

4.2.4 Technology security architecture
Technology is the last domain in the traditional BIT domains of EA. Because of the fact that technology is a very broad term that applies to many different kinds of solutions, services and products, it is very difficult to establish a generic “technology security architecture”. No examples could be found of an architecture that uses the classification “technology security architecture”.

A technology security architect[^32]:
- Assesses and baselines current security-specific technologies.
- Identifies trust and clearance levels of users, administrators, connections.
- Identifies minimal privileges.
- Identifies security measures justified by the risk analysis effort.

As indicated, examples of “technology security architecture” could not be found. Instead, security architectures have been found that focus on a specific field within the technology domain. The application, network and system security architectures are further elaborated in paragraphs 4.2.6 4.2.7 and 4.2.8.

4.2.5 Data (or information) security architecture
Data security architecture focusses on protecting the data. Therefore, these architectures are very single-purpose in nature, in contrary to the extended scope of an enterprise security architecture. With the focus on information, common subjects for data security architecture are encryption and logical access control.

A data security architecture:
- Limits itself to providing principles for protecting data.
- Addresses communication and data security requirements.
An example of a data security architecture is the Common Data Security Architecture (CDSA) [33][34].

### 4.2.6 Network security architecture

Similar to data security architecture, network security architecture is single-purpose in nature: protect the company information assets by deploying a robust network. It is important to note that there is a difference between deploying security solutions (such as firewalls and intrusion detection) into the network design and network security architecture. While the first activity is part of network security architecture, the latter should provide guidance for implementing such solutions. Architecture should not be confused with design, although the two subjects are related. Design is guided by architecture. In SABSA, architecture at the lower layers of abstraction is called design.

A network security architecture:
- Provides guidance for implementing security solutions in the network infrastructure.
- Provides principles and requirements that determine the need for- and type of security solutions and the technique used.
- Provides a set of security principles that are applied to the network design.

An example of an architecture that bears the name network security architecture is the SANS network security architecture for GIAC [35].

### 4.2.7 System security architecture

System security architecture deals with security at the system level. This excludes any applications that may be running on the system, as these are part of the application architecture. It also excludes any part of the network connections to the system beyond the actual physical interfaces in the system. This clearly signifies how the different security architectures within the technology domain interface. All such interfaces should be described and clearly owned in the architecture effort. A system security architecture attempts to describe the security of the system as a whole and should thus identify all relevant interfaces.

A system security architect:
- Assesses and baselines system-specific security settings;
- Create a secure systems design;
- Describe interfaces with other systems, applications and networks.

No generic architecture that can be considered a ‘system security architecture’ could be found. This is likely due to the fact that system security architecture is specific for the system itself. Therefore, very specific searches (such as “Windows 2008 server security architecture”) does provide results [36]. Guidelines and principles for system security architecture were found, but were inadequately referenced and not fully described [37].

### 4.2.8 Application security architecture

The last in the list the single-purpose security architectures is application security architecture. Software architecture is a very mature field as software developers greatly benefit from creating software modules that can be implemented in existing code segments to provide certain functionality. This speeds up the software development process, provides a means for standardisation and reduces the likelihood of errors or inconsistencies. The downside is that security errors introduced into these modules can become a common vector of attack if implemented widely.

An application security architecture:
- Should be technology independent
- Must protect the application from threats
- Support assurance by providing an accountability mechanism
• Should allow for agility to adapt to changing security infrastructures

An example of an architecture that bears the name application security architecture is the SANS institute Application Security Architecture [38].

4.3 A model for comparison
Now that the inventory has been established a model must be created or chosen to compare security architectures and classify them. This will be done using a model.

The four characteristics of this model are:
• It must provide some form of abstraction. As outlined in hierarchy, not all architectures act at the same layer of the organisation. A form of abstraction or layering must be chosen to differentiate between security architectures.
• It must provide some viewpoints to determine the primary focus of the architecture. Viewpoints have already been discussed, but will be analysed in further detail.
• It must provide the ability to determine the intended use of the architecture. Is it an architecture framework, a methodology, a process or a taxonomy? This determines the applicability of the security architecture to different situations. Additionally, the architecture under evaluation should be classified under one of the architecture types in paragraph 4.2.
• It must provide qualitative indicators for measuring the quality of the architecture. Such indicators may include as completeness, adoption rate, etc.

These characteristics need to be unified into a single classification model. The characteristics will be examined in further detail in the next paragraphs.

4.3.1 Abstraction
Abstraction has been mentioned as one of the identifying features of an enterprise security architecture [26]. The dictionary describes ‘abstract’ as “relating to or involving general ideas or qualities rather than specific people, objects, or actions” [39]. Abstraction enables the architect to remove detailed features so that a generically applicable layer remains.

Zachman abstraction
Zachman introduced abstraction into his enterprise architecture by defining different levels. Each of these levels represents a view that is applicable to a stakeholder or a number of similar stakeholders. For example, in the most abstract layer of the taxonomy (the conceptual layer), the architect attempts to create a high-level overview of the intended structure or system. This high-level overview (the concept) serves two main purposes:
1. To set constraints for the structure or system that needs to be created. These constraints are either technical in nature or may be derived from system or structure requirements.
2. To proof to the intended owner of the system that the architect has well understood the needs of the owner.

For these reasons, the conceptual layer is also called the “planners view” as this activity outlines the intention (the plan) of the architect towards the structure or system.

In total, Zachman was able to identify 5 different layers of abstraction. These were combined with the 6 descriptions of the system, as outlined in the introduction: what, where, who, when, why, how (also abbreviated as 5W1H). The combination of the layers as rows and the descriptions as columns yields a matrix or taxonomy with 30 (5 x 6) perspectives on the information system. Each of these perspectives is unique in nature as both the columns and the layers are unique. Zachman then proceeded to set 7 explicit rules for the taxonomy:
1. The columns have no order. All of the 5W1H columns are equally important. It must be noted that the rows do have an order as the order in which they are presented is the logical sequence for delivering an EA.

2. Each column has a basic, simple model. For each of the 5W1H columns, Zachman defined an abstract model that outlines the basic relationship of attributes that exists within that column. For example, for the ‘what’ column (also known as the ‘data’ column), the abstract model is “entity-relationship-entity”, while for the ‘where’ column (also known as the ‘network’ column), the abstract model is “node-link-node”.

3. The basic model of each column must be unique. Because each column has a unique description, the basic model of that column must be unique as well.

4. Each row represents a distinct, unique perspective. Similar to the uniqueness of the columns, every row is distinct from the other rows as well. In his first paper, Zachman states about this that: “each [row] has a different nature from the others. They are not merely a set of representations, each of which displays a level of detail greater than the previous one. Level of detail is an independent variable, varying within any one architectural representation.” In other words: it’s not about representing a level of detail, it’s about representing a different view on the system.

5. Each cell is unique. This has been covered twice already. Since rows and columns are unique, cells must therefore also be unique.

6. The composite or integration of all cell models in one row constitutes a complete model from the perspective of that row. This leads to the conclusion that all the cells in a row are connected somehow. Additional columns (should they be defined) must also add to the perspective created in that particular row.

7. The logic is recursive. Essentially, this means that the model applies to any form of architecture and is not limited in its use.

Figure 5 shows the framework defined by Zachman that adheres to the rules stated above (retrieved from http://www.zachman.com/images/ZI_Pics/ZF3.0.jpg, 26-7-2013).
When examining figure 5 in detail, there are a few things noticeable:

- The perspectives are slightly differently named than in the original framework.
- There is an additional perspective that has not been described yet. This is the enterprise perspective and describes the functional or operational enterprise. In the model, the layer has a different colour, as it does not concern itself with the creation of an information system (which was the original goal in the parallel drawn by Zachman to physical architecture \[1\]), but instead focuses on continued delivery of the service and maintenance of the system.

Since this thesis has a focus on delivery, rather than maintenance and operations, the ‘enterprise perspective’ is not considered to be useful in the context of this thesis.

**Other abstractions**

Besides the Zachman taxonomy, other EA frameworks have defined a similar kind of abstraction. A common other abstraction is based on three layers: the conceptual, contextual and implementation level. These layers of abstraction are then combined with different viewpoints (traditionally, the BIT viewpoints) to create a matrix or pyramid. An example of such an extraction is the Gartner EA Model, (source: Gartner, “Structure and Content of an Enterprise Information Security Architecture”, 2006), shown in Figure 6.
Gartner states that with each consecutive layer, less abstraction and more detail is provided \[^{40}\]. This statement is in contradiction with the rules that Zachman applied to his framework (rule number 4), where it is argued that each layer provides abstraction that is different in nature from the other levels of abstraction. The Zachman definition makes more sense, as there is no reason why a technical document should have more detail than a document describing business strategy.

Another abstraction that stands out is the SABSA Model for Security Architecture Development. This model, according to the writers of the SABSA book on Enterprise Security Architecture, “follows closely the work done by John A. Zachman in developing a model for enterprise architecture” \[^{17}\]. The model uses the same 5 basic layers of abstraction outlined in the initial papers by Zachman. However, the model has been extended with a ‘Facilities Manager’s View’ that represents the operational enterprise. This layer is also present in the latest incarnations of Zachman’s framework for EA. The main difference between SABSA and the Zachman framework regarding the layers of abstraction is that SABSA argues that the ‘Facilities Manager’s View’ (also known as the operational security architecture or the security service management architecture) is a vertical layer rather than a horizontal layer. This means it doesn’t provide a different level of abstraction, but that parts of the operational security architecture must be addressed in each of the other layers. Given the fact that this essentially means that it is not a layer of abstraction, the operational security architecture will not be considered. What remains are the same basic layers or viewpoints as the Zachman
framework, except that SABSA views these layers from a security perspective. The SABSA model is depicted in figure 7 (source: “Enterprise Security Architecture, White Paper”, April 2009).

![Figure 7: SABSA Model for Security Architecture Development](image)

In this thesis, the 5 layers of abstraction defined in the original Zachman model will be used. The Zachman framework has been chosen over the Gartner 3-layer framework due to the fact that the Zachman model has more detail; it breaks the ‘implementation’ level in the Gartner EA model down into three layers. It is possible that this level of detail is not represented in the architectures that will be analysed. Nonetheless, the usage of the Zachman model does not necessarily exclude the application of the Gartner EA model (this depends on the level of detail), while the other way around, this is true. In other words: if the classification of the architectures does not require differentiating between layers in the ‘implementation level’, the model can be adapted to aggregate these layers into a single layer.

The Zachman model has been chosen over the SABSA model, because it is essentially the same. SABSA, having a vertical layer breaks the basic rules of abstraction. This layer is not considered a layer of abstraction and is therefore discarded. Using similar logic, the operational enterprise viewpoint in the latest version of the Zachman framework will not be part of the model. It must be noted that the naming of the layers of abstraction will not follow the perspective naming, but rather the more common and widely adopted naming conventions (which are also part of the Zachman framework): context, concept, logic, physical and component. This naming convention is also used in many other frameworks.

Other frameworks have been examined, but are similar in nature to the models outlined in this paragraph. These models may use slightly different attributes or a different naming convention, but the essence is the same.

4.3.2 Viewpoints

Besides layers of abstraction, it is also important to integrate different viewpoints in the model. As argued before, the basic domain are Business, Information Systems and Technology (BIT). These basic domains are part of the TOGAF Framework as well as the Gartner EA Framework. All domains can be turned into viewpoints by providing a view for a specific stakeholder. It could be argued that additional viewpoints exist, such as the Application Viewpoint (TOGAF), Information viewpoint, the Security viewpoint (forming the BITS layers as seen in figure 3), the Governance viewpoint. The Cap
Gemini Integrated Architecture Framework (CG IAF) uses exactly these viewpoints to describe their model [41]. Figure 8 shows the CG IAF framework with these models and 4 out of the 6 attributes mentioned by Zachman in his papers.

When examining the viewpoints and layers of abstraction, the following conclusion can be drawn:

- The layers of abstraction are similar to those in the Zachman framework. A major difference is that Zachman addresses each of the attributes (why, what, etc.) in each of the layers of abstraction, while the CG IAF framework uses one of the attributes as a primary focal point for each layer.
- The basic viewpoints (Business, Information, Information Systems and Technology Infrastructure) are similar to what TOGAF offers and what the Gartner EA Framework uses as well.
- The overarching Security and Governance viewpoints apply and integrate with all of the basic viewpoints and the layers below the contextual abstraction. Although not outlined as such, TOGAF addresses security in each of the viewpoints as well.

Other models use similar viewpoints. As outlined earlier, the TOGAF framework is considered leading in terms of these viewpoints. However, since we will be comparing security architectures, the viewpoints need to be considered in a security context. Thus, the following viewpoints will be used:

- Business Security
- Information Security
- Application Security
- Technology Security

4.3.3 Architectural application

Architectures can vary in the way they can be applied or used within the enterprise. While some architectures provide a complete and detailed process for implementation, others describe a common practice or best practice approach. Some architectures provide a taxonomy including rules for classification, while others have no such feature. The following list of architectural applications will be used in analysis of the architectures [46]:

- Taxonomy. The architecture is considered to be a taxonomy or have a taxonomy when the architecture provides a means for classification of architectural artefacts according to a defined set of rules.
- Methodology. The architecture is considered a methodology when it describes the rules and guidelines for the development, deployment and maintenance of the architecture.
• Process. The architecture is considered a process when it provides a step-by-step instruction of each of the stages in the methodology. While being similar to a methodology, a process applies more standardization and detail.

• Practice. The architecture is considered a practice when it provides a set of tools that can be used in the implementation. This form implementation guidance is standardized, accurate and based on previous experience.

This list is completed with:

• Model. The architecture is considered to have a model when there is a schematic description of the architecture that incorporates all the basic elements and describes the relationship between those elements.

• Framework. The architecture is considered to be a framework when it adheres to the basic principles for frameworks as outlined in ISO/IEC/IEEE 42010-2011 [12].

These descriptions indicate the focus and applicability of the architecture and are not considered to be qualitative in nature. For example, a methodology is not considered better than a practice. Qualitative features are covered in the next paragraph and may be related to this list.

4.3.4 Quality attributes

The quality attributes provide information regarding the strengths and weaknesses of a security architecture. As all architectures are created with a specific purpose in mind, there will be difference in strengths and weaknesses. This thesis will not attempt to rate the architecture feature’s strength in detail (for example, using ‘++’ to ‘--’ ratings). This is due to the fact that this kind of rating requires a complete definition of when a feature can be considered ‘strong’ or ‘weak’. Instead, this qualitative rating only indicates whether a feature is present or not. A higher number of features that are present indicate a more useable framework. The attributes have been specifically chosen to assist in indicating the quality of a security architecture.

Primary quality attributes (security attributes)

The primary attributes for a security architecture represent the basic elements of information security. As described in paragraph 2.4 and published in the NIST standard FIPS PUB 199 [4], the well-known and widely accepted CIA triad represent the most basic elements of information security. This is confirmed by other publications, including the ISO/IEC 27000:2009 standard [42] and a publication from the Committee on National Security Systems, where ‘adequate security’ is described as having the three basic elements [43].

However, as indicated in paragraph 2.4 as well, these elements do not cover all basic information security concepts. Thus, additional attributes need to be identified and considered in order to obtain a full overview of information security architecture quality. These additional elements are debated. For example, Computer Security – Principles and Practice identifies ‘authenticity’ and ‘accountability’ as additional elements and recognizes that authenticity is already covered under integrity by FIPS PUB 199 and that non-repudiation supports accountability [44]. ISO/IEC identifies ‘authenticity’ and ‘accountability’ as well, but also adds ‘non-repudiation’ and ‘reliability’ [42]. A publication on basic concepts in secure systems mentions the CIA triad and adds the attributes ‘reliability’ and ‘safety’, where safety is explained as ‘absence of catastrophic consequences on the user(s) and the environment’ [45]. Thus, this is a more physical oriented attribute. In this thesis, the following basic security attributes will be used to assess the security architectures:

- Confidentiality. This attribute aims to ensure that information is only available to those authorized to see it. Associated techniques and keywords include encryption, access control and authorization.
- **Integrity.** This attribute aims to ensure that information is not modified in an unauthorized manner. Associated techniques are hashing, access control and digital signing.

- **Availability.** This attribute aims to ensure that information is available to authorized individuals when the information is required. Associated techniques include keywords are backup, redundancy & failover, connectivity and disaster recovery.

- **Authenticity.** This attribute aims to ensure that information is authentic. Thus, the recipient should be able to verify that the sender is who they claim to be and that the message that was sent is valid. Therefore, authenticity is related to integrity. Associated techniques and keywords include encryption, digital signing and digital certificates.

- **Accountability.** This attribute aims to ensure that actions can be traced to individuals. Associated techniques and keywords include logging, auditing and identity management.

- **Non-repudiation.** This attribute aims to ensure that actions carried out by individuals cannot be denied by those individuals. The concept is closely related to accountability. Associated techniques and keywords include encryption, digital signatures and public key infrastructures.

These attributes are the greatest common denominator between most publications. This also aligns well with the framework chosen for the secondary quality attributes, as explained in the following paragraph.

**Secondary quality attributes**

Several publications are available on generic architecture quality attributes. These publications include research into the characteristics of quality attributes [46], a case study for architecture quality attributes [47] and a technical report that focuses on architecture quality attributes [48]. Other publications have been found with similar content and results. These publications have two important things in common:

- They focus entirely on Enterprise Architecture. Since this thesis focuses on security architecture, the attributes will need to be evaluated for applicability. However, it is likely that there has already been a selection based on the context (which is EA), so that a selection based on this context fails to provide a representative overview.

- They focus on the evaluation and quality attributes of architecture implementations. This thesis does not focus on the actual implementation of an architecture, but instead focuses on the possibilities for implementation provided by the architecture. In other words, this thesis focuses on architecture capability and should thus be supported by quality attributes that can be used to evaluate such capability.

These features render these publications less useful as guidance in the context of this thesis.

Another article containing a qualitative comparison of EA frameworks was also considered as a starting point for the comparison of secondary attributes in this thesis as it embodies a closely related effort. The article uses a set of features and subsequently rates the strength of the framework on each feature, thereby providing a detailed qualitative analysis. These features are [49]:

- Taxonomy completeness
- Process completeness
- Reference-model
- Practice guidance
- Maturity model
- Business focus
- Governance guidance
- Partitioning guidance
- Prescriptive catalog
- Vendor neutrality
- Information availability
Unfortunately, there are a number of basic things wrong with the article:

- There is no strong argumentation for ‘Top Four’.
- There is no argumentation of why the list of features is chosen.
- There is no strong argumentation for the rating of the features.

Additionally, the focus of the article is again specifically on EA. The article is based on the experience of the author with each of the frameworks. Because strong argumentation is lacking, it is not sufficient within a scientific context. This means that this work cannot be used as a basis either.

Because of the lack of readily available and useful quality attributes specific for security architecture, a more generic quality assessment standard should be used. A common standard for quality assessment is the ISO 25010 standard. While this standard is actually designed for software product quality evaluation, its characteristics are generic in nature and can thus be applied to architecture capability evaluation as well. The standard describes 8 main characteristics and 38 sub-characteristics. Figure 9 (source: ISO/IEC, “Software engineering-Software product Quality Requirements and Evaluation (SQuaRE) Quality model”, 2008) shows the ISO25010 quality (sub-)characteristics.

![ISO 25010 quality attributes](image)

Figure 9: ISO 25010 quality attributes

Not all of the (sub-)characteristics above are relevant to security architecture. Also, not all (sub-)characteristics are relevant to architecture capabilities. For example, ‘attractiveness’ is something that is not relevant for architectures and also difficult to establish in an objective manner. Thus, a selection should be made that is appropriate for assessing security architecture capabilities. This selection is displayed in Figure 10.
The quality attributes in Figure 10 are discussed in the next paragraph. Note that, for consistency purposes, the American spelling from the standard is used for the attributes.

**Security architecture capability quality attributes**

For ‘functional suitability’, the attribute ‘appropriateness’ has been selected. An architecture implementation is considered appropriate when the implementation meets the business goals set for the architecture. For architecture capability, appropriateness is measured by how much the architecture framework focuses on meeting business goals. This attribute is similar to the ‘business focus’ attribute used by Sessions.

For ‘reliability’, the quality attribute ‘availability’ has been selected. Availability is one of the attributes in the traditional CIA (confidentiality, integrity and availability) triad. The other two attributes are addressed as well, although in a different attribute group. In this context, availability does not mean the availability of the architecture itself, but information availability as an important focus point being addressed by the architecture framework. Availability is separated from the other security requirements because availability, from a systems management point of view is operational availability. From the information security perspective, availability has a different context and is not entirely operational. To avoid conflicts regarding the context for this attribute, it has been moved to the ‘Security’ attribute group, as designated by the arrow in Figure 10.

For ‘performance efficiency’, the quality attribute ‘resource utilisation’ has been selected. In the context of evaluating security architectures, resource utilisation is mainly the human resources required in implementing the architecture. This is somewhat similar to the ‘Time to value’ attributes used by Sessions.

For ‘operability’, two attributes have been selected:

- Learnability. This attribute has to do with the availability and clarity of the documentation for the architecture methodology or framework. Without sufficient documentation, implementation will be most likely more time consuming and thus more expensive. This attribute is equal to the attribute ‘information availability’ used by Sessions. This attribute also greatly benefits from the availability of courses and other forms of support.
- Ease of use. This attribute describes how easy the architecture is to apply in projects and other initiatives. Although this depends on the implementation, the architecture framework
can have information on how to best use the architecture. This attribute is somewhat similar to the guidance attributes by Sessions: Practice guidance, governance guidance and partitioning guidance.

For ‘security’, all attributes apply. As indicated, the ‘availability’ attribute was moved to this group as the context is information availability, not system availability. The ‘security’ group contains all primary attributes, as discussed in the previous paragraph.

For ‘compatibility’, two attributes have been selected:

- Replaceability. This attribute is an indicator of how easy it is to move away from the architecture after implementation. For example, an architecture using and promoting open standards will be more likely to transfer easily to another architecture framework if required by the organisation. Any form of vendor lock-in caused by a security architecture that is not compatible with other architectures should be avoided to allow for future enterprise agility. This attribute is similar to the ‘vendor neutrality’ attribute used by Sessions.

- Interoperability. This attribute defines how well an architecture can integrate with other architectures. Again, the usage of open standards can be an important indicator of interoperability, although other indicators may exist.

A note on the attribute ‘co-existence’: a security architecture can co-exist with an enterprise architecture. However, this requires that architecture change management and alignment covers both architectures. An integrated architecture is therefore preferred.

For ‘maintainability’, four attributes have been selected:

- Modularity. The modularity attribute is an indicator of logical segmentation within the architecture. This attribute indicates if the architecture framework or methodology has elements that can be used separately from other elements.

- Reusability. An architecture should provide artefacts that can easily be re-used by the organisation. Therefore, the creation of artefacts should have some guidance so that they can be generically applied and are not too specific for any project or other initiative.

- Changeability. The architecture requires change to align with the changing goals and objectives of the organisation. This change should be governed to ensure that the architecture maintains its added value. This attribute is similar to the ‘Governance guidance’ attribute in the work from Sessions.

- Analyzability. This attribute is especially important for architecture governance, an attribute also covered by Sessions. The architecture must be subject for evaluation in order to complete the architecture lifecycle.

For ‘transferability’, the attribute ‘installability’ was selected. Architectures are not installed in the software sense, but they are deployed or implemented nonetheless. This quality attribute is an indicator of guidance for implementation. The more guidance the architecture provides, the more likely an ease deployment will be. The attribute is somewhat similar to the ‘process completeness’ attribute used by Sessions. The attribute ‘adaptability’ was also a candidate for selection, but is more accurately covered in ‘changeability’.

MoSCoW

All attributes described in this chapter will be used in the analysis. However, it is important to note that not all of these attributes are equally important indicators to the organisation of the quality of the architecture. If an organisation conducts a comparison using those attributes, they could be weighed using the MoSCoW technique \[50\] that indicate the necessity of each attribute:

- M: Must-have. This is an obligatory requirement, without which the architecture would be significantly less useful.
• S: Should-have. A workaround exists, not obligatory, but noticeable when not present.
• C: Could-have. This indicates a nice-to-have requirement.
• W: Won’t-have. These features will not be present. Naturally, none of the attributes are won’t-have requirements. This type of requirement exists because the MoSCoW method has its roots in application development in which won’t-have requirements are used to define the scope.

Table 1 shows an example MoSCoW table for the identified features.

<table>
<thead>
<tr>
<th>#</th>
<th>Feature</th>
<th>M</th>
<th>S</th>
<th>C</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Primary Security Architecture Attributes</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P.1</td>
<td>Confidentiality</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P.2</td>
<td>Integrity</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P.3</td>
<td>Availability</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P.4</td>
<td>Authenticity</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P.5</td>
<td>Accountability</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P.6</td>
<td>Non-repudiation</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Secondary Security Architecture Attributes</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Functional suitability</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S.1</td>
<td>Appropriateness</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Reliability</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S.2</td>
<td>Availability</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Performance Efficiency</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S.3</td>
<td>Resource utilisation</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Operability</strong></td>
<td></td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S.4</td>
<td>Learnability</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S.5</td>
<td>Ease of use</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Compatibility</strong></td>
<td></td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S.6</td>
<td>Replaceability</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S.7</td>
<td>Interoperability</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Maintainability</strong></td>
<td>V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S.8</td>
<td>Modularity</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S.9</td>
<td>Reusability</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S.10</td>
<td>Changeability</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S.11</td>
<td>Analyzability</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Transferability</strong></td>
<td>V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S.12</td>
<td>Installability</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The usage of a MoSCoW table has been investigated. However, this thesis will not be using a MoSCoW table, because the weight of each attribute depends heavily on the goals that the organisation has towards security architecture, the type of organisation, the maturity of the organisation and so on. Thus, a generically applicable MoSCoW table cannot be created.

4.3.5 Models
There are actually two parts to the model: the classification that can be applied to the framework and the other attributes (i.e. the qualitative and descriptive features). These attributes have been outlined in the previous paragraphs and will now be put into context for the thesis.
Classification Model

The classification model uses three different sets of features:
- The layers of abstraction provided in the architecture.
- The attributes within those layers.
- The viewpoints that are covered in the architecture.

Figure 11 shows the classification cube with each of the set of features depicted as a different axis in the model.

This model can be used to classify by omitting features that are missing from the architecture. For example, looking at Figure 8, we could conclude that the framework does not cover the ‘components’ abstraction. Additionally, not all attributes from the Zachman framework are part of the framework. At first glance, only ‘why’, ‘what’ and ‘how’ are covered. Figure 12 shows an example using a first look at the GC IAF framework. Note that this is only an example to show how the classification would work. It is not necessarily representative of the framework, as that would require a thorough analysis of the framework.
Qualitative and descriptive features
The qualitative (paragraph 4.3.4) and descriptive (4.3.3) features of the architectures will allow for a more detailed comparison of architectures. Table 2 represents the architecture assessment template that is the qualitative basis for the comparison. The architecture type is one of the types as outlined in paragraph 4.2.

Table 2: Architecture assessment template

<table>
<thead>
<tr>
<th>Architecture type</th>
<th>Taxonomy</th>
<th>Methodology</th>
<th>Process</th>
<th>Practice</th>
<th>Model</th>
<th>Framework</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architectural application</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#</td>
<td>Feature</td>
<td>Evaluation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Primary Security Architecture Attributes

Security
P.1 Confidentiality
P.2 Integrity
P.3 Availability
P.4 Authenticity
P.5 Accountability
P.6 Non-repudiation

Secondary Security Architecture Attributes

Functional suitability
S.1 Appropriateness

Reliability
Again, it must be noted that ‘architecture application’ itself is not an indicator of quality, merely of form.

4.3.6 Comparison

The architectures mentioned in paragraph 3.1.2 will be assessed using the classification model as well as the qualitative analysis template. The aim is to provide an objective overview of the features and intended scope of the architectures, not to assess whether an architecture is “better” than another. This thesis also does not consider the suitability of any architecture in any specific situation or for any existing architecture. Such suitability could be the focus of continued research and is further discussed in chapter 7.

Chapters 5 and 6 will focus on the analysis of the architectures and the assessment of the architectures using the classification cube and the analysis template.
5 Analysing the architecture frameworks

In this chapter, the architecture frameworks are explained by going into the details of their documentation. The references in this chapter are done using page numbers or sections rather than endnote references. This increases readability and checking of contents. The exception to this rule is when a reference is made to a different document than the primary source of information for that architecture framework.

5.1 SABSA

The Sherwood Applied Business Security Architecture was first introduced in 2005. It aims to provide a well-defined method to achieving results in security related IT projects and initiatives that are aligned with the goals of the enterprise (i.e. the organisation). The book wants to position itself as the “security architect’s bible” and provide the architect with guidance on the architecture process from defining a plan for security architecture, to structuring the creation of the architecture and measuring the process and the result of the effort (pp. xvii – xviii). The SABSA book makes repetitive use of use cases that apply architecture to real-world examples, thereby providing some practical guidance for implementation.

5.1.1 SABSA lifecycle phases

This architectural guidance as described in the previous paragraph is supported by the SABSA lifecycle model, which has four basic phases. These phases are:

- Strategy and planning;
- Design;
- Implement;
- Manage and measure.

Figure 13 shows the phases of the SABSA lifecycle (source: “Enterprise Security Architecture, White Paper”, April 2009).

Figure 13: SABSA lifecycle phases

Each of these phases will be examined in more detail.

Strategy and planning

This phase focusses on the definition of the results that the security architecture should provide. The methodology addresses the design of the contextual security architecture (chapter 9) and the design of the conceptual security architecture (chapter 10). The two architectures are positioned by the SABSA framework to be the first architectures to define. The remaining architecture layers depend on the outcome of these two layers, so they should be addressed first.

- The contextual security architecture is defined to address business requirements and threats that face the primary business focus. Note that this is different from IT threat management. SABSA provides a complete risk management methodology that should be used in the
creation of the contextual security architecture. The risk management methodology is extensively explained in chapter 15 (pp. 451 – 483).

- The conceptual security architecture is defined using the business attribute profiling technique; a requirements management technique created specifically for the SABSA security architecture framework. This technique is used to decide which high-level attributes are relevant to the business. These attributes can then be used to provide direction for the security architecture and measure added value of installed security components in the IT infrastructure. See paragraph 5.1.3 for more information on the business attribute profiling technique.

**Design**
In the design phase, the architecture is created. This creation is done using a structured process outlined in the methodology. Designing security architecture in the SABSA methodology covers three layers of abstraction of the SABSA model (see Figure 7):
- the logical security architecture (chapter 11)
- the physical security architecture (chapter 12)
- the component security architecture (chapter 13)

The differentiation between strategy and planning and design puts the SABSA security architecture development lifecycle into perspective: while architecture at each of the layers of abstraction is considered to be ‘architecture’, the process of creating architecture at the lower layers of the SABSA model is also called ‘design’. In the ‘design’ part of the methodology, the methodology still addresses how to architecture should be created, but also has a strong focus on actual technique and the way this technique should be addressed in architecture.

**Implementation**
After the design is completed and validated, the architecture designed in the previous step is implemented. This implementation is also described in the SABSA methodology, but only briefly. Some information for implementation and starting the information security architecture program is provided in Chapter 8, where guidance for several topics related to implementation is outlined:
- Getting sponsorship and budget (p. 148);
- Building the team (p. 149);
- Using workshops to define the goals (pp. 152 – 155);
- Programme planning and management (pp. 156 – 159).

The SABSA methodology does not specify any particular project management methodology for implementation and does not address the actual implementation project. Instead, it focusses more on operation after implementation and measuring the progress and effectiveness of the implementation. Where “strategy and planning”, “design” and “operations” are the main parts of the methodology (the book is built up out of 4 parts; first part is “introduction”), implementation is not covered centrally. Figure 7-19 (p. 133) shows the implementation process as proposed by the SABSA methodology. The deliverables outlined on page 132 should be part of the project. For the project methodology itself, the organisation should use the organisational standard. The authors specify that they have no intent to outline any project- and programme management guidance (p. 156).

Implementation also includes the implementation of the security organisation. The deploying of the security architectures and the security organisation supporting the architecture is the main focus of implementation from the operational security architecture viewpoint. This part of the implementation is covered in a fragmented manner in chapters 14 and 17.
A final note on implementation: in the SABSA methodology, it is argued that an enterprise security architecture is never implemented as a single effort. Instead, implementation is done in a fragmented manner. Therefore, it is important to establish and architecture board and integrate architectural governance into the project management system.

Manage and Measure
The “manage and measure” phase of the SABSA lifecycle is the main subject of part 4 of the SABSA methodology (operations). The management of security policy is addressed, as well as the management of risk, assurance management and operational security management. IN the methodology “management” is covered by the management of security and the management of architecture in the broadest sense, covering ITIL operational management in the context of security as well. Measurement of security is covered in the assurance as well. Additionally, metrics are addressed throughout the methodology. The most significant guidance on metrics and measurement is provided in the section covering how to measure the business attributes profile created in the conceptual security architecture (pp. 89 – 99).

5.1.2 SABSA lifecycle integration
The SABSA lifecycle approach to security architecture creation and maintenance is similar to the plan-do-check-act pitched during Dr. W. Edwards Deming’s lecture in Japan in 1950 \[51]\]. The main differences are that the Demming “plan” phase is divided into two steps (strategy & planning and design, both of which are architecture development efforts) and the fact that the SABSA “manage and measure” phase covers operations, evaluation of operations and actions taken based on the evaluation. Figure 14 shows how the Demming cycle can be plotted on the SABSA lifecycle.

**Figure 14: Demming cycle plotted on the SABSA lifecycle phases**

Since the SABSA lifecycle is generic in nature, it can also be plotted to other architectures. This helps in aligning architectural development efforts between different architectures, thereby supporting interoperability. Figure 15 (source: “TOGAF and SABSA Integration", The Open Group, October 2011) shows the plotting of the SABSA lifecycle phases to the TOGAF ADM.
Besides the SABSA lifecycle, two other important features of SABSA should be covered: the business attribute profile and the SABSA matrix.

5.1.3 SABSA business attribute profile
The SABSA business attribute profile is one of the main features of the SABSA methodology. The profile is a set of characteristics that best fit the company profile. These characteristics should be established together with people that understand the business. This is very important, as the business attribute profile is the way in which the security architecture is aligned with the business goals of the organisation. Therefore, it should not be an effort conducted without consulting the business owners or business analysts.

The establishment of the business attribute profile can be done using a workshop technique in which all stakeholders are brought together. The security architect is responsible for identifying the right stakeholders (in conjunction with the business owners) and setting up and leading the workshop. SABSA provides a default business attribute profile that can be used as a starting point in the workshop. During the workshop the most important attributes can be established per attribute group. The list provided by SABSA is not intended to be a ‘one-size-fits-all’ list of attributes. Not all attributes apply, and other attributes that are not part of the list may need to be included. In more detail, not all attributes may have equal value to the organisation. A differentiation can be made to determine which attributes are the most critical to the organisation, which are less critical, and so on. The business attribute profile is the requirements management methodology used in the SABSA methodology and is therefore central to the whole architecture effort. Figure 16 (source: “TOGAF and SABSA Integration”, The Open Group, October 2011) shows the SABSA taxonomy of ICT business attributes. More recently, an alternative taxonomy was created that has a pure business focus instead of and ICT related business focus.
5.1.4 SABSA matrix
The SABSA matrix is a two-dimensional matrix that maps the layers of abstraction to the different questions that need to be addressed at each of the layers. This taxonomy resembles the work by Zachman, as the same layers of abstraction and the same attributes for each layer are addressed. The usage of the matrix is somewhat different, as SABSA uses it as a means to provide completeness of addressing business requirements, by providing full traceability through the matrix from business requirements at the contextual layer to security products at the components layer. The layer can also be used the other way around to validate that the services that are part of the organisational security architecture have their rationale in business requirements, thereby adding to the business case for those solutions. Figure 17 (source: “White Paper Enterprise Security Architecture”, SABSA institute, 2009) shows the SABSA matrix. An additional matrix exists, which is a specialized matrix for security service management. In this matrix, security service management is not a separate layer, but is a viewpoint for each of the other layers that changes the context of the matrix slightly.
The SABSA methodology is a large body of work. However, the parts outlined in this chapter form the core of the methodology that is relevant to the context of this thesis.

5.1.5 SABSA security elements

SABSA, being a security architecture, focuses heavily on security elements. The methodology starts out with creating the business case and obtaining management support for the information architecture program, and then moves into actual architectures and security mechanisms, moving from the contextual layer down to the actual design layers.

In the contextual layer, the focus is mainly on business goals and risk management and risk analysis techniques, rather than actual security technology. The level of abstraction is too high for the layer to add value to the primary attributes directly. All other layers of abstraction do focus on information security mechanisms and thus add value directly to the primary security attributes.

Confidentiality

Confidentiality is one of the “security services” mentioned by SABSA to protect information in the application layer (p. 228) and supported by several of the “Six As” (p. 227): authorisation, authentication, access control and administration (related to access control). Confidentiality of data in databases is also covered by access control and authorisations (p. 231) and more explicitly by confidentiality protection (p. 232). The ‘authentication, authorisation and audit strategy’, as outlined in the methodology (pp. 239 – 244) also adds to this attribute as limiting access to confidential information to authorized and authenticated persons only will decrease the likelihood of unauthorized access. Thus, the strategy adds to ensure that confidential information remains confidential.

The examples in the previous section are all taken from the ‘conceptual’ layer. Of source, SABSA ensures that in the subsequent layers, the confidentiality attribute and the associated mechanisms...
are also covered. For example, the logical security architecture also deals with the relevant security services (p. 295) and pays special attention to public keys (p. 299), authorisation (p. 303, 309), authentication (p. 303, 306, 307) and explicitly to message content confidentiality (p. 308, 311). The physical security architecture deals with encryption (p. 332), specific security mechanisms such as stored data encryption in databases (pp. 343 – 347) and encryption mechanisms (pp. 349 – 358). Lastly, the component architecture lists security tools and products (pp. 391 – 392) and explains confidentiality services such as encryption in sections regarding S-HTTP, HTTPS, SSL/TLS (p. 402) and IPSEC (p. 404).

**Integrity**

Integrity is also one of the "security services" mentioned by SABSA to protect information in the application layer (p. 228). The conceptual architecture deals with integrity of information in databases (p. 231) and digital signing through PKI (pp. 251 – 265) as well as the access control mechanisms as mentioned in the previous section on confidentiality.

In the logical security architecture, information integrity is part of the security services overview (pp. 295 – 297) and is specifically addressed in message integrity protection (p. 308) and stored data integrity protection (p. 311). The physical security architecture addresses integrity as part of the database security overview (p. 336), in topics such as hashing and integrity checksums in the security mechanisms overview (pp. 343 – 347) and as part of the goals for cryptographic services (pp. 356 – 358). Lastly, the component security architecture mentions related components (pp. 391 – 392), and again addresses encryption services as outlined in the confidentiality section.

**Availability**

Integrity is not mentioned as one of the "security services" to protect information in the application layer (p. 228). Instead, it is addressed as part of the database management services (p. 231) and as part of the information transfer (network) layer (p.233), specifically in techniques such as redundancy and routing (p. 236). Availability is also briefly addressed in backup and recovery planning (p. 239). Given the fact that all topics related to access control help to keep information available by ensuring that it cannot be deleted by unauthorized individuals, the sections on access control also apply to availability.

In the logical security architecture, information availability is addressed in the security services overview (pp. 295 – 297), and more specifically in the data and software replication and backup sections (p. 313). Also, disaster recovery is an important process for information and service availability, and is covered in the logical security architecture as well (p. 318). In the physical security architecture, replication and backup are covered in the security mechanisms overview (pp. 343 – 347) and as part of the resilience section (pp. 364 – 365). Disaster recovery, with all its associated techniques is mentioned in the physical security mechanisms overview (p. 347). In the component architecture, availability is mentioned in the security tools and product overview as part of disaster recovery and business continuity. Again, all topics relating to access control that have been mentioned previously also apply.

Since availability is considered to be a more operational topic, SABSA also covers data backup and recovery (p. 524) in the Chapter 17: “Security Administration and Operations”.

**Authenticity**

As mentioned, authenticity is related to integrity. It is separately mentioned as one of the “security services” for application level data protection (p. 227). As authenticity can be established using PKI techniques, the section on PKI applies (pp. 251 – 265). Authentication can apply as well, but only when authentication is related to information, rather than users or devices. At the conceptual level,
this difference is not always made clear. However, subjects like request authenticity (p. 232) do indicate that there is a difference between user and information authentication.

At the logical security architecture level, authenticity is addressed as part of the information provided on PKI, but also specifically on message origin authentication (pp. 307 – 308) and is thus also part of the security services overview (pp. 295 – 297). At the physical security architecture level, this is explained in more detail in sections on encryption (pp. 348 – 349), data integrity provided by a message authentication code (p. 350) and digital signatures (pp 351 – 358). At the component security architecture level, authenticity is not separately mentioned, but is addressed in the more generic encryption techniques.

Accountability
Accountability is not mentioned as one of the “security services” to protect information in the application layer (p. 228). However, the closely related concept non-repudiation is mentioned. Accountability itself is covered in the “Authentication, Authorisation and Audit” strategy, mainly by the “Audit” topic. Audit ensures that all actions performed by individuals are logged and stored in a secure fashion. Accountability is further supported by the Role Based Access Control (RBAC) model as this provides central system for user management (pp. 241 – 243). Centralized user and identity management help to establish accountability for actions.

In the logical security architecture layer, accountability is mainly covered with techniques in the “assurance” part of the service overview (p. 297) and in more detail in the topics audit trails (p. 310) and system audit (p. 319). Naturally, the RABC related information, as described in the sections on directory services and authentication (pp. 300 – 310), indirectly adds to accountability. The physical security architecture addresses accountability in topics such as event logging and audit trails (p. 344). Lastly, the component security architecture mostly addresses accountability indirectly. For example, directory services supporting RBAC and PKI-based smartcards indirectly add to accountability as they help to tie identities to users.

Non-repudiation
Non-repudiation is the last of the “security services” mentioned by SABSA to protect information in the application layer (p. 228). Since non-repudiation can be achieved using PKI cryptography, the entire section on this topic and on trust applies (pp. 251 – 265).

The logical security architecture addresses non-repudiation again in the PKI section (p. 299). Non-repudiation is indirectly addressed by related techniques such as identification and authentication. At the physical security architecture level, non-repudiation is mainly addressed in the section regarding digital signature mechanisms (p. 351). The services overview (p. 344) indicates that other topics also apply to non-repudiation. These are: notarisation servers, transaction logs and trusted third part certification and arbitration. Lastly, the component security architecture addresses non-repudiation in the cryptographic tools mentioned in the tools and products overview (pp. 391 – 392).

5.2 O-ESA
The Open Enterprise Security Architecture (O-ESA) was first introduced as the Network Applications Consortium Enterprise Security Architecture (NAC ESA) in 2004. The members of the consortium proposed a policy-driven enterprise security architecture that is capable of dealing with the complexity and rapid change within the enterprise. In 2011, a revision was made of the NAC ESA, which was dubbed O-ESA, with the subtitle: “A framework and template for Policy-Driven Security”. The guide was updated to reflect the latest developments and the ISO27000 standard is now referenced [xi]. The essence, however, is the same. The guide describes an approach to enterprise
security architecture that is governed by automated decision making based on the security policy. This concept is described in the guide as ‘policy-driven security’.

5.2.1 O-ESA domains
The O-ESA guide describes several aspects or domains of an enterprise security architecture effort. These domains are:
- Business drivers
- Security program management
- Security governance
- Security technology architecture
- Security operations

These domains, and the way in which they interact, is displayed in Figure 18 (p.9).

![Figure 18: O-ESA Enterprise Security Program Framework](image)

The guide then proceeds to limit itself to the three inner areas: security governance, security technology architecture and security operations. This limits the architectural abstraction layers addressed by the guide. Other subjects are identified is important to security architecture, but are not addressed in detail. Instead, these subjects are assumed to be part of the security program, which has a different focus than the guide. The subjects are: risk management, physical security,
Comparing Security Architectures

Rob van Os, robvan-2

© Copyright 2014, Rob van Os

corporate IT governance and Enterprise architecture (p. 11). The guide primarily focuses on security operations resulting from technology-related security architecture, as indicated by Figure 18.

Each of the domains identified as part of O-ESA are discussed in detail in the O-ESA guide and explained in the following paragraphs.

5.2.2 O-ESA security governance

Governance is positioned as a means for setting the boundaries for the architectural implementation and checking the resulting security operations by performing an audit and comparing the status quo with the high-level intent. This intent is outlined in the guide in two areas:

- Principles, policies, guidelines and standards as part of the input for technology security architecture; and
- Policies and procedures as part of the input for security operations resulting from the architecture implementation.

Principles can be direct input into the technology security architecture, but can also be indirect input by providing guidance for policy development. This policy has a central role in O-ESA, as the automated enforcement of that policy is the primary focus for the guide. Figure 19 (source: p. 23) shows the elements of security governance as outlined above.

![Figure 19: O-ESA Generic policy framework](image)

The guide then continues to describe these documents and to provide guidance into what they should address. One of the striking things about security governance as outlined in O-ESA is that there is some guidance on what the policy should contain (p. 31-34), but there is little guidance on how to exactly develop such policy. Instead, the guide provides a reference to ISO27001/27002 for implementation. This provides a strong indicator of the scope of the guide: even though the focus is
automated enforcement of policy, the creation of such policy is not addressed. Thus, there is no
guidance on creating a policy that is fit for automation.

5.2.3 O-ESA technology security architecture
The O-ESA technology security architecture is divided into 5 subsequent elements:
- Conceptual framework
- Conceptual architecture
- Logical architecture
- Physical architecture
- Design & Development

The first element (conceptual framework) is outlined in the guide as the template or conceptual
representation of the automated enforcement of the policy. This conceptual framework outlines the
use of a policy management authority (PMA), a policy decision point (PDP), a policy enforcement
point (PEP) and the policy repository. It also outlined the security services protecting assets and the
assets under protection. This conceptual framework has an important role in the O-ESA architecture
as it represents the boundaries within which the conceptual architecture can be created.

The guide continues to outline the conceptual architecture that is applicable to the conceptual
framework. Figure 20 shows this conceptual architecture.

![Figure 20: O-ESA Policy-driven security conceptual architecture](image)

This conceptual architecture is then applied within two different contexts: identity management and
border protection. For both contexts, the conceptual, logical and physical architecture are created,
thereby providing an example of how to use the O-ESA architecture model for technical applications.
In contrast to the conceptual architecture, the logical and physical architecture are not addressed in
a generic fashion. Only a detailed representation is available within the previously mentioned
contexts.

Design and development are addressed as part of the guide. This design and development is an
overview of consideration and risks facing developers when creating and applications as well as
considerations and risks for administrators developing and maintaining the infrastructure. Although
this section addresses applications, it cannot be considered application architecture, as the main
focus is still on technology and there is no guidance on managing application complexity and coherence.

A note on the architecture as described in this chapter: the architecture is very different from the architecture as provided by SABSA. While SABSA addresses the way in which a conceptual architecture that is aligned with the contextual architecture can be developed and how attributes must be addressed, O-ESA simply provides a conceptual architecture view on policy-driven security. This works because the scope and use of O-ESA is very different from SABSA. While SABSA aims to provide a complete methodology and serve as a reference book, O-ESA is single-purpose in providing a means for implementing policy-based security. Therefore, the scope of conceptual architecture is limited to creating a generic conceptual reference architecture that needs to be made specific for each security service addressed by the architecture.

5.2.4 O-ESA security operations
The security operations section of the O-ESA guide addresses the operational security enterprise. Several security-related processes, such as vulnerability management, (security) event management and incident management are discussed. Additionally, evaluation is discussed through compliance, through security testing and through security metrics. This evaluation focuses on mainly the architecture implementation, not the architecture itself.

Besides the domains that have been discussed in the previous paragraphs, O-ESA also dedicates a section on the implementation of the policy-driven security concept. The implementation is explained using a vision and high-level model as well as a more detailed roadmap for development and implementation. An example of policy automation is provided as well.

5.2.5 O-ESA security elements
O-ESA, compared to SABSA, has a different purpose. Where SABSA focuses on the business case, and the design, implementation and operation of different security techniques, O-ESA focuses on implementing a framework for policy-driven security operations. However, the framework does address some security techniques in examples and security services.

Confidentiality
Confidentiality is mentioned as one of the overall objectives of information security (p. 5) and is briefly mentioned in the chapter on governance principles (p. 28). In the concrete architecture examples in paragraphs 4.4 (identity management architecture) and 4.5 (border protection architecture), confidentiality is addressed indirectly. The services mentioned support confidentiality by ensuring access to objects is controlled, authenticated and authorized. These technologies are addressed at the conceptual, logical and physical architecture.

Other services that help to protect information confidentiality are mentioned in paragraph 4.6: access management services (p. 66), access control services (p. 67), authentication services (p. 67) and authorization services (p. 68). However, the main contribution to confidentiality is addressed in the cryptographic services (pp. 73 – 74). There is a direct reference to confidentiality protection as a goal for the cryptographic services.

Integrity
Just like confidentiality, integrity is mentioned as one of the overall objectives of information security (p. 5) and is mentioned in more detail in the chapter on governance principles (p. 28). The concrete examples do not address integrity directly, although it can be argued that access control mechanisms protect information integrity, confidentiality and availability. Thus, both identity management architecture and border protection architecture add to preserving information integrity.
Other services that help to protect information integrity are mentioned in paragraph 4.6 as well: access management services (p. 66), access control services (p. 67), authentication services (p. 67), authorization services (p. 68) apply to integrity as well as confidentiality, although only indirectly. Similar to information confidentiality protection, the main direct contribution to preserving integrity comes from cryptographic services (pp. 73 – 74). Integrity protection is also mentioned as one of the main goals for cryptographic services.

Availability
Just like confidentiality and integrity, availability is mentioned as one of the overall objectives of information security (p. 5). Availability is more explicitly addressed in the section on resilience (p. 27). Information availability of course, depends on proper access control. Thus, the section identified previously that address access control and related technologies (authentication, authorization, identification) also apply to availability.

Authenticity
This element is not addressed by the security architecture. The element is also not mentioned as one of the basic goals for information security.

Accountability
This element is only indirectly addressed by the security architecture, although it is mentioned as one of the management goals for information security. There is a section on event management and auditability, but this does explicitly address accountability. Rather, this section focusses on information security incident management and compliance automation. It could be argued that identity management does add to accountability, but because basic other techniques (such as logging) are not addressed in the context of accountability, this element is not considered to be sufficiently addressed.

Non-repudiation
Non-repudiation is not mentioned in the architecture. The section on PKI in the encryption services part of the architecture does add to non-repudiation, but this is the only (brief) section that could apply. This is not considered to be a broad enough basis to support non-repudiation properly.

It must be noted that, although O-ESA does address some of the basic security elements, this is not done as structurally or as strongly as in the SABSA methodology. However, this thesis focusses on whether the elements are addresses (quantitatively) and not how well they are addressed (qualitatively), so the elements are considered to be addressed in the template.

5.3 OSA
The Open Security Architecture (OSA) is a community effort at security architecture. The statement on the OSA site is (as mentioned previously):

“OSA distils the know-how of the security architecture community and provides readily usable patterns for your application. OSA shall be a free framework that is developed and owned by the community.”

OSA is an architecture framework that is still in development. This analysis takes place on the current state dated 19-04-2014. Any changes to the framework after this date will not be considered. In this chapter and the next, any references to web pages will be done by referencing the section of the site and the page that the information was found on. The full URL will not be used, instead the relative URL will be used as a reference.
The OSA architecture community has defined some building blocks that make up the architecture framework. Not all of these building blocks are currently in place and not all are complete and fully mature. The following building blocks can be distinguished:

- **Generic information regarding the scope and use of OSA.** This information includes the security architecture landscape, the basic design principles on which OSA was built and a taxonomy.
- **Controls.** OSA provides a set of controls based upon the NIST 800-53 standard. Controls are stored in the OSA ‘control catalog’.
- **Patterns.** OSA provides a number of security patterns that can be used by a security architect to design a security solution. The patterns address a wide range of topics and indicate how they can be mapped to standards such as ISO27001 and COBIT. Patterns are stored in the OSA ‘pattern catalog’.
- **Threats.** One of the goals of security architecture is to ultimately protect against threats facing the enterprise. The OSA threat catalog is set up to contain a list of relevant threats and also provides a method of evaluating threats.

The building blocks will be examined in more detail in the next paragraphs. In every paragraph, a statement will be made regarding the completeness and perceived maturity of the contents.

### 5.3.1 OSA generic information

As indicated, the OSA framework provides some general information. The most vital part of this information is the security architecture landscape (p. foundations/osa-landscape) that defines the intended scope for the OSA framework. This security architecture landscape is displayed in Figure 21.
This architecture landscape seems to indicate a broad scope that addresses organisational security issues as well as IT security issues. However, it must be noted that the landscape is a proposal and a draft. The architecture scope as presented here would indicate that OSA is an enterprise security architecture, where business (strategy and processes), applications (application platforms and collaboration), information (data warehouse) and technology (network architecture) are addressed. In actuality, the current contents of the architecture do not address all parts of the security architecture landscape in an adequate manner. For example, security awareness is defined in the list of controls, but only contains references to other sources and does not add to or specify any further information regarding the topic.

The basic design principles that are currently outlined are also in ‘initial draft’ version (p. foundations/design-principles). There is no reference from any of the patterns or controls directly to these design principles. Additionally, there is no foundation for why these principles were chosen and no guidance on how to use the principles. Therefore, the actual added value of the principles is hard to evaluate.

The OSA taxonomy describes the different processes, attributes and entities within an organisation that affect the security architecture processes. It outlines a central role for both security architecture patterns and security controls, which are the main focus of OSA (p. foundations/osa-taxonomy)
Although the taxonomy provides a nice simplified overview of the different forces which exist within an enterprise that influence information security. OSA has stated that the main focus is currently on controls and patterns. The figure clearly outlines the role of these elements in the taxonomy, but also does not differentiate between security architecture and security architecture patterns.

Other building blocks are defined in the ‘foundations’ part of the OSA website. However, these other building blocks are not supported by any concrete results. For example, OSA intends to have a lifecycle, but has not chosen on which standard the lifecycle should be based. Therefore, these building blocks cannot be evaluated in this thesis.

### 5.3.2 OSA controls

One of the main focuses of the OSA community, according to the website, is the control catalogue that is based on the NIST 800-53. This catalogue describes a total of 172 controls in some detail. In the introduction, it is stated that all controls are now covered, but require an additional level of detail, that should be a focus for implementation in 2012. This seems to indicate that regular maintenance and updating of the control catalogue does not take place.

All controls contain a mapping to several standards: NIST 800-53, ISO 17799, PCI-DSS v2, COBIT 4.1. It must be noted that ISO17799 has been replaced by ISO27001 in 2005 (and has an updated version in 2013) and that a new version of COBIT is currently available (since 2012). Again, these are indicators that the control catalogue is not regularly updated.
5.3.3 OSA patterns
The OSA architectural pattern library (p. library/patternlandscape) contains a total of 22 patterns, 2 test patterns, 1 generic pattern and 3 reserved spaces for patterns that are still in creation. Some of these patterns are still in DRAFT version, awaiting comments by the community before being finalized.

Each of the patterns provides a high-level design in which the topic is worked out in detail. The patterns contain a number of controls that apply to that specific topic. These controls are all linked directly to the control library as outlined in the previous paragraph. Besides a high-level design, every pattern is described in additional detail. The form of the pattern description is fixed for all patterns. The following accompanying text is added to the pattern:

- The pattern legend in which the icons that were used in the pattern are explained;
- The pattern description in which the topic is worked out in more detail and the pattern is further contextualized;
- Assumptions that were used to create the pattern;
- Typical challenges for implementation and operation of the pattern;
- Indications of which organisations this pattern applies to;
- Contra-indications of which situations this pattern cannot be applied to;
- Resistance against threats;
- And some additional information including references, links to other patterns, version information, classification information and the authors and reviewers that have had a role in the development of the architectural pattern.

In some cases, additional information may be applicable, such as the design principles that apply to the architectural pattern (p. library/patternlandscape/259-pattern-data-security).

5.3.4 OSA threats
The threat catalogue as described by OSA is intended to provide a generic list of threats that may be applicable to the organisation. These threats can have controls applied to mitigate the risk and the controls can be implemented using the architectural patterns. Therefore, the catalogue has a central place in the OSA framework.

At this point, the threat catalogue does not contain any threats. A first attempt was made at analysing existing threat catalogues to build the OSA threat catalogue. This attempt is based on the Information Security Society Switzerland (ISSS) document on threat modelling [52]. The outcome of the analysis is provided as a contextual introduction. A statement is made on the OSA website that further work is required to create an actual list of threats (p. library/threat_catalogue).

5.3.5 OSA security elements
Basic security elements and services in OSA are mentioned in the control library. However, the control library limits itself to naming the control and referencing to security standards. Thus, for the control to be addressed, it must be present in one or more security patterns, as these security patterns form a central part of the OSA framework.

Confidentiality
Confidentiality is described by OSA as one of the main goals for information security (p. definitions/it-security).

In the control library, controls that address confidentiality are:
- Transmission Confidentiality (SC-09).
- Controls relating to encryption:
  - Cryptographic Key Establishment and Management (SC-12)
  - Use of Cryptography (SC-13)
Controls relating to Access Control, including:
   - Access Control Policies and Procedures (AC-01)
   - Access Enforcement (AC-03)

- Authentication and authorization related controls, including:
  - Account Management (AC-02)
  - Identification and Authentication Policy and Procedures (IA-01)
  - User Identification and Authorization (IA-02)

In the pattern library, these controls are implemented in several patterns, most importantly:
- The Identity Management Pattern (SP-010), which describes a technical architecture for identity management. Some of the controls that are implemented in the library add indirectly to information confidentiality. These controls include account management, user identification and authentication and access control policies and procedures.
- The Data Security Pattern (SP-013), which describes the protection of information in several stages (at rest, in transit, in use) and attempts to provide a complete picture of controls that can be applied to secure information. Concretely, these controls include cryptography, transmission confidentiality, media access, media sanitation and disposal (removing confidential information before disposing physical media), information leakage, physical access control, access enforcement and account management.
- The DMZ security pattern (SP-016). Border protection is meant to make sure unauthorized individuals are not able to penetrate the companies systems, thus indirectly helping in keeping information confidential. This is similar to ‘border protection’ in O-ESA, but worked out in more detail.
- Email Transport Layer Security Pattern (SP-020). The controls in the data security pattern apply here as well, only in a different context: a specific kind of information and a specific kind of protection (encryption).

Integrity
Integrity is also described by OSA as one of the main goals for information security (p. definitions/it-security).

In the control library, controls that address integrity are:
- Transmission Integrity (SC-08)
- System and Information Integrity Policy and Procedures (SI-01)
- Software and Information Integrity (SI-07)
- Controls relating to access control (see confidentiality)
- Controls relating to encryption (see confidentiality)
- Controls relating to authentication and authorization (see confidentiality)

In the pattern library, these controls are implemented in several patterns, most importantly:
- The Data Security Pattern (SP-013) as previously described. Controls that add to information integrity include transmission integrity, all access control related controls (media access control, access management, access enforcement, identification, authentication, authorization) and cryptography.
- The DMZ Security Pattern (SP-016), similar to confidentiality. This pattern adds indirectly to information integrity.
- Email Transport Layer Security (TLS) Pattern (SP-020), similar to confidentiality.

Availability
Availability is the last item described by OSA as one of the main goals for information security (p. definitions/it-security).
In the control library, controls that address information availability are:
- All contingency planning related controls, including:
  - Contingency Planning Policies and Procedures (CP-01)
  - Contingency Plan (CP-02)
  - Alternate Storage Site (CP-06)
  - Alternate Processing Site (CP-07)
  - Information System Backup (CP-09)
  - Information System Recovery and Reconstitution (CP-10)
- Denial of Service protection (SC-05)
- All access control related controls, see integrity and confidentiality

In the pattern library, these controls are implemented in several patterns, most importantly:
- The Data Security Pattern (SP-013), including the access control controls that have been identified previously. Specifically, Information System Backup, which includes information backup, addresses information and system availability.
- The DMZ Security Pattern (SP-016). Some controls help to ensure that the IT infrastructure as a whole and thus the information contained in the infrastructure remains available. Specific controls for availability include:
  - Distributed Denial-of-Service (DDoS) protection, which is an attack on availability.
  - Network disconnect and session termination, which make sure that the session tables cannot be flooded by an attacker, leading to unavailability of the service.
- The Public Webserver Pattern (SP-008). Specifically, contingency planning controls are identified that ensure that service is restored quickly in case of incidents and emergencies. Another control in this pattern, alternate processing site, is related to the contingency planning control. Other previously identified controls (most from the data security pattern) apply as well.

**Authenticity**
Authenticity is not mentioned as one of the main goals for information security. However, there are controls that specifically target authenticity:
- Information Accuracy, Completeness, Validity, And Authenticity (SI-10)
- Session authenticity (SC-23)

These controls are addressed in the pattern landscape in the following patterns:
- The DMZ Module Security Pattern (SP-016) and Email Transport Layer Security (TLS) Pattern (SP-020). These patterns specifically include the session authenticity pattern.
- The Data Security Pattern (SP-013). This pattern addresses all data security related mechanisms and specifically addresses the information authenticity control in data entry.

**Accountability**
Accountability is not mentioned as one of the main goals for information security. However, there are controls that specifically target accountability:
- Audit and Accountability Policy and Procedures (AU-01)
- All other auditing related controls:
  - Auditable events (AU-02)
  - Content of Audit Records (AU-03)
  - Auditable events (AU-03)
  - Et cetera (AU-04 to AU-09 and AU-11)

These controls are addressed in the pattern landscape, mainly in the following patterns:
- Advanced Monitoring and Detection (SP-025). This is the main pattern for audit related controls and addresses 4 controls in total.
- Identity Management pattern (SO-010). This pattern addresses identification, authorization and access enforcement. Audit records related to these security mechanisms greatly add to accountability of employees.

**Non-repudiation**

Non-repudiation is not mentioned as one of the main goals for information security. However, there is a specific control for non-repudiation in the audit and accountability control section: Non-repudiation (AU-10). Cryptographic controls can also add to non-repudiation, specifically public key related controls, such as Cryptographic Key Establishment and Management (SC-12) and Public Key Infrastructure Certificates (SC-17).

The generic control (SC-12) is

The AU-10 control is addressed in the following patterns:
- DMZ Module Pattern (SP-013). This pattern addresses many audit and accountability related controls. The context, however, is not completely clear as these controls are mapped to the external firewall. Since there is no connection to PKI or any form of authentication and identification, it is not clear how the external firewall would address non-repudiation.
- Board of Directors Pattern (SP-022). This pattern addresses non-repudiation (AU-10) in conjunction with supporting PKI and cryptographic controls (SC-12 and SC-17).
6 Assessing the architectures using the model
The chapter outlines the assessment of the three architectures using the classification and quality attribute description parts of the model.

6.1 SABSA
In this paragraph, the results of the SABSA architecture analysis are processed to fill in the model for assessment and classification.

6.1.1 Classification
Abstraction and taxonomy attributes
The two main elements of the classification cube are:
- the parts of the Zachman taxonomy that are addressed in the methodology; and
- The domains or viewpoints that are part of the methodology.

Since SABSA closely follows the work done by Zachman, it is expected that the whole taxonomy matrix is addressed. This is indeed the case, as shown in the SABSA matrix in Figure 17. Therefore, there is a one-to-one match between the Zachman taxonomy and the SABSA taxonomy.

Viewpoints
SABSA also covers all domains or viewpoints:
- Business. This is the primary focus of the SABSA methodology. Business architecture is mainly supported by the business attribute profiling technique.
- Information. Information architecture is addressed throughout the methodology by examining subjects such as information classification (pp. 416-417), data management (pp. 231-323) and backup and restore (p. 524).
- Application. Application architecture is addressed throughout the methodology by examining subjects such as middleware (pp. 325-326) and different security elements of application (309-313).
- Technology. Technology is mainly addressed in the ‘design’ part of the methodology. In the three layers of abstraction associated with the design part, several technological solutions and protocols are examined, among which are XML (pp. 380 – 381), SAML (pp. 396-397), HTTP(S) (401-402) and public key infrastructures (PKI, pp. 250-263).

Figure 23 shows the SABSA classification cube as derived from the previous indicators. The cube is a complete cube as all areas are addressed.
6.1.2 Descriptive and quality attributes

**Descriptive attributes**

The SABSA methodology describes itself as an enterprise security architecture. Given the fact that it covers all basic domains (see previous paragraph), this is indeed a well-supported claim. The name implies that it is a business security architecture. It can be said that business is the main focus for the security architecture, but the classification of enterprise security architecture is more accurate, because many other subjects besides merely business are addressed. The designation given to the security architecture for this thesis is: business-oriented enterprise security architecture.

The architecture methodology is an extensive body of work, which increases its application possibilities. SABSA provides a taxonomy (the SABSA matrix) a methodology (the lifecycle), a process (the SABSA development process), a model (the SABSA model) and a framework. The latter is supported by the characteristics and usage described in ISO/IEC/IEEE 42010-2011. Most importantly, SABSA addresses stakeholder identification and management and delivers principles, tools and guidance for the architect.

**Primary quality attributes (security attributes)**

The security attributes of SABSA have been discussed previously in paragraph 5.1.5.

**Secondary quality attributes**

**Appropriateness.** Appropriateness is ensured by the SABSA methodology and framework by presenting a strong business focus through the use of the business attributes profiling technique.

**Availability.** Availability is addressed in the SABSA methodology and framework by going into detail on business continuity management (pp. 550 – 555).

**Resource utilisation.** There is no detailed information on the effort required for implementation. Because of the fact that SABSA deals with enterprise-wide implementation, a lot of effort is expected. The method clearly states that “It is unlikely that a major strategic enterprise-wide security infrastructure will ever be implemented as a single project” (p. 132). This is an indicator of the effort...
involved in implementation, which is a direct result from the scope of the methodology (i.e. the whole enterprise).

**Learnability.** The SABSA methodology is extensively documented in the SABSA blue book. Supporting whitepapers about the use and integration of SABSA have been released to further support the methodology. Lastly, training is available to learn how to use the architecture.

**Ease of use.** Because the SABSA methodology is well documented, it is relatively easy to use. For example, the tools such as business attributes profiling are well described. Other tools, such as stakeholder workshops and operational risk management also receive significant attention, thus adding to ease of use. The book itself is well structured and suitable for use as a reference work. Lastly, the repetitive use of use cases as implementation examples add to this attribute.

**Replaceability.** This is not addressed as part of the SABSA methodology. No supporting documentation was found on replaceability either.

**Interoperability.** The SABSA methodology and framework itself does not cover interoperability of the architecture as a whole. Interoperability is addressed as an attribute in the ICT business attribute profile (Figure 16). The SABSA – TOGAF whitepaper describes in great detail how these two architectures can interoperate and integrate to complement each other. In more detail, the component security architecture also has a section on interoperability (p. 378) and mentions a number of standards to align with (pp. 381 – 390).

**Modularity.** Because of the fact that the layers are addressed independently, and the fact the several techniques used in the methodology (such as business attribute profiling and risk management) are separate activities, parts or modules from the SABSA architecture can be used by the architect.

**Reusability.** The layered approach that SABSA has to security architecture enables reusability. Changes in the lower layers of abstraction will not automatically result in changes in the higher layers of abstraction. This allows reuse of resources (artefacts) at higher layers of abstraction.

**Changeability.** Architecture changeability is supported by the SABSA lifecycle. The implementation of the lifecycle is evaluated using a set method for audit and assurance. The results from the evaluation then feed into the architecture management process and can lead to new activity in the strategy and planning phase, thus completing the SABSA lifecycle.

**Analyzability.** SABSA has a strong focus on adding value. Therefore, it also dedicates a chapter on measuring the added value of security architecture (chapter 6). Additionally, the SABSA methodology addresses the analysis of the architecture by providing guidance for assurance management, either stand-alone (chapter 16) or as part of the risk management methodology (chapter 14).

**Installability.** The SABSA methodology allows for implementation of parts of the architecture in information security projects and programmes. Installability is greatly supported by the numerous use cases outlined in the methodology that provide practical guidance to architecture implementation.

Table 3 shows the resulting assessment matrix based on the descriptive and qualitative features as outlined in this paragraph.
### Table 3: SABSA assessment matrix

<table>
<thead>
<tr>
<th>Architecture type</th>
<th>Business-oriented Enterprise Security Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Taxonomy</td>
</tr>
<tr>
<td>Architectural application</td>
<td>V</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#</th>
<th>Feature</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Primary Security Architecture Attributes</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P.1</td>
<td>Confidentiality</td>
<td>Yes</td>
</tr>
<tr>
<td>P.2</td>
<td>Integrity</td>
<td>Yes</td>
</tr>
<tr>
<td>P.3</td>
<td>Availability</td>
<td>Yes</td>
</tr>
<tr>
<td>P.4</td>
<td>Authenticity</td>
<td>Yes</td>
</tr>
<tr>
<td>P.5</td>
<td>Accountability</td>
<td>Yes</td>
</tr>
<tr>
<td>P.6</td>
<td>Non-repudiation</td>
<td>Yes</td>
</tr>
</tbody>
</table>

| **Secondary Security Architecture Attributes** | | |
| Functional suitability | | |
| S.1 | Appropriateness | Yes |

| Reliability | | |
| S.2 | Availability (see P.3) | See P.3 |

| Performance Efficiency | | |
| S.3 | Resource utilisation | No |

| Operability | | |
| S.4 | Learnability | Yes |
| S.5 | Ease of use | Yes |

| Compatibility | | |
| S.6 | Replaceability | No |
| S.7 | Interoperability | Yes, through additional information |

| Maintainability | | |
| S.8 | Modularity | Yes |
| S.9 | Reusability | Yes |
| S.10 | Changeability | Yes |
| S.11 | Analyzability | Yes |

| Transferability | | |
| S.12 | Installability | Yes |

### 6.2 O-ESA
In this paragraph, the results of the O-ESA architecture analysis are processed to fill in the model for assessment and classification.

#### 6.2.1 Classification

*Abstraction and taxonomy attributes*
O-ESA provides a clear point on the layers of abstraction. The business drivers are not considered and therefore, the contextual abstraction layer of the security architecture is not present. This is also supported by the outline of the scope of O-ESA’s enterprise security architecture in Figure 18. The ‘components’ layer of abstraction is also not addressed in the guide. Instead, abstraction ends with the overview of components in the system and the interactions that they have (the physical architecture).

The layers of abstraction applicable within O-ESA have now been established. However, since O-ESA does not provide a taxonomy, the attributes within those layers are not immediately evident. Instead, they have to be derived from the architectural examples provided and the accompanying text. For this, the Zachman definition of attributes is used [53]:

- **What**: material description. This attribute yields the bill of materials.
- **How**: functional description. This attribute provides the functional specification.
- **Where**: location description. This attribute has a focus on flows and interfaces.
- **Who**: people description. This attribute outlines roles and responsibilities.
- **When**: time description. This attribute outlines time dependencies.
- **Why**: motivation description. This attribute indicates necessity.

The existence of these attributes is evaluated for the services identity management (IdM) and border protection (BP).

**Identity Management**

At the conceptual level, the function (how) is mainly examined. Other attributes that are also addressed are: ‘what’ (the identity repository), ‘who’ (administrators and users subject to IdM and administration) and ‘where’ (location of storage) is also addressed.

At the logical level, the bill of materials (p. 53) is mainly examined. This is the ‘what’ according to Zachman. It is also analogous to ‘how’ from the CG IAF at the same level of abstraction. In essence, this view by CG IAF is a simplified view on the layer. Looking at figure 4.6 and the accompanying text, the following attributes can be identified: ‘what’ (the services), ‘how’ (account creation, maintenance, registration), ‘where’ (the internet and internal network) and ‘who’ (administrators, users).

At the physical level, the main focus is on using the bill of material to build the infrastructure: products to fulfil logical functions. This is again, the ‘what’ according to Zachman. It is also analogous to the ‘with what’ from CG IAF. In figure 4.7 (p57) and the accompanying text, the attributes material (what), how (function), where (location) are considered.

**Border protection**

The BP service is similar to IdM at the conceptual and logical layer and addresses the same attributes. The physical layer is not addressed in the guide. In short, the BP service does not yield any additional attributes, so the IdM representation must be considered leading for the classification cube.

**Viewpoints**

As argued in the analysis, the main focus of O-ESA is on technology architecture. Very briefly, applications are also addressed, but from a technical point of view, rather than from an information systems point of view (the applications view is part of the information systems view in a BIT domain approach, see Figure 1). Therefore, only the technology viewpoint is properly addressed.
6.2.2 Descriptive and qualitative features

Descriptive attributes

The O-ESA architecture describes itself as a framework and template for policy-driven security. The main focus of the architecture is technology, or more specifically, the technological implementation of automated policy enforcement. This focus makes the security architecture a single-purpose security architecture rather than a generic security architecture (such as SABSA). O-ESA bears the name ‘Enterprise Security Architecture’. This term is probably applied by the architecture itself because policy-driven usually means business-driven (because security policy is derived from business policy). However, the architecture does not ensure that the security policy is business-driven (the architecture does not focus on the development of policy). Additionally, it is expected from an enterprise architecture to have a broader scope and to address multiple domains. Because of its limited scope and focus on technology, the architecture is designated in this thesis as: single-purpose technology security architecture. This finding contradicts the text in paragraph 4.2.4. In that paragraph, it is indicated that no technology security architectures could be found based on the reference that they make to themselves. This is still true, as O-ESA refers to itself as enterprise security architecture. The reason for classifying it as a technology security architecture is that it provides a generic template that can be applied to any security service, whether it’s a network related service (e.g. border protection), a system security architecture (e.g. IdM) or an application security architecture (e.g. anti-virus service).

O-ESA cannot be considered a methodology for security architecture, as it does not outline the actions and stakeholders required to create a full architecture. It does, however, provide a process and template for implementation of the architecture. O-ESA can be considered a framework because it provides architectural principles and implementation guidance. Stakeholders are briefly addressed in the governance section of the book.

Primary quality attributes (security attributes)

These attributes have been extensively discussed in paragraph 5.2.5.

Secondary quality attributes

Appropriateness. Appropriateness is addressed in O-ESA by focussing on security policy. This security policy should be derived from business policy and should thus be business-focussed (p. 109). Because
of the fact that this business-focus is mentioned, but not ensured, the quality attribute is considered to be only partially addressed.

**Availability.** Availability is designated by the architecture as one of the key elements of information security (p. 5).

**Resource utilisation.** The architecture does not address the effort required for implementation. Given the fact that the architecture is single-purpose, the implementation can most likely be done with relative minor effort, but this is an assumption and not based on information from the guide. Therefore, this attribute is considered ‘not covered’.

**Learnability.** The O-ESA guide details the process for implementation and dedicates an entire chapter on the implementation process (chapter 6, “toward policy-driven security architecture”). The concepts are clear and with a specific focus in mind. This feature, combined with the well-written guide, make this an architecture that is fairly easy to understand and learn.

**Ease of use.** Because of the implementation examples provided throughout the book, the usage of the methodology for implementation purposes obtains strong support.

**Replaceability.** Replaceability is not addressed by the security architecture. Because of the fact that the architecture is very specific in its purpose, the likelihood of finding an alternative is low.

**Interoperability.** The O-ESA architecture is based on standards. This makes sense, as it is provided by The Open Group, a group promoting the use of (open) standards. The implementation of policy that the guide suggests is based in the ISO27001 standard (which is not an open standard), while the implementation of policy automation is suggested using the XACML standard.

**Modularity.** Modularity is applicable when parts of the architecture can be used, separately from other parts. Given the fact that this architecture is single-purpose in nature, modularity is not applicable and not implemented.

**Reusability.** The O-ESA provides “a framework and template”. The template mentioned is the conceptual architecture template to fit the conceptual framework. This template can be applied to a variety of services, as outlined in the guide.

**Changeability.** Changeability is not directly addressed by the architecture. Changes in policy can be easily reflected in the implementation by updating the policy repository. However, changes in the IT landscape and therefore the actual architecture (most likely: the logical and physical architecture) are not covered. Given the fact that changeability primary focusses on the architecture itself, this topic is considered ‘not covered’.

**Analyzability.** Metrics are defined in the O-ESA guide, mainly in the security operations section (pp. 96-105). The problem with these measures is that they are not aimed at the architectural implementation. In other words: there are almost no metrics that aim to measure the efficiency and effectiveness of policy-driven security architecture. Similarly, governance is described in the book, but the lifecycle that is required for architectural evaluation and follow-up (check and act in the Demming PDCA cycle) are not addressed. Therefore, this attribute is considered ‘not covered’.

**Installability.** Implementation guidance is provided throughout the book by providing examples and a process that explains how business policy is translated to automated technical enforcement of access control policy.
Table 4 shows the resulting assessment matrix based on the descriptive and qualitative features as outlined in this paragraph.

Table 4: O-ESA assessment matrix

<table>
<thead>
<tr>
<th>Architecture type</th>
<th>Single-purpose Technology Security Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Taxonomy</td>
</tr>
<tr>
<td>Architectural application</td>
<td>X</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#</th>
<th>Feature</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Security Architecture Attributes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P.1 Confidentiality</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>P.2 Integrity</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>P.3 Availability</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>P.4 Authenticity</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>P.5 Accountability</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>P.6 Non-repudiation</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

| Secondary Security Architecture Attributes |
| Functional suitability | |
| S.1 Appropriateness | Partial |
| Reliability | |
| S.2 Availability (see P.3) | See P.3 |
| Performance Efficiency | |
| S.3 Resource utilisation | No |
| Operability | |
| S.4 Learnability | Yes |
| S.5 Ease of use | Yes |
| Compatibility | |
| S.6 Replaceability | No |
| S.7 Interoperability | Yes |
| Maintainability | |
| S.8 Modularity | No |
| S.9 Reusability | Yes |
| S.10 Changeability | No |
| S.11 Analyzability | No |
| Transferability | |
| S.12 Installability | Yes |
6.3 OSA

In this paragraph, the results of the OSA architecture analysis are processed to fill in the model for assessment and classification.

6.3.1 Classification

Abstraction and taxonomy attributes

The layers of abstraction and taxonomy attributes that are addressed in the OSA architecture framework are somewhat difficult to identify. This is due to the fact that the framework does not address the traditional layers of abstraction and is also a work in progress.

When examining the patterns, several attributes can be identified. Not all patterns contain all attributes. The following list is an accumulated list of attributes that is created by analysing all patterns:

- Actors;
- Systems, such as servers and network devices;
- Services;
- Locations. This attribute can be found for example, in the DMZ pattern; and
- Interfaces.

Additionally, for each of these items, the controls that apply to that item are also outlined in the pattern.

These items can be translated to the Zachman attributes: what (systems), how (services, interfaces), where (locations), who (actors). The architectural patterns themselves act the logical layer: each of the patterns describes a logical solution. Some of the topics outlined in Figure 21 are often part of contextual and conceptual security architecture (e.g. strategy, governance and risk management). However, at this point in time, the patterns and other guidance provided by OSA are limited to the logical abstraction layer.

Viewpoints

When analysing the patterns, two basic patterns types can be distinguished:

- Technical patterns. Of this pattern type, there are 20 in total, ranging from specific patterns (for example: the public web server pattern) to more generic patterns (for example: the data security pattern).
- Organisational patterns. Of this pattern type, there are 2 in total: the awareness pattern and the ISMS.

The organisational patterns are an indication of the architecture acting at the business domain of security architecture. There are two main reasons to not consider the business domain adequately covered:

1. The patterns themselves do not provide sufficient guidance;
2. The scope of the business domain covered by the patterns is too limited. Only the ISMS and awareness are covered.

Therefore, the OSA architecture framework in its current state is considered to be a technical framework.

Figure 23 shows the OSA classification cube as derived from the previous indicators. Only 4 attributes are listed and only 1 abstraction layer could be identified at this moment.
6.3.2 Descriptive and qualitative features

Descriptive attributes

The OSA architecture describes itself as a free framework that consists of a number of readily usable security architecture patterns that can be applied to the organisation (p. /cms/). The main focus of the architecture is technology acting at the logical level. Thus, it is similar to O-ESA. The main difference between the two is the fact that O-ESA is a single purpose security architecture framework, where OSA aims to be a framework that addresses a broad range of topics.

While the goal of OSA is to act at different levels of the organisation and address technology topics as well as organisational topics (such as governance and risk management), the current pattern library is restricted to mostly technical patterns. In order for OSA to become an enterprise security architecture, OSA cannot be considered a methodology as it provides little guidance for implementation. Though each of the patterns in the pattern library has some description and provides some guidance, there is no generic guidance of generic approach to architecture. A security architect could use a pattern from the OSA library and implement it, taking into consideration all the controls that are outlined in the pattern. The implementation itself, as well as the integration of the pattern into the existing architecture, must be guided using some other architecture methodology or framework. OSA does not provide a concrete approach.

As indicated in the explanation of frameworks, they must provide principles, implementation guidance and address all stakeholders. At the moment, OSA does not sufficiently address any of those qualities. Therefore, OSA cannot be considered a true framework. In this thesis, OSA will be referenced as ‘technical architectural pattern and control library’.

Primary quality attributes (security attributes)

Secondary quality attributes

Appropriateness. Appropriateness is not addressed by OSA. If the whole of the intended security architecture landscape would have been implemented in the pattern library, then some of the
artefacts in the governance domain could add to appropriateness. As is, this attribute must be considered ‘not covered’.

**Availability.** Availability requirements are addressed in many of the controls in the control library. Examples of such controls are the controls ‘alternate processing site’ and ‘Security Planning Policy And Procedures’.

**Resource utilisation.** Similar to the other architectures, OSA does not address the effort required for implementation. Implementation effort will vary between different patterns, but none of the patterns addresses implementation effort either. Therefore, this attribute must be considered ‘not covered’.

**Learnability.** The OSA security architecture provides little generic guidance on how to use the architecture. As indicated, architecture patterns could be taken from the pattern library and applied into the organisation. OSA does not provide any guidance on how to implement a pattern and how to check if the implementation was successful. Therefore, learnability is considered ‘not covered’ in the architecture.

**Ease of use.** The OSA architecture currently focuses on controls and on patterns. OSA does not provide any examples or use cases for implementation. However, the fact that OSA uses a website provides very easy access to the control and pattern libraries. Therefore, the architecture can be considered easy to use in terms of navigation and finding the right pattern and associated controls to apply.

**Replaceability.** As the architecture currently lacks a process description, methodology and lifecycle, attributes such as replaceability are currently not addressed. This attribute requires a higher level of maturity than then architecture is currently at.

**Interoperability.** OSA currently focuses on controls and patterns. These patterns are positioned as ‘readily useable’. Because of the fact that OSA currently does not attempt to provide a full methodology for architecture development and implementation, these patterns require the leveraging of an architecture methodology for implementation. The fact that the patterns can be used in conjunction with other architecture frameworks and methodologies supports interoperability attribute. Additionally, the control library is based on and aligns with several standards: COBIT, ISO27001 and NIST 800-53, thus allowing integration and alignment with most organisations security programs.

**Modularity.** Modularity is one of the strong features of OSA. This is because of the fact that each pattern can be taken from the library and implemented as a separate architecture module. This modularity is further supported by the fact that the patterns have references to other patterns if dependencies exist. Thus, a set of dependent architectural patterns can also be considered to be a module.

**Reusability.** As indicated several times previously, the architecture currently provides patterns and controls. The patterns are not reusable, as they are solution-centred and single-purpose. The controls, however, are reusable as a single control may apply to multiple architectural patterns. It could be argued that the controls are not specific to OSA as they are based on the NIST 800-53 standard, but the exact definition and positioning of the controls in the architecture can be considered to be unique.
Changeability. This attribute is not addressed by the architecture. OSA does not provide a methodology, process or lifecycle that deals with such topics. Additionally, the patterns and controls are static. The pattern description does not describe implementation of the architecture and also does not elaborate on the impact of architectural change to the pattern. Therefore, this attribute is considered ‘not covered’.

Analyzability. For an architecture to be analysed, some metrics should be present. Alternatively, a lifecycle could be described that provides guidance on how to perform analysis and improvement of the architecture. The OSA landscape does address some of these items as high-level processes and a first attempt is made at trying to find a standard for a lifecycle. However, in its current state, the scope of OSA is limited to patterns and controls, and thus analyzability is currently considered ‘not covered’.

Installability. As indicated previously, OSA does not describe implementation of patterns and architecture and therefore does not address this attribute. This attribute is considered ‘not covered’.

Table 5 shows the resulting assessment matrix based on the descriptive and qualitative features as outlined in this paragraph.

Table 5: OSA assessment matrix

<table>
<thead>
<tr>
<th>Architecture type</th>
<th>Technical architectural pattern and control library</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Taxonomy</td>
</tr>
<tr>
<td>Architectural application</td>
<td>V</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#</th>
<th>Feature</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>Feature</td>
<td>Evaluation</td>
</tr>
<tr>
<td>Primary Security Architecture Attributes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P.1</td>
<td>Confidentiality</td>
<td>V</td>
</tr>
<tr>
<td>P.2</td>
<td>Integrity</td>
<td>V</td>
</tr>
<tr>
<td>P.3</td>
<td>Availability</td>
<td>V</td>
</tr>
<tr>
<td>P.4</td>
<td>Authenticity</td>
<td>V</td>
</tr>
<tr>
<td>P.5</td>
<td>Accountability</td>
<td>V</td>
</tr>
<tr>
<td>P.6</td>
<td>Non-repudiation</td>
<td>V</td>
</tr>
<tr>
<td>Secondary Security Architecture Attributes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Functional suitability</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S.1</td>
<td>Appropriateness</td>
<td>No</td>
</tr>
<tr>
<td>Reliability</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S.2</td>
<td>Availability (see P.3)</td>
<td>See P.3</td>
</tr>
<tr>
<td>Performance Efficiency</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S.3</td>
<td>Resource utilisation</td>
<td>No</td>
</tr>
<tr>
<td>Operability</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S.4</td>
<td>Learnability</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Ease of use</td>
<td>Yes</td>
</tr>
<tr>
<td>---</td>
<td>------------------</td>
<td>-----------</td>
</tr>
<tr>
<td><strong>Compatibility</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S.5</td>
<td>Replaceability</td>
<td>No</td>
</tr>
<tr>
<td>S.6</td>
<td>Interoperability</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Maintainability</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S.7</td>
<td>Modularity</td>
<td>Yes</td>
</tr>
<tr>
<td>S.8</td>
<td>Reusability</td>
<td>Yes</td>
</tr>
<tr>
<td>S.9</td>
<td>Changeability</td>
<td>No</td>
</tr>
<tr>
<td>S.10</td>
<td>Analyzability</td>
<td>No</td>
</tr>
<tr>
<td><strong>Transferability</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S.11</td>
<td>Installability</td>
<td>No</td>
</tr>
</tbody>
</table>
7 Summary and conclusions
This chapter provides a summary and conclusion for this thesis and continues with critical notes about the comparison and possible continuation of research.

7.1 Summary
Evaluating security architectures can be done by using a model for classification and performing quality evaluation based on a set of quality attributes that are specific for security architecture capability. The SABSA, O-ESA and OSA architectures that have been under evaluation each provided unique results for the combination of classification and quality evaluation.

The only way to determine the right classification and evaluate the quality properly is by going into the details of each of the security architectures. These details must then be put into context. This requires some basic understanding of security architectures. The assessment can be carried out generically, but can also easily be made more specific to the organisation by applying a prioritization to the quality attributes. The classification should remain the same, no matter the organisational context.

7.2 Overview
This paragraph provides the overviews of the analysis in a single representation. This representation is a visual aggregate of the results from chapter 6.

7.2.1 Classification cube overview
Figure 26 shows the full overview of classifications in a single colour-coded cube. The following colour coding is used:
- Blue with yellow border: shared between all architectures
- Yellow with red border: exclusive to SABSA
- Green with black border: shared between SABSA and O-ESA

Thus, the following is generic:
- Technology security viewpoint
- Logical layer of abstraction
- ‘What’, ‘how’, ‘where’ and who attributes (for the logical abstraction layer)
The following is exclusive to SABSA:
- Business security, application security and information security viewpoints
- Contextual and Components layer of abstraction.
- ‘When’ and ‘why’ attributes (for all layers of abstraction)

The following is shared between SABSA and O-ESA:
- Conceptual and physical architecture (for the technology security viewpoint only and, in case of the physical architecture, the ‘what’, ‘how’ and ‘where’ attributes)

Nothing is exclusive to OSA or exclusively shared between the OSA framework and other frameworks.

7.2.2 Qualitative assessment overview
Table 6 shows the overview of the assessment template for all architectures.

Table 6: Overview of architecture assessments

<table>
<thead>
<tr>
<th>Architecture type: SABSA</th>
<th>Business-oriented Enterprise Security Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture type: O-ESA</td>
<td>Single-purpose Technology Security Architecture</td>
</tr>
<tr>
<td>Architecture Type: OSA</td>
<td>Technical architectural pattern and control library</td>
</tr>
</tbody>
</table>

| Architectural application: SABSA | V | V | V | X | V | V |
| Architectural application: O-ESA | X | X | V | X | V | V |
| Architectural application: OSA  | V | X | X | V | X | X |

<table>
<thead>
<tr>
<th>#</th>
<th>Feature</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Security Architecture Attributes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security</td>
<td>SABSA</td>
<td>O-ESA</td>
</tr>
<tr>
<td>P.1 Confidentiality</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td>P.2 Integrity</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td>P.3 Availability</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td>P.4 Authenticity</td>
<td>V</td>
<td>X</td>
</tr>
<tr>
<td>P.5 Accountability</td>
<td>V</td>
<td>X</td>
</tr>
<tr>
<td>P.6 Non-repudiation</td>
<td>V</td>
<td>X</td>
</tr>
</tbody>
</table>

<p>| Secondary Security Architecture Attributes | | |
| Functional suitability | SABSA | O-ESA | OSA |
| S.1 Appropriateness | Yes | Partial | No |
| Reliability | SABSA | O-ESA | OSA |
| S.2 Availability | See P.3 | See P.3 | See P.3 |
| Performance Efficiency | SABSA | O-ESA | OSA |</p>
<table>
<thead>
<tr>
<th></th>
<th>Resource utilisation</th>
<th>Operability</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S.3</td>
<td>No</td>
<td>SABSA</td>
<td>O-ESA</td>
<td>OSA</td>
</tr>
<tr>
<td>S.4</td>
<td>Learnability</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>S.5</td>
<td>Ease of use</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Operability</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S.3</td>
<td>No</td>
<td>SABSA</td>
<td>O-ESA</td>
</tr>
<tr>
<td>S.4</td>
<td>Learnability</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>S.5</td>
<td>Ease of use</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Operability</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S.4</td>
<td>Learnability</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>S.5</td>
<td>Ease of use</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Operability</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S.6</td>
<td>Replaceability</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>S.7</td>
<td>Interoperability</td>
<td>Yes (through additional information)</td>
<td>Yes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Operability</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S.7</td>
<td>Interoperability</td>
<td>Yes (through additional information)</td>
<td>Yes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Operability</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S.8</td>
<td>Modularity</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>S.9</td>
<td>Reusability</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>S.10</td>
<td>Changeability</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>S.11</td>
<td>Analyzability</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Operability</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S.8</td>
<td>Modularity</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>S.9</td>
<td>Reusability</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>S.10</td>
<td>Changeability</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>S.11</td>
<td>Analyzability</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Operability</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S.12</td>
<td>Installability</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Operability</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S.12</td>
<td>Installability</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### 7.3 Conclusions

The question this thesis set out to answer was:  
“How can security architecture frameworks be compared to one another?”

In this thesis, a model was created for classification and quality evaluation to support the comparison. In the light of comparison, the most useful part of the model is the classification cube. Using this cube, it can quickly be established what the scope of the security architecture capability is. An organisation looking for an “enterprise security architecture” can thus quickly establish the right candidates. However, this does require a good understanding by the architect of what an “enterprise security architecture” really is. The definitions in this thesis can be used as a guideline.

When analysing the results, it is clear that there is no distinct relationship between the classification cube and the assessment template. It could be argued that SABSA, since it has a full score on the classification cube, will likely also score well in the assessment template. While this indeed may be likely, there is no direct relationship or ‘rule’ that would support this. For example, although O-ESA addresses a larger part of the classification cube than OSA, OSA scores better on the primary quality attributes in the assessment.

This thesis has yielded two distinct results:
- architectures that are different in classification and details; and
- architectures that are similar in classification, but different in details, scope and quality attributes.

As indicated, the difference in classification can be used to create a shortlist of candidates for selection. The difference in details can be used to compare results at a more detailed level. These details include the exact Zachman layers of abstraction and attributes as an indicator of taxonomy completeness and the differences in quality per attribute. This allows for a detailed comparison that uses the business requirements of the target organisation as input. It must be noted that it may not be possible to determine the value that should be set for all quality attributes, simply because
insufficient information is available. In such a case, expert interviews may be used to gain further insight into these quality attributes. An example of such an attribute is the effort required in implementation, something that is not often detailed in any document.

The detailed comparison method using the qualitative features of a security architecture depends on the classification as the context changes when the scope of the architecture changes. For example, an architecture may have to capabilities provide good results for a very specific purpose (as is the case for O-ESA), while another provides good results for any part of the architecture effort (as is the case for SABSA). While comparing O-ESA to SABSA at the classification level is useful to determine the differences in scope, comparison at the detailed level is less useful as the two architectures should be applied in very different situations. Comparing O-ESA and OSA makes more sense, as their classifications are more similar (though not equal).

To summarize, the classification model acts at two distinct levels:

1. To determine the scope and taxonomy completeness of the architecture
2. To determine the ability of the architecture to address specific quality attributes, both directly security related attributes and other quality attributes.

These two levels together provide a valuable combination because it allows for comparison and selection at a high and at a low level. If this model was to be used in practice, it is recommended to perform classification for all possible candidates first, and subsequently perform a detailed quality analysis on the remaining short-list of candidates after selection. The classification is often fairly easy to deduce without the need to go into the details of the architecture.

In short, the answer to the research question should be:
“By classifying and evaluating architecture frameworks using a predefined classification scheme and using quality attributes specific for security architecture capability”.

7.4 Critical notes about the comparison

Some critical notes can be made about the comparison:

- The architectural domains (viewpoints) and layers of abstraction at which the architectures act is the most important piece of information in the classification cube. The Zachman attributes (what, how, where, who, when, why) provide less information as it is not immediately clear what that impact of absence of an attribute is on the architecture capability. The initial proposal of the Zachman framework only addressed what, how and where is an important indicator of the importance of these attributes. All architectures were found to address these three attributes, at least for the technology viewpoint.

- Some of the secondary quality attributes are difficult to address. For example, the attribute ‘resource utilisation’ is not described in any of the analysed architectures. It must be stated that this is not an unexpected result: a methodology rarely makes statements about the effort required as this depends on many factors (such as the complexity of the organisation and the experience of the employees involved). Thus, such an attribute can really only be objectively established when performing interviews with security architects that are familiar with the implementation of that particular architecture and who are able to put it into perspective with other architectures. The same applies to the attribute ‘replaceability’, which cannot be expected to be answered by the architecture itself.

- The evaluation of the qualitative features is still prone to some level of subjectivity. This could be avoided by further detailing the qualitative attributes (especially the primary attributes) and more clearly establishing metrics for when an attribute is sufficiently addressed and how well the attribute is addressed. This model does not provide that level of detail. Therefore, the evaluation should be carried out by a single person. This will still have
some level of subjectivity, but at least the subjectivity will be consistent between the different architectures being evaluated.

- The secondary quality attributes that were used in the comparison are quality attributes that apply to software quality evaluation. Naturally, software is different from architecture. Therefore, not all attributes from the standard are applicable. In this thesis, a selection was made from the available attributes. By limiting the possible attributes to those presented in the standard, other attributes that are relevant to architecture but not to software quality architecture may have been overlooked. Since there is no standard for architecture quality attributes, this can only be established by performing expert interviews. In one of these interviews, an attribute that is potentially lacking was identified: ‘communicateability’, the ability to communicate about the architecture with different architects. TOGAF, for example, integrates well with the architecture visualization tooling archimate, which allows architects to conduct architecture activities and discussions using the same ‘language’. It could be argued that this is the ‘interoperability’ requirement from a different viewpoint: the architect’s viewpoint instead of the architecture viewpoint.

- Some of the secondary quality attributes have a dependency at a higher level. For example, looking at the OSA architecture, it is clear that this architecture is not supported by any methodology, lifecycle or proper implementation guidance and process. This means that the attributes: analyzability, learnability, installability and changeability are not addressed. By mapping these dependencies on higher level artefacts, the analysis could be performed more efficiently. This means that the analyst will look for specific artefacts that address attributes, rather than look for proof of attributes. Table 6 contains an example mapping based on the experience with architecture comparison.

Table 7: Example mapping of secondary attributes to artefacts

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Requires artefact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appropriateness</td>
<td>Business requirements</td>
</tr>
<tr>
<td>Availability</td>
<td>Security description</td>
</tr>
<tr>
<td>Resource utilisation</td>
<td>External source, possibly expert interview</td>
</tr>
<tr>
<td>Learnability</td>
<td>Documentation, support, training</td>
</tr>
<tr>
<td>Ease of use</td>
<td>Use cases, examples,</td>
</tr>
<tr>
<td>Security attributes</td>
<td>Security description</td>
</tr>
<tr>
<td>Replaceability</td>
<td>External source, possibly expert interview</td>
</tr>
<tr>
<td>Interoperability</td>
<td>Open standards</td>
</tr>
<tr>
<td>Modularity</td>
<td>Various</td>
</tr>
<tr>
<td>Reusability</td>
<td>Templates, generic artefacts</td>
</tr>
<tr>
<td>Changeability</td>
<td>Architecture lifecycle</td>
</tr>
<tr>
<td>Analyzability</td>
<td>Architecture lifecycle</td>
</tr>
<tr>
<td>Installability</td>
<td>Implementation guidance</td>
</tr>
</tbody>
</table>

Modularity can have various artefacts, as modularity can act at different layers: modules can be separated patterns (such as in OSA) or separate layers of abstraction. It must be emphasized that this is an example mapping. A more definitive and complete mapping could be created as continued research.

7.5 Research continuation

Besides the creation of a definitive and complete mapping of artefacts to attributes, other further research into comparison of security architectures can be performed. This paragraph lists some suggestions for continued research.
7.5.1 Extending the primary attributes
The attributes chosen as primary attributes have been based on existing research regarding these topics. Other primary attributes can be identified, leading to a more complete view of security attributes for each architecture. For example, additional primary attributes could be: legality, compliance, privacy and safety. It must be noted that, as the list grows, the effort required to perform the analysis will also increase.

7.5.2 Detailed qualitative analysis
The thesis uses 17 attributes (6 primary and 11 secondary) to determine what each of the architectures has to offer in terms of functionality. These features are indicators of architecture quality and have been analysed for presence in the architecture. The research could be continued by also determining the quality of each of the features. This provides a more in-depth detail of the strengths and weaknesses of each of the architectures. This research requires an objective definition for each of the features to determine its strength. For example, the architectural feature ‘Have architecture implementation guidance’ could be further detailed by indicating that strong implementation guidance means that a process should exist in which every step of the implementation is detailed. When the implementation guidance does not detail every step of the implementation, but merely provides high-level guidance, this feature is considered to be less strong. If such an analysis at providing more detail is attempted, it should be noted that the measurement of the strength of each of the features may be difficult to execute and prone to subjectivity, especially if there’s little or no experience with these architectures. Interviews with subject matter experts that have extended experience with the implementation of at least two or more architectures should be considered.

7.5.3 Quality evaluation of architecture implementations
The evaluation and classification methodology created in this thesis uses a set of quality attributes for determining the quality of the architecture capabilities. This evaluation part of the methodology could also be applied to architecture implementations to determine if the architecture was implemented in a way that ensures high quality. Evaluation of the architecture implementation requires a slightly different set of attributes, as the focus is different. After all, the architecture has already been implemented, so features that focus on the implementation or are especially important during the implementation (such as the availability of framework documentation) are less important. Using this new set of quality attributes, the architecture can be evaluated for strengths and weaknesses, which can be used as input in the architecture governance process.

Figure 27 shows the list of proposed attributes for quality evaluation for architecture implementations. Continued research can establish whether these are the right attributes for a correct evaluation and can also provide guidance on how to act on the output of such evaluations.
As the figure clearly shows, compliance is now an important part of the architecture quality: how well does the architecture implementation comply with its goals? It must be noted that, although some attributes have not changed, the context is entirely different. For example, resource utilisation is no longer an indicator of the effort required for implementation, but an indicator of the effort required for maintenance of the architecture.

### 7.5.4 Architecture applicability

In this thesis, the architectures have been analysed and compared in an objective manner. The applicability of each of the architectures to different kinds of businesses and business maturity levels has not been discussed, but is valuable to investigate. With this maturity and applicability research, a matching could be made between companies and the most suitable security architecture. For this purpose, the following should be considered:

- The company business model
- The company maturity level

For the basic business models, the value disciplines as defined by Treacy and Wiersema \[54\] could be used. These basic business models are:

- Operation excellence
- Product leadership
- Customer Intimacy

For the company maturity level, the Capability Maturity Model Index (CMMI) levels could be used. The basic CMMI levels are numbered from 1 to 5, with 5 being the most mature. The CMMI levels differentiate three different areas:

- Development \[55\]
- Services \[56\]
- Acquisition \[57\]

Table 8 shows the matrix that could be used for the analysis. For each of the architectures, the fields should be filled in that indicate that the architecture can be used at that maturity level and for that particular CMMI area within a business model.
Additionally, the existing architecture itself could be used to determine applicability. As outlined in this thesis, integration with existing architectures is an important driver for choosing an architecture. Not all architectures integrate equally well with other architectures. To determine applicability of a security architecture for an organisation, this integration aspect would need to be investigated. The following could be considered in such an investigation:

- The architecture currently in place in the organisation
- The maturity of the existing architecture.

The evaluation of the current architecture, including the maturity, can be performed using a wide range of assessment models that are available. For example, to assess the architecture maturity level, generic assessment models (such as the Gartner ITscore maturity assessment \[^{[58]}\] or the US Department of Commerce Architecture Maturity Model \[^{[59]}\]) or architecture-specific architecture assessment methodologies (such as the Institute for Enterprise Architecture Developments (IFEAD) Extended Enterprise Architecture Maturity Model (E2AMM) \[^{[60]}\]) can be used. Alternatively, the System Security Engineering Capability Maturity Model (SSE-CMM) could also be used. Lastly, the SABSA methodology also briefly addresses maturity models (pp. 100 – 101) for assessing security architectures but does not make a definitive statement on which model to use.
8 Reflections and a final word

All in all, the process of getting this thesis together was harder than I initially imagined. There is a lot of work associated with writing a thesis and obtaining properly founded results. Choosing a subject with which I had little expertise did not make it any easier.

But this thesis was written as part of a Master study, and study is all about learning. Although the learning curve was a bit steeper than I had initially envisioned, the creation of this thesis has enabled me to learn a lot about security architecture in a relatively short timeframe. It has also further awakened my interest in security architecture. And what’s more, I was able to translate this newfound knowledge directly to my working environment by providing consultancy and guidance on using features of security architectures to create a more robust and secure enterprise architecture.

For the next thesis (the master thesis), I have identified a few lessons learned:

- Preparation is very important. Getting the research question completely clear and creating an outline of what the results should be and how to get there is a vital part of writing a thesis in a smooth way. My own preparation, looking back, was insufficient. For the next thesis, I should have a lot of information gathered and read to ensure the research outline is completely clear before starting the writing process.
- There are no shortcuts to quality. Although this thesis was ‘only’ the magister thesis, it needed to be just as well-founded and set up as the master thesis. This requires a lot of information gathering and processing.
- Choosing a subject you are not familiar with will increase the effort, but will also provide more satisfactory study results. As mentioned, the amount of time required was higher than initially estimate. Nonetheless, I would not have it any other way.
- Using iterative review, rather than holistic review will help to avoid having to put a lot of effort into potential structural mistakes in the thesis. This would have prevented the revision that was required after the initial release.

In short, the ride was enjoyable, but still I’m glad I’ve arrived.
Annex A: Overview of tables and figures

A.1 Tables
Table 1: Example MoSCoW table of architectural quality features ...................................................... 28
Table 2: Architecture assessment template .......................................................................................... 30
Table 3: SABSA assessment matrix ........................................................................................................ 55
Table 4: O-ESA assessment matrix ........................................................................................................ 59
Table 5: OSA assessment matrix ........................................................................................................... 63
Table 6: Overview of architecture assessments .................................................................................. 66
Table 7: Example mapping of secondary attributes to artefacts .......................................................... 69
Table 8: maturity levels versus business models .................................................................................. 72

A.1 Figure
Figure 1: hierarchy of security architectures, based on EA ............................................................... 11
Figure 2: addressing security at each of the BIT layers ....................................................................... 11
Figure 3: security as a separate viewpoint in EA ............................................................................... 12
Figure 4: security as a separate domain in EA .................................................................................... 13
Figure 5: Zachman Framework for EA ............................................................................................... 19
Figure 6: Gartner EA model ................................................................................................................ 20
Figure 7: SABSA Model for Security Architecture Development ..................................................... 21
Figure 8: Cap Gemini Integrated Architecture Framework .............................................................. 22
Figure 9: ISO 25010 quality attributes ............................................................................................... 25
Figure 10: security architecture capability quality attributes ........................................................... 26
Figure 11: Classification cube .............................................................................................................. 29
Figure 12: Example classification ........................................................................................................ 30
Figure 13: SABSA lifecycle phases ...................................................................................................... 32
Figure 14: Demming cycle plotted on the SABSA lifecycle phases .................................................... 34
Figure 15: SABSA lifecycle phases plotted on the TOGAF ADM ....................................................... 35
Figure 16: SABSA taxonomy of ICT business attributes .................................................................... 36
Figure 17: SABSA matrix ...................................................................................................................... 37
Figure 18: O-ESA Enterprise Security Program Framework ............................................................ 40
Figure 19: O-ESA Generic policy framework ...................................................................................... 41
Figure 20: O-ESA Policy-driven security conceptual architecture ................................................... 42
Figure 21: OSA security architecture landscape ................................................................................ 46
Figure 22: OSA taxonomy ................................................................................................................... 47
Figure 23: SABSA classification cube ................................................................................................. 53
Figure 24: O-ESA classification cube ................................................................................................. 57
Figure 25: OSA classification cube ...................................................................................................... 61
Figure 26: Overview of classifications ................................................................................................. 65
Figure 27: proposed architecture implementation quality attributes ............................................... 71
Annex B: literature and references

B.1 Literature
As mentioned in chapter 3, the following literature and sources were used in the analysis of the architectures:

- http://www.opensecurityarchitecture.org

B.1 References
These are all the references used in the thesis. When a reference was quoted multiple times from different pages or parts of the reference, these are considered to be separate references.

2 J.F. Sowa; J.A. Zachman, “Extending and formalizing the framework for information systems architecture”, IBM SYSTEMS JOURNAL, VOL 31, NO 3, 1992, pp. 590-616
8 The Open Group, “TOGAF® version 9.1, an open standard”, 2011, chapter 3.12
11 The Open Group, “The Open Group Architecture Framework 9.1; part IV: Architecture Content Framework”, 1-12-2012, chapter 35.1
14 Y.T. Lee, “INFORMATION MODELING: FROM DESIGN TO IMPLEMENTATION”, 1999, p. 2
24 The Open Group TOGAF-SABSA Integration Working Group, “TOGAF® and SABSA® Integration”, October 2011