

DESIGNING SECURE ENTERPRISE ARCHITECTURES

A comprehensive approach: framework, method, and modelling language

MASTER THESIS SANDER VAN DEN BOSCH

DESIGNING SECURE ENTERPRISE ARCHITECTURES

Enschede, 2 May 2014

Author

Sander van den Bosch

Programme: MSc Business Information Technology
Institute: University of Twente
Faculty of Electrical Engineering,
Mathematics and Computer Science
Enschede, the Netherlands
Student number: s0175358
E-mail: s.f.vandenbosch@alumnus.utwente.nl
Date: 2 May 2014

Graduation Committee

dr. ir. Marten van Sinderen

Department Services, Cybersecurity and Safety
E-mail m.j.vansinderen@utwente.nl

UNIVERSITY OF TWENTE.

dr. Maria-Eugenia Iacob

Department Industrial Engineering and
Business Information Systems
E-mail m.e.iacob@utwente.nl

UNIVERSITY OF TWENTE.

ir. Sander van Wijk

Department Deloitte Consulting B.V.,
Enterprise Architecture
E-mail svanwijk@deloitte.nl

Deloitte.

PREFACE

This master thesis describes the research to conclude my master study 'Business & IT' at the University of Twente. It also means the end of my time as a student, which I enjoyed very much and in which I developed myself in both a personal as well as a professional way.

The research has been established in cooperation with Deloitte Consulting in the Netherlands. Specifically I worked together with the experts of the service line Enterprise Architecture, who helped me very much in discussing the interesting subject, as well as in making the translation from theory to practice.

First, I would like to thank my university supervisors Marten van Sinderen and Maria Iacob for their support. They gave me guidance, room for developing ideas and shared their experiences and views on this research.

Furthermore, I would like to thank Deloitte Consulting for providing me the opportunity doing my research in cooperation with them. I would really like to thank Sander van Wijk for his supervision, enthusiasm and critical view on the research when needed. Also I would like to thank Eric Onderdelinden for the inspiring discussions about the subject and bringing me into contact with several experts within the enterprise architecture discipline. Next to that, I would like to thank the experts I interviewed during the validation of this research: Lourens Bordewijk, Rob Faber, Aaldert Hofman, Sorin Iacob, Paul Samwel, and Jan Joris Vereijken. Their interesting views on the subject significantly improved the quality of the research. Also special thanks to Jacco Roest, the discussions we had significantly boosted the quality of this thesis.

Finally I would like thank my family and girlfriend Suzanne, who supported me in writing this thesis as well as during my years of studying. Last but not least, thanks to the many friends who made my time in Enschede a time I'll never forget.

I hope you really enjoy reading this thesis and, in case if you have any questions, please feel free to contact me.

Sander van den Bosch

MANAGEMENT SUMMARY

Although acceptance is increasing of the notion that security should be considered throughout the entire design cycle of enterprise architecture, both disciplines are still separated to a large extent. Moreover, no systematic approach exists to integrate the two.

In this thesis, an approach for designing secure enterprise architectures is proposed. Both security and enterprise architecture discipline could potentially benefit from this approach. From a security perspective, the likelihood that security requirements will be addressed throughout the enterprise could be increased. From an enterprise architecture perspective, the intended benefits are expected to be realized by providing a holistic view on the organisation where security is adequately addressed. This is demanded, because the impact of security on the organisation is still growing.

The approach to designing secure enterprise architectures as developed in this thesis consists of three elements: a framework, a method, and a modelling language.

- The *framework* structures the architecture viewpoints. Zachman is often used for enterprise architecture in this regard, where for security purposes SABSA is frequently employed. This thesis describes how these two frameworks are related. Although the two frameworks have an identical structure, it is still valuable to use these frameworks side by side, as they complement each other regarding their content.
- The *method* provides a step-wise prescriptive approach for developing an architecture. This thesis describes how the TOGAF Architecture Development Method and the SABSA Lifecycle can be integrated. The SABSA Lifecycle enriches the ADM with relevant security aspects per development phase.
- The *modelling language* defines the concepts for describing an architecture. Since security concepts are currently not covered in ArchiMate, a security extension is proposed. The extension provides the various concepts needed to include security in the architecture specification, specifically: vulnerability, threat, risk, security mechanism, and security policy.

Experts from the enterprise architecture and security discipline were interviewed to validate the proposed approach. Overall, the approach is considered to be both usable and useful, since it includes the correct ingredients and they are well integrated. Also, the appropriate constructs are deemed to be included in the modelling language to support security modelling in enterprise architecture.

In summary, results of the research include:

- A description of the relation between the Zachman and the SABSA framework
- An integration of the TOGAF ADM and the SABSA Lifecycle
- An identification of constructs for modelling security in enterprise architecture and an ArchiMate extension based on these constructs

The overall contribution of this research is the approach to designing secure enterprise architectures.

TABLE OF CONTENTS

Preface	V
Management Summary	VII
1 Research Introduction	1
2 Background	2
2.1 Enterprise Architecture	2
2.2 Security	3
2.3 Secure Enterprise Architecture	3
3 Research Design	5
3.1 Problem Statement	5
3.2 Research Objectives	6
3.3 Research Questions	6
3.4 Research Methodology	7
4 Literature Review	9
4.1 Approach	9
4.2 Enterprise Security Architecture	11
4.3 Analysis	14
4.4 Model	17
4.5 Design	21
4.6 Hyperconnectivity and Interoperability	23
4.7 Conclusion	23
5 Solution Design	25
5.1 Line of Reasoning	25
5.2 Framework	26
5.3 Method	29
5.4 Modelling Language	40
5.5 An Integrated Approach	54
6 Demonstration	55
6.1 ArchiSurance Case	55
6.2 Application of Solution Design	55
6.3 Demonstration Conclusion	67
7 Validation	68
7.1 Method	68

7.2	Research Rigor.....	69
7.3	Research Relevance.....	70
7.4	Validation Conclusions	72
8	Conclusion	75
8.1	Research Questions.....	75
8.2	Contributions.....	77
8.3	Limitations and Suggestions for Further Research	79
9	Bibliography.....	80
	List of Figures	82
	List of Tables	84
	Appendix A – Literature Review Short List	85
	Appendix B – Security Related Concepts in ArchiMate	87
	Appendix C – Interview Questions.....	90

1 RESEARCH INTRODUCTION

The subject of this research is the integration of the enterprise architecture and security disciplines: two research fields which are still too separated, but have more in common than you would say at first sight. When executed together properly, these fields are able to reinforce each other (Kreizman & Robertson, 2006). Although the enterprise architecture has evolved into a mature discipline, still not much attention has been paid to integrate non-functional aspects in general, and in particular to integrate security (Shariati, Bahmani, & Shams, 2011).

Enterprise Architecture is the discipline that focuses on organizing logic for business processes and IT infrastructure, reflecting the integration and standardization requirements of the firm's operating model. The purpose of enterprise architecture is to optimize the often fragmented legacy of processes across the enterprise into an integrated environment that is responsive to change and supportive of the delivery of the organisations strategy (The Open Group, 2011b). Within the enterprise architecture discipline, over the last years several frameworks, methods and modelling languages have been defined to achieve this goal (Iacob, Jonkers, Quartel, Franken, & Berg, 2012).

Security aims at ensuring that risks and controls are in balance (Anderson, 2003). In order to achieve this a holistic approach to security is needed encompassing the complete organisation (Ekstedt & Sommestad, 2009; Lang & Schreiner, 2008). 'Secure by design' is an important credo within this discipline, stating that artefacts should be designed from the ground up to be secure. The underlying hypothesis is that once the design has been produced, security problems can seldom be fixed by adding new functionalities, and generally the solution lies in redesign, which can be both costly and time-consuming.

An integration of the two disciplines can be promising for both, and that is exactly what this research aims at.

2 BACKGROUND

In this chapter high-level background information on the research is provided. It starts with an elaboration on enterprise architecture in section 2.1, followed by security in section 2.2, and considers the various concepts that are related to an integration between the two. One of these concepts is a definition of secure enterprise architectures, which is discussed in section 2.3.

2.1 ENTERPRISE ARCHITECTURE

Within this research, the definition of enterprise architecture provided by Engelsman, Quartel, Jonkers, and van Sinderen (2011) is used:

“a design or a description that makes clear the relationships between products, processes, organisation, information services and technological infrastructure; it is based on a vision and on certain assumptions, principles and preferences; consists of models and underlying principles; provides frameworks and guidelines for the design and realisation of products, processes, organisation, information services, and technological infrastructure.”

Enterprise architecture examines more than just the technological architecture, or just the business architecture, or even the two side by side. It investigates the phenomena that emerge when the two interact (Engelsman et al., 2011; Innerhofer-Oberperfler & Brey, 2006).

The term enterprise in enterprise architecture covers all kinds of business organisations, including public or private sector organisations and an entire business or corporation. Also a business unit as part of a whole and a conglomerate of several organisations is included. The enterprise can be both national and international oriented.

In general, a distinction between four layers of architecture is made: business, information / data, application, technological / infrastructure, as outlined in Figure 1. The processes and activities in the *business* make use of *information* or data, which need to be collected, organized and distributed, using *applications*, which run on *technology* or *infrastructure* such as a computer system. Sometimes, the information/data and application layer are integrated into one layer, often called the information systems layer.

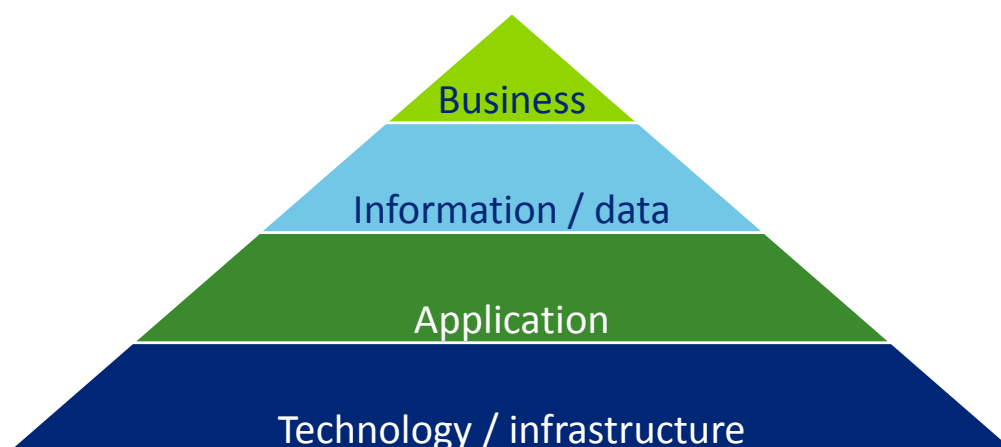


Figure 1: Four common layers in Enterprise Architecture

2.2 SECURITY

The definition for information security within the context of this thesis is the definition of Anderson (2003):

“A well-informed sense of assurance that information risks and controls are in balance.”

Security refers to minimizing the risk of exposure of assets and resources to vulnerabilities and threats of various kinds. This applies to any vulnerable and valuable asset, e.g. a person, community or organisation. Within the context of security, a distinction is made between physical security, IT security and information security. The physical security refers to the measures taken to deny unauthorized access to facilities, equipment and resources. IT security is mainly aimed at protecting applications and technical measures. Information security is concerned with the protection of information, both analogue and digital. Cybersecurity is information security as applied to computers and computer networks. Within the context of this thesis, the focus is on information security, so digital as well as analogous information are in scope.

Three fundamental qualities of information are vulnerable to risk, namely confidentiality, integrity and availability. *Confidentiality* aims at ensuring that information is accessible only to those authorized to have access. *Integrity* safeguards the accuracy and completeness of information and processing methods. The *availability* aspect ensures that authorized users have access to information and associated assets when required (Johansson & Johnson, 2005; Kim & Leem, 2004).

On the other hand are the costs and effort incurred in minimizing the risk. There is a need to balance these costs with the business needs and regulatory compliance. The concept of balance also includes the notion of cost effectiveness (Anderson, 2003).

2.3 SECURE ENTERPRISE ARCHITECTURE

A lot of concepts are related to ‘secure enterprise architecture’. An overview of these concepts is provided, and it is described to what extent they will or will not be incorporated within this thesis. A distinction is made between ESA (enterprise security architecture), TSA (technical security architecture), and SEA (secure enterprise architecture). These are the terms discussed in literature most often.

A. Enterprise Security Architecture (ESA)

An enterprise security architecture is defined as a document (or a layered set of documentation) that links an accepted vision for information security in the enterprise to blueprints for implementing security controls (including processes, policies and technology) (Scholtz, 2006). The identification, analysis and prioritization of business security requirements, the risk and threats and the choice of a portfolio of the best integrated enterprise security solutions are done based on this architecture (Shariati et al., 2011).

Sometimes an author refers to the term 'enterprise information security architecture' instead of 'enterprise security architecture'. Within this thesis they are treated as synonyms.

The ESA is concerned with the security architecture throughout the organisation.

B. Technical Security Architecture (TSA)

The technical security architecture typically defines standards for protection settings that can be implemented as a technical mechanism to minimize information risks. In contrast to the enterprise security architecture the prime focus is on technical elements.

A technical security architecture focuses on the mapping between the control architecture and the protection processes on a technical level.

C. Secure Enterprise Architecture (SEA)

A secure enterprise architecture encompasses the enterprise architecture as well as the security risks and measurements on all levels of enterprise architecture: business, information, application and technology architecture. By combining the enterprise architecture and security, it is possible to provide an integrated description of an organisation's structure, processes and underlying IT landscape (Innerhofer-Oberperfler & Breu, 2006).

A secure enterprise architecture describes the security risks and measures in relation to the business, information, application and technology architecture.

The secure enterprise architecture is the concept discussed in this thesis.

3 RESEARCH DESIGN

This chapter describes the research design. The problem statement is outlined in section 3.1. In order to solve the problem, the research objectives are derived and discussed in section 3.2. In section 3.3 the research questions are provided, and the final section 3.4 describes the research methodology.

3.1 PROBLEM STATEMENT

Both practice and literature state that the fields of security and enterprise architecture are still too separated (Oda, Fu, & Zhu, 2009). In theory, the role of security experts is in the profile and planning phase by performing a risk assessment and developing principles, procedures, policies and measurements accordingly. Based on this risk analysis, the enterprise architect's expertise is to implement these principles, procedures, policies and measurements in a target architecture. However in practice, the synergistic benefits between these fields have not been captured yet.

Although several attempts have been made to integrate these worlds, still no comprehensive approach exists. Integrating and improving the relationship of enterprise architecture and security is mainly relevant because of two reasons:

- **From a security point of view:** According to (Kreizman & Robertson, 2006), "security architecture work is performed outside of the EA group in most organisations. Combining work efforts of EA staff and security architects improve the probability that security requirements will be addressed throughout the enterprise."
- **From an enterprise architecture point of view:** Security becomes an integrated part of the enterprise, while its impact is still growing. For the enterprise architecture, claiming to provide a holistic view, security is an essential aspect. The growing impact of security on the organisation is illustrated by the rise of the cost of data security breaches over the past five years, as shown in Figure 2. The impact on the organisation of not having security and EA aligned and integrated becomes bigger. The number between the brackets indicates the amount of data breaches per capita in the corresponding year (Ponemon Research Institute, 2013).

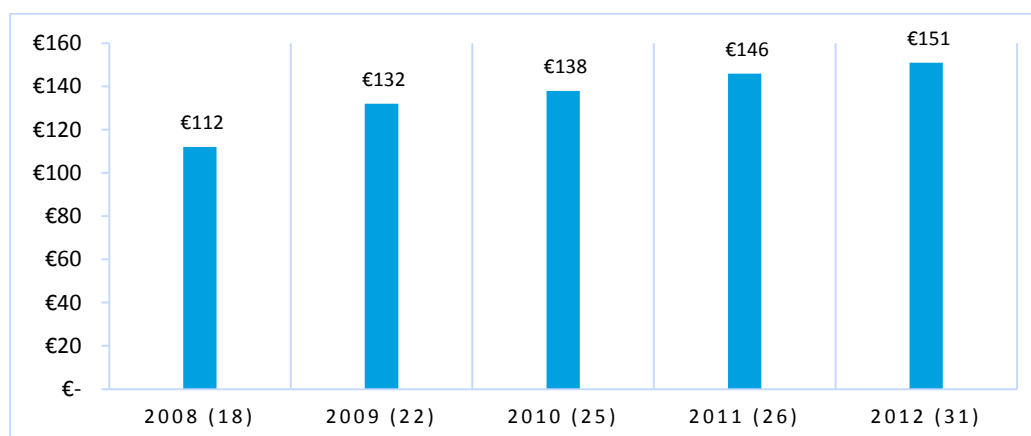


Figure 2: The average per capita cost of data breach over five years in Germany (Ponemon Research Institute, 2013)

Currently, no comprehensive approach exists for developing enterprise architecture with integrated security. In order to realize the above mentioned benefits, an integrated approach of enterprise architecture and security is needed. The development of this approach will be the main objective of this research.

3.2 RESEARCH OBJECTIVES

The main objective of this research is to develop an integrated approach of enterprise architecture and security. From literature it is derived, as identified later in chapter 4, that a comprehensive approach consists of three ingredients: a framework, a method and a modelling language. The research focuses on an integration between these elements separately, and on how these elements relate to each other to provide a comprehensive approach.

As identified in the literature review in chapter 4, the language-aspect of security is not very well defined in the literature yet. A focus on this aspect is needed in order to make the integration between enterprise architecture and security possible.

To conclude, the research objectives are to:

- Provide a description on how an integrated approach of security and enterprise architecture would look like, by:
 - Providing an integration of each of the ingredients separately;
 - Closing the missing gap of not having a graphical security modelling language;
 - Providing a description of the full approach.
- Validate this approach in the field by means of a case study.

3.3 RESEARCH QUESTIONS

The research questions are derived from the research objectives. Two main research questions are central in the research conducted. The first research question focuses on designing the integrated approach of security and enterprise architecture, including the validation of this approach. The second research question focuses on the integration of security in the graphical modelling language of enterprise architecture. The research questions and their sub questions are outlined below.

1. “What is a validated, comprehensive and integrated approach for designing secure enterprise architecture?”
 - 1.1. What is the current state of the Enterprise Architecture and Security discipline and their relation?
 - 1.2. Which elements are needed to provide a comprehensive approach, and what are their requirements?
 - 1.3. What does an integration of these elements look like?
 - 1.4. How can the proposed approach be demonstrated in a real-life situation?
 - 1.5. How can the proposed approach be validated?

2. “How can an enterprise architecture language be extended to incorporate security aspects?”
 - 2.1. What elements are needed to specify a modelling language?
 - 2.2. Which security concepts need to be merged into the enterprise architecture language?
 - 2.3. How can the language be validated?

3.4 RESEARCH METHODOLOGY

Since the question is a design problem, design science is conducted. One of the research methodologies for this type of research is the design science research methodology as proposed by Peffers, Tuunanen, Rothenberger, and Chatterjee (2007). A graphical representation of the approach is shown in Figure 3.

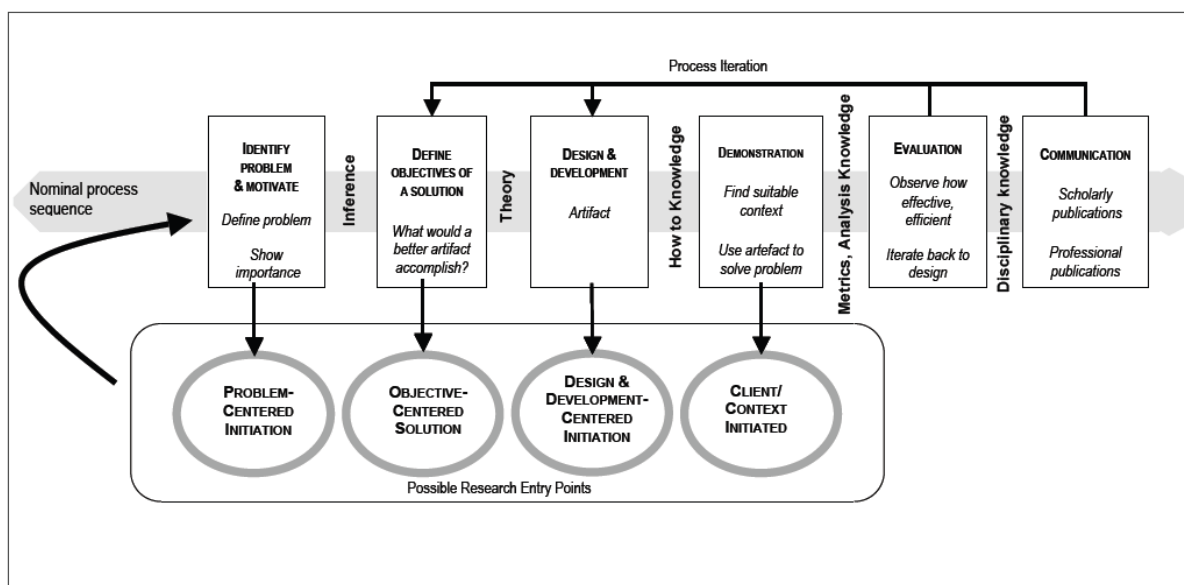


Figure 3: Design Science Research Methodology (DSRM) process model (Peffers et al., 2007)

An overview of the process steps, related research questions, and the corresponding sections is provided in Table 1.

A. Identify problem & Motivate

Define the specific research problem and justify the value of a solution. In section 3.1, the research problem is defined, and is further explored in the literature review in chapter 4.

B. Define objectives of a solution

Infer the objectives of a solution from the problem definition and knowledge of what is possible and feasible. The objectives of the solution are derived from the problem statement and described in section 3.2.

C. Design & Development

Create the artefact. The outcome of this phase is an approach for designing secure enterprise architectures. The approach consists of an integrated framework, method and a modelling language. The underlying design choices are explained in chapter 5.

D. Demonstration

Demonstrate the use of the artefact to solve one or more instances of the problem. The artefact will be demonstrated by applying the proposed integrated approach to a case. This demonstration is provided in chapter 6.

E. Evaluation

Observe and measure how well the artefact supports a solution to the problem. The evaluation is aimed at comparing objectives of the solution to the actual results derived in the design & development and evaluation phase. During this phase several experts in the field were interviewed. The outcomes of the interviews is discussed in chapter 7.

F. Communication

Communicate the problem and its importance, the artefact, its utility and novelty, the rigor of its design, and its effectiveness to researchers and other relevant audiences. The communication of the problem, its importance and the solution is done in this report.

Table 1: Relation between process, research questions and thesis outline

Process step	Relates to research question(s)	Discussed in section / chapter	Outcome
Identify problem & motivate	1.1	3, 4	Problem outline, research motivation
Define objectives of a solution	1.1	3, 4	Research objectives
Design & development	1.2, 1.3, 2.1, 2.2	5	Approach for designing secure enterprise architectures
Demonstration	1.4	6	A case where the approach is demonstrated
Evaluation	1.5, 2.3	7	An evaluation of the approach and its demonstration
Communication			Report

4 LITERATURE REVIEW

For this literature review, 27 publications from both enterprise architecture and security discipline are used. These papers were derived from a structured literature research, as explained in section 4.1. The literature is discussed in sections 4.2 till 4.6, and a conclusion is provided in section 4.7.

4.1 APPROACH

The aim of this literature review is to identify the state-of-the-art in the enterprise architecture and security disciplines and their relations. In order to do so, the EA-related security concepts as well as the security-related EA concepts are identified. With this approach, a view from both enterprise architecture and security discipline is provided. Within this view a distinction is made between analysis techniques, modelling techniques and design methods. The differentiation in analysis, model and design is also used by Gregor (2006), where they differentiate between I. Analysis, II. Explanation, III. Prediction, IV. Explanation and Prediction and V. Design and Action, in order to identify the state of research fields.

The approach is based on the systematic literature review approach as described by Webster and Watson (2002). According to them, two reasons for conducting a structured review exist. First, authors could deal with a mature topic where an accumulated body of research exists that needs analysis and synthesis. In this case, they would conduct a thorough literature review and then propose a conceptual model that synthesizes and extends existing research. Second, authors could tackle an emerging issue that would benefit from exposure to potential theoretical foundations. Here, the review of current literature on the emerging topic would, of necessity, be shorter. The author's contribution would arise from the fresh theoretical foundations proposed in developing a conceptual model. This research focuses on the latter type and by adopting a replicable scientific and transparent process, fresh theoretical foundations can be proposed.

Two types of literature review exist: author-centric and concept-centric. An author-centric approach would compare authors, based on the concepts they describe and the review then essentially presents a summary of the articles. The concept-centric approach compares the concepts, where authors and their statements on the same concept are compared. The concept-centric approach is used in this literature review, so concepts determine the organizing framework of the synthesis.

The search process for literature has been divided in four steps and three deliverables, as shown in Figure 4. A blue colour indicates a step in the process, where a green colour indicates a deliverable.

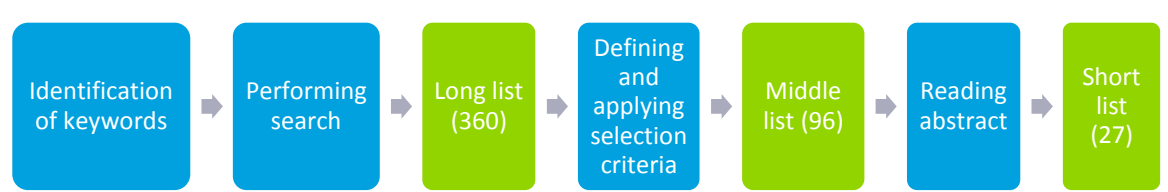


Figure 4: Literature review search process

4.1.1 LONG LIST

In the first step of the search for literature, Scopus and Google Scholar are used. An exploratory search is conducted in order to identify the keywords. The used search terms are 'enterprise architecture' and 'information security'. During the inspection of the results, it already appeared that the two fields have some overlap in the concept "enterprise (information) security architecture".

As described in the approach of the literature review, the identification of analysis techniques, modelling techniques and design methods is the main goal of the literature review. Therefore the respective terms "analysis", "model" and "design" are added to the term "enterprise security architecture". Furthermore, a search on these terms is repeated, but 'security' is left out of the quotes, in order to extend the view to security related papers.

The used search queries and the outcomes of the searches are depicted in Table 2.

Table 2: Search queries used in literature review

Search query	# Results in Scopus	# Results in Google Scholar
"enterprise security architecture"	22	468 (50)
"enterprise security architecture" analysis	6	379 (50)
"enterprise security architecture" model	9	379 (50)
"enterprise security architecture" design	4	395 (50)
"enterprise architecture" information security	89	~ 13.200 (50)
"enterprise architecture" information assurance	16	4.820 (50)
"enterprise architecture" security analysis	49	~ 12.100 (50)
"enterprise architecture" security model	68	~ 13.200 (50)
"enterprise architecture" security design	33	~ 13.100 (50)

The query in Scopus is performed in title, author and abstract, while Google Scholar searches in the full text. This partially explains the difference in the amount of results. It already appeared that Google Scholar returned far more results than could be considered due to the time constraints. Because the results in Google Scholar are ranked according to their relevance to the search query, weighing the full text of each document, only the top 50 results is taken into account. This approach is acceptable for the review, since the ranking is mainly based on how often and how recently it has been cited in other scholarly literature, as well as on where it was published, the author, and is furthermore based on the h-index of the authors. The top 50 results were all unique results within the search query (Beel & Gipp, 2009; Google Scholar, 2013).

Subsequently, the results of Scopus and Scholar are standardized in order to filter out the duplicates. After this filtering step, 360 unique publications remain in the long list.

4.1.2 MIDDLE LIST

The long list is used as input to create the middle list. A filter on the long list is needed to make sure only time is invested in the relevant literature in order to create a short list. A first cut is made based on three criteria. First, and foremost, is that the article is likely to contribute to answering the research questions. Second, it is important that the articles deals with the research fields of enterprise architecture and/or security, because those are the investigated research fields. Third, as some articles are superseded, it is important to take the most recent version of the papers into account.

The amount of citations is left out of the selection criteria, because recent developments within this research area might be missed when this criterion is applied.

Concluding, the long list is filtered to the middle list by applying the following criteria:

- R1: The article is likely to contribute to answering the literature review research questions.
- R2: The article is primarily dealing with Enterprise Architecture and/or Security.
- R3: The article has not been superseded.

These criteria are applied to the long list based on the title, the year of publication and the journal of publication. The relevancy for answering the research questions (R1) and research discipline (R2) is based on the title of the publication and the journal in which it is published. The year of publication is also taken into account to decide whether or not the article is likely to contribute to answering the research questions, since a more recent article will provide more information on recent developments in the disciplines. The year of publication is also used to determine whether or not the article has been superseded. After this filtering step, 96 publications remain in the middle list.

4.1.3 SHORT LIST

In order to create the short list, the publications from the middle list are retrieved and the abstract is considered. In order to select the relevant papers, requirements R1 and R2 are used again. Articles primarily focusing at enterprise architecture, as well as articles which did not mention the word 'security' or related concepts are left out of the scope.

After this selection, 27 articles remain in scope. A list of these articles is included in Appendix A – Literature Review Short List.

4.2 ENTERPRISE SECURITY ARCHITECTURE

The field of enterprise security architecture describes the security architecture throughout the enterprise. Framework and process are the elements found in this field of expertise. These elements are discussed in section 4.2.1 and 4.2.2 respectively.

4.2.1 ENTERPRISE SECURITY ARCHITECTURE FRAMEWORK

Several enterprise security architectures are currently accepted by the industry, these include:

- Gartner EISA
- Zachman framework
- SABSA framework & methodology

Three viewpoints are often included in the frameworks: business, information systems and technology. The *business architecture* models the organisation's business processes, roles, responsibilities and structure. It reflects the "business of security" and how information security interrelates with the way the organisation functions. The *information systems architecture* includes data, integration and application models used to operate the organisation. The *technology layer* encompasses the organisation's IT infrastructure, consisting of hardware, software and security requirements derived from the information architecture. None of the layers function in isolation or independently of other layers. Rather, the correlation across all three layers is the crux (Montelibano & Moore, 2012; Oda et al., 2009; Shariati et al., 2011).

The Gartner EISA framework, introduced in 2006, was the first attempt of introducing an enterprise security architecture by considering the compatibility of EISA with EA program and insisting on the collaboration of these two. This framework focuses on a description of the structure, but does not include a specific methodology for implementation (Shariati et al., 2011).

The Zachman framework is a layered architecture, consisting of six horizontal layers: contextual, conceptual, logical, physical, component, and functioning enterprise layer. Across these six horizontal layers, six vertical columns are placed, representing six essential aspects of the viewpoints: what, how, where, who, when, and why.

The Zachman framework does not specifically address security aspects. Nevertheless, due to the fact that Zachman is a comprehensive framework that is used within lots of businesses, it is adapted to include security nowadays. One of the advantages in this case is the use of an automated tool that provides consistent perspective of the enterprise architecture. Key players are able to add or remove parts of the model as business and technology change. However, an automated process cannot determine information security needs and requirements on its own; the tool is meant to manage complex information provided by key players (Burkett, 2012; Heaney et al., 2002; Oda et al., 2009).

SABSA is the most outstanding attempt of a holistic EISA. It has an identical structure compared to the Zachman framework, but differs in the sixth horizontal layer. Within SABSA, the sixth horizontal layer aims at security service management. The service management layer is for example concerned with the assurance of operation continuity and management of the environment.

SABSA offers a framework and methodology in such a way that it guarantees the security of enterprise information through a continuous process. Compared to the Gartner framework, which is rather abstract and theoretical, SABSA is more practical and includes its own specific

method for carrying out requirement engineering. The SABSA Matrix is included in Figure 5 (Burkett, 2012; Shariati et al., 2011; John Sherwood, Clark, & Lynas, 2005).

	ASSETS (What)	MOTIVATION (Why)	PROCESS (How)	PEOPLE (Who)	LOCATION (Where)	TIME (When)
CONTEXTUAL ARCHITECTURE	Business Decisions	Business Risk	Business Processes	Business Governance	Business Geography	Business Time Dependence
	Taxonomy of Business Assets, including Goals & Objectives	Opportunities & Threats Inventory	Inventory of Operational Processes	Organisational Structure & the Extended Enterprise	Inventory of Buildings, Sites, Territories, Jurisdictions, etc.	Time dependencies of business objectives
CONCEPTUAL ARCHITECTURE	Business Knowledge & Risk Strategy	Risk Management Objectives	Strategies for Process Assurance	Roles & Responsibilities	Domain Framework	Time Management Framework
	Business Attributes Profile	Enablement & Control Objectives; Policy Architecture	Process Mapping Framework; Architectural Strategies for ICT	Owners, Custodians and Users; Service Providers & Customers	Security Domain Concepts & Framework	Through-Life Risk Management Framework
LOGICAL ARCHITECTURE	Information Assets	Risk Management Policies	Process Maps & Services	Entity & Trust Framework	Domain Maps	Calendar & Timetable
	Inventory of Information Assets	Domain Policies	Information Flows; Functional Transformations; Service Oriented Architecture	Entity Schema; Trust Models; Privilege Profiles	Domain Definitions; Inter-domain associations & interactions	Start Times, Lifetimes & Deadlines
PHYSICAL ARCHITECTURE	Data Assets	Risk Management Practices	Process Mechanisms	Human Interface	ICT Infrastructure	Processing Schedule
	Data Dictionary & Data Inventory	Risk Management Rules & Procedures	Applications; Middleware; Systems; Security Mechanisms	User Interface to ICT Systems; Access Control Systems	Host Platforms, Layout & Networks	Timing & Sequencing of Processes and Sessions
COMPONENT ARCHITECTURE	ICT Components	Risk Management Tools & Standards	Process Tools & Standards	Personnel Man'ment Tools & Standards	Locator Tools & Standards	Step Timing & Sequencing Tools
	ICT Products, including Data Repositories and Processors	Risk Analysis Tools; Risk Registers; Risk Monitoring and Reporting Tools	Tools and Protocols for Process Delivery	Identities; Job Descriptions; Roles; Functions; Actions & Access Control Lists	Nodes, Addresses and other Locators	Time Schedules; Clocks, Timers & Interrupts
SERVICE MANAGEMENT ARCHITECTURE	Service Delivery Management	Operational Risk Management	Process Delivery Management	Personnel Management	Management of Environment	Time & Performance Management
	Assurance of Operational Continuity & Excellence	Risk Assessment; Risk Monitoring & Reporting; Risk Treatment	Management & Support of Systems, Applications & Services	Account Provisioning; User Support Management	Management of Buildings, Sites, Platforms & Networks	Management of Calendar and Timetable

Figure 5: SABSA Matrix (John Sherwood et al., 2005)

To summarise, Gartner’s EISA framework tries to integrate security and EA, however it is not as detailed as, e.g. the Zachman framework or SABSA. Zachman is more specified, but does not explicitly address security aspects. SABSA focuses on security, however it does not include EA elements.

4.2.2 ENTERPRISE SECURITY ARCHITECTURE METHOD

Specific methods for designing enterprise security architecture have been developed. In this literature review, two methods were identified: the RISE methodology, and the SABSA Lifecycle.

The RISE methodology is rather a method than a framework, so it prescribes how to develop an enterprise security architecture by incorporating security and privacy features into business processes. RISE encompasses three phases: profile, plan and protect. The profile-phase bases a risk assessment on the as-is architecture. Accordingly, requirements & policies are created, and control alternatives and policies are derived. These form the basis for the To-Be architecture. Finally, plans, systems and schedules form the operation deployment in the protection-phase. Although it is very comprehensive from the processes point of view, the downside is that it is not based on one specific framework (Shariati et al., 2011).

Also, the SABSA approach includes a method, which is called the SABSA lifecycle. It consists of four parts: strategy & planning, design, implement and manage & measure. The first phase sets the goal and vision of the architecture. The design phase embraces the design of the logical, physical, component and service management architectures. The third activity is 'Implement', followed by 'Manage and Measure' (J Sherwood, Clark, & Lynas, 2009).

4.3 ANALYSIS

In order to manage and improve something, it is necessary to be able to analyse the current state of affairs. This analysis is at the core of making rational decisions about information systems and enterprise architecture. In order to perform an analysis, information about the involved systems and their organisational context is required for a good understanding of their data quality. It is for instance reasonable to believe that a firewall has a positive influence on the probability that a system or network of systems is more secure. The availability of the firewall is thus one factor that has an effect on the security and should therefore be recorded in a scenario model. The person that carries out the analysis and builds the model should therefore have an understanding of the context, what information to gather and also ensure that this information is collected and modelled accordingly (Buschle, Ullberg, Franke, Lagerström, & Sommestad, 2011; Johansson & Johnson, 2005).

Analysis can be divided in qualitative and quantitative analysis, and each type of analysis has its advantages and disadvantages associated. Quantitative analysis aims at a statically reliable and generalizable result, based on observations. Qualitative analysis aims at a complete and detailed description, with no attempt to assign frequencies to the features which were identified. Both types of analysis can be found in the literature on the topic of enterprise security architecture. Some analysis models are specifically designed for this type of architectures, others are more generalized, but use the subject of security or cyber security to provide an example or validation to the audience.

Most frameworks are based on standards like ISO 27000 series, or NIST (NIST, 2012). A common denominator for all these guides and standards is that they can all be considered as theoretical frameworks for how to achieve security. None of these works are ensuring "complete" security. A common problem with standards like ISO 27000 series or NIST, is that it is not clear how all the different promoted features and mechanisms are related to each other and if some are more important than others. For example, firewalls have a positive effect on the level of security, but are firewalls more or less important than a security awareness program? Is it taken into account that an awareness program might increase the chance that a firewall is correctly configured? The strength and structure of the causal relations are typically not addressed in the standards and guides (Ekstedt & Sommestad, 2009).

4.3.1 QUANTITATIVE ANALYSIS

A quantitative risk assessment provides results in numbers that management can understand, whereas a qualitative approach, although easier to implement, makes it difficult to trace generalized results. A quantitative final risk measure allows for testing, improvement, comparing and budgeting as opposed to attributes such as high, medium, or low, which cannot be managed or quantified numerically for an objective assessment (Sahinoglu, 2005).

A. Assessment with Architecture Theory Diagram

Johansson and Johnson (2005) propose a single quantitative estimate of the level of Enterprise Information Security in a company by providing a list of 1365 questions from standards: ISO/IEC, NIST, ISF and OCTAVE. The questions are clustered by dimension Scope (technical, organisational, environmental), Purpose (preventive, detective, responsive) and Time (planning, operational, controlling). Answering (a selection of) the questions provides a result on a scale which indicates the level of Enterprise Information Security in the researched company. In a validation attempt of the method, the enterprise CISO confirmed that the result from this survey did correspond well to their common feeling of the possible Enterprise Information Security level (Johansson & Johnson, 2005).

B. Attack and Defence trees

Attack trees are a graphical notation evolved from fault trees, where the main goal of an attacker is depicted as the root of a tree. The steps to reach the goal are broken down into sub-goals (nodes) of the attack through “AND” and “OR” relationships, which represent mandatory or optional steps respectively. These trees can be used to answer questions about the current security status and facilitate comparison with previous measurements, but does not answer the question how to improve security. A natural extension of the attack trees are the defence trees, where countermeasures are also included (Ekstedt & Sommestad, 2009). An example of an attack and defence tree is included in Figure 6.

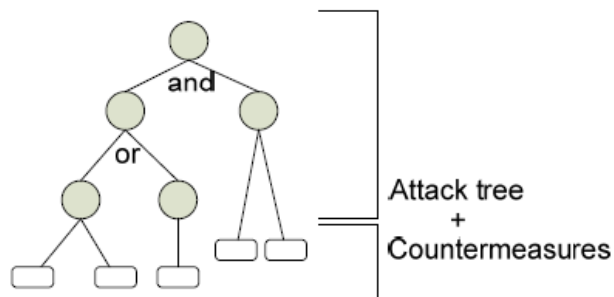


Figure 6: The defence tree concept (Ekstedt & Sommestad, 2009)

Sahinoglu (2005) proposes a security meter, which helps calculating residual risk based on the lack of countermeasures identified in a decision-tree. The decision-tree, of which an example is included in Figure 7, encompasses the vulnerabilities (indicated with ‘Vx’), the threats (indicated with ‘Tx’), and the cost of countermeasures (CM) or lack of countermeasures (LCM). Without the use of this probabilistic framework such as the one suggested, the conclusions to assess a risk’s severity might be misleading and costly due to over- or underestimation of the risk scenario.

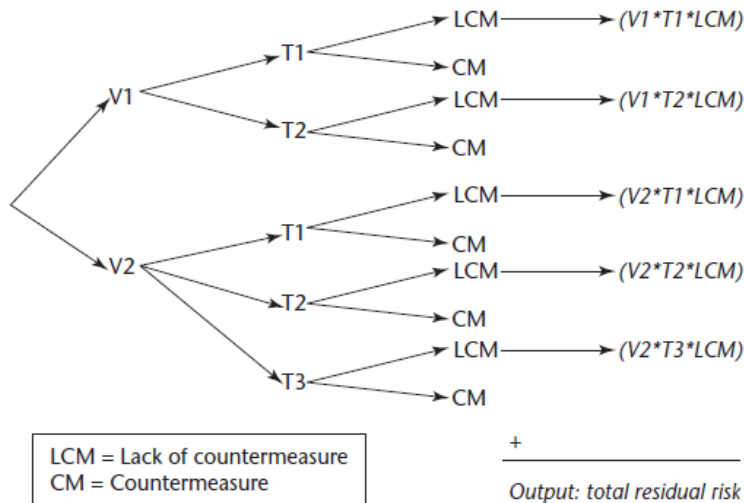


Figure 7: A general-purpose decision-tree diagram example for the Security Meter model (Sahinoglu, 2005)

The decision tree frameworks as discussed here are aimed at measuring security, but do not include enterprise architecture aspects yet. The drawback of attack and defence trees is that they can grow extensively if several goals and sub goals are of interest. Therefore, Bayesian networks have been proposed.

C. Bayesian Networks and extended influence diagrams

A Bayesian network is a probabilistic graphical model that represents a set of variables and their dependencies. Influence diagrams are an enhancement of Bayesian networks and a powerful modelling approach. These are used to depict and analyse complex causal interplay between properties. As illustrated in the example diagram in Figure 8, extended influence diagrams can be used to represent defence trees. A utility node can be used to represent the consequence of successful attacks and steps required for their success can be decomposed in a number of sub steps. Attack steps will assume the state “Success” or “Failure”, each with a certain probability, influenced by the countermeasures. Based on the scenario chosen the states of the countermeasures will differ represented by decision nodes that influence the state of countermeasures (Ekstedt & Sommestad, 2009).

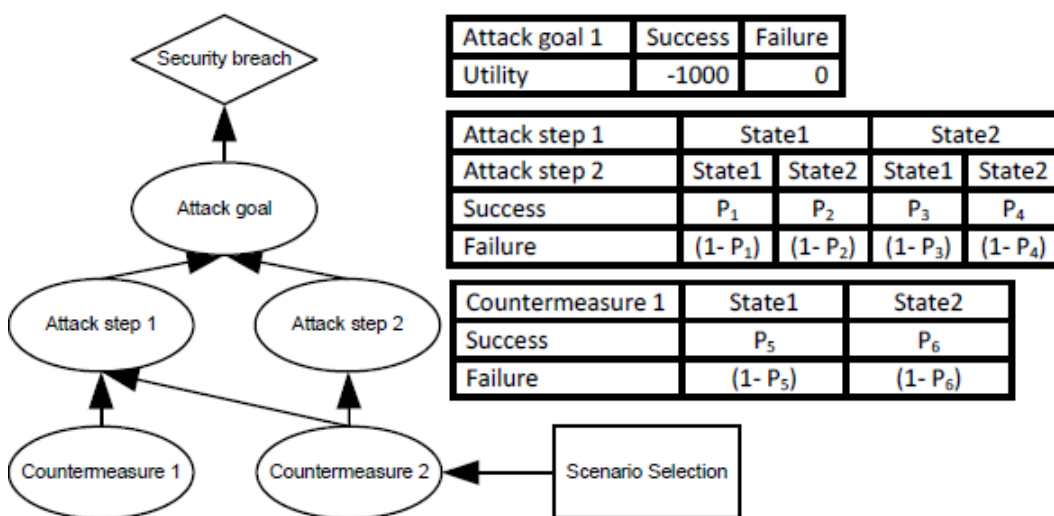


Figure 8: Syntactic elements of extended influence diagrams and a simple example (Ekstedt & Sommestad, 2009)

In the example outlined in Figure 8, the cost of a security breach is 1000. This is indicated by the utility node of attack goal 1; when this goal is successfully reached this result in a loss of 1000. The goal can be decomposed in attack steps with different chances of success and corresponding failure. The chance of successfully executing the attack steps depends on the countermeasures taken. In this example, countermeasure 1 can have two different states (state 1 and state 2) indicating, for example, a weak and a strong firewall (P5 and P6). The state of the countermeasure has an effect on the successfulness of reaching the attack goal through the various attack steps, resulting in the various chances of success (P1 – P4)

One important feature of the Bayesian formalism is the possibility to learn from previous data and create powerful statistical models for accurate assessments (e.g. on cyber security). Since influence diagrams include decision and utility nodes, predicted losses from successful attacks can be included in the models.

4.3.2 QUALITATIVE ANALYSIS

Kim (2011) proposes a reference model for IT compliance, based on best practices which are mainly described in BS7799, ISO27000 and ISO13335 standards. A best practice mentioned in the paper is for example ‘Mission critical systems and record management systems should be located in an environmentally friendly area. Access to computer hardware, wiring, displays and network should be controlled by rules of least privilege. Systems which monitor and audit a physical access to computer hardware, wiring, displays and networks should be implemented’. The corresponding mitigating controls then include Environment control, Power protection, Water damage controls, HVAC, Smart cards, Fire prevention and suppression and Physical access controls. The best practices can be found in some parts of BS7799 and ISO27000.

By using this reference model for IT compliance, one can improve quality of internal auditing plans and is able to analyse the current state of affairs for information security within the company. Furthermore, the model can be used to manage an auditing project or train auditors to improve the auditors’ capability.

4.4 MODEL

The current approach to protect distributed systems in large organisations is based on a security policy document. This document balances the functional business requirements of the application, the security requirements, legal and regulatory aspects and other factors like costs. This approach raises several issues however, for example policy correctness and consistency, even for quite simple client/server applications. It is the question how to ensure that the high level policy is correctly mapped to a high number of access control rules and configurations of security systems. This approach is error prone, causes high maintenance costs and requires a lot of resources. Furthermore, the challenge in security is not in the elements as encryption or access control, but in the protection of the system as a whole (Buschle, Holm, Sommestad, Ekstedt, & Shahzad, 2012; Lang & Schreiner, 2008).

Therefore the current approach to security will not be sufficient anymore: in order to provide a consistent view, security requirements and the enterprise architecture need a holistic modelling approach. Without knowledge of the whole, knowledge about the details serves

little purpose in a complex world. This is also demanded by regulatory and legal requirements, which require a higher level of security, including the proof that the system is sufficiently protected. A model transforms part of the real world, e.g. an organisation, into model conditions (Ekstedt & Sommestad, 2009; Hensel & Lemke-Rust, 2010; Lang & Schreiner, 2008).

Currently, information security is modelled in one of the two approaches. Frequently, the security aspects of a system are designed and analysed separately from the rest of the architecture, and are therefore not well integrated, leading to a potential of new vulnerabilities. Alternatively, the security aspects are designed implicitly in the architecture, so that they cannot be extracted. This leads to a lack of analysability of the security aspects. Thus there is a need for practical guidance on how enterprise engineers not qualified as security engineers can include security aspects on the holistic approach of architectures (Heaney et al., 2002; Hensel & Lemke-Rust, 2010; Lang & Schreiner, 2008; Shin & Gomaa, 2007).

A major difficulty of implementing security policies is in the fact that these policies are expressed at a high level of abstraction, that is organisation-, business-, or information-centric, but often not IT-centric or expressed in IT-terms. Furthermore, compliance monitoring is hard, how can an organisation demonstrate that it complies with regulations? Doing this manually is too slow, costly and error-prone. These two difficulties are even harder to tackle within distributed IT environments which get reconfigured regularly.

Benefits of model driven security include a regulation of information flows and resource access between the various systems of an organisation and its users; it helps aligning business security requirements and policy-driven technical security enforcement. Furthermore, the cost/effort savings can be significant because multiple rules can be generated and maintained automatically while also providing a link to the business enterprise architecture, which ensures that the needs of the business are reflected (Lang & Schreiner, 2008).

In order to do so, several approaches exist: Heaney et al. (2002) propose the use of security patterns, Buschle et al. (2012) provides a tool for automatic enterprise architecture modelling with a running example on cyber security.

A. Using patterns

Heaney et al. (2002) also mentions the importance of linking the IT element of information security to the enterprise business needs, because the purpose of enterprise architecture is to insure that IT effectively supports the needs of the business.

It is important to address all levels of conceptualization, both on system level but also on enterprise level, and to address all levels of composition, from a system of systems to the smallest unit. This can be achieved by using patterns. Patterns have been successfully used in systems and software communities to capture and share knowledge about well-known and successful solutions to common technical problems. By providing an integrated system of patterns and their detailed representations, the level of security can be improved and understood by enterprise engineers not qualified as security engineers.

A detailed example of the pattern for Identification & Authentication (I&A) is included in Figure 9. The framework indicates the various level of abstraction, ranging from Scope and

Business Model, to system model, technology model and the detailed representations. Accordingly, characteristics can be defined per level and solution. For example for the solution option 'Biometrics', the costs are moderate to high per user and there are high costs per entry point, but it provides a solution when tokens or passwords are not acceptable and the entry point is physically insecure (Heaney et al., 2002).

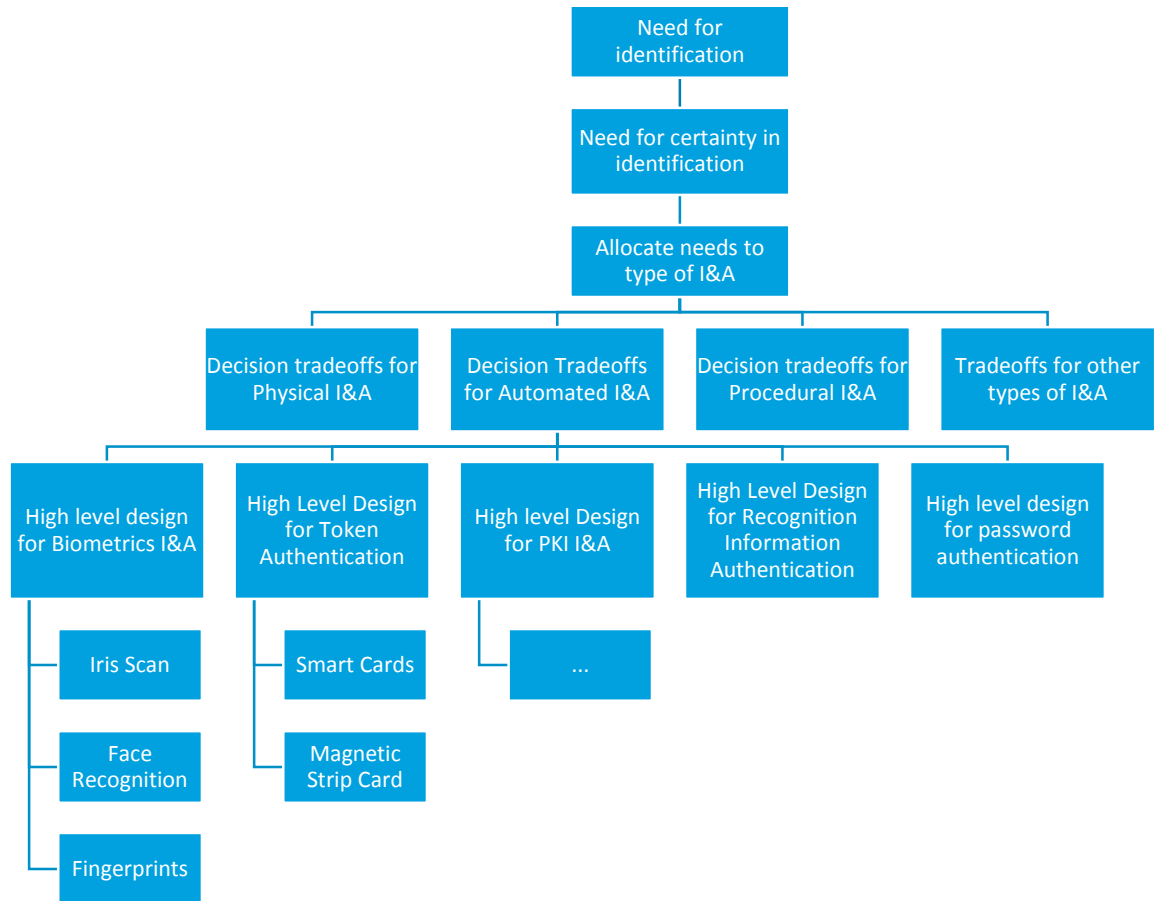


Figure 9: Identification & Authentication Pattern Tree (Heaney et al., 2002)

B. Automatic generation

Another approach for designing an enterprise security architecture is by generating it automatically. In a study conducted by Buschle et al. (2012), a tool is proposed to automatically instantiate elements in enterprise architecture models based on results from network scans. This approach mainly focuses on the application and technology layer of the organisation.

The information needed to construct the architecture is gathered through the application of a vulnerability scanner that evaluates the structure of an enterprise.

For this purpose, a vulnerability scanner is needed. Examples of this tool include NeXpose, Nessus, and OpenVAS. In the research conducted by Buschle et al. (2012) NeXpose is used, which is an active vulnerability scanner capable of both authenticated and unauthenticated scans. Active means that it queries remote hosts for data and it does this by providing user accounts to hosts (authenticated) or without (unauthenticated). Authenticated scans are typically less disturbing to normal operations and provide a higher degree of accuracy, but

credentials are not always available. The result of the scan is a network architecture including all devices communicating over TCP or UDP, e.g. firewalls and printers, and their operating systems or firmware and any services that are running. If the scanner is given credentials, all applications and versions, and user / administration accounts are also included. It is able to detect both software flaws and configuration errors from the security point of view. It scans for approximately 53.000 vulnerabilities.

This tool proves to be extremely useful for automatic generation of the infrastructure and application layer of the enterprise architecture. In a case study, it took less than an hour to create the EA model for a network of 20 physical computers and 28 virtual machines (Buschle et al., 2012).

Other tools for modelling an enterprise security architecture are mentioned in section 4.2.1.

C. Modelling language

Known graphical modelling languages in security discipline are Rei, UMLsec, and SI* (Ekstedt & Sommestad, 2009). Rei is a policy modelling language. It allows for specifying different types of policies in terms of rights, obligations, dispensations, and prohibitions (Kagal, Finin, & Joshi, 2003).

UMLsec is an extension to UML and has a focus on security requirements for systems engineering. It allows for integrating security related information in UML specifications, by enabling software developers with a background in security to make use of security engineering knowledge and capsule it in a widely used design notation (Jürjens, 2002).

SI* is a security requirements engineering modelling language, which encompasses constructs for actors, roles, goals and their dependencies. It is an agent-oriented methodology, which supports the modelling of the social context in which the system-to-be will operate, and is therefore especially useful in requirements-engineering (Massacci, Mylopoulos, & Zannone, 2010).

Other current modelling languages that have been tailored for security, like UMLsec, secure UML and Misuse cases, provide good support for detailed modelling of concerns such as access control, and formal validation of security design. However, they lack a holistic scope and do not represent the broad spectrum of security. Also, alignment with other system topics of interest such as maintainability, performance, functionality, and business alignment is lacking. This requires an enterprise wide architecture viewpoint (Ekstedt & Sommestad, 2009).

A well-known modelling language in the enterprise architecture discipline is ArchiMate (M.E. Iacob, 2012). This language is becoming widely accepted as the de-facto standard for the specification of enterprise architecture models and views. Three layers are distinguished: the business layer, the application layer, and the infrastructure layer. In addition, the language considers structural, behavioural, and informational aspects within each layer.

The drawback of the discussed security modelling languages is that they are limited in scope on policies (Rei) or software development (UMLsec and SI*), and do not cover the enterprise architecture discipline. ArchiMate, on the other hand, does not include security-specific concepts.

4.5 DESIGN

Designing information security from a purely technical perspective is believed to have a high tendency of failure. Both literature and the industry agree upon having a more holistic approach is vital in order to secure the enterprise and its assets. Similar to how a building architect designs security into a building, an enterprise security architecture has to assimilate security elements throughout the layers of an organisation and has to align these with the current enterprise architecture (Burkett, 2012; Montelibano & Moore, 2012; Park, Ahmad, & Ruighaver, 2010).

Today, the security department is often seen as the business prevention department, keeping the business from innovation and creating value. This is in contrast to what most security architects aim for: security as a business enabler. Information security should enable a business to take the risks it is prepared to take on, by designing and deploying countermeasures that allow for sensible business risk (Liu, Sullivan, & Ormaner, 2001; Peterson, 2007).

All approaches to designing enterprise security architectures apply the pattern as outlined in Figure 10. Important in this process is the continuous loop, where most process models aim for. Defining and updating the enterprise security architecture within the organisation is a continuous process, because the environment is constantly changing. Kim and Leem (2004) argue for an environment analysis prior to the development of AS-IS and TO-BE architectures, identifying both business and technical environment. The business environment focuses both on external and internal environment, including threat of entry, and powerful suppliers; and buyers, management planning, financing, and research. They include this environment analysis in order to identify the competitive environment of the business and the technical trends, which have an impact on the risk analysis in the TO-BE architecture (Kim & Leem, 2004; Liu et al., 2001).

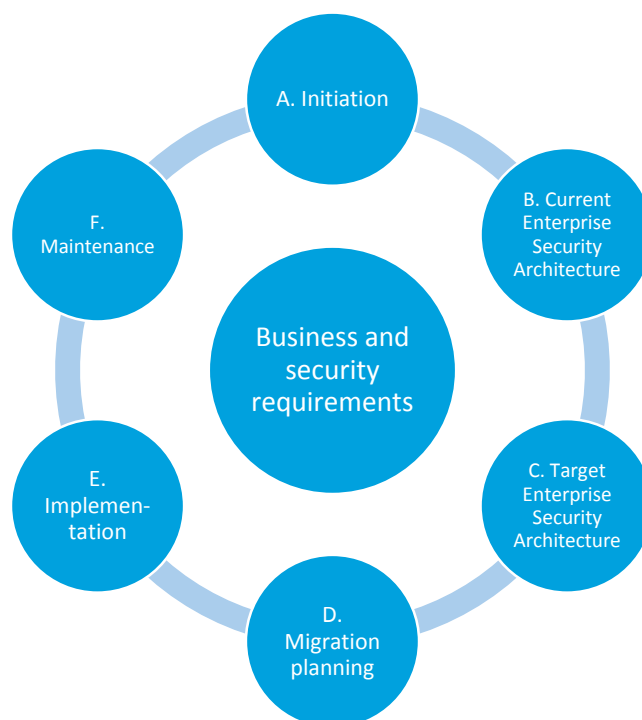


Figure 10: Enterprise Security Architecture design process, adapted from Liu et al. (2001)

Liu et al. (2001) divide the activities for each step of the process in technical and business concerns. These concerns are summed up in Table 3. For each process step in Figure 10, the table indicates which technical and business activities one should consider.

This overview balances the technical and business concerns, where the involvement of the business is primarily required for the preparation of implementation in the business. Current approaches require more involvement of the business in weighing up risks and countermeasures (Burkett, 2012; Liu et al., 2001).

Table 3: Activities in six-step process to develop Enterprise Security Architecture, as outlined in Figure 10. (Liu et al., 2001)

Process step	Technical activities	Business activities
A. Initiate the effort	Develop a high-level security policy and guidelines. Establish the security team	Create a readiness for enterprise IT security. Overcome resistance to change. Identify and influence stakeholders. Encourage open participation and involvement
B. Describe where you are	Conduct detailed risk analysis. Characterize the baseline security architecture	Reveal discrepancies between current and desired states. Make it clear to everyone why the organisation must change. Convey credible expectations
C. Identify where you would like to be	Develop the target security architecture	Communicate the target security architecture's valued outcomes and features Energize commitment
D. Plan how to get there	Develop transition plans	Create a plan for transition activities. Communicate the transition plan. Establish a sound management structure
E. Execute plan to go there	Implement the target security architecture	Build support for the security team. Develop new security competencies and skills. Practice security incident responses.
F. Keep the initiative alive and well	Operate and monitor security operations. Enhance the target security architecture	Reinforce security practices Communicate the valued outcomes.

Other tools for designing an enterprise security architecture are mentioned in section 4.2.1.

4.6 HYPERCONNECTIVITY AND INTEROPERABILITY

The emergence of internetworked systems has given corporations and government agencies the opportunity to share information in unprecedented fashion, recognized by Haigh (1995) already. However, there are significant security implications in this trend. An enterprise must not only protect the confidentiality, integrity and availability of its own information, but also of the virtual organisation to which it belongs. The sharing is not only geographically across a single enterprise, but it can also be distributed across several enterprises. This raises several complex research questions on privacy, information security and trust (Pulkkinen, Naumenko, & Luostarinen, 2007).

Consequently, security is inherently suffering from a weakest-link syndrome (Ekstedt & Sommestad, 2009). Enterprise Architecture is proposed as a means for comprehensive and coordinated planning and management of corporate ICT and the security infrastructure. Pulkkinen et al. (2007) conducted a study which provides an example of security architecture planning based on enterprise architecture, which aligns the development of technological solutions with business goals. By choosing for an enterprise architecture approach, the planning of business and ICT developments is combined.

Shariati et al. (2011) conducted a review on Enterprise Information Security Architectures from an interoperability perspective. Three interoperability aspects were included: technical, organisational and semantic interoperability. The result was that the role of information security in interoperability was often neglected. It seems that much practical research should be done so that the two incompatible quality attributes of security and interoperability could be implemented along with each other. A summary of the results is included in Figure 11.

Interoperability Aspects	Frameworks				
	Gartner	SABSA	RISE	AGM-based Model	Intelligent SOA-based EISA
Technical	UK	ES	NI	NI	ES
Organizational	IS	ES	IS	IS	ES
Semantic	ES	IS	UK	IS	ES

IS=Implicit Support, ES= Explicit Support, NI= Not Included, UK= Unknown

Figure 11: The comparison of prominent EISA frameworks from interoperability perspective (Shariati et al., 2011)

4.7 CONCLUSION

The goal of the literature review was to identify the state of the combined research field of enterprise architecture and security and the relation between these concepts. This goal is reached by discussing the various subjects in the preceding sections.

By researching the various existing frameworks, it becomes apparent that architectures for security exist, architectures for enterprises exist, but enterprise architectures with integrated security do not. At least this was not discovered within this structured literature review. However, by examining the applications of a structured security approach, especially in analysis and modelling techniques, the relation between the business processes, information objects and IT assets on the one hand, and security on the other hand, seems very useful.

Analysis can be done both quantitative and qualitative, where each of these has its advantages and disadvantages. Both of these approaches discuss security in terms of attacks and countermeasures related to the various aspects of an enterprise architecture. Once the enterprise architecture components and their relations are determined, a risk analysis can be conducted and countermeasures can be put in place.

In order to achieve this application, a comprehensive secure enterprise architecture approach is needed. Within the enterprise architecture discipline, three ingredients are identified; a framework, a method, and a language (Iacob, Jonkers, et al., 2012). These ingredients correspond to the core components of any information system design theory, as described by Gregor and Jones (2007): principles of form and function, principles of implementation, and constructs.

- A *framework* for the subdivision of an architecture in different domains, sometimes including the relationships between these domains.
- A *method*, or a way of working, which is in most cases a step-wise prescriptive method for developing architectural descriptions.
- A (modelling) *language*, defining the concepts for describing an architecture, including a (preferably graphical) *representation* of these concepts.

These are the three ingredients used in the solution design.

5 SOLUTION DESIGN

The solution design consists of three components: framework, method and modelling language. The concepts are described and investigated in section 5.2, 5.3, and 5.4 respectively. The solution design is then discussed in section 5.5.

5.1 LINE OF REASONING

According to Iacob, Jonkers, et al. (2012), a comprehensive approach for enterprise architecture consists of three essential ingredients: a framework, a method, and a modelling language. Moreover, it should become clear in the approach how these three concepts relate to each other. The concepts are described in further detail below.

A *framework* provides the various existing viewpoints on an architecture, and subdivides the architecture in different domains, sometimes complemented with the relationships between these domains. In enterprise architecture, examples include the Zachman Framework, NIST Enterprise Architecture, and Federal Enterprise Architecture Framework.

A *method* provides a step-wise prescriptive approach for developing the architecture, from scratch or from existing models. In the enterprise architecture field, the TOGAF ADM approach is well-known for example. But also the Federal Enterprise Architecture covers a development method.

The final ingredient of a comprehensive approach is a *language*, which defines the concepts for describing an architecture. This can be both in natural language or graphically, where the latter one is preferred. Examples include UML and ArchiMate.

The relation between these ingredients is depicted in Figure 12.

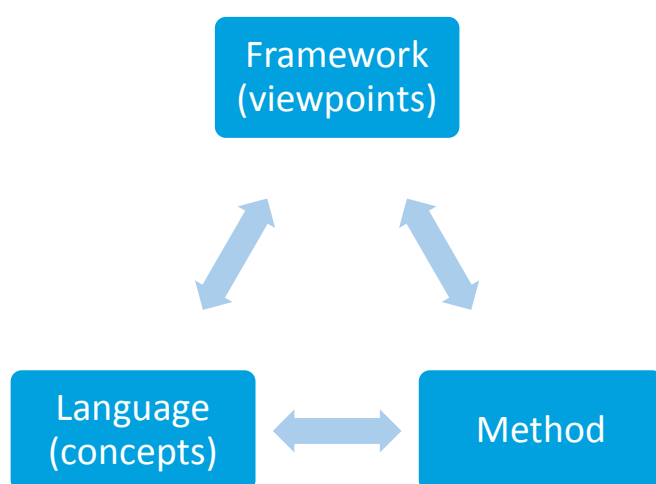


Figure 12: Ingredients of an Enterprise Architecture approach (Iacob, Jonkers, et al., 2012)

A comprehensive approach, where security and enterprise architecture are integrated, would also consist of a framework, method and modelling language. Furthermore, for each of these ingredients, it should be described how they relate to the ingredient of the other discipline, and also how they relate to the full approach.

The literature review conducted and described in chapter 4 provides insight in the current state of affairs. A framework often used in security discipline is SABSA, which has the same structure as the Zachman framework. As a method, one could use the ESA development cycle (Liu et al., 2001), which describes the common steps in order to create an enterprise security architecture. A suitable graphical security modelling language is not discovered in the extensive literature review. Some languages support parts of security though, but none of them covers the broad topic of security. Therefore, a definition of a language or a language extension that allows modelling security, and describe its relation to ArchiMate is demanded. An elaboration on the choice of the framework, method and language is provided in the following sections. An overview is depicted in Figure 13.

	Enterprise Architecture	Security
Framework	Zachman	SABSA
Method	TOGAF	Security implementation approach / SABSA Lifecycle
Language	ArchiMate	<i>To be defined</i>

Figure 13: Essential ingredients of an integrated approach for EA and Security

5.2 FRAMEWORK

As discussed in chapter 4, several architecture frameworks exists. In order to accomplish the formulation of the framework, an integration of Zachman and SABSA is chosen.

The most important reason for using the Zachman framework is because of its wide acceptance in the enterprise architecture discipline, both in research and industry. It is relatively old, but still relevant, used and well-known. Furthermore, it is comprehensive in the sense that it provides several views. Currently, security is not explicitly addressed in the Zachman framework. (van Gansewinkel & Hofman, 2012).

The most important reason for using SABSA is because it is an open standard, comprising a number of security artefacts. It is also well-known and the de facto standard for security frameworks.

The Zachman framework is discussed first, then the SABSA framework will be elaborated, followed by a description of the relation between these two frameworks.

5.2.1 EA: ZACHMAN FRAMEWORK

The original concept of the Zachman Framework has been introduced by John Zachman in 1987 and was one of the first frameworks proposed for an architecture. It is extended in 1992 and still popular. The framework is a structured set of essential components of an object for which explicit expressions are necessary for creating, operating, and changing the object. It is explicitly not a methodology for creating the implementation of the object, this would be the process. An outline of the framework is depicted in Figure 14.

	WHAT	HOW	WHERE	WHO	WHEN	WHY
	DATA	FUNCTION	NETWORK	PEOPLE	TIME	MOTIVATION
SCOPE (Contextual) Planner	List of things important to the business Entity = Class of business things	List of processes the business performs Process = Class of business process	List of locations in which the business operates Node = Major business locations	List of organisations important to the business People = Major business unit	List of event cycles significant to the business Time = Major Business Event Cycle	List of business goals/strategies End/Mean = Major Business Goal/Strategy
BUSINESS MODEL (Conceptual) Owner	e.g., Semantic Model Entity = Business Entity Relationship = Business	e.g., Business Process Model Process = Business IO = Business Resource	e.g., Business Logistics System Node = Business Location Link = Business Linkage	e.g., Workflow Model People = Organisation unit Work = Work Product	e.g., Master Schedule Time = Business Event Cycle = Business Cycle	Business Plan End = Business Objective Means = Business Strategy
SYSTEM MODEL (Logical) Designer	e.g., Logical Data Model Entity = Data Entity Relationship = Data Relationship	e.g., Application Architecture Process = Application Function IO = User Views	e.g., Distributed System Model Node = I/S Function Relationship = Line Characteristics	e.g., Human Interface Architecture People = Role Work = Deliverable	e.g., Processing Structure Time = System Event Cycle = Processing Cycle	e.g., Business Rule Model End = Structural Assertion Means = Action Assertion
TECHNOLOGY MODEL (Physical) Builder	e.g., Physical Data Model Entity = Segment/Table Relationship = Pointer/Key	e.g., System Design Process = Computer Function IO = Data Elements/sets	e.g., Technology Architecture Node = H/w /System s/w Relationship = Line Specifications	e.g., Presentation Architecture People = User Work = Screen Formats	e.g., Control Structure Time = Execute Cycle = Component Cycle	e.g., Rule Design End = Condition Means = Action
DETAILED REPRESENTATIONS (Out-of-context) Subcontractor	e.g., Data Definition Entity = Field Relationship = Address	e.g., Program Process = Language Statement IO = Control Block	e.g., Network Architecture Node = Address Link = Protocol	e.g., Security Architecture People = Identity Work = Job	e.g., Timing Definition Time = Interrupt Cycle = Machine Cycle	e.g., Rule Specification End = Sub-condition Means = step
FUNCTIONING ENTERPRISE	e.g DATA	e.g FUNCTION	e.g NETWORK	e.g ORGANISATION	e.g SCHEDULE	e.g STRATEGY

Figure 14: The Zachman framework for Enterprise Architecture (version 2003)

The framework contains six rows, each providing a different viewpoint; and six columns, where the focus is on the same fundamental questions. The rows will be discussed first, followed by an elaboration on the columns.

The rows represent different viewpoints, but do not imply a hierarchical structure. An upper row does not necessarily have a more comprehensive understanding than a lower perspective. Each row represents a distinct, unique viewpoint. Still, each viewpoint should take the requirements of the other viewpoints into account and the constraints that these viewpoints impose. The constraints are additive, so the requirements of the higher viewpoint do affect the constraints of the lower rows, while the opposite is not necessarily true.

- Planner's Viewpoint (Contextual) – In this viewpoint, the “ballpark view” is defined. It starts with some concepts, which are specifications for the “ballpark” in which they intend to manufacture. It contains for example specifications for the product that it will fly so high, so fast, so far, and for which purpose it is constructed.
- Owner's Viewpoint (Conceptual) – In this viewpoint, the work breakdown structure is outlined. Within this viewpoint it is specified what work will be accomplished in terms of components and systems.

- Designer’s Viewpoint (Logical) – The designer translates the work breakdown structure into physical products. Within this view, the detailed requirements are created.
- Builder’s Viewpoint (Physical) – The builder must redraw the plans of the designer to represent de builder’s perspective, including all technical details and constraints of supporting technology.
- Subcontractor Viewpoint (Detailed representations) – The subcontractors build the pieces from the detailed representations.
- Actual System (Functioning enterprise) – Finally, the functioning enterprise is built.

The Zachman Framework also contains six columns, focussing on six fundamental questions: What, How, Where, Who, When and Why. Each of these viewpoints provides an unique answer to the question (Zachman, 1987, 2008).

5.2.2 SECURITY: SABSA FRAMEWORK

The SABSA Model follows the work done by Zachman closely, although it has been adapted to security. Each layer represents the view of a different player in the process of specifying, designing, constructing and using ‘the building’ (J Sherwood et al., 2009). The SABSA matrix is outlined in Figure 15.

	ASSETS (What)	MOTIVATION (Why)	PROCESS (How)	PEOPLE (Who)	LOCATION (Where)	TIME (When)
CONTEXTUAL ARCHITECTURE	Business Decisions	Business Risk	Business Processes	Business Governance	Business Geography	Business Time Dependence
	Taxonomy of Business Assets, including Goals & Objectives	Opportunities & Threats Inventory	Inventory of Operational Processes	Organisational Structure & the Extended Enterprise	Inventory of Buildings, Sites, Territories, Jurisdictions, etc.	Time dependencies of business objectives
CONCEPTUAL ARCHITECTURE	Business Knowledge & Risk Strategy	Risk Management Objectives	Strategies for Process Assurance	Roles & Responsibilities	Domain Framework	Time Management Framework
	Business Attributes Profile	Enablement & Control Objectives; Policy Architecture	Process Mapping Framework; Architectural Strategies for ICT	Owners, Custodians and Users; Service Providers & Customers	Security Domain Concepts & Framework	Through-Life Risk Management Framework
LOGICAL ARCHITECTURE	Information Assets	Risk Management Policies	Process Maps & Services	Entity & Trust Framework	Domain Maps	Calendar & Timetable
	Inventory of Information Assets	Domain Policies	Information Flows; Functional Transformations; Service Oriented Architecture	Entity Schema; Trust Models; Privilege Profiles	Domain Definitions; Inter-domain associations & interactions	Start Times, Lifetimes & Deadlines
PHYSICAL ARCHITECTURE	Data Assets	Risk Management Practices	Process Mechanisms	Human Interface	ICT Infrastructure	Processing Schedule
	Data Dictionary & Data Inventory	Risk Management Rules & Procedures	Applications; Middleware; Systems; Security Mechanisms	User Interface to ICT Systems; Access Control Systems	Host Platforms, Layout & Networks	Timing & Sequencing of Processes and Sessions
COMPONENT ARCHITECTURE	ICT Components	Risk Management Tools & Standards	Process Tools & Standards	Personnel Man’ment Tools & Standards	Locator Tools & Standards	Step Timing & Sequencing Tools
	ICT Products, including Data Repositories and Processors	Risk Analysis Tools; Risk Registers; Risk Monitoring and Reporting Tools	Tools and Protocols for Process Delivery	Identities; Job Descriptions; Roles; Functions; Actions & Access Control Lists	Nodes, Addresses and other Locators	Time Schedules; Clocks, Timers & Interrupts
SERVICE MANAGEMENT ARCHITECTURE	Service Delivery Management	Operational Risk Management	Process Delivery Management	Personnel Management	Management of Environment	Time & Performance Management
	Assurance of Operational Continuity & Excellence	Risk Assessment; Risk Monitoring & Reporting; Risk Treatment	Management & Support of Systems, Applications & Services	Account Provisioning; User Support Management	Management of Buildings, Sites, Platforms & Networks	Management of Calendar and Timetable

Figure 15: SABSA Matrix (J Sherwood et al., 2009)

5.2.3 RELATION BETWEEN ZACHMAN AND SABSA

An integration in the sense of combining the two matrices into one comprehensive matrix is not strictly necessary nor useful. The concepts considered in the cells of each framework are relevant for a comprehensive approach, however this does not require them to be in one and the same matrix.

The Zachman- and SABSA matrices provide the views for describing an enterprise architecture and security respectively. They also provide the relationships between the concepts in the matrix. The relationship between the concepts in the Zachman- and SABSA matrix is provided by combing the intersection of a row and a column in both matrices. For example the intersection of the conceptual layer and function column in Zachman provides the view: Business Process Model. The intersection of the conceptual architecture and Process column in SABSA provides the view: Strategies for Process Assurance. The two views complement each other in the sense that the Business Process Model contains the business process of the organisation, captured in the enterprise architecture, whereas the Strategies for Process Assurance contains the strategies that assure these processes.

Both authors, Sherwood and Zachman, advise to fill in all concepts in both matrices. If this is not accomplished, one could miss out on an important part of enterprise architecture or security. However in the industry it is more common to fill in only the relevant parts for the current assignment or development of the architecture. Especially for a more experienced professional, who is capable of determining the impact of the various concepts, it should be sufficient to select the relevant concepts (J Sherwood et al., 2009; Zachman, 1987).

In order to fill in the views in Zachman and SABSA, several approaches exist. One could fill in all the relevant parts in Zachman, and then take a look at SABSA to think about the security-part of the design choices. However, that brings with it the risk that architectural choices are made which have a negative impact on security, and have to be resolved later on. Exactly that is what should be prevented by applying the integrated approach. More suitable would be an iterative approach where, once the contextual layer of Zachman is filled in, the complementing layer in SABSA is also constructed. This prevents making architecture decisions with a negative influence on security in an early stage.

Now that the framework is defined, it is time to focus on the method-part of the approach.

5.3 METHOD

A widely accepted standard in enterprise architecture development is the TOGAF ADM, the architecture development method, designed by the Open Group. It is the de-facto industry standard and provides a structured approach for designing enterprise architectures. Since it is also an open standard, it is well suited for the purpose of the method.

According to the literature review in chapter 4, no single development method is the industry standard. However, several development methods have been identified, of which the Security implementation method from Liu et al. (2001) is a general method. Also, the SABSA lifecycle is well known and aligns with the SABSA matrix. For that reason, the Security implementation method and the SABSA lifecycle are chosen as complementing method for the TOGAF ADM.

5.3.1 EA: TOGAF ADM

The TOGAF Architecture Development Method consists of eight development phases, complemented with the preliminary phase where framework and principles are defined. The ADM is set up as an iterative process model, where the final stage indicates the start of a new iteration. The current available architecture descriptions are considered as the baseline architecture. The TOGAF ADM is outlined in Figure 16.

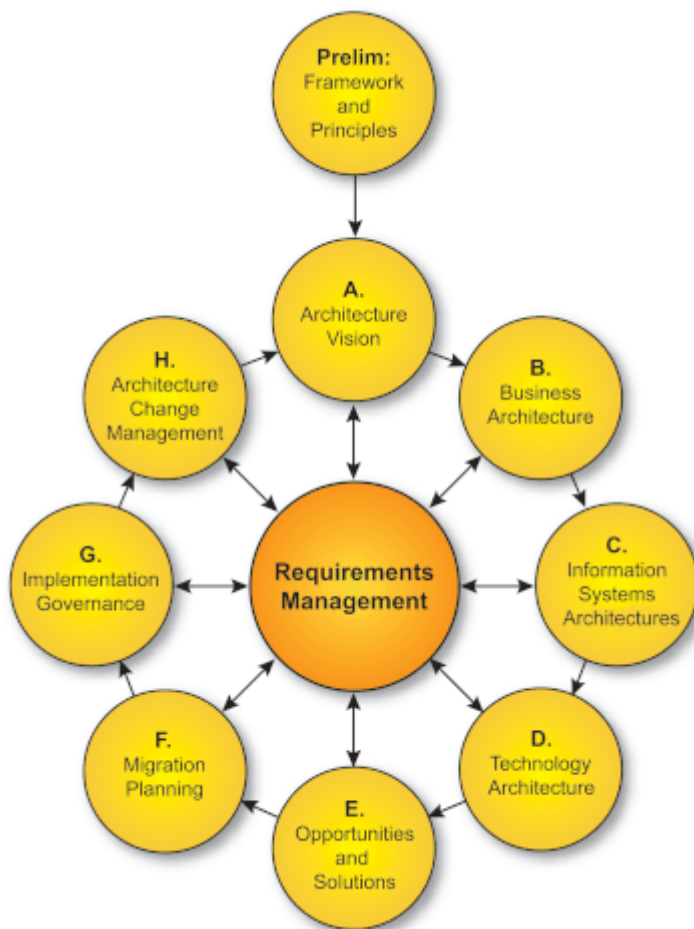


Figure 16: TOGAF Architecture Development Method

In the following sections, the various phases of the TOGAF ADM method are described.

A. Architecture Vision

Phase A starts with a request for architecture work from the sponsoring organisation to the architecture organisation. It defines what is in and what is outside of the scope, which decisions should be made on the basis of practical assessment of resource and competence availability.

Input for this phase is for example, but is not limited to: the organisational model for enterprise architecture, with budget requirements and requests for change, and the populated architecture repository which contains existing architectural documentation.

Output for Phase A may include an approved statement of architecture work, refined statements of business principles, goals and drivers, architecture principles and a draft architecture definition document.

B. Business Architecture

In short, the business architecture describes the product and/or service strategy and the organisational, functional, process, information, and geographic aspects of the business environment. The scope in this phase will depend mostly on the enterprise environment.

Input for phase B includes existing architecture descriptions, business goals, -drivers and -principles, architecture repository.

Output consists of a refined and updated version of the architecture vision phase deliverables, draft architecture definition document and a draft architecture requirements specification. Also the business architecture components of an architecture roadmap is part of the output of this phase, which may include catalogues, matrices and diagrams.

C. Information Systems Architectures

The objectives of the Information Systems Architectures phase are to develop the target information systems architecture, for data and application level, describing how the enterprise's information systems architecture will enable the business architecture and the architecture vision, in a way that addresses the Request for Architecture Work and stakeholder concerns. Furthermore, also the gaps between baseline and target information systems architectures are identified.

Input for this phase is the output of phase B, accompanied by the application and data principles.

Output for this phase are refined and updated versions of the earlier deliverables, results of the gap analysis between the baseline and the target architecture, and the information systems components of an architecture roadmap.

D. Technology Architecture

Objectives of the Technology phase are to develop the target technology architecture that enables the logical and physical application and data components and the architecture vision, and to identify candidate components based upon gaps between the baseline and target architecture.

E. Opportunities and Solutions

Phase E is focused at generating the initial complete version of the architecture roadmap, based upon the gap analysis and candidate architecture roadmap components from phases B, C, and D.

F. Migration Planning

Migration planning is considered with the finalization of the architecture roadmap and the supporting implementation and migration plan. These plans also have to be coordinated with the enterprise's approach to managing and implementing change in the overall portfolio.

G. Implementation Governance

This phase ensures conformance with the Target Architecture by implementation projects and performs architecture governance functions for the solution and any implementation-driven change requests.

H. Architecture Change Management

The closing phase of the loop makes sure that the architecture lifecycle is maintained, that the architecture governance framework is executed and that the capabilities meets the requirements.

Within TOGAF, the role of security is acknowledged but not completed. The work of the enterprise architecture and security practitioner is often separated while needing to be fully integrated in it. A security architecture has its own discrete security methodology, views and viewpoints, it often addresses non-normative flows through systems and among applications and calls for its own unique set of skills and competences of the enterprise and IT architects. TOGAF ADM mentions some security aspects per phase, which will be discussed at the integration section (The Open Group, 2011b).

5.3.2 SECURITY: ESA DEVELOPMENT CYCLE & SABSA LIFECYCLE

The security implementation approach by Liu et al. (2001) is outlined in Figure 17.

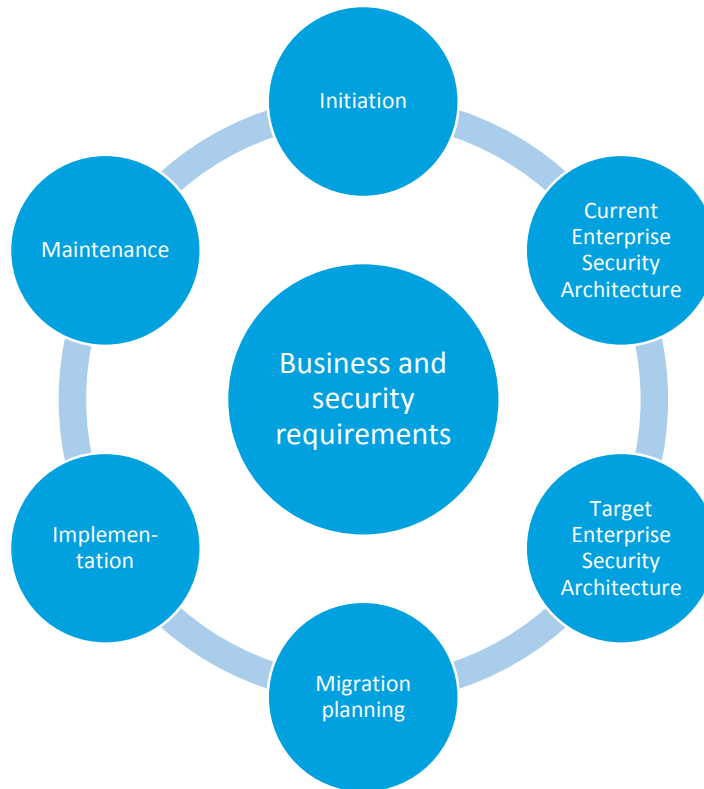


Figure 17: Security implementation approach (Liu et al., 2001)

An extensive description of the security implementation approach has already been outlined in Table 3 in section 4.2. The process aims at developing and implementing an enterprise security architecture. This is a security architecture for the enterprise, instead of a secure enterprise architecture. A secure enterprise architecture also contains elements from the enterprise, besides security elements.

Roughly, the process steps are comparable to the steps mentioned in the TOGAF ADM. A difference is in the designing phase, where TOGAF distinguishes between the business, information systems (data / application) and technology architecture, the approach by Liu et al. (2001) distinguishes a current and target architecture. Furthermore, the preliminary phase and architecture vision activities in TOGAF are, in less detail, described in the initiation phase.

The SABSA lifecycle phases also contain roughly the same phases, although not much information is publicly available on the exact interpretation of the phases. The lifecycle has much in common with the plan-do-check-act-cycle. The SABSA lifecycle is outlined in Figure 18.

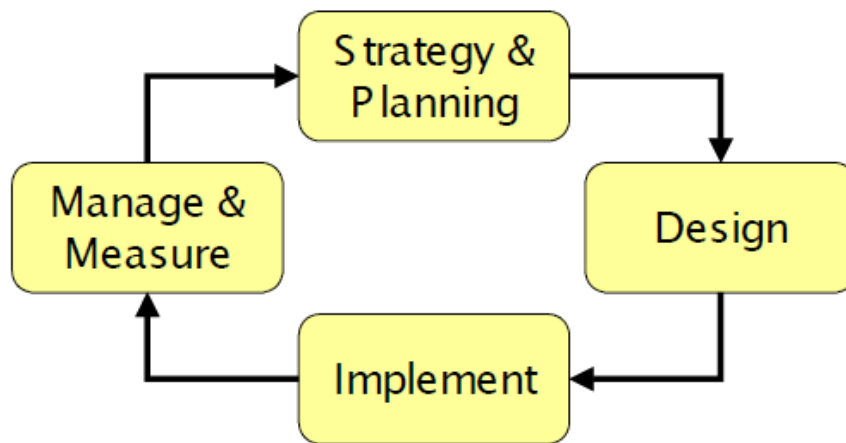


Figure 18: The SABSA Lifecycle

5.3.3 RELATION TOGAF ADM TO ZACHMAN AND SABSA FRAMEWORK

The TOGAF Architecture Development Method aligns with the first four rows of the Zachman and SABSA frameworks. For the output of each of the phases, it is illustrated which views in Zachman and SABSA are applicable. Describing the relation between the TOGAF ADM and the used frameworks helps in executing the method and realising which views are relevant to consider.

A. Preliminary phase

The outputs of the preliminary phase in TOGAF ADM and its relation to the Zachman and SABSA framework is outlined and illustrated in Table 4 and Figure 19 respectively.

Table 4: Relation of TOGAF ADM Preliminary phase to Zachman and SABSA

TOGAF ADM output	Zachman / SABSA Framework
Organisational model	Contextual/How; Conceptual/How
Architecture principles	Contextual/All
Business principles, goals and drivers	Contextual/Why; Conceptual/Why

	WHAT HOW WHERE WHO WHEN WHY							ASSETS (What) MOTIVATION (Why) PROCESS (How) PEOPLE (Who) LOCATION (Where) TIME (When)					
	DATA	FUNCTION	NETWORK	PEOPLE	TIME	MOTIVATION		CONCEPTUAL ARCHITECTURE	CONCEPTUAL ARCHITECTURE	LOGICAL ARCHITECTURE	PHYSICAL ARCHITECTURE	COMPONENT ARCHITECTURE	SERVICE MANAGEMENT ARCHITECTURE
SCOPE (Contextual)	List of things important to the business	List of processes the business performs	List of locations in which the business operates	List of organizations responsible for the business	List of event cycles significant for the business	List of business goals/strategies	Business Decisions	Business Risk	Business Processes	Business Governance	Business Geography	Business Time Dependence	
Owner	Entity = Class of business things	Process = Class of business processes	Node = Major business facilities	People = Major business staff	Time = Major Business Event Cycle	Endstate = Major Business Goal/Strategic	Taxonomy of Business Assets, including Goals & Objectives	Opportunities & Threats Inventory	Inventory of Operational Processes	Organizational Structure & the Extended Enterprise	Inventory of Buildings, Sites, Territories, Jurisdictions, etc.	Time dependencies of business objectives	
BUSINESS MODEL (Conceptual)	e.g. Logical Data Model	e.g. Business Process Model	e.g. Organizational System	e.g. Organizational Chart	e.g. Business Event Cycle	Business Risk	Business Attributes Profile	Risk Management Objectives	Strategies for Process Assurance	Roles & Responsibilities	Domain Framework	Time Management Framework	
Designer	Entity = Business Entity Relationships & Entities	Process = Business IO = Business Resources	Node = Business Location	People = Organization and Work Units/Individuals	Time = Business Event Cycle	End = Business Objective/Means = Business Strategy	Business Attributes Profile	Enablement & Control Objectives, Policy Architecture	Process Mapping Framework, Architectural Strategies for ICT	Owners, Custodians and Users; Service Providers & Customers	Security Domain Concepts & Framework	Through-Life Risk Management Framework	
SYSTEM MODEL (Logical)	e.g. Logical Data Model	e.g. Business Process Model	e.g. Organizational System	e.g. Organizational Chart	e.g. Business Event Cycle	Business Risk	Information Assets	Risk Management Policies	Services	Entity & Trust Framework	Domain Maps	Timeline & Timetable	
Builder	Entity = Business Entity Relationships & Entities	Process = Business IO = Business Resources	Node = Business Location	People = Organization and Work Units/Individuals	Time = Business Event Cycle	End = Business Objective/Means = Business Strategy	Inventory of Information Assets	Domain Policies	Information Flows, Functional Transformations, Service Oriented Architecture	Entity Schemas, Trust Models, Privilege Profiles	Domain Definitions, Inter-domain associations & integrations	Start Times, Lifetimes & Deadlines	
TECHNOLOGY MODEL (Physical)	e.g. Physical Data Model	e.g. Business Process Model	e.g. Organizational System	e.g. Organizational Chart	e.g. Business Event Cycle	Business Risk	Data Assets	Risk Management Practices	Process Mechanisms	Human Interfaces	ICT Infrastructure	Processing Schedule	
Subcontractor	Entity = Business Entity Relationships & Entities	Process = Business IO = Business Resources	Node = Business Location	People = Organization and Work Units/Individuals	Time = Business Event Cycle	End = Business Objective/Means = Business Strategy	Data Dictionary & Data Inventory	Risk Management Rules & Procedures	Applications, Middleware, Systems, Security Mechanisms	User Interfaces to ICT Systems, Access Control Systems	Host Platforms, Layout & Networks	Time Scheduling, Sequencing of Processes and Sessions	
Detailed Representations (Operational)	e.g. Data Dictionary	e.g. Program	e.g. Network Architecture	e.g. Security Architecture	e.g. Timing Definition	e.g. Risk Specification	ICT Components	Risk Management Tools & Standards	Process Tools & Standards	Personnel Management Tools & Standards	Locator Tools & Standards	Step Timing & Sequencing Tools	
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANISATION	e.g. SCHEDULE	e.g. STRATEGY	ICT Products, including Data Repositories and Processors	Risk Analysis Tools, Risk Registers, Risk Monitoring and Reporting Tools	Tools and Protocols for Process Delivery	Identities, Job Descriptions, Roles, Functions, Actions & Access Control Lists	Nodes, Addresses and other Locations	Time Schedules, Clocks, Timers & Interrupts	
							Service Delivery Management	Operational Risk Management	Process Delivery Management	Personnel Management	Management of Environment	Time & Performance Management	

Figure 19: Relation of TOGAF ADM Preliminary phase to Zachman and SABSA

B. Phase A: Architecture Vision

The outputs of the preliminary phase in TOGAF ADM and its relation to the Zachman and SABSA framework for the Architecture Vision phase is outlined and illustrated in Table 5 and Figure 20 respectively.

Table 5: Relation of TOGAF ADM Phase A to Zachman and SABSA

TOGAF ADM output	Zachman / SABSA Framework
Approved statement of architecture work (Scope)	Contextual/All
Business principles, goals and drivers	Contextual/Why
Architecture principles	Contextual/All
Baseline & Target Business / technology / data / application architecture	Conceptual/All; Logical/All; Physical/All

	WHAT	HOW	WHERE	WHO	WHEN	WHY						
	DATA	FUNCTION	NETWORK	PEOPLE	TIME	MOTIVATION						
SCOPE (Contextual)	List of things important to the business Entity = Class of business things	List of processes the business performs Process = Class of business process	List of locations in which the business operates Node = Major business location	List of organizations important to the business People = Major business unit	List of event cycles significant to the business Time = Major Business Event Cycle	List of business goals/strategies End = Major Business Goal/Strategy	ASSETS (What) Business Decisions Taxonomy of Business Assets, including Goals & Objectives	MOTIVATION (Why) Business Risk Opportunities & Threats Inventory	PROCESS (How) Business Processes Inventory of Operational Processes	PEOPLE (Who) Business Governance Organizational Structure & the Extended Enterprise	LOCATION (Where) Business Geography Inventory of Buildings, Sites, Territories, Jurisdictions, etc.	TIME (When) Business Time Dependence Time dependencies of business objectives
BUSINESS MODEL (Conceptual)	e.g. Semantic Model Entity = Business Entity Relationship = Business	e.g. Business Process Model Process = Business Process = Business Resource	e.g. Business Logical System Node = Business Location Link = Business Usage	e.g. Workflow Model People = Organization unit Work = Major Process	e.g. Greater Schedule Time = Business Event Cycle = Business Cycle	Business Plan End = Business Objective Means = Business Strategy	CONCEPTUAL ARCHITECTURE Business Knowledge & Risk Strategy Business Attributes Profile	Risk Management Objectives Enablement & Control Objectives; Policy Architecture	Strategies for Process Assurance Process Mapping Framework; Architectural Strategies for ICT	Roles & Responsibilities Owners, Custodians and Users; Service Providers & Customers	Domain Framework Security Domain Concepts & Framework	Time Management Framework Through-Life Risk Management Framework
SYSTEM MODEL (Logical)	e.g. Logical Data Model Entity = Data Entity Relationship = Data Relationship	e.g. Application Architecture Process = Application Functionality = User Interface	e.g. Information System Model Node = IT Functionality = Data Characteristics	e.g. System Information Architecture People = System User Work = System Element	e.g. Processing Structure Time = System Event Cycle = Processing Cycle	e.g. Business Rule Model End = Business Rule Means = Action Assertion	LOGICAL ARCHITECTURE Information Assets Inventory of Information Assets	Risk Management Policies Domain Policies	Process Maps & Services Information Flows; Functional Transformations; Service Oriented Architecture	Entity & Trust Framework Entity Schema; Trust Models; Privilege Profiles	Domain Maps Inter-domain associations & interactions	Calendar & Timetable Start Times, Lifetimes & Deadlines
TECHNOLOGY MODEL (Physical)	e.g. Physical Data Model Entity = Segment/Table Relationship = Primary/Foreign Key	e.g. System Design Process = Computer Functionality = Data Event/Message	e.g. Technology Architecture Node = Physical System or Subsystem	e.g. Information Architecture People = User Work = System Element	e.g. Control Structure Time = Logical Cycle = Component Cycle	e.g. Rule Design End = Control Means = Action	PHYSICAL ARCHITECTURE Data Assets Data Dictionary & Data Inventory	Risk Management Practices Risk Management Rules & Procedures	Process Mechanisms Applications; Middleware; Systems; Security Mechanisms	Human Interface User Interface to ICT Systems; Access Control Systems	ICT Infrastructure Host Platforms, Layout & Networks	Processing Schedule Timing & Sequencing of Processes and Events
DETAILED REPRESENTATIONS (Out of context)	e.g. Data Definition Entity = Field Relationship = Address	e.g. Program Process = Program Control = Data	e.g. Network Architecture Node = Address Link = Connection	e.g. Security Architecture People = Identity Role = Job	e.g. Timing Diagram Time = Event Cycle = Message Cycle	e.g. Rule Specification End = Condition Means = Job	COMPONENT ARCHITECTURE ICT Components ICT Products, including Data Representations and Processes	Risk Management Tools & Standards Risk Analysis Tools Risk Registers; Risk Monitoring and Reporting Tools	Process Tools & Standards Tools and Protocols for Process Delivery	Personnel Management Tools & Standards Identities, Job Descriptions, Roles, Functions, Actions & Access Control Lists	Locator Tools & Standards Nodes, Addresses and other Locations	Step Timing & Sequencing Tools Time Schedules, Clocks, Times & Intervals
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANISATION	e.g. SCHEDULE	e.g. STRATEGY	SERVICE MANAGEMENT ARCHITECTURE Service Delivery Management	Operational Risk Management Risk Assessment	Process Delivery Management Management & Control	Personnel Management Account	Management of Environment Management of	Time & Performance Management

Figure 20: Relation of TOGAF ADM Phase A to Zachman and SABSA

C. Phase B: Business Architecture

The outputs of the Business Architecture phase in TOGAF ADM and its relation to the Zachman and SABSA framework is outlined and illustrated in Table 6 and Figure 21 respectively.

Table 6: Relation of TOGAF ADM Phase B to Zachman and SABSA

TOGAF ADM output	Zachman / SABSA Framework
Architecture Vision	Contextual/All
Business goals and objectives	Contextual/Why; Conceptual/Why
Organisation structure	Contextual/Where; Conceptual/Where; Contextual/Who; Conceptual/Who
Business functions	Contextual/How; Conceptual/How
Business services	Conceptual/Why; Logical/Why

Business processes	Contextual/How; Conceptual/How; Contextual/When; Conceptual/When
Business roles	Contextual/Who; Conceptual/Who
Business data model	Contextual/What; Conceptual/What
Location catalogue	Contextual/Where; Conceptual/Where
Process flow diagram	Contextual/How; Conceptual/How
Event diagram	Contextual/When; Conceptual/When
Technical requirements	Logical/Why

	WHAT	HOW	WHERE	WHO	WHEN	WHY
	DATA	FUNCTION	NETWORK	PEOPLE	TIME	MOTIVATION
SCOPE (Contextual)	List of things important to the business e.g. Class of business Range	List of processes the business performs Process = Class of business process	List of locations in which the business operates Node = Major Business Location	List of organisations important to the business People = Major business unit	List of event cycles significant to the business Time = Major Business Event Cycle	List of business goals/strategies Endstate = Major Business Goal/Strategy
BUSINESS MODEL (Conceptual)	e.g. Semantic Model Entity + Business Entity Relationships + Business	e.g. Business Process Model Process + Business ID + Business Reference	e.g. Business Location System Node + Business Location Link + Business Linkage	e.g. Interflow Model People + Organisation unit Work + roles/Producer	e.g. Master Schedule Time + Business Event Cycle + Business Cycle	Business Plan End + Business Objective Means + Business Strategy
SYSTEM MODEL (Logical)	e.g. Logical Data Model Entity + Time Relationship + Time	e.g. Application Architecture Process + Application Function ID + Role/View	e.g. Network Architecture Node + Link + Network Link + Network Linkage	e.g. Human Interface Model People + User Interface + User Interface Linkage	e.g. Performance Model Time + Business Event Cycle + Business Cycle	e.g. Business Rule Model End + Business Objective Means + Business Strategy
TECHNOLOGY MODEL (Physical)	e.g. Physical Data Model Entity + Equipment/Tools Relationship + Possibilities	e.g. System Design Process + Computer System ID + Data Element/Unit	e.g. Technology Architecture Node + How/Address Link + Network Linkage	e.g. Presentation Architecture People + User Interface + User Interface Linkage	e.g. Operational Model Time + Business Event Cycle + Business Cycle	e.g. Risk Model End + Business Objective Means + Business Strategy
DETAILED REPRESENTATIONS (Detailed/Logical)	e.g. Data Dictionary Entity + Field Relationship + Address	e.g. Program Process + Program ID + Control/Block	e.g. Network Architecture Node + Address Link + Network Linkage	e.g. Security Architecture People + Identity Link + Role	e.g. Timing Definition Time + Interval Cycle + Business Cycle	e.g. Risk Specification End + Sub-condition Means + Step
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANISATION	e.g. SCHEDULE	e.g. STRATEGY

	ASSETS (What)	MOTIVATION (Why)	PROCESS (How)	PEOPLE (Who)	LOCATION (Where)	TIME (When)
CONCEPTUAL ARCHITECTURE	Business Decisions Taxonomy of Business Assets, including Goals & Objectives	Business Risk Opportunities & Threats Inventory	Business Processes Inventory of Operational Processes	Business Governance Organisational Structure & the Extended Enterprise	Business Geography Inventory of Buildings, Sites, Territories, Jurisdictions, etc.	Business Time Dependence Time dependencies of business objectives
CONCEPTUAL ARCHITECTURE	Business Knowledge & Risk Strategy Business Attributes Profile	Risk Management Objectives Enablement & Control Objectives, Policy Architecture	Strategies for Process Assurance Process Mapping Framework, Architectural Strategies for ICT	Rules & Responsibilities Owners, Custodians and Users; Service Providers & Customers	Domain Framework Security Domain Concepts & Framework	Time Management Framework Through-Life Risk Management Framework
LOGICAL ARCHITECTURE	Information Assets Inventory of Information Assets	Risk Management Policies Domain Policies	Process Maps & Services Information Flows; Functional Transformations; Service Oriented Architecture	Entity & Trust Framework Entity Schema, Trust Models, Privilege Profiles	Domain Maps Domain Definitions, Inter-domain associations & interactions	Calendar & Timetable Start Times, Lifetimes & Durations
PHYSICAL ARCHITECTURE	Data Assets Data Dictionary & Data Inventory	Risk Management Practices Rules & Procedures	Process Mechanisms Applications; Middleware; Systems; Security Mechanisms	Human Interface User Interface to ICT Systems; Access Control Systems	ICT Infrastructure Host Platforms, Layout & Networks	Processing Schedule Timing & Sequencing of Processes and Sessions
COMPONENT ARCHITECTURE	ICT Components ICT Products, including Data Repositories and Processes	Risk Analysis Tools Risk Registers; Risk Monitoring and Reporting Tools	Tools and Protocols for Process Delivery Operational Risk Management	Personnel Management Tools & Standards Identifies, Job Descriptions, Roles, Functions, Actions & Access Control Lists	Locator Tools & Standards Nodes, Addresses and other Locations	Step Timing & Sequencing Tools Time Schedules; Clocks, Timers & Interrupts
SERVICE MANAGEMENT ARCHITECTURE	Service Delivery Management Assurance of	Process Delivery Management Risk Assessment;	Process Delivery Management Management &	Personnel Management Account	Management of Environment	Time & Performance Management

Figure 21: Relation of TOGAF ADM Phase B to Zachman and SABSA

D. Phase C: Information Systems Architecture

The outputs of the Information Systems Architecture phase in TOGAF ADM and its relation to the Zachman and SABSA framework is outlined and illustrated in Table 7 and Figure 22 respectively. This is done for both the data and application architecture.

Table 7: Relation of TOGAF ADM Phase C to Zachman and SABSA

TOGAF ADM output	Zachman / SABSA Framework
Business data model	Conceptual/What
Logical data model	Logical/What
Data management process model	Logical/How; Logical/Who
Data entity / Business function matrix	Conceptual/How; Conceptual/What; Logical/What
Conceptual data diagram	Conceptual/What
Logical data diagram	Logical/What
Data security diagram	Logical/What
Data interoperability requirements	Logical/All

Process systems model	Logical/How
Place systems model	Logical/Where
Time systems model	Logical/When
People systems model	Logical/Who
Application interoperability requirements	Logical/All
Implications on Business Architecture	Conceptual/What; Conceptual/How; Conceptual/Where; Conceptual/Who; Conceptual/When
Constraints on Technology Architecture	Logical/Why

	WHAT	HOW	WHERE	WHO	WHEN	WHY
	DATA	FUNCTION	NETWORK	PEOPLE	TIME	MOTIVATION
SCOPE (Contextual)	Use of business processes for business	Use of capabilities for business	Use of business systems for business	Use of business people for business	Use of business time for business	Use of business why for business
SCOPE (Owner)	Entry + Core of Business	Process + Core of Business	Node + Core of Business	People + Core of Business	Time + Core of Business	End + Core of Business
BUSINESS MODEL (Conceptual)	e.g. Semantic Model	e.g. Business Process Model	e.g. Business Logistics System	e.g. Workflow Model	e.g. Scheduler Schedule	Business Plan
BUSINESS MODEL (Owner)	Entry + Business Entry	Process + Business Process	Node + Business Location	People + Business Organization	Time + Business Event	End + Business Objective
SYSTEM MODEL (Supplier)	e.g. Logical Data Model	e.g. Application Architecture	e.g. Distributed System Model	e.g. System Interface Architecture	e.g. Processing Structure	e.g. Business Rule Book
SYSTEM MODEL (Designer)	Entry + Data Entry	Process + Application Process	Node + User Interface	People + System Interface	Time + System Cycle	End + System Objective
TECHNOLOGY MODEL (Physical)	e.g. Physical Data Model	e.g. System Design	e.g. Technology Architecture	e.g. System Interface Architecture	e.g. Processing Structure	e.g. Business Rule Book
TECHNOLOGY MODEL (Builder)	Entry + System Data Model	Process + System Design	Node + System Architecture	People + System Interface	Time + System Cycle	End + System Objective
DETAILED REPRESENTATIONS (Qualification)	e.g. Data Dictionary	e.g. Program	e.g. Network Architecture	e.g. Security Architecture	e.g. Timing Definition	e.g. Risk Specification
DETAILED REPRESENTATIONS (Subcontractor)	Entry + Data Dictionary	Process + Program	Node + Network Architecture	People + Security Architecture	Time + Timing Definition	End + Risk Specification
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANISATION	e.g. SCHEDULE	e.g. STRATEGY

Figure 22: Relation of TOGAF ADM Phase C to Zachman and SABSA

E. Phase D: Technology Architecture

The outputs of the Technology Architecture phase in TOGAF ADM and its relation to the Zachman and SABSA framework is outlined and illustrated in Table 8 and Figure 23 respectively.

Table 8: Relation of TOGAF ADM Phase D to Zachman and SABSA

TOGAF ADM output	Zachman / SABSA Framework
Technology components	Logical/What; Physical/What
Technology platforms	Logical/How; Physical/How
Environments and locations	Logical/Where; Physical/Where
Expected processing load	Logical/How; Physical/How
Physical network communications	Logical/Where; Physical/Where; Logical/Who; Physical/Who; Logical/When; Physical/When; Logical/Why; Physical/Why
Hardware and network specifications	Logical/Where; Physical/Where

	WHAT	HOW	WHERE	WHO	WHEN	WHY
	DATA	FUNCTION	NETWORK	PEOPLE	TIME	MOTIVATION
SCOPE (Contextual)	List of things important to the business Entity = Major Business Strategic	List of processes the business performs Process = Class of Business Process	List of locations in which the business operates Node = Major Business Location	List of organizations responsible for the business People = Major Business Unit	List of event cycles important to the business Time = Major Business Event Cycle	List of business goals or objectives End = Business Goal/End State
BUSINESS MODEL (Conceptual)	e.g. Semantic Model Entity = Business Entity Relationship + Business	e.g. Business Process Model Process = Class of Business Process	e.g. Business Location System Node = Business Location Link + Business Location	e.g. Business Unit Model People = Organization Unit + Business Unit	e.g. Business Event Cycle Time = Business Event Cycle + Business Cycle	e.g. Business Goal Model End = Business Goal/End State + Business Strategy
SYSTEM MODEL (Logical)	e.g. Logical Data Model Entity = Data Entity Relationship + User Requirement	e.g. Applicable Architecture Process = Computer Function + User View	e.g. Distributed System Model Node = IT Function Relationship + Line Relationship	e.g. Human Interface Architecture People = User Work + Control Panels	e.g. Processing Structure Time = Event Cycle + Component Cycle	e.g. Business Rule Model End = Structural Assent/Means + Actor Assent
TECHNOLOGY MODEL (Physical)	e.g. Physical Data Model Entity = Segment/Field Relationship + Relationship	e.g. System Design Process = Computer Function + Data Element/sets	e.g. Technology Architecture Node = IT Function Relationship + Line Relationship	e.g. Presentation Architecture People = User Work + Control Panels	e.g. Control Structure Time = Event Cycle + Component Cycle	e.g. Risk Design End = Location Node + Action
DETAILED REPRESENTATIONS (Out-of-context)	e.g. Data Definition Entity = Field Relationship + Address	e.g. Program Process = Language + Control Block	e.g. Network Architecture Node = Address Link + Protocol	e.g. Security Architecture People = Entity Work + Job	e.g. Timing Structure Time = Event Cycle + Multiple Cycle	e.g. Risk Definition End = Assent/Means + Job
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANISATION	e.g. SCHEDULE	e.g. STRATEGY

	ASSETS (What)	MOTIVATION (Why)	PROCESS (How)	PEOPLE (Who)	LOCATION (Where)	TIME (When)
CONTEXTUAL ARCHITECTURE	Business Decisions Taxonomy of Business Assets, including Goals & Objectives	Business Risk Opportunities & Threats Inventory	Business Processes Inventory of Operational Processes	Business Governance Organisational Structure & the Extended Enterprise	Business Geography Inventory of Buildings, Sites, Territories, Jurisdictions, etc.	Business Time Dependence Time dependencies of business objectives
CONCEPTUAL ARCHITECTURE	Business Knowledge & Risk Strategy Business Attributes Profile	Risk Management Objectives Enablement & Control Objectives, Policy Architecture	Strategies for Process Assurance Process Mapping Framework, Architectural Strategies for ICT	Roles & Responsibilities Owners, Custodians and Users; Service Providers & Customers	Domain Framework Security Domain Concepts & Framework	Time Management Framework Through-Life Risk Management Framework
LOGICAL ARCHITECTURE	Information Assets Inventory of Information Assets	Risk Management Policies Domain Policies	Process Maps & Services Information Flows; Functional Transformations; Service Oriented Architecture	Entity & Trust Framework Entity Schemas; Trust Models; Privilege Profiles	Domain Maps Domain Definitions, Inter-domain associations & interactions	Calendar & Timetable Start Times, Lifetimes & Deadlines
PHYSICAL ARCHITECTURE	Data Assets Data Dictionary & Data Inventory	Risk Management Practices Risk Management Rules & Procedures	Process Mechanisms Applications; Middleware; Systems; Security Mechanisms	Human Interface User Interface to ICT Systems; Access Control Lists	ICT Infrastructure Host Platforms, Layout & Networks	Processing Schedule Timing & Sequencing of Processes and Cases
COMPONENT ARCHITECTURE	ICT Components ICT Products, including Data Repositories and Processors	Risk Management Tools & Standards Risk Analysis Tools; Risk Registers, Risk Monitoring and Reporting Tools	Process Tools & Standards Tools and Protocols for Process Delivery	Personnel Management Tools & Standards Identities; Job Descriptions; Roles; Functions; Actions & Access Control Lists	Locator Tools & Standards Nodes, Addresses and other Locations	Step Timing & Sequencing Tools Time Schedules, Clocks, Timers & Interrupts
SERVICE MANAGEMENT ARCHITECTURE	Service Delivery Management Assurance of	Operational Risk Management Risk Assessment;	Process Delivery Management Management &	Personnel Management Account	Management of Environment Management of	Time & Performance Management Management of

Figure 23: Relation of TOGAF ADM Phase D to Zachman and SABSA

F. Overview

The phases E to H are also part of the TOGAF ADM cycle, but are not considered architectures and are therefore not mapped to the Zachman and SABSA frameworks. The phases A to D align with the upper four rows of both Zachman and SABSA, as illustrated in Figure 24. The mapping illustrates how the deliverables of the TOGAF phases relate to both the considered frameworks.

	WHAT	HOW	WHERE	WHO	WHEN	WHY
	DATA	FUNCTION	NETWORK	PEOPLE	TIME	MOTIVATION
SCOPE (Contextual)	List of things important to the business Entity = Major Business Strategic	List of processes the business performs Process = Class of Business Process	List of locations in which the business operates Node = Major Business Location	List of organizations responsible for the business People = Major Business Unit	List of event cycles important to the business Time = Major Business Event Cycle	List of business goals or objectives End = Business Goal/End State
BUSINESS MODEL (Conceptual)	e.g. Semantic Model Entity = Business Entity Relationship + Business	e.g. Business Process Model Process = Class of Business Process	e.g. Business Location System Node = Business Location Link + Business Location	e.g. Business Unit Model People = Organization Unit + Business Unit	e.g. Business Event Cycle Time = Business Event Cycle + Business Cycle	e.g. Business Goal Model End = Business Goal/End State + Business Strategy
SYSTEM MODEL (Logical)	e.g. Logical Data Model Entity = Data Entity Relationship + User Requirement	e.g. Applicable Architecture Process = Computer Function + User View	e.g. Distributed System Model Node = IT Function Relationship + Line Relationship	e.g. Human Interface Architecture People = User Work + Control Panels	e.g. Processing Structure Time = Event Cycle + Component Cycle	e.g. Business Rule Model End = Structural Assent/Means + Actor Assent
TECHNOLOGY MODEL (Physical)	e.g. Physical Data Model Entity = Segment/Field Relationship + Relationship	e.g. System Design Process = Computer Function + Data Element/sets	e.g. Technology Architecture Node = IT Function Relationship + Line Relationship	e.g. Presentation Architecture People = User Work + Control Panels	e.g. Control Structure Time = Event Cycle + Component Cycle	e.g. Risk Design End = Location Node + Action
DETAILED REPRESENTATIONS (Out-of-context)	e.g. Data Definition Entity = Field Relationship + Address	e.g. Program Process = Language + Control Block	e.g. Network Architecture Node = Address Link + Protocol	e.g. Security Architecture People = Entity Work + Job	e.g. Timing Structure Time = Event Cycle + Multiple Cycle	e.g. Risk Definition End = Assent/Means + Job
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANISATION	e.g. SCHEDULE	e.g. STRATEGY

	ASSETS (What)	MOTIVATION (Why)	PROCESS (How)	PEOPLE (Who)	LOCATION (Where)	TIME (When)
CONTEXTUAL ARCHITECTURE	Business Decisions Taxonomy of Business Assets, including Goals & Objectives	Business Risk Opportunities & Threats Inventory	Business Processes Inventory of Operational Processes	Business Governance Organisational Structure & the Extended Enterprise	Business Geography Inventory of Buildings, Sites, Territories, Jurisdictions, etc.	Business Time Dependence Time dependencies of business objectives
CONCEPTUAL ARCHITECTURE	Business Knowledge & Risk Strategy Business Attributes Profile	Risk Management Objectives Enablement & Control Objectives, Policy Architecture	Strategies for Process Assurance Process Mapping Framework, Architectural Strategies for ICT	Roles & Responsibilities Owners, Custodians and Users; Service Providers & Customers	Domain Framework Security Domain Concepts & Framework	Time Management Framework Through-Life Risk Management Framework
LOGICAL ARCHITECTURE	Information Assets Inventory of Information Assets	Risk Management Policies Domain Policies	Process Maps & Services Information Flows; Functional Transformations; Service Oriented Architecture	Entity & Trust Framework Entity Schemas; Trust Models; Privilege Profiles	Domain Maps Domain Definitions, Inter-domain associations & interactions	Calendar & Timetable Start Times, Lifetimes & Deadlines
PHYSICAL ARCHITECTURE	Data Assets Data Dictionary & Data Inventory	Risk Management Practices Risk Management Rules & Procedures	Process Mechanisms Applications; Middleware; Systems; Security Mechanisms	Human Interface User Interface to ICT Systems; Access Control Lists	ICT Infrastructure Host Platforms, Layout & Networks	Processing Schedule Timing & Sequencing of Processes and Cases
COMPONENT ARCHITECTURE	ICT Components ICT Products, including Data Repositories and Processors	Risk Management Tools & Standards Risk Analysis Tools; Risk Registers, Risk Monitoring and Reporting Tools	Process Tools & Standards Tools and Protocols for Process Delivery	Personnel Management Tools & Standards Identities; Job Descriptions; Roles; Functions; Actions & Access Control Lists	Locator Tools & Standards Nodes, Addresses and other Locations	Step Timing & Sequencing Tools Time Schedules, Clocks, Timers & Interrupts
SERVICE MANAGEMENT ARCHITECTURE	Service Delivery Management Assurance of	Operational Risk Management Risk Assessment;	Process Delivery Management Management &	Personnel Management Account	Management of Environment Management of	Time & Performance Management Management of

Figure 24: Overview of mapping TOGAF ADM to Zachman and SABSA

In book chapter 39 published by the Open Group (see (The Open Group, 2007)), a mapping from TOGAF to Zachman is also provided. Some differences exist due to the use of TOGAF version 8.1.1 instead of version 9.1.1 as was used in this thesis.

5.3.4 INTEGRATION OF TOGAF ADM AND ESA DEVELOPMENT CYCLE

The update of TOGAF on security is motivated by the fact that TOGAF treated security and risk either implicitly through stakeholder requirements or through a limited set of techniques (The Open Group, 2011a). The relation of the TOGAF ADM to both Zachman and SABSA framework is provided in section 5.3.3. In this section an overview on how the integrated method looks like is provided.

A. Integration TOGAF ADM & SABSA Lifecycle

An integration of the TOGAF ADM and SABSA lifecycle is outlined in Figure 25.

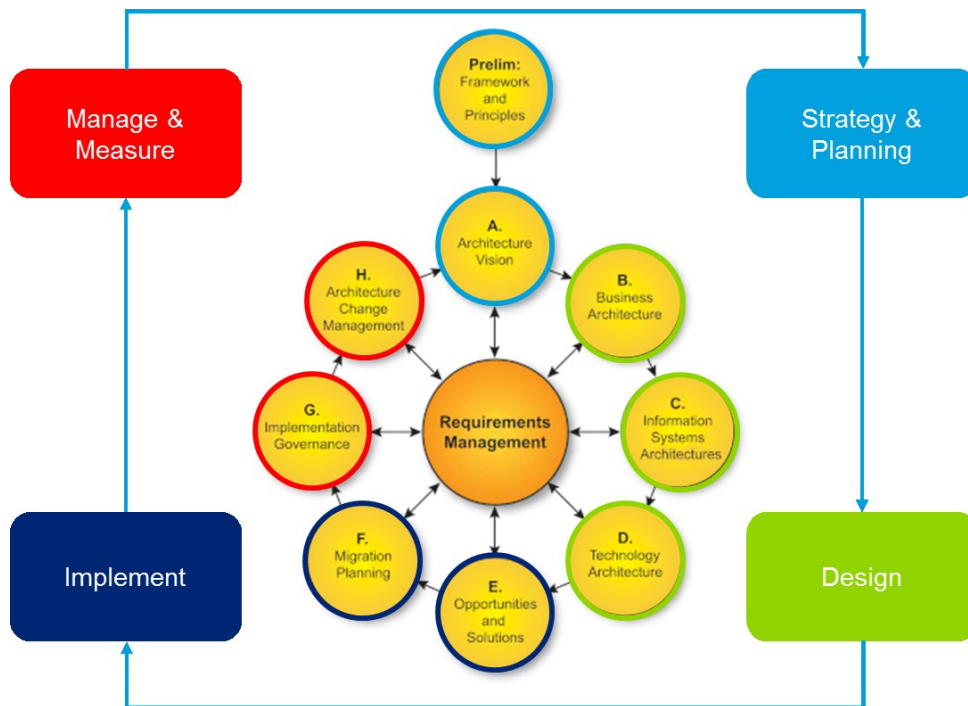


Figure 25: Relation between SABSA lifecycle and TOGAF ADM

The activities conducted in the Strategy & Planning phase of the SABSA Lifecycle are considered to be in the same category of TOGAF ADM's preliminary phase and architecture vision. In these TOGAF phases the framework and principles are chosen and a strategy or vision is developed.

For the design of the architecture, TOGAF distinguishes between three levels: business, information systems and technology architecture. In this phase, the architecture is designed and aligned to the chosen strategy / vision.

In the implementation phase the developed architecture is implemented into the organisation. TOGAF ADM distinguishes two phases: Opportunities and Solutions and Migration Planning.

Finally the assessment on successfulness has to be completed in order to determine whether or not the implementation was successful. The results should be measured against the original strategy and vision. This is executed in the Implementation Governance phase. This step might lead to new insights to change the vision or to new changes to the architecture and a new iteration of the cycle might be started. In the final step, Architecture Change Management, internal and external changes to architecture, organisation or its environment is monitored. Based on this observation a new iteration of the architecture development method might be started.

B. Integration TOGAF ADM & Security implementation cycle

The Security Implementation Cycle (SIC) as described by Liu et al. (2001) is comparable to the TOGAF ADM cycle. The preliminary phase is not explicitly included in Liu's process model, however the activities identified for this step correspond to the preliminary phase in TOGAF ADM.

Another difference between the SIC and the TOGAF ADM is the difference in the design phase. The SIC distinguishes between the current and target architecture, but does not differentiate between a business, information systems, and technology architecture. The TOGAF ADM also distinguishes between current (baseline) and target architecture, but in contrast to the SIC on each separate level (business, information systems, and technology). In order to provide a close alignment to the enterprise architecture, a separation of business, information systems, and technology is useful. Therefore, the TOGAF ADM best suits the method aspect of the approach.

Concluding, the method for developing a secure enterprise architecture is guided by TOGAF ADM.

5.4 MODELLING LANGUAGE

The modelling language component is the final ingredient of the approach to design secure enterprise architectures. A modelling language is needed to unambiguously specify and describe components and their relationship. Within the context of this thesis specifically, a language is used to describe EA- and security-components and their relationship. It is useful to specify the aspects of both fields in the same language.

For the language component of the approach ArchiMate was chosen, according to the ArchiMate 2.1 specification with its motivation and migration extensions. ArchiMate being an open standard, and – as well as the TOGAF ADM – being supported by the Open Group is the main reason for choosing this standard. It is widely accepted and tools for modelling this standard are available. It has a good fit with both the TOGAF and Zachman Framework, and provides traceability for design choices via the motivation extension. Furthermore, it provides a basis for analysis. The drawback of using ArchiMate is that although it is widely accepted within the enterprise architecture discipline, it is not yet accepted by the entire organisation.

In the security discipline, currently no graphical modelling language is widely accepted. Most security documents and policies are described in a natural language. This has the implication that it does not provide a good fit with the enterprise architecture models, nor does it provide a basis for analysis. For these reasons a security extension for the ArchiMate modelling language is provided.

5.4.1 ARCHIMATE

The ArchiMate enterprise architecture specification language is an open standard, and is maintained and supported by the Open Group since 2008 (The Open Group, 2013). It provides a language to describe for example business processes, organisational structures, information

flows, IT-systems and technical infrastructures. It distinguishes the business, the application, and the technology layer. Within these layers three components are distinguished: passive structure, behaviour, and active structure. The active structure aspect represents the structural concepts like actors, components, and devices. The behaviour aspect represents the processes, functions, events, and services performed by the active structure aspects. The passive structure aspect represents the objects on which the behaviour is performed, which are for example information objects. This combines to the architectural framework as displayed in Figure 26.

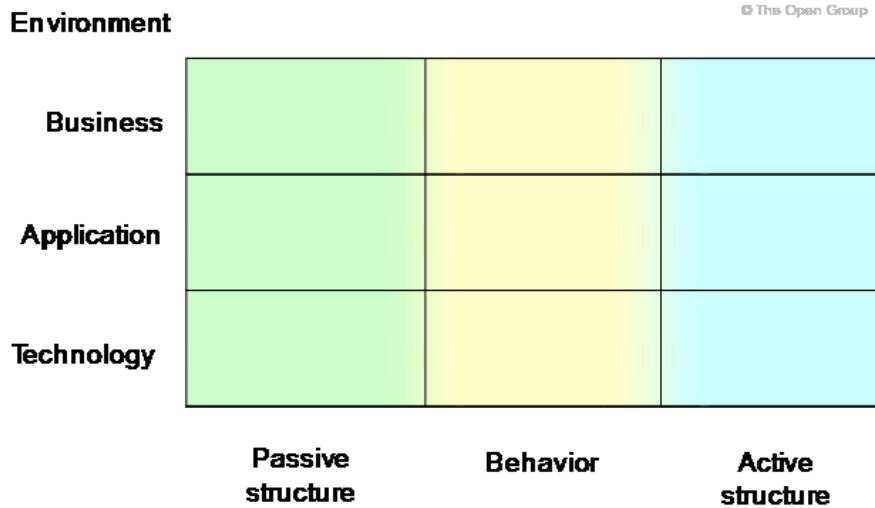


Figure 26: ArchiMate Architectural Framework (The Open Group, 2013)

The relation between the TOGAF ADM and the ArchiMate specification language is illustrated in Figure 27. Both the ADM cycle and ArchiMate distinguish the business, information systems and technology layer.

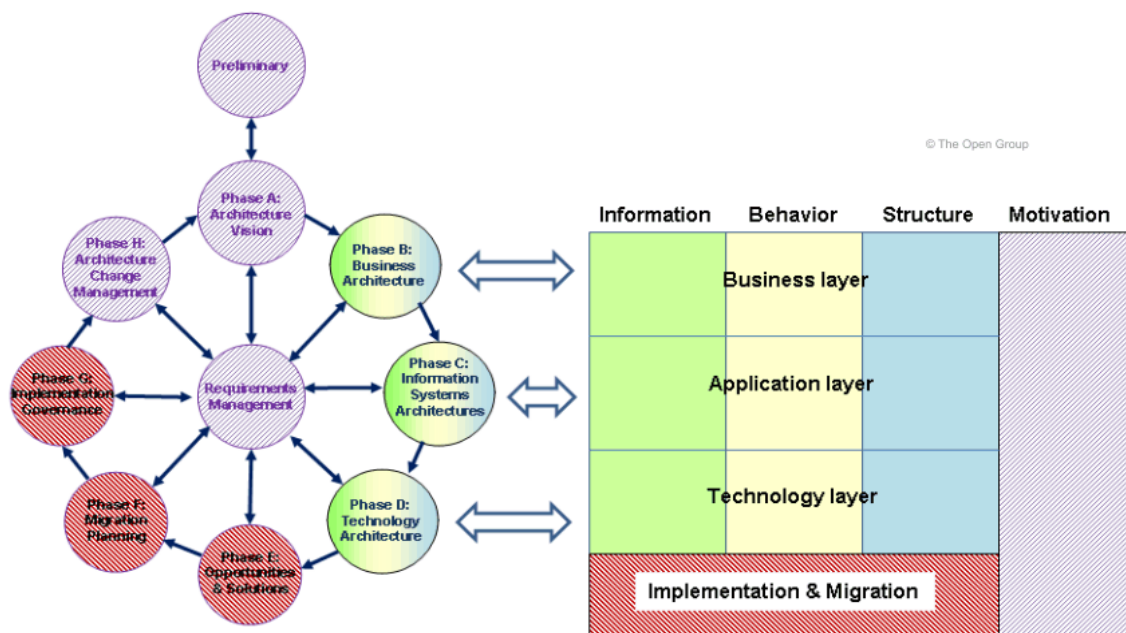


Figure 27: Correspondence between ArchiMate and TOGAF ADM (The Open Group, 2013)

ArchiMate currently contains several concepts. Some of these concepts are of specific relevance in the security setting. These concepts are included in Appendix B – Security Related Concepts in ArchiMate.

Having identified the relevant concepts in the current ArchiMate language, it is interesting to determine whether or not relevant security concepts have been missed.

5.4.2 ARCHIMATE SECURITY EXTENSION

In order to extend the ArchiMate specification language to include security aspects, three key questions need to be answered (in accordance with research question 2.1 – 2.3):

1. **What elements are needed to specify a modelling language?**
2. **Which security concepts need to be merged into the enterprise architecture language?**
3. **How can the language be validated?**

A key challenge in the development of a language is to strike a balance between the specificity of the language for the various domains, and the general set of concepts. This is illustrated in Figure 28 (The Open Group, 2013).

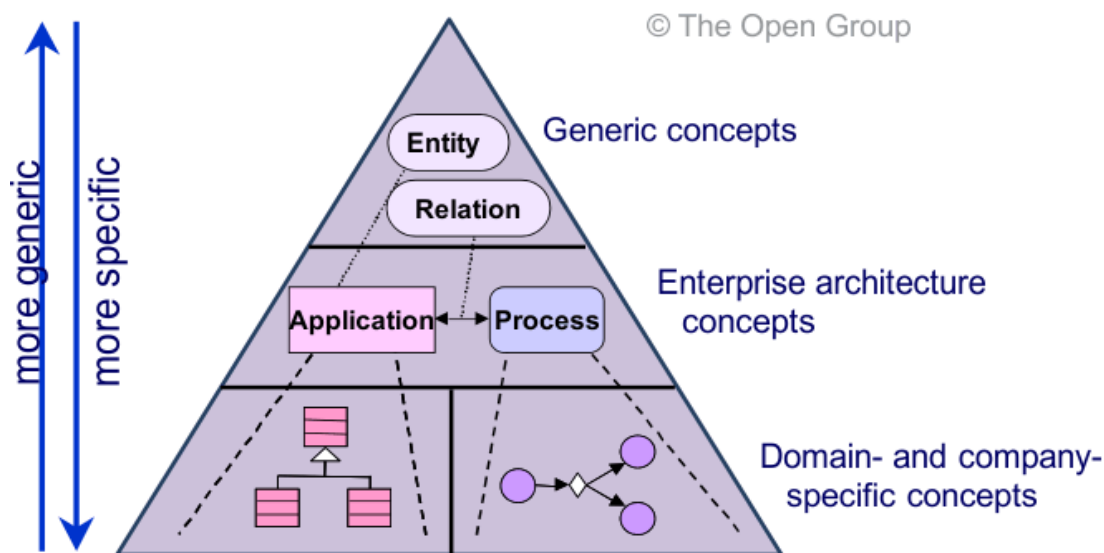


Figure 28: Metamodels at Different Levels of Specificity (The Open Group, 2013)

A modelling language is defined by a consistent set of rules. The rules are used to interpret the meaning of the concepts and their relationships. When defining new concepts, there is a need to define to which rules it should apply, and what a relationship with the concept means in general.

5.4.2.1 CONCEPT SELECTION

In order to identify which concepts are desired to represent in a security modelling language in relation with the enterprise architecture, the SABSA framework is used. In Figure 29 the concepts for each of the SABSA viewpoints are extracted. The **bold** concepts are already included in the ArchiMate specification language, the non-bold concepts are currently not covered and are candidates for the extension language.

	What	Why	How	Who	Where	When
Contextual	Business assets; goals and objectives	Risks; Vulnerabilities; Opportunities; Threats; Requirements	Business processes	Organisational structure	Location	Time (performance, sequence)
Conceptual	Business attributes	Control and enablement objectives; risk assessment	Security mechanisms	Roles, responsibilities	Security domain (logical and physical), domain boundaries and security associations	Business time-management framework
Logical	Business information assets	Policies	Processes	Entities; relationships (actors)	Domains	Timetable
Physical	Business data model	Risk management rules & procedures	Security mechanisms	User Interface to ICT systems, Access control systems	Platforms, networks	Sequences, events, lifetimes
Component						
Service management						

Figure 29: SABSA concepts and its relation to ArchiMate

Selection

A selection on these concepts is based upon the following criteria:

- R1: level of semantic overlap in enterprise architecture and security discipline
- R2: level of generality

The first selection criterion aims at identification of constructs which have a shared meaning in both disciplines. The second selection criterion is important because security is a cross-cutting concern in enterprise architecture. Therefore concepts modelling security, should be usable in all architectural layers and levels of abstraction in an enterprise architecture.

Furthermore, similarly with other past extension proposals of ArchiMate, the following general principles for modelling language extensions are also considered (Iacob, Quartel, & Jonkers, 2012):

- Alignment with ArchiMate: the proposed language fragment should be aligned with the current ArchiMate metamodel specification;
- Parsimony and ease of use. The number of additional concepts is kept to a minimum. Whenever possible, existing ArchiMate concepts and relationships are reused or specialized. The new concepts are easy to learn, understand and use;
- The new concepts easily accommodate model-based valuation techniques.

A list of concepts that are currently not covered in ArchiMate (extracted from Figure 29) is included in Table 9.

Table 9: Concepts in SABSA, not covered in ArchiMate and the motivation for in-/exclusion

	Inclusion based on	Exclusion based on
Risk	R1; R2	
Vulnerability	R1; R2	
Opportunity		R1
Threat	R1; R2	
Time		R1
Business attribute		R1
Security mechanism	R1; R2	
Security domain definitions/ boundaries / associations		R2
Business time-management framework		R2
Policy	R1; R2	
Timetable		R2
Risk management rules & procedures		R1
Access control systems		R2
Lifetime		R2

Thus, the concepts selected to include in the security modelling language extension are: vulnerability, threat, risk, security mechanism, and security policy.

Example

An example is provided on how the included concepts on the proposed extension of ArchiMate relate to each other.

One of the concepts is ‘security mechanism’. Examples of a security mechanism are: ‘role based access control’ and ‘intrusion detection system’. A security mechanism can be both technical and non-technical. A way in which this concept can be used is illustrated in the following example.

A well-known concept within the security area is the concept of ‘attack and defence trees’, as already identified in the literature study and described in section 4.3.1. Assuming a car requires protection, an identification of what steps a thief should undertake to steal (threat) the car (asset) is helpful. In order to steal a car, the thief has to both unlock the door, and start the engine. Unlocking the door can be done by either picking the lock, or smashing the window (vulnerabilities). Starting the engine can be done by either hotwiring the contact, or by putting a screwdriver in the ignition. So, in order to protect the car the vulnerabilities should be mitigated by implementing security mechanisms. An attack tree based on this example would look like Figure 30 .

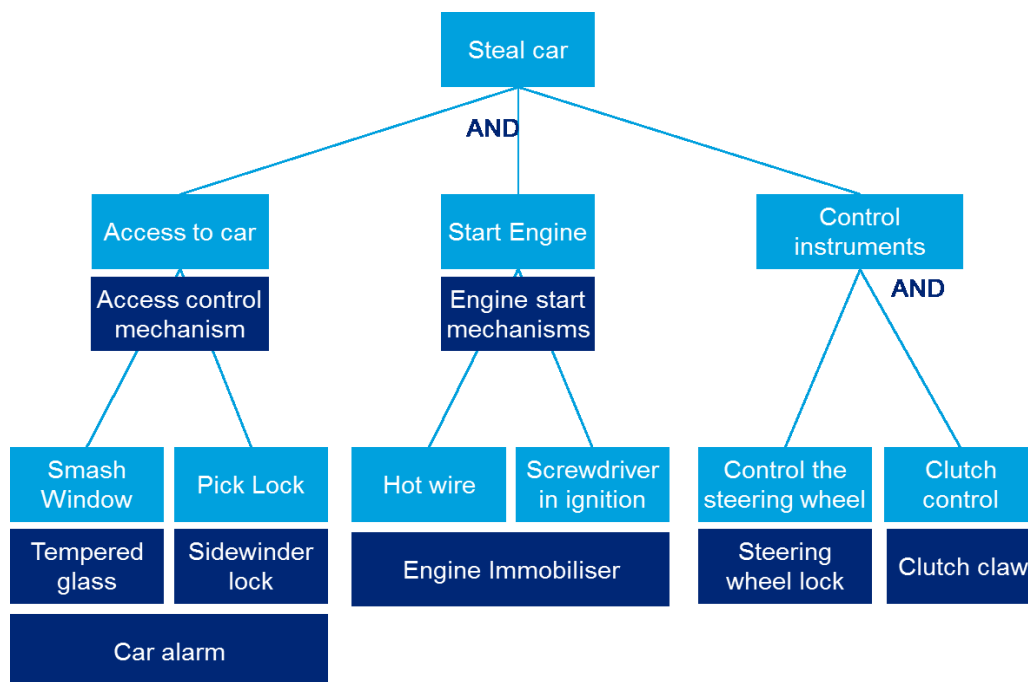


Figure 30: Attack and defence tree on stealing a car

The risk of having a door unlocked by a smashed window is mitigated by installing bullet proof glass or a car alarm. The way this can be implemented may vary per car, or brand, but the mechanism remains the same. This is the level of specificity aimed for in the language. By providing an attack tree, the motivation for –for example– installing a car alarm is provided.

The specification for the car alarm is the security policy related to it. This describes what is allowed and what is not allowed to do. Other concepts and their definition are specified and illustrated in the language specification.

5.4.2.2 LANGUAGE SPECIFICATION

This chapter provides a specification of the concepts in the proposed security extension. The following concepts are described: vulnerability, threat, risk, security policy, and security mechanism. These are the concepts selected in the previous section.

A. Vulnerability

A vulnerability is the flaw or weakness in system security procedures, design, implementation or internal controls that could be intentionally exploited and result in a security breach or a violation of the system's security policy (NIST, 2012).

The weakness in the system becomes a vulnerability when it is exposed to someone or something. The act of speaking for a large public in itself is not a vulnerability, it becomes one, because you might be exposed to for example criticism. The notation for vulnerability is outlined in Figure 31.

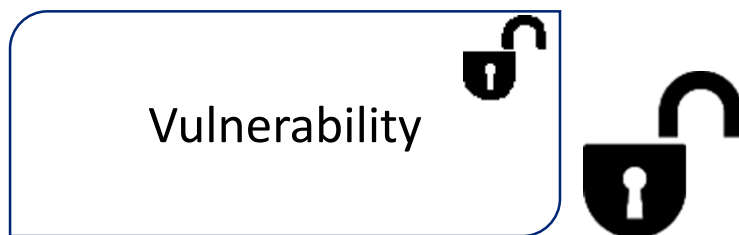


Figure 31: Vulnerability notation

Example

The use of a vulnerability concept is illustrated by the example of the claim handling process at ArchiSurance, an insurance provider. The process comprises the following steps: claim register, accept, valuate, and pay. A vulnerability in the process step 'accept' is that the acceptance of a claim is done by just one employee. This vulnerability allows for certain threats. The example is illustrated in Figure 32.

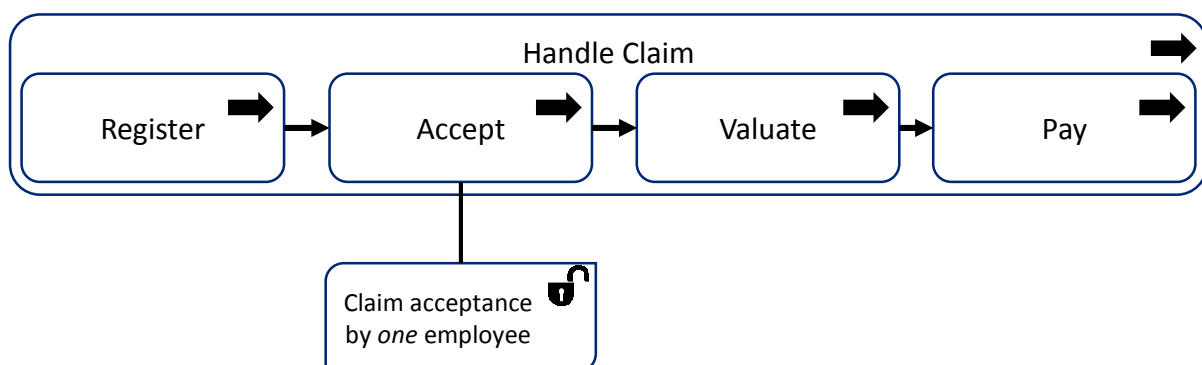


Figure 32: Example of vulnerability

B. Threat

A threat is the potential for an actor with a certain motivation to exploit a vulnerability (NIST, 2012).

The actor in this definition can be, but is not limited to, a person, an organisation or a government. The motivation for this actor to exploit the vulnerability can be financial, political, boredom or for publicity. The actor and motivation together form a threat source.

By stating that the threat actor has a certain motivation, it is also indicated that the action is performed intentionally. The notation for threat is outlined in Figure 33.



Figure 33: Threat notation

Example

An example of a threat for the specific vulnerability 'claim acceptance by *one* employee' is that the employee accepts their own claim. The example is illustrated in Figure 34.

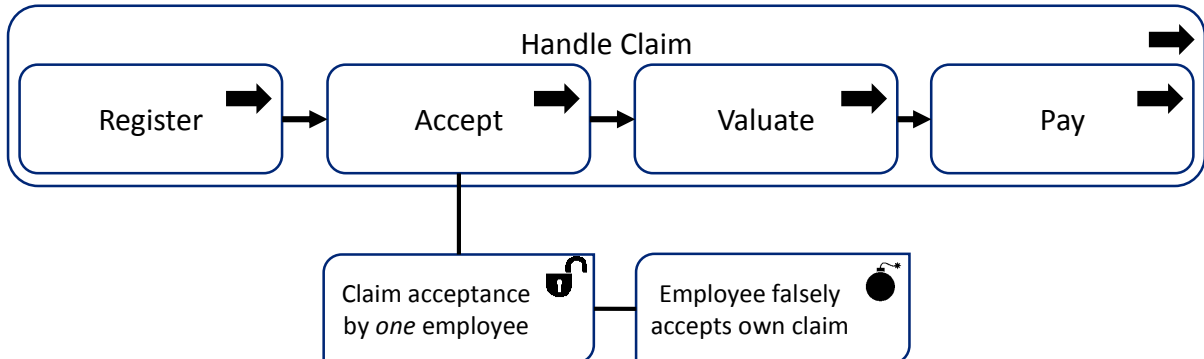


Figure 34: Example of threat

C. Risk

Risk is the net mission impact considering (i) the probability that a particular threat-source will intentionally exploit a particular vulnerability and (ii) the resulting impact if this should occur (NIST, 2012).

Often risk is seen as the combination of threats, vulnerabilities and (business) impact. Risk might for example be a financial loss or a loss of reputation. The notation for risk is outlined in Figure 35.



Figure 35: Risk notation

Example

A risk when an employee accepts its own claim lies in the financial loss. Multiple types of risk can be associated with threats. The example is illustrated in Figure 36.

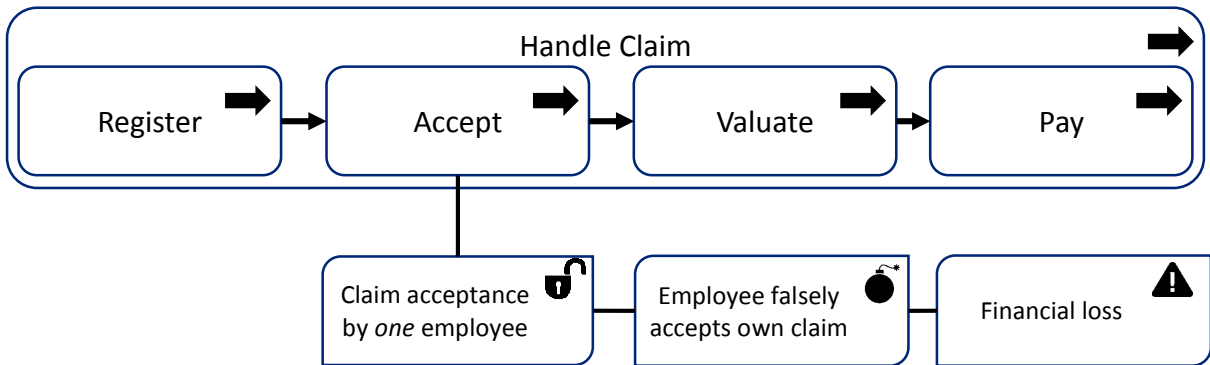


Figure 36: Example of risk

D. Security mechanism

A security mechanism is a method, tool, or procedure for enforcing a security policy. It is designed to detect, prevent, or recover from a security attack (Bishop, 2004). The notation for security mechanism is outlined in Figure 37.



Figure 37: Security mechanism notation

Security mechanism versus security control

Within security discipline an often used concept is ‘security control’. Instead of security control, the term security mechanism has been chosen. There is a slight but important difference between these two concepts, which has to do with the abstraction level mentioned in the concept selection phase. A security mechanism defines *how* the risk is mitigated, and a security control describes by *what measure* the risk is mitigated. A security mechanism can be implemented in various ways, resulting in a different security controls, while the mechanism remains the same. An example of the difference between security mechanism

and control came up in the car stealing example outlined in Figure 30. The access control mechanism prevents the car from being accessed. This can be implemented by various security controls, for example by tempered glass to prevent the window from being smashed, or by a sidewinder lock to prevent lock picking. The mechanism remains identical, while the control varies. The higher abstraction level is useful to reason about the pros and cons of the mechanism.

Example

An example of a security mechanism to mitigate the threat ‘employee accepts own claim’, which defines that two pair of eyes are needed to approve a certain action. The example is illustrated in Figure 38.

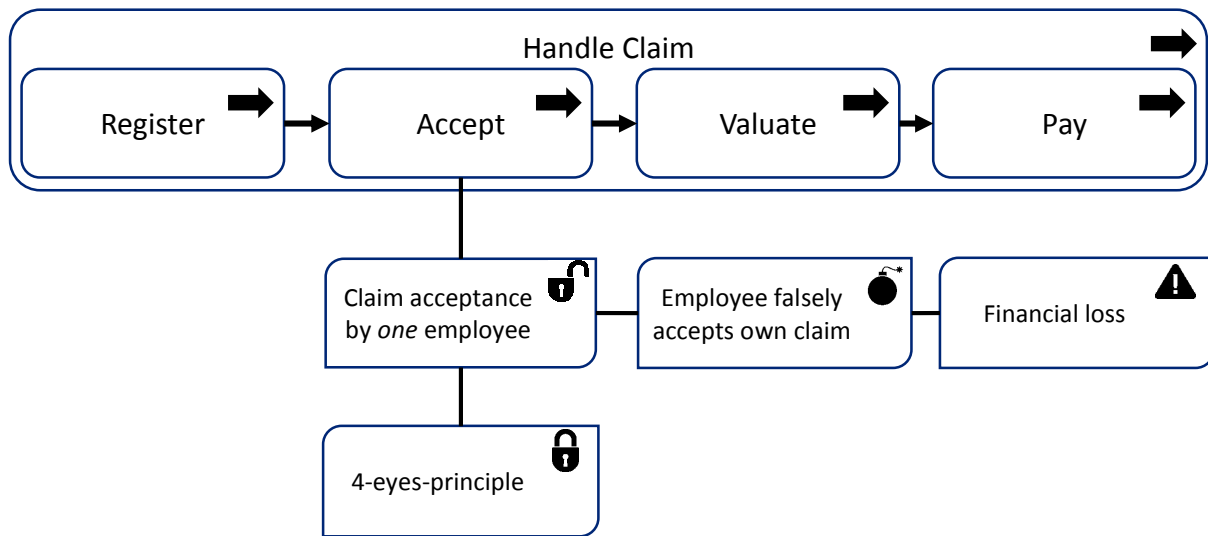


Figure 38: Example of security mechanism

E. Security policy

A security policy is a statement of what is, and what is not allowed (Bishop, 2004; NIST, 2012).

In the policy might be defined that a certain action requires approval, or an information disclosure is only allowed for people with a certain role in the organisation. The notation for security policy is outlined in Figure 39.



Figure 39: Security policy notation

Example

An example of a security policy for the specific security mechanism '4-eyes-principle'. It specifies when two pair of eyes are needed to accept a claim. The example is illustrated in Figure 40.

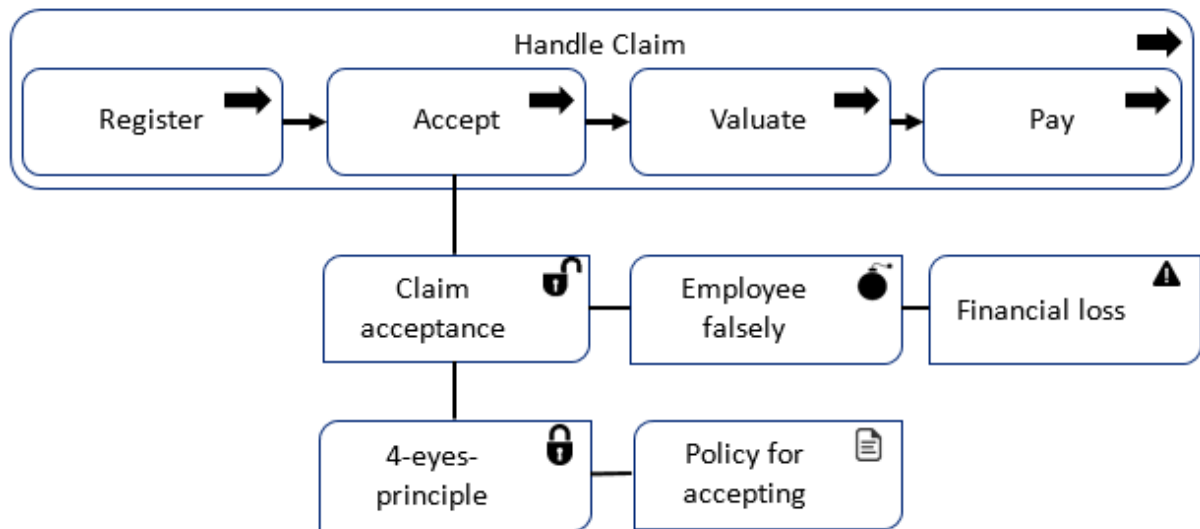


Figure 40: Example of security policy

5.4.2.3 SECURITY EXTENSION METAMODEL

To illustrate the relation among the extension concepts, and those between them and the ArchiMate core, a metamodel is developed. The model is outlined in Figure 41 and explained below.

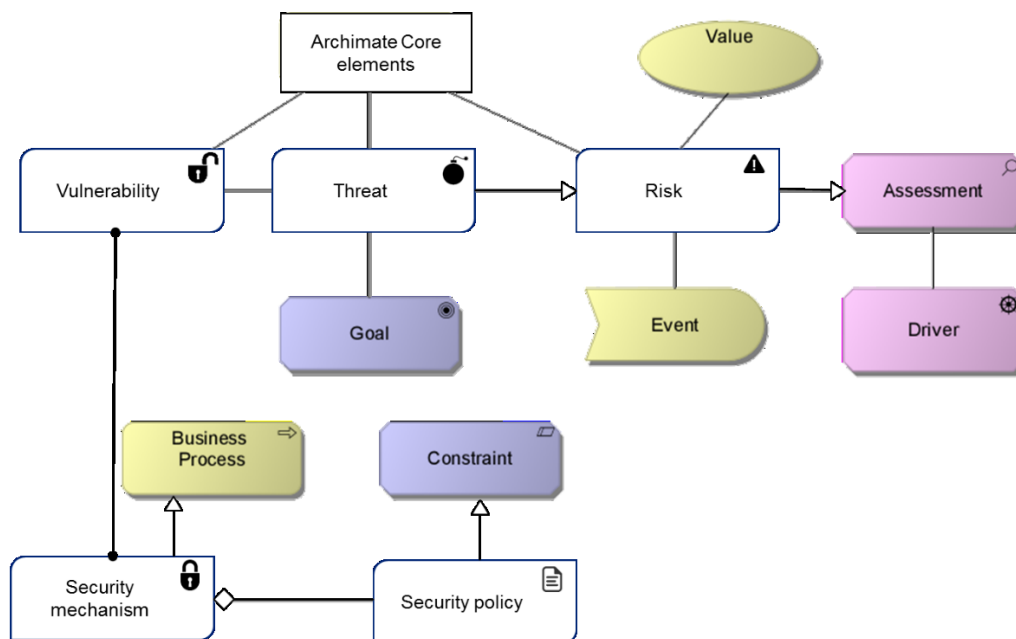


Figure 41: ArchiMate security extension metamodel

As indicated in the definition, a *vulnerability* can be associated to any ArchiMate core element. Furthermore, a vulnerability is associated to a threat in the sense that a threat exercises a vulnerability to result in a security breach.

A *threat* is a specialization of risk. It is specialized in the sense that it has a certain likelihood and impact, but is also linked to a vulnerability, and has a certain goal.

Risk is the threat's probability multiplied by the impact on the organisation. The probability is indicated with the 'Event' construct from ArchiMate, and the impact is indicated by the construct 'Value'. It is a specialized form of assessment.

A security policy is, by definition, a specialized form of a constraint, as it prescribes how a security mechanism works.

A security mechanism is a specialized form of a (business) process because, by definition, it is a method, tool, or procedure that enforces a security policy, and thus essentially a process. It is related to the vulnerability by an assignment relation, because it addresses a certain vulnerability.

5.4.3 VIEWPOINTS

Establishing and maintaining an architecture is a complex task, because of the involvement of different stakeholders. In order to handle this complexity, the concept of viewpoints is introduced. A viewpoint defines an abstraction on the set of models representing the enterprise architecture, aimed at a particular type of stakeholder and addressing a particular set of concerns. Viewpoints can be used to view certain aspects in isolation, and to relate two or more aspects. Viewpoints help to eliminate complexity by focussing on particular aspects of the architecture. What is shown and what is not shown in a view depends on the scope of the viewpoint and on what is relevant to the concerns of the stakeholder (The Open Group, 2013).

The viewpoints can be classified by their purpose and by their level of abstraction. The purpose can either be to design, to decide, and to inform. *Design* viewpoints support architects and designers in the design process from sketch to design. *Decision* viewpoints assist managers in the process of decision-making by offering insight into cross-domain architecture relationships. *Informing* viewpoints help to inform any stakeholder about the enterprise architecture in order to achieve understanding, obtain commitment, or convince adversaries.

The level of abstraction can be an overview, coherence or detailed view. The overview abstraction level typically addresses multiple layers and aspects. In the coherence view multiple layers or multiple aspects are spanned. And the detailed view typically considers one layer and one aspect from the framework. In Figure 42 an overview of the classification of Enterprise Architecture viewpoints is depicted, and typical stakeholders are addressed for each purpose (The Open Group, 2013).

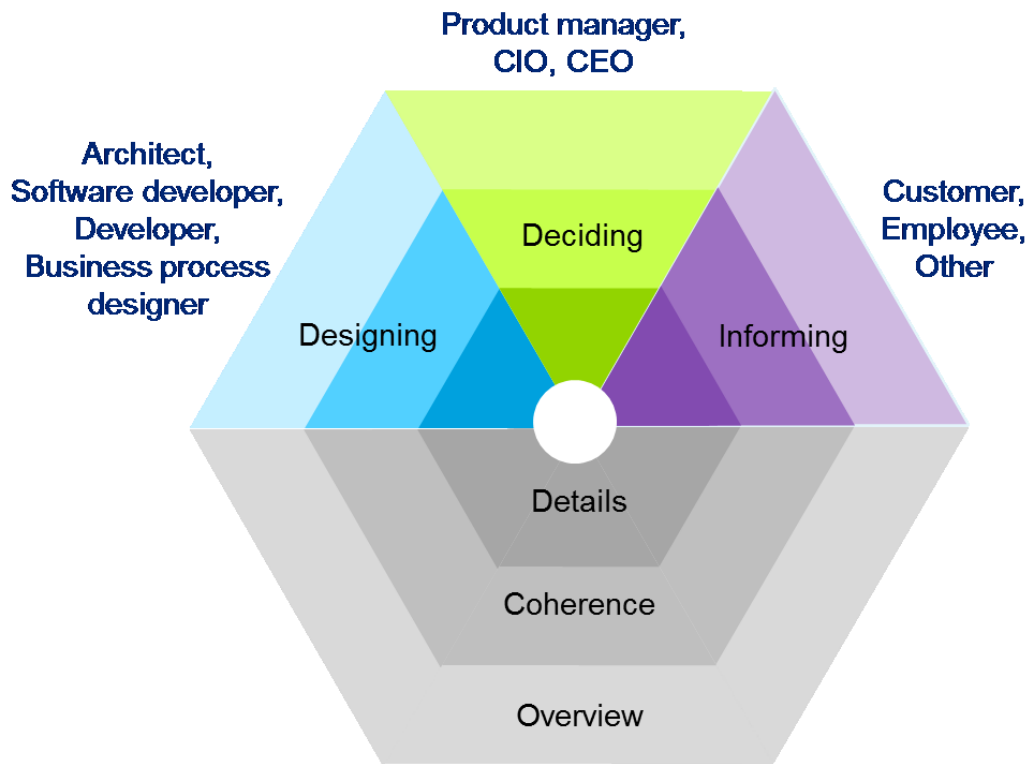
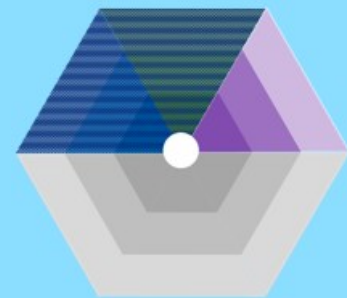
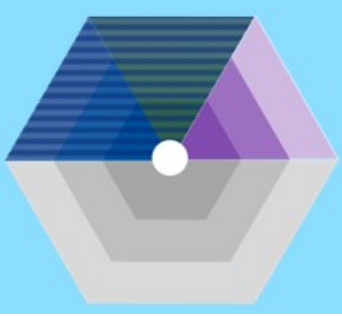


Figure 42: Classification of Enterprise Architecture Viewpoints (The Open Group, 2013)

Within the context of the security extension two additional viewpoints have been defined, and related to stakeholders. The viewpoints are summarized below and indicate stakeholders and their concerns, the purpose of the view, its abstraction level, the corresponding ArchiMate layer and encompassed aspects.

Risk analysis viewpoint	
Stakeholders	Security experts, enterprise architects, business managers
Concerns	Identification of vulnerabilities, threats, and risks within the organisation
Purpose	Designing, deciding
Abstraction level	Overview, coherence, detail
Layer	Business, information, application, and technology layer
Aspects	Active structure, behaviour, passive structure



Risk mitigation viewpoint		
Stakeholders	Security experts, enterprise architects, business managers	
Concerns	Balancing the risks and controls within the organisation	
Purpose	Designing, deciding	
Abstraction level	Overview, coherence, detail	
Layer	Business, information, application, and technology layer	
Aspects	Active structure, behaviour, passive structure	

The risk analysis and risk mitigation viewpoint are also related to Zachman and SABSA. They are introduced to describe the viewpoints on a contextual, conceptual, logical, and physical architecture level, answering the specific questions ‘how’, ‘who’, and ‘why’.

Zachman / SABSA	What	How	Where	Who	When	Why
Contextual						
Conceptual						
Logical						
Physical						
Out-of-context / component						
Functioning enterprise / Service management						

Figure 43: Relation of risk analysis and risk mitigation viewpoint to Zachman and SABSA framework

5.5 AN INTEGRATED APPROACH

To conclude, in this chapter an integrated framework, method, and modelling language are specified. These are considered to be the main ingredients for an approach to design secure enterprise architectures. In order to do so, common standards as Zachman, SABSA, TOGAF ADM, and ArchiMate are used and a security extension to ArchiMate is proposed.

Furthermore, the relation among the standards themselves and between these standards has been described. This combines to an integrated approach, which has two main purposes:

- Provide assistance on assessing the current enterprise architecture, mapping the security measures onto the enterprise architecture;
- Provide assistance on modelling a target architecture based on the baseline architecture and a risk analysis.

Both purposes were enabled by providing the approach and common language between the enterprise architecture and security discipline.

6 DEMONSTRATION

This section provides a demonstration of the approach developed and described in chapter 5. It starts with a description of the case in section 6.1, and subsequently demonstrates how the developed approach can be applied to the case in section 6.2.

6.1 ARCHISURANCE CASE

The ArchiSurance case (Jonkers, Band, & Quartel, 2012) is chosen to illustrate the approach. The case is frequently used within the enterprise architecture discipline. ArchiSurance is a fictitious company, providing all kinds of insurances: home, travel, car, and legal aid. It sells its services through a network of intermediaries. Its primary operations are (1) maintaining customer and intermediary relations, (2) contracting, (3) claims handling, (4) financial handling, and (5) asset management.

One of the drivers of the board of ArchiSurance is a secure claim handling process. In order to reach a secure claim handling process, the baseline architecture is determined. Subsequently a security analysis on the architecture is performed, and risks are identified. In order to mitigate these risks, a target architecture is proposed accordingly.

The scope of the case is limited to the process of claim handling in ArchiSurance, and covers the business process and services, the supporting applications and the IT infrastructure that enables the claim handling process.

The approach as demonstrated (consisting of a framework, method and modelling language) provides an enterprise architecture with integrated security analysis in the baseline architecture, as well as in the process of designing the target architecture.

6.2 APPLICATION OF SOLUTION DESIGN

This section is structured according to the phases of the proposed method, based on TOGAF ADM. For each of the phases the relevant viewpoints of the Zachman and SABSA framework are used, and the architecture is specified according to the ArchiMate specification language and the proposed security extension. The relation of the phases to the frameworks, and ArchiMate is described in section 5.3.3.

For the purpose of demonstration, output as defined in the method description of the solution design is provided. The provided output is not exhaustive. For each specific case various deliverables may be necessary, and are not limited to the output provided in this example.

6.2.1 PRELIMINARY PHASE

The preliminary phase is for constructing the stakeholder model, identifying business principles, goals and –drivers, and the identification of key risk areas.

Stakeholder view

The stakeholder view describes the stakeholders to be considered in the architecture project, related to the drivers they have. The stakeholder set encompasses everyone with a stake in the architecture project, and is therefore not limited to actors in the architecture project only.

A fragment of the stakeholder view is depicted in Figure 44. For the purpose of illustration, the scope of the stakeholder view is limited to one stakeholder; the board of the organisation. The stakeholder 'Board' has several concerns, one of which is the security of the claim handling process. The resulting driver: a secure claim handling process, contributes to a controlled finance.

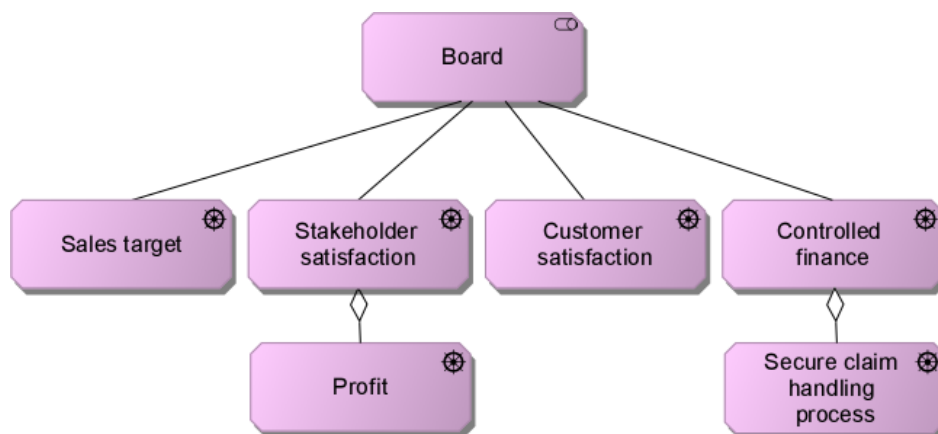


Figure 44: Fragment of Stakeholder view for ArchiSurance

In order to reach a secure claim handling process, the baseline architecture is constructed, a security analysis is performed, and a target architecture is proposed.

Business goals and drivers

The driver 'Secure claim handling process' and its related goals are outlined in Figure 46. This viewpoint is described in the Zachman and SABSA framework by the intersection of the contextual perspective and the question 'why': list of business goals / strategies (Zachman), and the business risks (SABSA). This is illustrated in Figure 45.

The elaboration of the diagram in Figure 46 is on a lower abstraction level than the diagram outlined in Figure 44. The viewpoints in the frameworks Zachman and SABSA do not take this difference in abstraction level into account.

Zachman / SABSA	What	How	Where	Who	When	Why
Contextual						
Conceptual						
Logical						
Physical						
Out-of-context / component						
Functioning enterprise / Service management						

Figure 45: Zachman and SABSA view on the outlined business goals and drivers

Main drivers include ensuring information confidentiality and integrity. Both are positively influenced by a prevention of unauthorised access, and the integrity is also positively influenced by a prevention of synchronization errors.

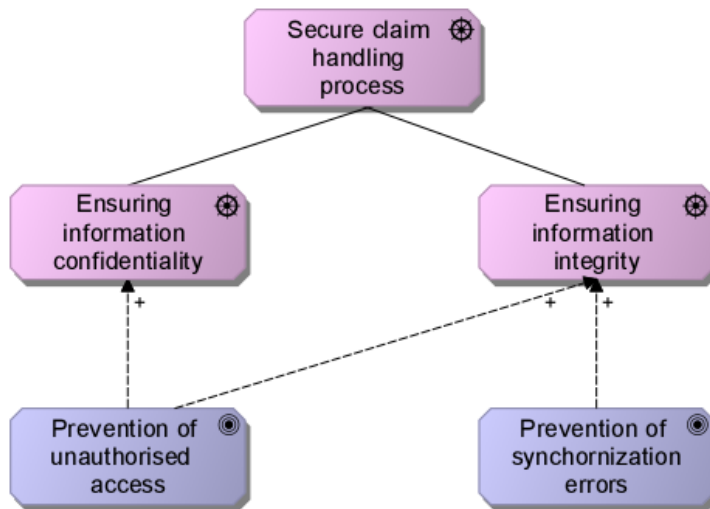


Figure 46: Business goals associated with the driver Secure handling of information

Risk appetite

The identified risks within ArchiSurance are mapped in the risk analysis matrix, defining the impact of the risk on the x-axis, and the likelihood of the risk on the y-axis. This risk analysis matrix has the scope of the claim handling process within ArchiSurance.

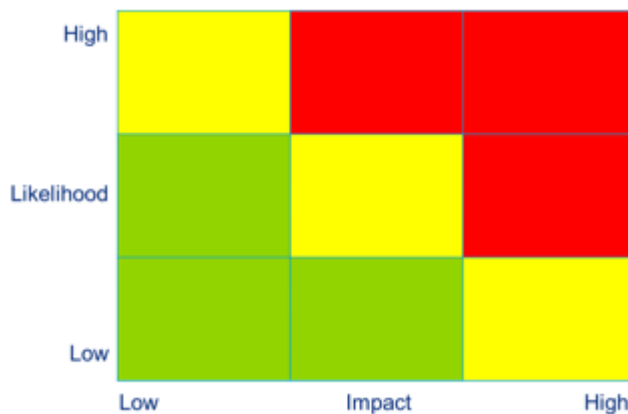


Figure 47: Risk analysis matrix

The risk analysis matrix is depicted in Figure 47. The risks having an impact and likelihood classification of ‘high’ (red cells) should be mitigated by default. For the risks plotted in the yellow cells (impact high and likelihood low, or likelihood high but impact low), it should be decided by the board whether or not measures should be taken. The risks plotted in the green cells (likelihood and impact low), fall by definition within the risk appetite of ArchiSurance and therefore no mitigating measures are taken.

Request for architecture work

The request for architecture work is based on the concerns of the board regarding the secure claim handling process, one of the drivers for controlled finance. The request is formulated as follows:

“Reassess the security of the claim handling process and, where applicable, identify how the corresponding security mechanisms can be implemented in the enterprise architecture”

The request is performed in the following sections.

6.2.2 PHASE A: ARCHITECTURE VISION

The objectives of the architecture vision phase are to develop a high-level aspirational vision of the capabilities and business value to be delivered as a result of the proposed enterprise architecture. Furthermore, an approval has to be obtained for a Statement of Architecture Work that defines a program of works to develop the architecture outlined in the Architecture Vision.

Statement of work

In order to assess the security of the claim handling process and, where applicable, identify how the corresponding security mechanisms can be implemented in the enterprise architecture, the following activities will be carried out:

- Construct the baseline architecture;
- Perform security analysis;
- Construct the target architecture.

The reason for assessing the current state of enterprise architecture, and constructing a target state of the architecture, is the fact that the board is concerned about the security of the claim handling process. The scope of the architecture project is limited to this business process, and their supporting information systems and technology architecture.

6.2.3 PHASE B: BUSINESS ARCHITECTURE

The business architecture phase focuses on constructing the business architecture, the baseline as well as the target architecture.

Baseline Business Architecture

For constructing the baseline business architecture, the conceptual perspective on the questions ‘how’ and ‘who’ are used. This is illustrated in Figure 48.

Zachman / SABSA	What	How	Where	Who	When	Why
Contextual						
Conceptual						
Logical						
Physical						
Out-of-context / component						
Functioning enterprise / Service management						

Figure 48: Zachman and SABSA view on the outlined business architecture

The constructed architecture viewpoint including the security analysis is depicted in Figure 49.

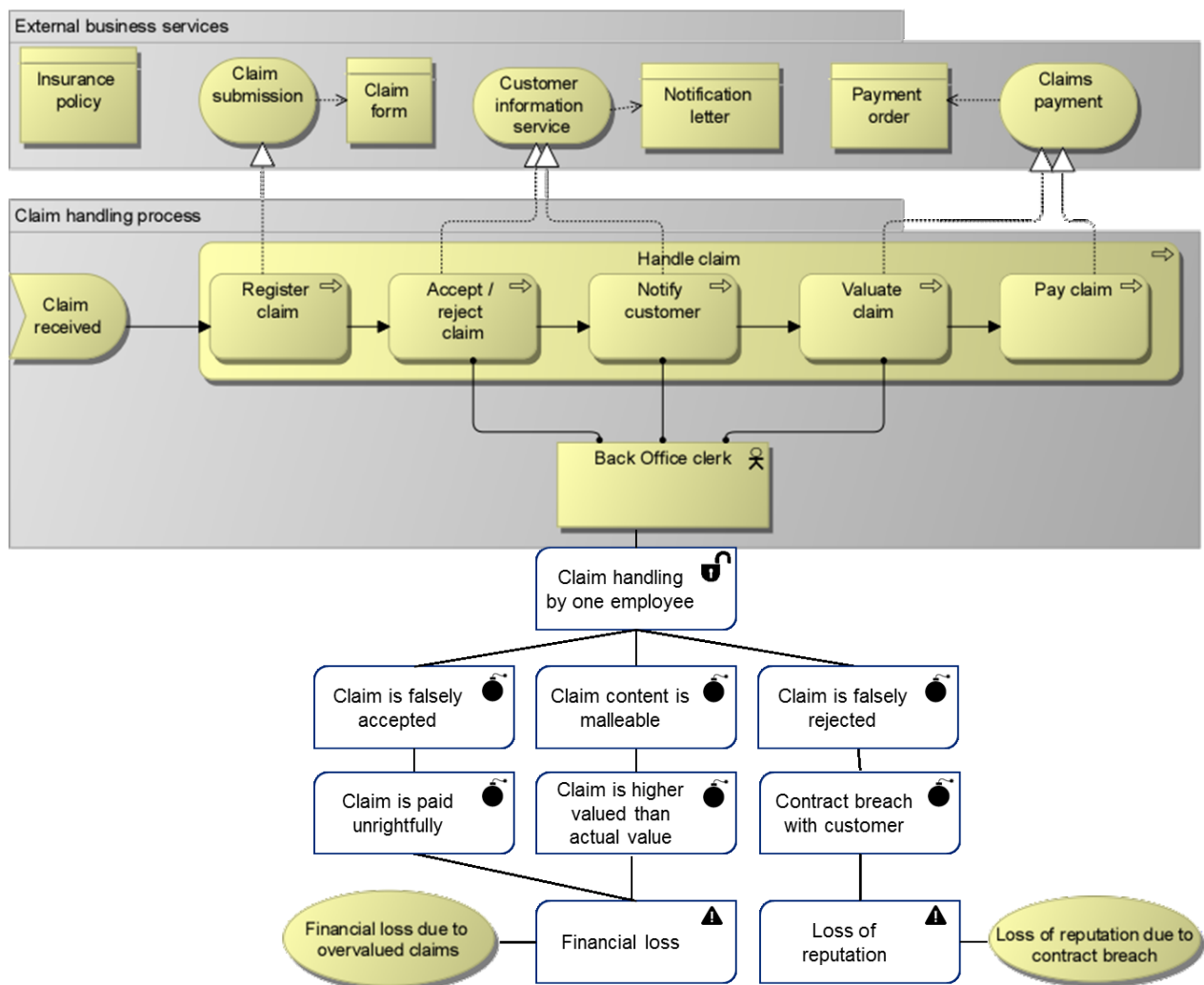


Figure 49: ArchiSurance’s baseline business architecture

The security analysis has revealed three vulnerabilities in the claim handling process, based on the vulnerability that the claim handling process is accomplished with one employee:

- 1) The claim is falsely accepted, resulting in the claim being paid unrightfully
- 2) The claim content is malleable, resulting in that the claim might be higher valued than the actual claim value.
- 3) The claim is falsely rejected, resulting in a contract breach with the customer.

For each of these threats a risk analysis is performed, defining the likelihood that the event occurs and the resulting impact of the threat. Based on the analysis, it is decided whether or not the risks are within the risk appetite of the organisation. A green cell indicates the risk is within the risk appetite, a yellow cell indicates that a decision of the board is needed, and a red cell indicates that a corresponding security mechanism is needed to mitigate the risk in such a way that it falls within the risk appetite. The corresponding viewpoint in SABSA is the risk management viewpoint, which is illustrated in Figure 50. The risk analysis is depicted in Figure 51.

SABSA	What	How	Where	Who	When	Why
Contextual						
Conceptual						
Logical						
Physical						
Out-of-context / component						
Functioning enterprise / Service management						

Figure 50: SABSA view on the outlined risk analysis

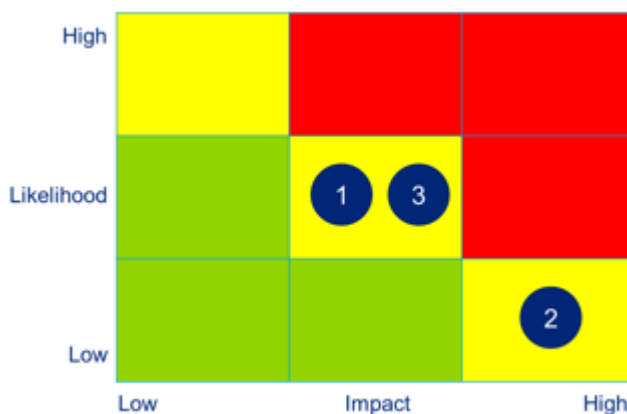


Figure 51: Risk analysis on Archinsurance's business architecture

Based on the risk analysis, it is decided to mitigate the risks with corresponding security mechanisms. These are outlined in the target business architecture, depicted in Figure 52.

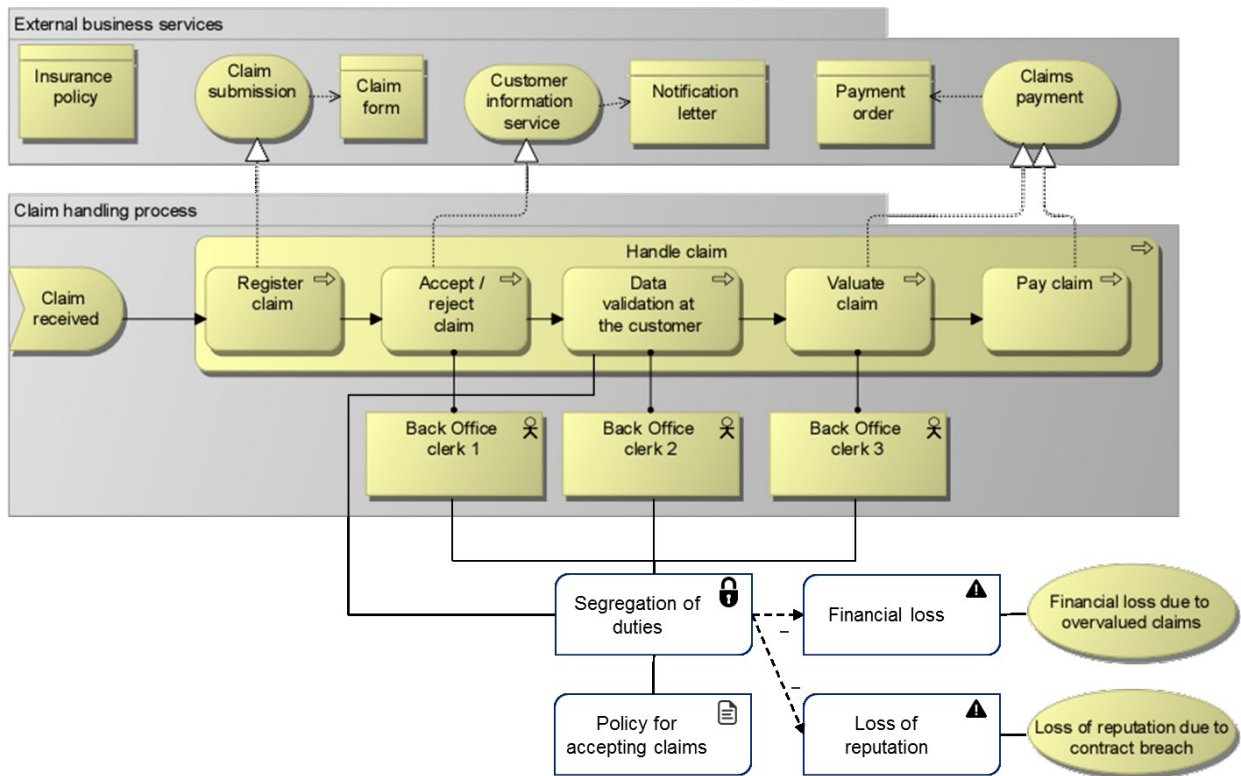


Figure 52: Archinsurance's target business architecture

In the target architecture, the risks discovered in the risk analysis on the baseline architecture are mitigated, by introducing the security mechanism 'separation of duties'. The security mechanism can be implemented in two ways, function apart or combined:

- The duties can be segregated within the organisation, by requiring three back office clerks to accept the claim
- The duties can be separated in combination with an external factor: data validation at the customer. The customer then must validate that the claim was accepted with the right information, mitigating the risk of claim acceptance of rejecting when this should not have been the case.

The same security mechanism can be implemented in two ways. This makes clear the difference between security mechanism, and security control, referring to the discussion mentioned in section 5.4.2.2 at the security mechanism specification.

This has direct impact on the business architecture by enforcing that it takes three back office clerks to accept one claim, or that the customer is required to validate the claim acceptance.

6.2.4 PHASE C: INFORMATION SYSTEMS ARCHITECTURE

The information systems architecture aims at constructing the data and application architecture. For this domain, the baseline as well as the target architecture is constructed.

While executing this phase, two activities are important:

1. Ensuring that the mitigated risks in the business target architecture are not be undone in the information systems architecture, so the business and information systems architecture must be aligned.
2. Identifying and mitigating new risks within this architecture.

Baseline Information Systems Architecture

For constructing the baseline information systems architecture, the conceptual and logical perspective on the question ‘how’ is used. This is illustrated in Figure 53.

Zachman / SABSA	What	How	Where	Who	When	Why
Contextual						
Conceptual						
Logical						
Physical						
Out-of-context / component						
Functioning enterprise / Service management						

Figure 53: Zachman and SABSA view on the outlined information systems architecture

The constructed architecture viewpoint including the security analysis is depicted in Figure 54.

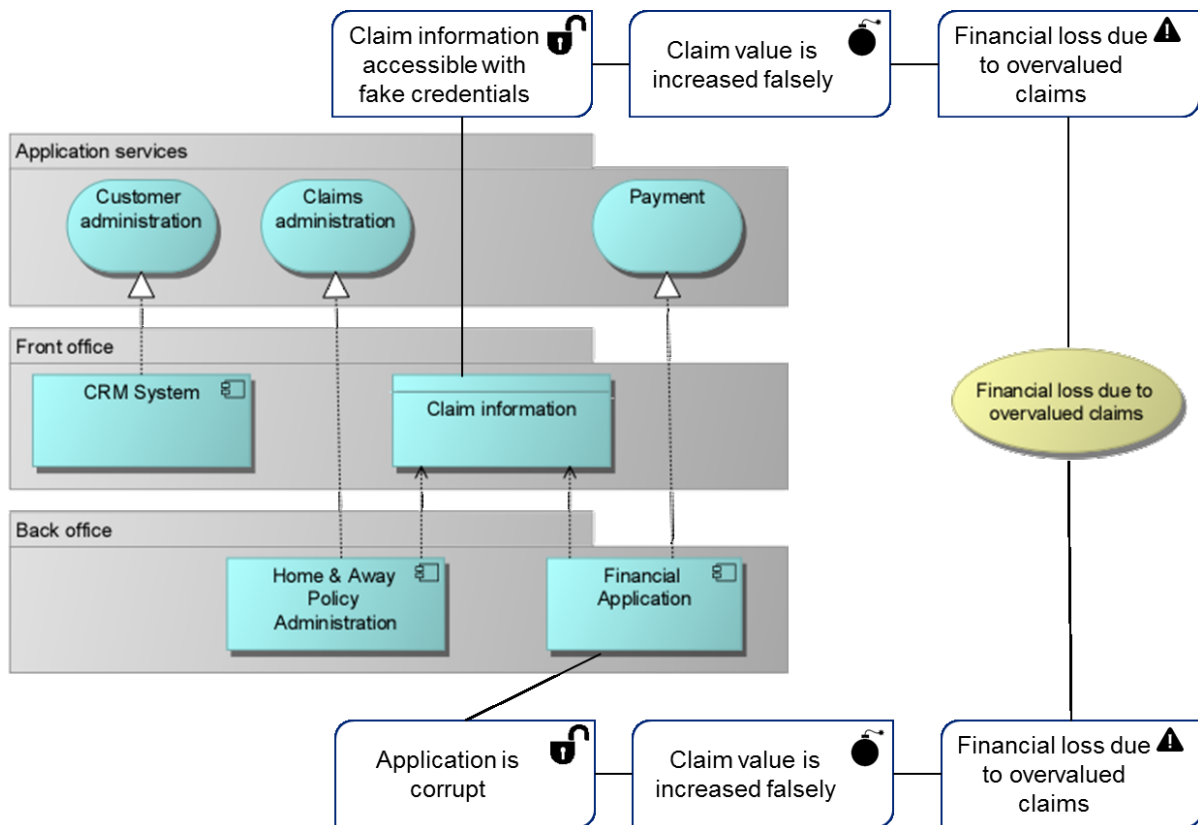


Figure 54: Archinsurance’s baseline information services architecture

The security assessment on the information services architecture has resulted in two vulnerabilities and its corresponding threats and risks:

- 1) The claim information is accessible with fake credentials. This might result in that a non-authorized actor gains access to claim information, and breaks the mechanism of segregation of duties as defined in the business architecture.
- 2) The application is corrupt, for examples due to a programming error the claim value can be increased falsely.

Both vulnerabilities and corresponding threats lead to risks, in the form of a financial loss.

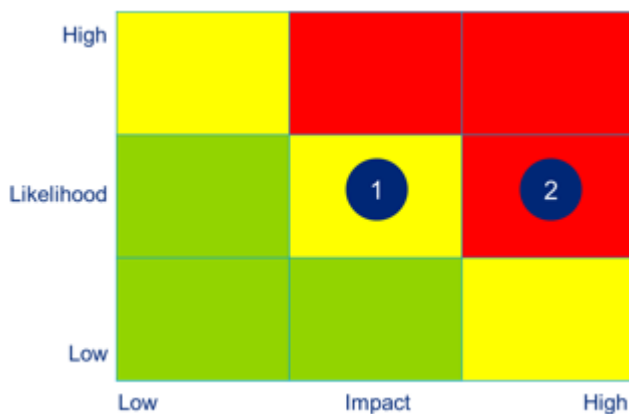


Figure 55: Risk analysis on Archisurance's information services architecture

The risk management viewpoint is outlined in Figure 55. Both risk 1 & 2 are not considered to be within the risk appetite of the process and need appropriate countermeasures. These are depicted in the target information systems architecture in Figure 56.

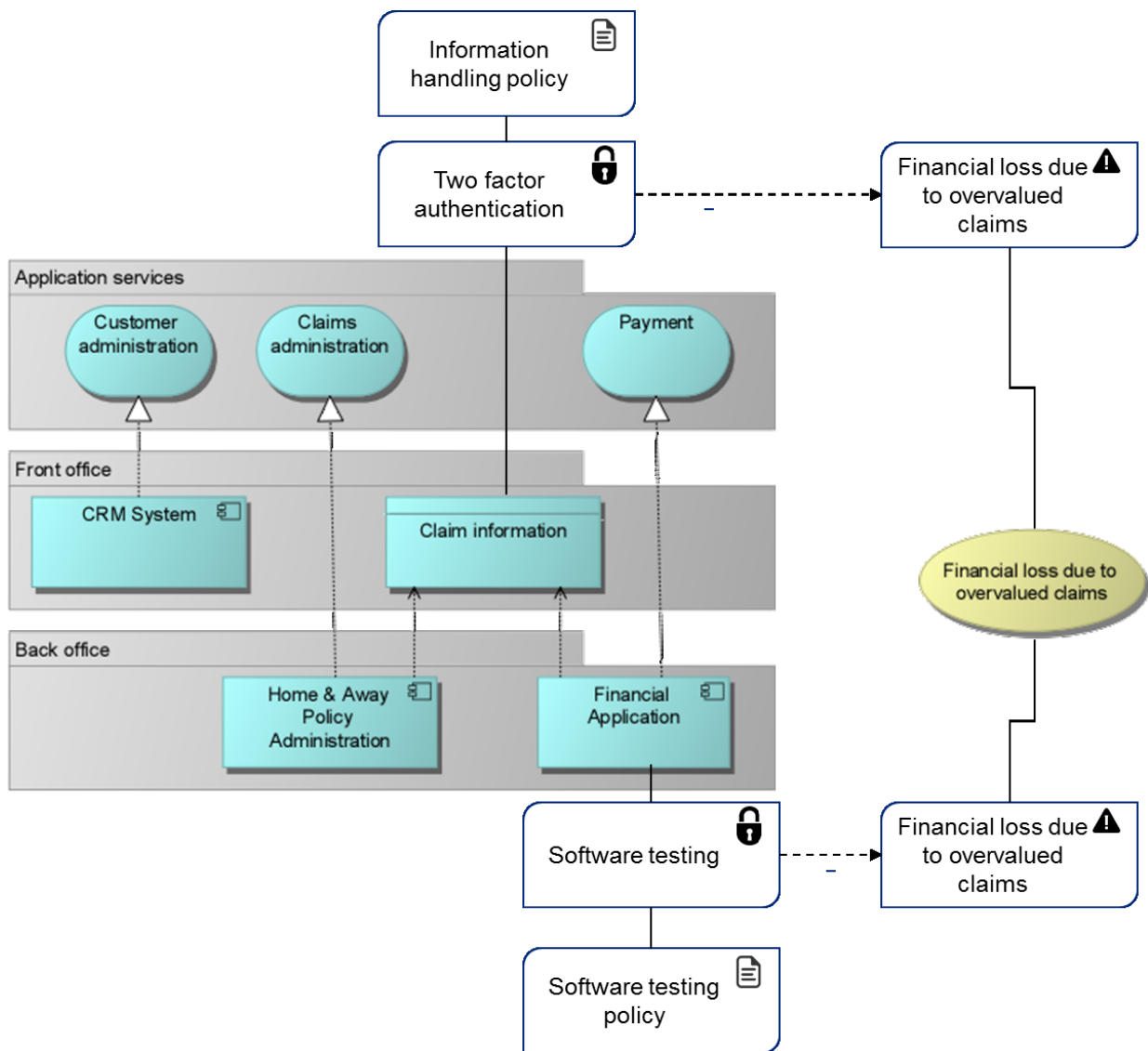


Figure 56: Archinsurance's target information systems architecture

6.2.5 PHASE D: TECHNOLOGY ARCHITECTURE: RISK ANALYSIS VIEWPOINT

The technology architecture is outlined in Figure 58. Also, the security analysis is carried out here and shown in the architecture.

While executing this phase, two activities are important:

1. Ensuring that the mitigated risks in the business & information systems target architecture can't be undone in the technology architecture, so all architecture domains must be aligned;
2. Identifying and mitigating new risks within this architecture.

The viewpoint described here is the physical perspective on 'how'. This is illustrated in Figure 57.

Zachman / SABSA	What	How	Where	Who	When	Why
Contextual						
Conceptual						
Logical						
Physical						
Out-of-context / component						
Functioning enterprise / Service management						

Figure 57: Zachman and SABSA view on the outlined technology architecture

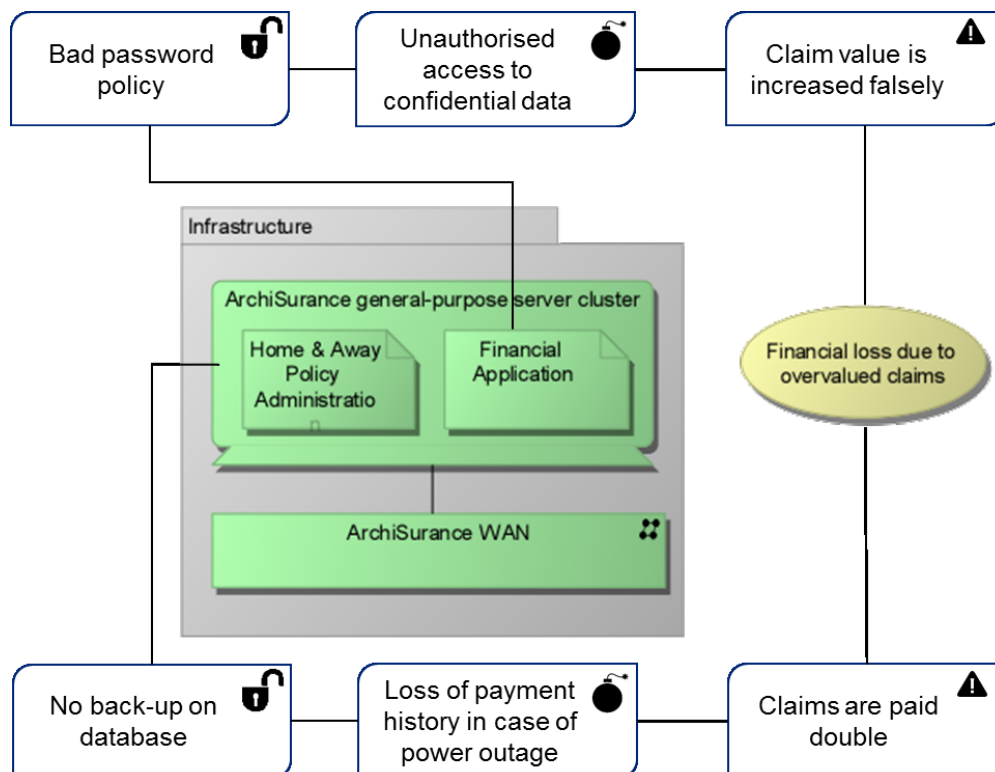


Figure 58: Technology Architecture

Two security risks are identified in the architecture:

- By having a bad password policy, unauthorised access to confidential data stored on the server might be gained. This may result in a deletion, a duplication or alteration of the data;
- By having no back-up mechanism on the server, the payment history might be lost in case of a power outage, resulting in claims being paid double.

The risks are plotted in the risk analysis matrix in Figure 59.



Figure 59: Risk analysis on Archisurance's technology architecture

In order to mitigate this risk, several security mechanisms are possible. One of the options is a two-factor authentication to gain access to the server. This security mechanisms is plotted in the architecture and depicted in Figure 60.

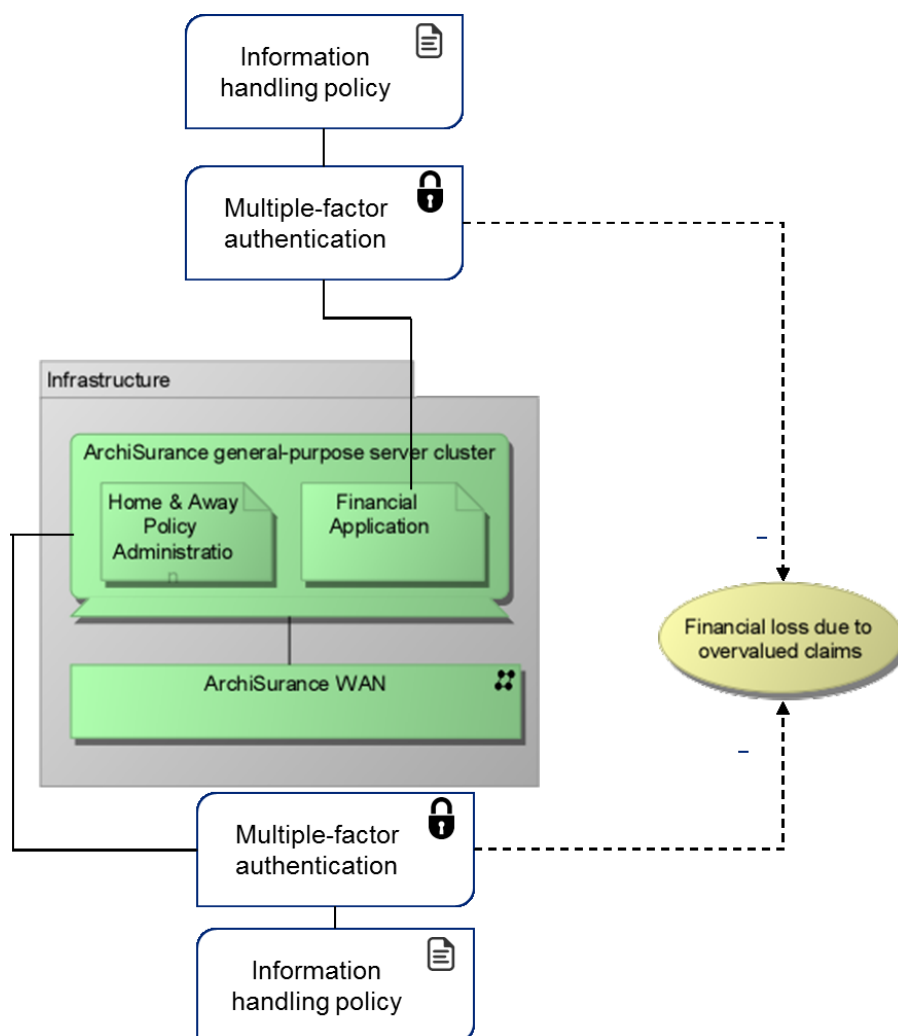


Figure 60: Archisurance's target technology architecture

The ADM phases E, F, G, and H are not considered in the demonstration, since they do not differ from regular enterprise architecture projects.

6.3 DEMONSTRATION CONCLUSION

The introduced example demonstrates the application of the secure enterprise architecture approach on the case of ArchiSurance. Although it is a bit simple, it demonstrates the application of the approach within the several layers of the enterprise architecture. It encompasses the framework of the approach with its different viewpoints, and illustrates the ArchiMate language in relation to the developed security extension. Furthermore, it indicates the several activities conducted, related to the phases of the TOGAF Architecture Development Method.

7 VALIDATION

Validating the proposed integrated approach for secure enterprise architectures is important, because it is only worth to invest in an implementation of the approach if the proposed benefits are likely to be achieved. Wieringa (2012-2013) states “A solution design is valid if the designed solution is expected to reduce the gap between experiences and desires that it set out to reduce”.

This chapter discusses the validity of the proposed secure enterprise architecture approach. The following section elaborates on the method used to validate the approach. Section 7.2 explains the research rigor of the approach. The research relevance, discussed in expert interviews, is summarized in section 7.3. Finally, section 7.4 discusses the conclusions of the validation.

7.1 METHOD

In design science, the solutions to the design problems are evaluated by their degree of utility; the degree to which they solve the previously stated problem, the gap between experiences and desires that it set out to reduce (Wieringa, 2012-2013). In order to determine whether or not the solution design is likely to solve the problem, the research rigor and the research relevance are discussed.

Research rigor in qualitative research is the ability to determine whether the conclusions are trustworthy. That makes it comparable to validity and reliability in quantitative research. The research rigor can be evaluated with the criteria specified by Whetten (1989).

The options for discussing the *research relevance* of the proposed approach are: (1) formal verification, (2) expert opinion, (3) single-case mechanism experiment, and (4) statistical difference-making experiment (Wieringa, 2012-2013). The first method is a formal verification method: it verifies the artefact specification against the requirements. Method 2, 3, and 4 are empirical validation methods.

- (1) *Formal verification* requires a formal specification of the artefact, a formal specification of the context, a formal specification of the effects that contribute to the goals and proves that the artefact in context leads to the desired properties. This might lead to interesting properties of the artefact not discoverable by informal analysis;
- (2) *Expert opinion* consists of the researcher asking practitioners about perceived usability and utility of the new artefact in the contexts that they know first-hand. It can be done with an interview, a questionnaire or a focus group;
- (3) *Single-case mechanism experiment* includes bringing the proposed approach into practice and evaluating whether or not it leads to the desired goals;
- (4) *Statistical difference-making experiment* is done when the proposed approach is brought into practice in multiple cases, and the outcome is evaluated. This provides an objective evaluation, but requires a large group of cases to be statistically significant.

For the validation of this approach, expert opinion is used. This option is most suitable because opinions from various contexts can be discussed, which is useful for measuring utility and usability. Statistical difference-making would also be an option given this criterion, however this would require too much time to complete within the boundaries of the graduation project.

Several experts from the security and enterprise architecture discipline were interviewed. Both the components of the approach as well as its overall design were discussed in this interview.

7.2 RESEARCH RIGOR

According to Whetten (1989) a theoretical contribution consists of four essential elements: What, How, Why, and the limitations of the theory by questions Who, Where, and When. These elements are discussed within the context of the secure enterprise architecture design approach.

What

This section answers the question: Which elements logically should be considered as part of the solution design and how are these elements described?

The elements of the proposed approach are: framework, method, and language. Each of these elements are defined in section 5.1 (Iacob, Jonkers, et al., 2012). The approach uses standards that are defined individually: Zachman (Zachman, 1987), SABSA (J Sherwood et al., 2009), TOGAF (The Open Group, 2011b), SABSA Lifecycle (J Sherwood et al., 2009), and ArchiMate (The Open Group, 2013). Besides these elements, an ArchiMate Security Extension is introduced. The constructs in the language are derived from the SABSA Framework and defined by using definitions from NIST (2012) and Bishop (2004)

How

After a set of elements has been identified, a description is required of these elements are related.

The relation between the elements 'framework', 'method', and 'modelling language' is defined as follows: a framework provides the viewpoints on the enterprise architecture, the modelling language defines the concepts describing the viewpoints on the architecture, and the method provides the way of working to develop architectural descriptions.

The relation among the modelling language extension concepts, and those between them and the existing ArchiMate modelling language are specified in the metamodel in Figure 41.

Why

After defining the elements and the relations among the elements, the underlying psychological, economic, or social dynamics that justify the selection of these elements and their relations should be described.

The selection of elements ‘framework’, ‘method’, and ‘modelling language’ is motivated by common use of these elements in the enterprise architecture discipline. In this discipline these elements, or a combination of these elements, are considered to be the building blocks of enterprise architecture work. Within the security discipline, these elements are also used to provide a structured approach to security work, although not all elements are specified in standards. Because the disciplines share these elements, they were selected to develop an approach to designing secure enterprise architectures with.

The selection of concepts for inclusion in the modelling language extension are derived from the SABSA framework, which describes various viewpoints on security. By selecting the concepts from this framework, it is ensured that the viewpoints described in SABSA can be modelled. Consequently, in order to select the concepts that are relevant to include in the architecture, a selection made from the SABSA concepts, by evaluating the level of semantic overlap and the level of generality the concepts encompass. Furthermore general selection criteria for describing ArchiMate language extensions are taken into consideration.

Who, where, and when

These conditions place limitations on the propositions generated from a theoretical model. These temporal and contextual factors set the boundaries of generalizability.

The boundaries of the approach are limited to organisations that are conducting enterprise architecture. Small businesses such as the local grocery store are not included, even though this type of business also has to deal with security. The secure enterprise architecture approach has a focus on medium to large firms, which conduct enterprise architecture and typically have difficulty to manage security due to complexity and scale.

7.3 RESEARCH RELEVANCE

For the external validation, several experts were interviewed one-on-one, during 1 – 1.5 hour sessions. During the interviews, several questions were asked about the validity and utility of the proposed approach. The list of interviewed experts, their role within the organisation and the sector they are active in, is included in Table 10. The questions discussed during the interview are included in Appendix C – Interview Questions.

Table 10: Validation expert interview list

Expert	Role within organisation	Organisation / Sector
Expert 1	Enterprise Security Architect	Financial services - banking
Expert 2	Enterprise Security Architect	Financial services - insurance
Expert 3	Researcher	Research institute / electronics
Expert 4	Enterprise Architect	Aviation
Expert 5	Manager Risk Services	Professional services
Expert 6	Chief Security Architect	Financial services - banking

The results of the validation interviews are discussed per question.

7.3.1 GENERAL USEFULNESS OF APPROACH

All interviewed experts agree on the usefulness of an approach where enterprise architecture and security are integrated. Experts from the enterprise architecture as well as security discipline agree on having an integrated approach is useful. The approach would enrich the enterprise architecture discipline with security, and would include security in the design cycle.

7.3.2 INGREDIENTS OF THE APPROACH

The ingredients of the approach: framework, method, and language are the appropriate ingredients according to most interviewed experts. One of the interviewed experts suggested the use of 'roadmaps' as an additional ingredient to the approach. A roadmap describes which viewpoints from the framework, and which phases from the method are used in specific cases.

One of the interviewed experts did not consider the addition of language to be valuable for the approach. According to the interviewee, an integration of the security components into the enterprise architecture language increases the complexity of the already complex views on the architecture.

7.3.3 FRAMEWORK

To the knowledge of the interviewees, Zachman and SABSA and its relation are correctly described and used. One of the remarks was that the vocabularies of Zachman and SABSA do not always correspond. This is the case for example with the use of 'services'. Zachman limits the use of this concept to describing the value offering from the business to the customer, while SABSA has a broader use of this concept. This is important to keep in mind when using both frameworks.

Not all interviewees use Zachman and/or SABSA in their organisation. This is mainly because their organisation did not conform to the framework. Two of the interviewed organisations have chosen to use the Novius model instead.

7.3.4 METHOD

The use of TOGAF ADM and its integration with the SABSA lifecycle is correctly done and applicable to practice. Especially its wide-spread and intuitive use is in favour of using this method.

7.3.5 LANGUAGE

The introduction of language is generally considered useful. One of the experts did not agree upon adding security concepts to the enterprise architecture modelling language, mainly because of the increased complexity in the architecture drawings. For those experts who

considered the language component useful in the proposed approach, specific concepts are discussed below.

The construct *Vulnerability* is introduced logically and correctly used. The definition is in line with both disciplines and the construct is considered useful for modelling in an enterprise architecture.

The construct *Threat* is introduced logically and correctly used. An additional notion of the capability of the attacker (actor) could be taken into account, which would be necessary for calculating the likelihood of a successful attack.

The construct *Risk* is defined as the combination of likelihood of the occurrence of the event and the resulting impact might this occur. After reducing the risk by the introduction of a security mechanism, a residual risk might still exist. Two interviewed experts indicated that it would be useful to model this residual risk.

The construct *Security Mechanism* is logically introduced and correctly used. One of the experts indicated that the term *Mechanism* might be confusing, because in the security industry the construct 'security control' is often used instead.

The construct *Security Policy* is introduced logically and correctly used. One of the experts indicated that this construct is often forgotten and indeed an important part of the security solution.

7.3.5.1 METAMODEL

The introduced metamodel, in which the relation between the introduced concepts and the existing concepts is described, is correct. One interviewee indicated that it is useful to also take into account the moderating effect of a security mechanism on risk.

One of the key points indicated by the interviewees is the fact that multiple vulnerabilities might be mitigated by one security mechanism, and multiple security mechanisms might be needed to mitigate one vulnerability. This might result in a complex graph, reducing the overview and insights.

7.4 VALIDATION CONCLUSIONS

The approach is validated both internally and externally by reviewing the design process and expert interviews. Overall the feedback was positive, however several remarks were made by the experts. The remarks are summarized in Table 11, along with an indication of how to process the feedback.

Table 11: Validation conclusions

Remark	Number of experts	Implications for the approach
Used framework does not fit the organisation	2	Provide integration of other frameworks with the secure enterprise architecture design approach.
Model the residual risk	2	Indicate the residual risk with a certain colour or specialization of risk.
Inclusion of security concepts in enterprise architecture language increases the complexity of the view	1	In theory, the approach is also applicable without the language component. However, the introduction of security constructs into the language has two main advantages: (i) it allows for analysing the dependencies between the various layers of the architecture, and (ii) the introduction of a common language allows for better understanding and re-usability. Viewpoints are introduced in order to decrease the complexity, allowing to focus on certain parts of the architecture.
Rephrase the term security mechanism to security control	1	The construct security mechanism could be rephrased to security control. However it should be considered whether security control has the same meaning and is on the same abstraction level as security mechanism. A discussion on this topic is included in section 5.4.2.2.
Indicate the moderating effect of security mechanism on risk	1	The metamodel can be changed to incorporate the moderating effect of a security mechanism on risk, because the main purpose of a security mechanism is to mitigate the risk. ArchiMate currently contains an influence relationship, but it is confined to ArchiMate's motivation extension. One could (mis)use this type of relationship, but currently it is not supported by the ArchiMate core.
Suggestion: include roadmaps as an ingredient for the approach	1	Roadmaps could be included to the approach, once specific cases arise. This would require input from experts on frequently encountered cases.

Implications for the proposed approach

The remarks made by the experts have some implications for the proposed approach. In general it can be stated that the approach is evaluated as usable and useful. All experts state that an integrated approach to security and enterprise architecture would significantly improve both disciplines.

The individual elements are considered useful and well integrated. Some adjustments might improve the usability and the usefulness of the approach, these comments are summarized in Table 11.

The integration of the elements making up the total approach is well described and solves the problem that triggered the research. To conclude, the approach is determined to be useful as well as usable, considering the limitations mentioned in this section.

8 CONCLUSION

This chapter describes the conclusions of this research as drawn from the literature review, the development of the secure enterprise architecture approach, its demonstration and its validation. The main research questions are answered in this chapter. The research started with two main research questions:

1. **“What is a validated, comprehensive and integrated approach for designing secure enterprise architecture?”**
2. **“How can an enterprise architecture language be extended to incorporate security aspects?”**

In order to answer these main research questions, several sub questions have been formulated, as stated in section 3.3. This chapter also discusses the contributions of this research to both theory and practice in section 8.2. Furthermore, the limitations and suggestions for further research are outlined in section 8.3.

8.1 RESEARCH QUESTIONS

The first main research question has been subdivided in five sub questions.

- 1.1. *What is the current state of the Enterprise Architecture and Security discipline and their relation?*

By examining the literature in the enterprise architecture and security discipline and their interfaces, it has become apparent that an integrated approach for enterprise architecture and security was not yet available. Several attempts have been made to integrate frameworks, but an integrated method and modelling language in order to achieve the envisioned benefits were lacking.

- 1.2. *Which elements are needed to provide a comprehensive approach, and what are their requirements?*

A design theory is comprised of three elements. These are the elements needed to provide a comprehensive approach:

- A *framework* for the subdivision of an architecture in different domains, sometimes including the relationships between these domains.
- A *method*, or a way of working, which is in most cases a step-wise prescriptive method for developing architectural descriptions.
- A *language*, defining the concepts for describing an architecture, including a (preferably graphical) *representation* of these concepts.

- 1.3. *What does an integration of these elements look like?*

An integrated framework, method, and modelling language are specified as the main ingredients for an approach to design secure enterprise architectures. In order to accomplish the formulation of this approach, the standards Zachman, SABSA, TOGAF ADM, and ArchiMate are used and integrated, and a security extension to the ArchiMate modelling language is defined.

1.4. How can the proposed approach be demonstrated in a real-life situation?

The proposed approach is demonstrated with the ArchiSurance case. Although it has a limited scope in the sense that it only encompasses one process of the organisation, it demonstrates the usefulness and usability of the approach to combine work efforts of the enterprise architecture and security discipline.

1.5. How can the proposed approach be validated?

The approach is validated to demonstrate its research rigor and research relevance. The research rigor is demonstrated according to the principles of Whetten (1989) by answering the questions What, How, Why and defining the limitations of the approach. The research relevance is discussed with several experts in the enterprise architecture and security disciplines, who were interviewed in expert opinion sessions.






The remarks made by the experts have some implications for the proposed approach. The element 'framework' is completed by a combination of the Zachman and SABSA framework. Not all organisations use these frameworks, and some prefer others such as the Novius model. The approach does not outline how other frameworks relate to the proposed method and modelling language, so this limits the usability of the approach.

2.1. What elements are needed to specify a modelling language?

Elements needed to specify a modelling language include a definition of the concepts and a description of how the concepts relate to each other and to existing concepts.

2.2. Which security concepts need to be merged into the enterprise architecture language?

The security concepts which needed to be merged into the enterprise architecture language include vulnerability, threat, risk, security mechanism, and security policy. The notation and definition of these concepts is presented below.

 Vulnerability	Flaw or weakness that could be exploited residing in system security procedures, design, implementation or internal controls.
 Threat	Potential for an actor with a certain motivation to exploit a vulnerability.
 Risk	Net mission impact considering (1) the probability that a particular threat-source will exploit a particular vulnerability and (2) the resulting impact.
 Security mechanism	A method, tool, or procedure for enforcing a security policy, designed to detect, prevent or recover from a security attack.
 Security policy	A statement of what is, and what is not allowed.

2.3. How can the language be validated?

The language and the metamodel are validated by interviews with experts from both the enterprise architecture and the security discipline.

One of the remarks mentioned by the experts was the absence of the concept 'residual risk'. This was mentioned by two interviewed experts. This absence can be resolved by differentiating the colour of the 'risk'-concept for residual risk specifically.

8.2 CONTRIBUTIONS

This research has both theoretical and practical relevance. Therefore the contribution of this research is divided into two parts. First the contribution to theory is elaborated, followed by the contributions to practice.

8.2.1 CONTRIBUTIONS TO THEORY

The first main contribution to theory is the comparison and integration of frameworks, methods, and modelling languages in the enterprise architecture and security discipline.

- This thesis describes how the Zachman and SABSA framework are related. Although the two frameworks have an identical structure, it is valuable to use these frameworks side by side, since they complement each other regarding their content.
- Furthermore, an integration of the TOGAF Architecture Development Method and the SABSA Lifecycle is provided. The SABSA Lifecycle enriches the ADM with relevant security aspects per development phase.
- The constructs for modelling security in enterprise architecture have been identified. The concepts have been selected from the SABSA framework, and are included in an ArchiMate extension.
- The relation between the Zachman and SABSA framework on the one hand, and the TOGAF ADM on the other hand is described.
- Also, the relation between the Zachman and SABSA framework on the one hand, and the ArchiMate modelling language on the other hand has been described.

The second main contribution is the proposal of the ArchiMate security extension. The constructs are derived from the SABSA framework, and the extension is validated among several experts from both the enterprise architecture and the security discipline.

8.2.2 CONTRIBUTIONS TO PRACTICE

The main contribution to practice is the developed Secure Enterprise Architecture approach. The approach consists of a combined framework, an integrated method, and an extended language.

The combined *framework* provides a collection of viewpoints as defined in the Zachman and SABSA framework. Moreover, it is described how the frameworks relate to each other while taking into account the advantages and disadvantages of the frameworks.

The integrated *method* provides a step-wise approach to designing secure enterprise architectures, prescribing the steps to be taken to deliver a secure enterprise architecture.

The extended *modelling language* provides a means to include security aspects in architecture specifications. The relevant concepts are selected and it is indicated how these concepts relate to the ArchiMate language.

The approach provides the security discipline with a means to specify the security aspects in relation to the enterprise, which might improve the probability that security requirements will be addressed throughout the organisation. Furthermore, it provides the EA discipline with an approach to embed security in the enterprise architecture, making it a more holistic discipline in order to specify its blueprints more accurately regarding the security aspect.

Implications for Deloitte

This research is conducted in cooperation with Deloitte Consulting, and in particular with the service line ‘Enterprise Architecture’. Currently, security is seen as a stakeholder in the enterprise architecture projects, however an integrated approach to secure enterprise architecture development was lacking. Another part of the Deloitte organisation is Risk Services, and more specifically the service line ‘Security & Privacy’. Experts from this service line acknowledged the lack of a common approach for the two disciplines. In this manner, the research has brought together the two service lines Enterprise Architecture and Security & Privacy in order to create new business with this integrated approach to designing secure enterprise architectures.

The approach to designing secure enterprise architectures supports Deloitte during three phases of a project: at the proposal, planning, and execution phase. Addressing security explicitly can be a key difference compared to competitors during the *proposal phase* of an enterprise architecture project. Especially in sectors where security issues are a main concern of the business, this can be a crucial differentiating factor. The approach also enables a focus at the most important aspects of the architecture in the *planning* phase. By having security integrated in the enterprise architecture, critical points in the architecture can be detected. Furthermore, the approach provides guidance in the *execution* phase of the project. This is accomplished by offering a framework to determine the viewpoints, a method to designing a secure enterprise architecture, and the modelling language to model the integrated views.

8.3 LIMITATIONS AND SUGGESTIONS FOR FURTHER RESEARCH

Although the validation contains generally positive feedback, the proposed approach has several limitations. The limitations and directions for future work are discussed below.

The approach only contains a limited set of standards. It uses the Zachman framework as the standard for EA framework, but it does not include directions to use another framework, for example the Novius framework or the TOGAF framework. Although it is likely that the proposed approach for designing secure enterprise architectures is compatible with other standards too, this has not been demonstrated nor validated. However, as stated previously, the Zachman framework is well-known and in use with many organisations. This is the reason for using Zachman in the first place, other frameworks can be added when the approach is in use.

The demonstration is done by a case with a relatively limited scope, and has not been applied to the full context of an organisation. Nevertheless, it has shown the application of the approach on the various levels of enterprise architecture. Furthermore it has demonstrated the role of frameworks, method and modelling language in the approach.

The proposed approach has not been applied to one or more real-life cases due to limitations in time. Future research is needed in order to validate the usability of the approach in a real-life setting. Furthermore, future research could quantitatively measure to what extent the intended benefits of the proposed approach are reached.

In order to be adopted by the industry, acceptance of the ArchiMate security extension is needed. Submission of a white paper at the Open Group could significantly increase the likelihood of adoption. Furthermore, tool support is needed to create the architecture views.

Future work is also required in order to make the modelling language suitable for cyber security analysis as in the work proposed by Ekstedt and Sommestad (2009).

9 BIBLIOGRAPHY

1. Anderson, J. M. (2003). Why we need a new definition of information security. *Computers & Security*, 22(4), 308-313.
2. Beel, J., & Gipp, B. (2009). *Google Scholar's ranking algorithm: The impact of citation counts (An empirical study)*. Paper presented at the Research Challenges in Information Science, 2009. RCIS 2009. Third International Conference on.
3. Bishop, M. (2004). Introduction to Computer Security.
4. Burkett, J. S. (2012). Business Security Architecture: Weaving Information Security into Your Organization's Enterprise Architecture through SABSA\textregistered. *Information Security Journal: A Global Perspective*, 21(1), 47--54.
5. Buschle, M., Holm, H., Sommestad, T., Ekstedt, M., & Shahzad, K. (2012). A Tool for automatic Enterprise Architecture modeling (pp. 1--15): Springer.
6. Buschle, M., Ullberg, J., Franke, U., Lagerström, R., & Sommestad, T. (2011). A tool for enterprise architecture analysis using the PRM formalism (pp. 108--121): Springer.
7. Ekstedt, M., & Sommestad, T. (2009). *Enterprise architecture models for cyber security analysis*. Paper presented at the Power Systems Conference and Exposition, 2009. PSCE'09. IEEE/PES.
8. Engelsman, W., Quartel, D., Jonkers, H., & van Sinderen, M. (2011). Extending enterprise architecture modelling with business goals and requirements. *Enterprise Information Systems*, 5(1), 9-36.
9. Google Scholar. (2013). About Google Scholar. Retrieved 25 november, 2013, from <http://www.google.com/intl/en/scholar/about.html>
10. Gregor, S. (2006). The nature of theory in information systems. *Mis Quarterly*, 30(3), 611-642.
11. Gregor, S., & Jones, D. (2007). The Anatomy of a Design Theory. *Journal of the Association for Information Systems*, 8(5).
12. Haigh, T. (1995). *Virtual enterprises and the enterprise security architecture*. Paper presented at the New Security Paradigms Workshop, 1995. Proceedings.
13. Heaney, J., Hybertson, D., Reedy, A., Chapin, S., Bollinger, T., Williams, D., & Kirwan Jr, M. (2002). *Information assurance for enterprise engineering*. Paper presented at the Proceedings of the 9th Conference on Patterns Language of Programming (PLoP'02).
14. Hensel, V., & Lemke-Rust, K. (2010). *On an Integration of an Information Security Management System into an Enterprise Architecture*. Paper presented at the Database and Expert Systems Applications (DEXA), 2010 Workshop on.
15. Iacob, M. E., Jonkers, D., Quartel, H., Franken, H., & Berg, H. (2012). *Delivering Enterprise Architecture with TOGAF® and ArchiMate®*: BIZZdesign.
16. Iacob, M. E., Quartel, D., & Jonkers, H. (2012). *Capturing business strategy and value in enterprise architecture to support portfolio Valuation*. Paper presented at the Enterprise Distributed Object Computing Conference (EDOC), 2012 IEEE 16th International.
17. Innerhofer-Oberperfler, F., & Breu, R. (2006). *Using an Enterprise Architecture for IT Risk Management*. Paper presented at the ISSA.
18. Johansson, E., & Johnson, P. (2005). Assessment of enterprise information security-an architecture theory diagram definition. *Proc. of CSER*, 5.
19. Jonkers, H., Band, I., & Quartel, D. (2012). The ArchiSurance Case Study. *The Open Group Case Study (Document Number Y121)(January 2012)*.
20. Jürjens, J. (2002). UMLsec: Extending UML for secure systems development <<UML>>2002—*The Unified Modeling Language* (pp. 412-425): Springer.

21. Kagal, L., Finin, T., & Joshi, A. (2003). *A policy language for a pervasive computing environment*. Paper presented at the Policies for Distributed Systems and Networks, 2003. Proceedings. POLICY 2003. IEEE 4th International Workshop on.
22. Kim, S. (2011). Auditing methodology on legal compliance of enterprise information systems. *International Journal of Technology Management*, 54(2), 270--287.
23. Kim, S., & Leem, C. S. (2004). An information engineering methodology for the security strategy planning (pp. 597--607): Springer.
24. Kreizman, G., & Robertson, B. (2006). *Integrating Security Into the Enterprise Architecture Framework*: Stamford CT: Gartner Inc.(G00137069).
25. Lang, U., & Schreiner, R. (2008). *Model driven security management: Making security management manageable in complex distributed systems*. Paper presented at the Modeling Security Workshop in Association with MODELS.
26. Liu, S., Sullivan, J., & Ormaner, J. (2001). A practical approach to enterprise IT security. *IT Professional*, 3(5), 35--42.
27. M.E. Iacob, H. J., M.M. Lankhorst, H.A. Proper, D.A.C. Quartel. (2012). *ArchiMate 2.0 Specification*: The Open Group.
28. Massacci, F., Mylopoulos, J., & Zannone, N. (2010). Security requirements engineering: the SI* modeling language and the secure tropos methodology *Advances in Intelligent Information Systems* (pp. 147-174): Springer.
29. Montelibano, J., & Moore, A. (2012). *Insider threat security reference architecture*. Paper presented at the System Science (HICSS), 2012 45th Hawaii International Conference on.
30. NIST. (2012). *Guide for Conducting Risk Assessments*.
31. Oda, S. M., Fu, H., & Zhu, Y. (2009). *Enterprise information security architecture a review of frameworks, methodology, and case studies*. Paper presented at the Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on.
32. Park, S., Ahmad, A., & Ruighaver, A. B. (2010). *Factors Influencing the Implementation of Information Systems Security Strategies in Organizations*. Paper presented at the Information Science and Applications (ICISA), 2010 International Conference on.
33. Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems*, 24(3), 45-77.
34. Peterson, G. (2007). *Security architecture blueprint*. Arctec Group LLC.
35. Ponemon Research Institute. (2013). 2013 Cost of Data Breach Study: Germany. Retrieved October 14, 2013, from http://www.symantec.com/content/en/us/about/media/pdfs/b-cost-of-a-data-breach-germany-report-2013.en-us.pdf?om_ext_cid=biz_socmed_twitter_facebook_marketwire_linkedin_2013Jun_worldwide_CostofaDataBreach
36. Pulkkinen, M., Naumenko, A., & Luostarinen, K. (2007). Managing information security in a business network of machinery maintenance services business--enterprise architecture as a coordination tool. *Journal of Systems and Software*, 80(10), 1607--1620.
37. Sahinoglu, M. (2005). Security meter: A practical decision-tree model to quantify risk. *Security & Privacy, IEEE*, 3(3), 18--24.
38. Scholtz, T. (2006). Structure and Content of an Enterprise Information Security Architecture. *Gartner Research*, January, 23.
39. Shariati, M., Bahmani, F., & Shams, F. (2011). Enterprise information security, a review of architectures and frameworks from interoperability perspective. *Procedia Computer Science*, 3, 537--543.

40. Sherwood, J., Clark, A., & Lynas, D. (2005). *Enterprise security architecture: a business-driven approach*: Backbeat Books.
41. Sherwood, J., Clark, A., & Lynas, D. (2009). Enterprise Security Architecture White Paper. *SABSA Limited*.
42. Shin, M. E., & Gomaa, H. (2007). Software requirements and architecture modeling for evolving non-secure applications into secure applications. *Science of Computer Programming*, 66(1), 60--70.
43. The Open Group. (2007). ADM and the Zachman Framework *TOGAF Version 8.1.1*.
44. The Open Group. (2011a). *TOGAF and SABSA Integration (White Paper)*.
45. The Open Group. (2011b). *TOGAF Version 9.1*.
46. The Open Group. (2013). *ArchiMate 2.1 Specification*: Van Haren.
47. van Gansewinkel, R., & Hofman, A. (2012). Hebt u ze op een rijtje? (Dutch). *Informatiebeveiliging*(4), 36 - 40.
48. Webster, J., & Watson, R. T. (2002). ANALYZING THE PAST TO PREPARE FOR THE FUTURE: WRITING A. *Mis Quarterly*, 26(2).
49. Whetten, D. A. (1989). What constitutes a theoretical contribution? *Academy of management review*, 14(4), 490-495.
50. Wieringa, R. J. (2012-2013). Chapter 8. Design Validation *Design Science Methodology [Lecture notes]*.
51. Zachman, J. A. (1987). A framework for information systems architecture. *IBM systems journal*, 26(3), 276-292.
52. Zachman, J. A. (2008). John Zachman's Concise Definition of The Zachman Framework™. *Zachman International, USA*.

LIST OF FIGURES

Figure 1: Four common layers in Enterprise Architecture	2
Figure 2: The average per capita cost of data breach over five years in Germany (Ponemon Research Institute, 2013)	5
Figure 3: Design Science Research Methodology (DSRM) process model (Peffer et al., 2007)	7
Figure 4: Literature review search process.....	9
Figure 5: SABSA Matrix (John Sherwood et al., 2005)	13
Figure 6: The defence tree concept (Ekstedt & Sommestad, 2009).....	15
Figure 7: A general-purpose decision-tree diagram example for the Security Meter model (Sahinoglu, 2005)	16
Figure 8: Syntactic elements of extended influence diagrams and a simple example (Ekstedt & Sommestad, 2009)	16
Figure 9: Identification & Authentication Pattern Tree (Heaney et al., 2002)	19
Figure 10: Enterprise Security Architecture design process, adapted from Liu et al. (2001)..	21
Figure 11: The comparison of prominent EISA frameworks from interoperability perspective (Shariati et al., 2011).....	23
Figure 12: Ingredients of an Enterprise Architecture approach (Iacob, Jonkers, et al., 2012)	25

Figure 13: Essential ingredients of an integrated approach for EA and Security	26
Figure 14: The Zachman framework for Enterprise Architecture (version 2003)	27
Figure 15: SABSA Matrix (J Sherwood et al., 2009)	28
Figure 16: TOGAF Architecture Development Method	30
Figure 17: Security implementation approach (Liu et al., 2001)	33
Figure 18: The SABSA Lifecycle	34
Figure 19: Relation of TOGAF ADM Preliminary phase to Zachman and SABSA.....	34
Figure 20: Relation of TOGAF ADM Phase A to Zachman and SABSA	35
Figure 21: Relation of TOGAF ADM Phase B to Zachman and SABSA	36
Figure 22: Relation of TOGAF ADM Phase C to Zachman and SABSA	37
Figure 23: Relation of TOGAF ADM Phase D to Zachman and SABSA.....	38
Figure 24: Overview of mapping TOGAF ADM to Zachman and SABSA	38
Figure 25: Relation between SABSA lifecycle and TOGAF ADM	39
Figure 26: ArchiMate Architectural Framework (The Open Group, 2013).....	41
Figure 27: Correspondence between ArchiMate and TOGAF ADM (The Open Group, 2013)	41
Figure 28: Metamodels at Different Levels of Specificity (The Open Group, 2013).....	42
Figure 29: SABSA concepts and its relation to ArchiMate.....	43
Figure 30: Attack and defence tree on stealing a car	45
Figure 31: Vulnerability notation	46
Figure 32: Example of vulnerability	46
Figure 33: Threat notation	47
Figure 34: Example of threat	47
Figure 35: Risk notation	48
Figure 36: Example of risk.....	48
Figure 37: Security mechanism notation	48
Figure 38: Example of security mechanism	49
Figure 39: Security policy notation	49
Figure 40: Example of security policy	50
Figure 41: Archimate security extension metamodel	50
Figure 42: Classification of Enterprise Architecture Viewpoints (The Open Group, 2013).....	52
Figure 43: Relation of risk analysis and risk mitigation viewpoint to Zachman and SABSA framework.....	53
Figure 44: Fragment of Stakeholder view for ArchiSurance.....	56
Figure 45: Zachman and SABSA view on the outlined business goals and drivers.....	57

Figure 46: Business goals associated with the driver Secure handling of information	57
Figure 47: Risk analysis matrix	57
Figure 48: Zachman and SABSA view on the outlined business architecture	59
Figure 49: ArchiSurance’s baseline business architecture	59
Figure 50: SABSA view on the outlined risk analysis	60
Figure 51: Risk analysis on Archisurance's business architecture	60
Figure 52: Archisurance's target business architecture	61
Figure 53: Zachman and SABSA view on the outlined information systems architecture	62
Figure 54: Archisurance’s baseline information services architecture	62
Figure 55: Risk analysis on Archisurance's information services architecture	63
Figure 56: Archisurance's target information systems architecture	64
Figure 57: Zachman and SABSA view on the outlined technology architecture	65
Figure 58: Technology Architecture.....	65
Figure 59: Risk analysis on Archisurance's technology architecture	66
Figure 60: Archisurance's target technology architecture	66

LIST OF TABLES

Table 1: Relation between process, research questions and thesis outline	8
Table 2: Search queries used in literature review	10
Table 3: Activities in six-step process to develop Enterprise Security Architecture, as outlined in Figure 10. (Liu et al., 2001)	22
Table 4: Relation of TOGAF ADM Preliminary phase to Zachman and SABSA	34
Table 5: Relation of TOGAF ADM Phase A to Zachman and SABSA.....	35
Table 6: Relation of TOGAF ADM Phase B to Zachman and SABSA.....	35
Table 7: Relation of TOGAF ADM Phase C to Zachman and SABSA.....	36
Table 8: Relation of TOGAF ADM Phase D to Zachman and SABSA	37
Table 9: Concepts in SABSA, not covered in ArchiMate and the motivation for in-/exclusion	44
Table 10: Validation expert interview list.....	70
Table 11: Validation conclusions	73
Table 12: ArchiMate concepts related to security	87

APPENDIX A – LITERATURE REVIEW SHORT LIST

1. Burkett, J. S. (2012). Business Security Architecture: Weaving Information Security into Your Organisation's Enterprise Architecture through SABSA. *Information Security Journal: A Global Perspective*, 21(1), 47--54.
2. Buschle, M., Holm, H., Sommestad, T., Ekstedt, M., & Shahzad, K. (2012). A Tool for automatic Enterprise Architecture modeling (pp. 1--15): Springer.
3. Buschle, M., Ullberg, J., Franke, U., Lagerström, R., & Sommestad, T. (2011). A tool for enterprise architecture analysis using the PRM formalism (pp. 108--121): Springer.
4. Dai, L., & Cooper, K. (2007). Using FDAF to bridge the gap between enterprise and software architectures for security. *Science of Computer Programming*, 66(1), 87--102.
5. Ekstedt, M., & Sommestad, T. (2009). Enterprise architecture models for cyber security analysis. Paper presented at the Power Systems Conference and Exposition, 2009. PSCE'09. IEEE/PES.
6. Haigh, T. (1995). Virtual enterprises and the enterprise security architecture. Paper presented at the New Security Paradigms Workshop, 1995. Proceedings.
7. Heaney, J., Hybertson, D., Reedy, A., Chapin, S., Bollinger, T., Williams, D., & Kirwan Jr, M. (2002). Information assurance for enterprise engineering. Paper presented at the Proceedings of the 9th Conference on Patterns Language of Programming (PLoP'02).
8. Hensel, V., & Lemke-Rust, K. (2010). On an Integration of an Information Security Management System into an Enterprise Architecture. Paper presented at the Database and Expert Systems Applications (DEXA), 2010 Workshop on.
9. Innerhofer-Oberperfler, F., & Breu, R. (2006). Using an Enterprise Architecture for IT Risk Management. Paper presented at the ISSA.
10. Johansson, E., & Johnson, P. (2005). Assessment of enterprise information security-architecture theory diagram definition. *Proc. of CSER*, 5.
11. Johnson, P., Johansson, E., Sommestad, T., & Ullberg, J. (2007). A tool for enterprise architecture analysis. Paper presented at the Enterprise Distributed Object Computing Conference, 2007. EDOC 2007. 11th IEEE International.
12. Kim, S. (2011). Auditing methodology on legal compliance of enterprise information systems. *International Journal of Technology Management*, 54(2), 270--287.
13. Kim, S., & Leem, C. S. (2004). An information engineering methodology for the security strategy planning (pp. 597--607): Springer.
14. Lang, U., & Schreiner, R. (2008). Model driven security management: Making security management manageable in complex distributed systems. Paper presented at the Modeling Security Workshop in Association with MODELS.
15. Liu, S., Sullivan, J., & Ormaner, J. (2001). A practical approach to enterprise IT security. *IT Professional*, 3(5), 35--42.
16. Montelibano, J., & Moore, A. (2012). Insider threat security reference architecture. Paper presented at the System Science (HICSS), 2012 45th Hawaii International Conference on.
17. Moral-García, S., Moral-Rubio, S., Fernández, E. B., & Fernández-Medina, E. (2012). A new enterprise security pattern: Secure Software as a Service (SaaS).
18. Oda, S. M., Fu, H., & Zhu, Y. (2009). Enterprise information security architecture a review of frameworks, methodology, and case studies. Paper presented at the

- Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on.
19. Park, S., Ahmad, A., & Ruighaver, A. B. (2010). Factors Influencing the Implementation of Information Systems Security Strategies in Organisations. Paper presented at the Information Science and Applications (ICISA), 2010 International Conference on.
 20. Peterson, G. (2007). Security architecture blueprint. Arctec Group LLC.
 21. Pulkkinen, M., Naumenko, A., & Luostarinen, K. (2007). Managing information security in a business network of machinery maintenance services business--enterprise architecture as a coordination tool. *Journal of Systems and Software*, 80(10), 1607--1620.
 22. Sahinoglu, M. (2005). Security meter: A practical decision-tree model to quantify risk. *Security & Privacy, IEEE*, 3(3), 18--24.
 23. Shariati, M., Bahmani, F., & Shams, F. (2011). Enterprise information security, a review of architectures and frameworks from interoperability perspective. *Procedia Computer Science*, 3, 537--543.
 24. Sherwood, J., Clark, A., & Lynas, D. (2005). Enterprise security architecture: a business-driven approach: Backbeat Books.
 25. Sherwood, J., Clark, A., & Lynas, D. (2009). Enterprise Security Architecture White Paper. SABSA Limited.
 26. Shin, M. E., & Gomaa, H. (2007). Software requirements and architecture modeling for evolving non-secure applications into secure applications. *Science of Computer Programming*, 66(1), 60--70.
 27. Sommestad, T., Ekstedt, M., & Johnson, P. (2008). Combining defense graphs and enterprise architecture models for security analysis. Paper presented at the Enterprise Distributed Object Computing Conference, 2008. EDOC'08. 12th International IEEE.

APPENDIX B – SECURITY RELATED CONCEPTS IN ARCHIMATE

An overview of the security related concepts in ArchiMate and their description is provided in Table 12. A full description and representation of the constructs can be found in The Open Group (2013).

Table 12: ArchiMate concepts related to security

Concept	Description	Security aspect
Business actor	An organisational entity that is capable of performing behavior.	Potentially vulnerable object
Business interface	A point of access where a business service is made available to the environment.	Potentially vulnerable object
Location	A conceptual point or extent in space.	Potentially vulnerable object
Business process	A behavior element that groups behavior based on an ordering of activities. It is intended to produce a defined set of products or business services.	Property: CIA classification
Business function	A behavior element that groups behavior based on a chosen set of criteria (typically required business resources and/or competences).	Property: CIA classification
Business event	Something that happens (internally or externally) and influences behavior.	Security event
Business service	A service that fulfills a business need for a customer (internal or external to the organisation).	CIA classification
Business object	A passive element that has relevance from a business perspective.	Potentially vulnerable object
Value	The relative worth, utility, or importance of a business service or product.	Potential loss
Product	A coherent collection of services, accompanied by a contract/set of agreements, which is offered as a whole to (internal or external) customers.	Potentially vulnerable object
Contract	A formal or informal specification of agreement that specifies the rights and obligations associated with a product.	CIA classification Potentially vulnerable object

Concept	Description	Security aspect
Application component	A modular, deployable, and replaceable part of a software system that encapsulates its behavior and data and exposes these through a set of interfaces.	Potentially vulnerable object
Application interface	A point of access where an application service is made available to a user or another application component.	Potentially vulnerable object
Application function	A behavior element that groups automated behavior that can be performed by an application component.	Potentially vulnerable object
Application service	A service that exposes automated behavior.	Potentially vulnerable object
Data object	A passive element suitable for automated processing.	Potentially vulnerable object CIA classification
Node	A computational resource upon which artifacts may be stored or deployed for execution.	Potentially vulnerable object
Device	A hardware resource upon which artifacts may be stored or deployed for execution.	Potentially vulnerable object
Network	A communication medium between two or more devices.	CIA classification
Communication path	A link between two or more nodes, through which these nodes can exchange data.	CIA classification
Infrastructure interface	A point of access where infrastructure services offered by a node can be accessed by other nodes and application components.	Potentially vulnerable object
System software	A software environment for specific types of components and objects that are deployed on it in the form of artifacts.	Potentially vulnerable object CIA classification
Infrastructure function	A behavior element that groups infrastructural behavior that can be performed by a node.	Potentially vulnerable object
Infrastructure service	An externally visible unit of functionality, provided by one or more nodes, exposed through well-defined interfaces, and meaningful to the environment.	Potentially vulnerable object

Concept	Description	Security aspect
Artifact	A physical piece of data that is used or produced in a software development process, or by deployment and operation of a system.	Potentially vulnerable object CIA classification
Driver	Something that creates, motivates, and fuels the change in an organisation.	CIA classification
Assessment	The outcome of some analysis of some driver.	Vulnerability / threat
Requirement	A statement of need that must be realized by a system.	Security requirements

APPENDIX C – INTERVIEW QUESTIONS

1. Where do you think this approach is useful?
2. Do you consider framework, process and language as the right ingredients, what is missing?
3. Framework
 - a. Do you consider the frameworks to be relevant and correctly used?
 - b. Is it complete, what is missing?
4. Process
 - a. Is the approach (and integration) relevant and to what extent do you recognise this within your context?
 - b. Is it complete, what is missing?
5. Language
 - a. What do you think of the language extension?
 - b. Is an integration with the ArchiMate language useful?
 - c. Do you miss any concepts from a security perspective?
 - d. Is it used at the right level of abstraction?
6. All together
 - a. (How) Would you use this approach in practice?
 - b. Does it solve the problem as outlined in the research trigger?