Visualization and Support of Information Security Architectures on Atlas

Gonçalo Nuno Marques Ramos

Thesis to obtain the Master of Science Degree in

Telecommunications and Informatics Engineering

Supervisor: Prof. Pedro Manuel Moreira Vaz Antunes de Sousa

Examination Committee

Chairperson: Prof. Ricardo Jorge Fernandes Chaves
Supervisor: Prof. Pedro Manuel Moreira Vaz Antunes de Sousa
Member of the Committee: Prof. André Ferreira Ferrão Couto e Vasconcelos

January 2021
Dedicated to friends and family.
Acknowledgments

I would like to thank Professor Pedro Manuel Moreira Vaz Antunes de Sousa for all the knowledge shared and assistance provided in this master thesis, as well as a special thanks to all of Linkconsulting’s personnel for taking me in during this journey.
Resumo

A protecção de informação organizacional (seja esta física ou lógica) contra o acesso ou dano não desejado, podendo este ser malicioso ou acidental, é vital para a saúde de uma organização. Como tal, não abordar este contexto pode em muitas situações dar origem a consequências agravadas a longo prazo, sendo a organização afectada nas suas componentes operacionais, legais, financeiras ou sofrer danos causados à sua reputação. Este documento visa o suporte realizado sob a solução do Atlas de forma a descrever e visualizar Arquitecturas de Segurança Empresarial. Para este efeito conceitos chave aplicados ao contexto de segurança são identificados, sendo posteriormente feita uma análise identificando como estes podem ser descritos através de uma prática arquitectural. O suporte realizado sobre a solução do Atlas é efectuado através do desenvolvimento de uma visualização matricial de carácter genérico, sendo que assim é assegurada a sua reutilização independentemente do seu modelo de dados, de forma a facilitar a descrição tanto de análises de risco como dos seus controlos de segurança. A matriz é complementada por um conjunto de funcionalidade com o intuito de suportar uma prática arquitectural através de múltiplas configurações aos seus componentes de forma a gerar ou modificar a sua representação, assim como o impacto do ciclo de vida dos seus elementos sob esta representação. O trabalho realizado é então validado na apresentação de como esta visualização pode facilmente descrever conceitos identificados num contexto de segurança.

Abstract

The protection of enterprise information (being either physical or logical) against unauthorized access/damage (malicious or accidental) is vital for an organization’s health. As such, failure to address this issue often resolves on aggravated long-term consequences such as operationally, legally, financially or damage to the organization’s reputation. This paper addresses the support over the Atlas solution to describe and visualize Enterprise Security Architectures. As such, key concepts related to its security context are identified and how their representation can be performed on an architectural practice is evaluated. The support over the platform is mainly accomplished with the development of a generic matrix view, thus ensuring its reuse regardless of the data model, to facilitate the description of its risk analysis and security controls. The matrix is integrated with a set of features able to support an architectural practice, multiple configurations over its components as well as the impact of the lifecycle over its visualization. The work presented is then evaluated with the presentation of its visualization over different concepts related to the security context.

Keywords: Enterprise Architecture, Enterprise Security Architecture, Information Security, Visualization.
Contents

Acknowledgments ............................................................................................................. v
Resumo ........................................................................................................................... vii
Abstract ........................................................................................................................ ix
List of Tables .................................................................................................................. xiii
List of Figures .................................................................................................................. xv
Nomenclature .................................................................................................................. 1
Glossary ........................................................................................................................... 1

1 Introduction ................................................................................................................. 1

2 Background ................................................................................................................. 3
  2.1 Enterprise Architecture .......................................................................................... 3
  2.2 Enterprise Security Architecture .......................................................................... 4
  2.3 Modeling Languages and Frameworks ................................................................... 4
  2.4 TOGAF-9 ............................................................................................................... 6
    2.4.1 TOGAF-9 in Security Architecture ................................................................ 7
    2.4.2 Security Architecture and ADM ..................................................................... 10
  2.5 Lifecycle .................................................................................................................. 18
  2.6 Overview of Existing Frameworks in Enterprise Architecture ............................ 18
    2.6.1 Zachman Framework ...................................................................................... 18
    2.6.2 Department of Defense Architecture Framework (DoDAF) ......................... 19
    2.6.3 Federal Enterprise Architecture Framework (FEAF) .................................. 19
    2.6.4 Treasury Enterprise Architect Framework (TEAF) .................................... 19
    2.6.5 Comparison of Enterprise Architecture Frameworks ................................. 19
  2.7 Enterprise Security Architecture Tools ................................................................... 21
  2.8 Matrices in EA Tools ............................................................................................. 24
  2.9 Background Conclusion ......................................................................................... 25

3 Implementation ............................................................................................................ 27
  3.1 Description of the Solution and Motivation .......................................................... 27
  3.2 Methodology Applied for the Development .......................................................... 28
  3.3 Technologies .......................................................................................................... 28
# List of Tables

2.1 Seven Principles of Information Security [5]. . . . . . . . . . . . . . . . . . . . . . . . . . . . 5  
2.2 Recommendations applicable to the Preliminary Phase [5]. . . . . . . . . . . . . . . . . . 11  
2.3 Recommendations applicable to the Phase A [5]. . . . . . . . . . . . . . . . . . . . . . . . 12  
2.4 Recommendations applicable to the Phase B [5]. . . . . . . . . . . . . . . . . . . . . . . . 13  
2.5 Recommendations applicable to the Phase C [5]. . . . . . . . . . . . . . . . . . . . . . . . 14  
2.6 Recommendations applicable to the Phase D [5]. . . . . . . . . . . . . . . . . . . . . . . . 15  
2.7 Recommendations applicable to the Phase E [5]. . . . . . . . . . . . . . . . . . . . . . . . 16  
2.8 Recommendations applicable to the Phase F [5]. . . . . . . . . . . . . . . . . . . . . . . . 16  
2.9 Recommendations applicable to the Phase G. [5] . . . . . . . . . . . . . . . . . . . . . . . 17  
2.10 Recommendations applicable to the Phase H [5]. . . . . . . . . . . . . . . . . . . . . . . 17  
2.11 Comparison by Views/Perspectives [10]. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20  
2.12 Comparison by Abstractions [10]. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20  
2.13 Comparison by SDLC Phases [10]. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
List of Figures

2.1 Enterprise Architecture Components [5] ................................. 3
2.2 Overview of Risk Analysis [1] ........................................... 4
2.3 ArchiMate Risk and Security ‘Overlay’ Example [1] .................. 5
2.4 Structure of the TOGAF Standard [4] .................................. 6
2.5 Examples of Visualization Types Supported by EA tools [12] ........ 23
2.6 Information Model Binding to Create a Visualization [12] ............ 23
2.7 Examples of Available Configurations Supported by EA Tools [12] .. 24
2.8 Customization of Visual Parameters [12] ................................. 24

3.1 Matrix Layout ................................................................. 36
3.2 Direct Relationship Example ............................................. 39
3.3 Indirect Relationship Example .......................................... 40
3.4 Representation of the Lifecycle on the Visualization .................... 42

4.1 Authentication System with the Matrix Visualization .................. 50
4.2 Access Control Visualization ............................................ 50
4.3 Authorizations for User 1 at File G ...................................... 51
Chapter 1

Introduction

Over the years, with the growth of information related software, information systems have taken a major role within corporations due to its inapt ability to create competitive advantages in the market. As a direct function of this, information security is now being viewed as one of the most compelling concerns within an organization. This notion of information security relates to the rightful management and protection of both information and information systems alike, and is focused on the organization's processes, systems, services and technology.

In order to successfully manage the complexity of a large organization and/or systems, an architecture is needed [1]. This architecture is defined as an enterprise architecture. The architectural concept, first introduced in 1987 by John Zachman, was in its early stages implemented to describe an information system architecture but was later extended to represent the whole enterprise. It encapsulates principals, methods and models, used in the architecture design with the sole purpose of representing its organizational structure, business processes, information systems, and infrastructures. As of now, the information era, more enterprises are taking this approach into practical use to manage change, aligning IT with business, reducing costs, decreasing complexity, improving information quality, and managing stakeholders’ concerns [2]. The architectural stakeholders can range from Business Analysts, CxOs (i.e., CEO, CFO, CIO), Enterprise Architects, Business and IT Managers and Solutions Architects.

This approach serves as means to identify and address enterprises’ main concerns. Therefore, frameworks and methods to support the design of an Enterprise Architecture must also consider all of the enterprise's security risks, and by doing so, giving birth to an Enterprise Security Architecture.

Enterprise security architectures are viewed as nothing more than a unified framework and reusable services with the purpose of implementing policies, standards, and risk management decisions. This is done by identifying and target major areas of concern related to the architecture's security, while presenting decision criteria and context for each domain. These frameworks, such as TOGAF-9, serve as guidelines, and are used to avoid faults while describing its major security issues [3][4].

This research focal point centers on the identification of requirements and risks related to a particular environment, and its security policies and how these security key concepts can better be described within an architecture through the development of a matrix visualization, and consequently integrated
on the Atlas solution. These security issues and how they can be represented is identified with the evolution of Enterprise Security Architectures present in the state of art (Chapter - 2.4.2), namely, by analysis performed over the TOGAF-9’s framework Architecture Development Method (ADM) and the visualizations available on multiple existing enterprise management tools.

This dissertation presents the development (Chapter - 3) as the development of a generic matrix visualization, to support the Altas solution to facilitate the description of security-related areas of concern under an architectural practice on a security context. Due to its generic characteristics, the visualization developed is independent of a specific data model. This development raises some questions regarding how this representation can be configured initially and how its information and visual models can be customized in accordance with the analysis of other management tools (Chapter - 3.7) to represent and modify relationships over multiple objects, and thus providing an efficient support in both decision-making IT security planning within organizations. Moreover, architectural view representing the lifecycle of its components (Chapter -2.5) also has an impact over the matrix architectural view. Therefore, the matrix visualization is not only responsible for the generation of its architectural view, but also empowered with run-time features granting data model modifications able to restructure the matrix configuration and provide model adaptability and how the visual model is constructed to better visualize all its objects, relations, and lifecycle impact.

The development performed must be evaluation as form of validation (Chapter - 4). To this extent multiple presentations were made until the final approval of members of the Atlas development team and thus concluding the development phase. Moreover, examples were provided as proof of concept in the support made over a security context and how its security issues can easily be described with its application.

The contribution to this master’s thesis consists of the development of matrix visualization integrated on Enterprise Architecture solution able represent relations composed by multiple objects and the impact of the lifecycle concept can be visually represented on this representation, granted a generic context.
Chapter 2

Background

2.1 Enterprise Architecture

In order to address an Enterprise Security Architecture, we must first understand the view and concepts behind an Information Architecture, an Enterprise Architecture.

Over the years, enterprises have become increasingly more data reliable on the information derivative from organizational inputs. As a direct result from this behavior, companies view as major priority to make improvements over their information activities in order to better position themselves within its respective market. The main practice capable of guiding organizations through business, information, process and technology changes rely on the development of an Enterprise Information Architecture to conduct enterprise analysis, design, planning and implementation. Meaning, Architecture is viewed as a set of elements in order to support all information processes within said enterprise and is mostly used as an instrument in order to manage a company’s daily operations and future development [1].

![Figure 2.1: Enterprise Architecture Components](image)

Ever since the shift towards dynamic enterprise strategy approach, the need to re-think an organization’s structure has increased to improve flexibility, scalability, and finally adaptability. Therefore, frameworks are provided to develop and maintain the organization’s Enterprise Architecture to achieve these goals.

Commonly, an Enterprise Architecture is viewed as the description or detailed plan of an organization, its components, their interrelationships, and the principles governing their design [5]. Following this
concept, any enterprise strategy focusing information security, should be considered in the context of the organization's Enterprise Architecture.

2.2 Enterprise Security Architecture

An organization's Enterprise Security Architecture is viewed as its security design, focusing on coping the requirements, and more importantly, the risks associated to a particular environment, specifying what controls should be applied and where. In order to successfully implement such design, a proven methodology is required in its application to ensure its completeness and reliability.

![Figure 2.2: Overview of Risk Analysis [1].](image)

The architectural process begins with the definition of the business context, able to provide balance between the business drivers and acceptable risks. This business context is established as the direct result of the decisions made after performing an analysis over internal and external organizational factors. In consequence, security policies can be viewed as guidelines. Hereafter, the resulting architecture provided from these policies relates to the combination of processes and technology. Furthermore, this architecture must fit the business context for the enterprise to provide security, and to provide legal and regulatory compliance [5].

Architects entrusted with conducting the Security Architecture work must be able to define the detailed technical requirements in an information security context, and design, document and maintain these architectures with the rightful security technology and process components, as well as validate that the solution conceived meets the Security requirements [6].

2.3 Modeling Languages and Frameworks

To provide more insights into different aspects that an enterprise architecture model may encompass, it is outlined a number of well-known architecture frameworks. Frameworks structure architecture description techniques by identifying and relating different architectural viewpoints and the modeling
Principles of Information Security

Integral to Enterprise Security

Impacts the entire Organization.

Risk Management defines information security requirements.

Security accountability should be defined and acknowledged.

Must consider internal and external stakeholders.

Requires understanding and commitment.

Requires continual improvement.

Table 2.1: Seven Principles of Information Security [5].

techniques associated with them. They do not provide the concepts for the actual modeling, although some of the frameworks are closely connected to a specific modeling language or set of languages. These frameworks, through their guidelines, can be accurate to establish what elements from the organization should be a part of the enterprise architecture.

Meanwhile, the correct use of a proven framework may not be sufficient in order to guarantee the quality of an enterprise architecture during its life cycle. To achieve this feat, all the relations between the different types of domains, views, or layers of the architecture must remain clear, and any change should be carried through methodically in all of them [1][2].

Up until recent years, something was missing. A uniform language to allow both the modeling and visualization of the enterprise architectures. That was until the modeling language known as ArchiMate arrived to fill in those needs. This language offers tools, very much needed, helping enterprise architects to describe, analyze and visualize relations among domains. It offers a common language, easy to understand, to represent organizations’ business processes and structures, the information flows in said processes, and the IT systems and technical infrastructures. Its purpose is to assist stakeholders making the best possible decision, by designing, assessing and better communicate its consequences and changes of these business domains [2].

Figure 2.3: ArchiMate Risk and Security 'Overlay' Example [1].
ArchiMate is now The Open Group's modeling language for enterprise architectures and along with the TOGAF's framework it enables architects with a powerful approach, supported by methods, modeling languages and tools [7].

2.4 TOGAF-9

The Open Group Architecture Framework solution, commonly referred as TOGAF-9, is an architecture framework, providing an already proven methodology and a set of supporting tools with the focus of developing enterprise architectures, improving business efficiency. This framework is regarded as a very reliable enterprise architecture standard, able to ensure the application of consistent standards, methods, and communication amidst enterprise architecture professionals with the sole purpose of helping specialists to avoid being locked into proprietary methods, apply resources in a more efficient and effective manner, and make a large return on investment [3][4].

![Figure 2.4: Structure of the TOGAF Standard](image)

Unlike older frameworks, such as Zachman's Framework, the TOGAF Framework is much more process-driven, presenting us with ways to codify architectural patterns by introducing a seven-part structure and re-organizing the framework models constructed by objectives. By doing so, it allows future models to evolve at different rates or speed and more importantly, limits the impact of those on the architecture's blueprint, allowing the architecture to define compartments that operate independently [3].

Part I - Introduction

Firstly, it is provided an introduction to key concepts within an enterprise architecture and more importantly, the TOGAF approach, defining the terms used in the framework along with release notes stating changes over its previous version.
Part II - Architecture Development Method

Commonly referred as ADM, it is the framework’s core. Describes the architecture development method through a step-by-step explanation in order to successfully develop an enterprise architecture.

Part III - ADM Guidelines and Techniques

This structure offers the guidelines and techniques available to apply TOGAF and its ADM.

Part IV – Architecture Content Framework

Describes the content framework to increase consistency in the outputs generated through the application of the Architecture Development Method and includes a structured meta-model for the architectural artifacts, the use of architecture building blocks and an overview of architectural deliverables.

Part V – Enterprise Continuum and Tools

This part focus on the taxonomies and tools to be used to categorize and store the outputs of architectural activity within an enterprise.

Part VI – TOGAF Reference Models

Architecture reference models are provided, including the TOGAF Foundation Architecture and the Integrated Information Infrastructure Reference Model.

Part VII – Architecture Capability Framework

It is discussed the organization, processes, skills, roles, and responsibilities required in order to successfully establish and operate an architecture function within an enterprise. This division into these independent parts, offer different areas of specialization that may be addressed in isolation.

2.4.1 TOGAF-9 in Security Architecture

The design concerning Security Architectures focus on addressing the requirements, and more importantly, the risks inherited in a specific scenario and determines which security controls should be applied where. As such, all of the stakeholders within an organization should have security concerns. The areas of concern related to the development of a Security Architecture are as follow [3]:

- Authentication: Covers a person (or entity) identity, related to a system.
- Authorization: Specifies the permissions relegated to a person or entity already authenticated.
- Audit: Provides a forensic data indication that the system was used in conformance with the already established security policies.
• Assurance: Tests the system and proves it has the security attributes required to uphold the security policies.

• Availability: The system must be able to function without interruption, even in the presence of abnormal or malicious events.

• Asset Protection: the information should be protected from loss and resources from unauthorized use.

• Administration: Adding or modifying security policies, how these are implemented in the system and the persons or entities related to said system.

The security policies are a product of considering what is at risk, and how said risk can be reduced, guided by corporate decisions in this regard based on the context of the business. These must be individually crafted and are in reality a set of layers of policies on top of procedures and practices with the purpose of providing the framework at use with the technical aspects of the Security Architecture. A standard and proven methodology should be used to identify the risks, mitigation, and developing the security policies [8].

At the top of these layers of security policies is the corporate security policy that focus on the organization as a whole, representing a general statement regarding the security goals. This should be both static and non-technical, goal-driven, and should not specify technologies. It serves as guidance for the organization, where the more dynamic and technical details are to be specified at lower policy levels [8].

When considering the implementation of a security policy, these are the categories to be considered along with their objectives [8]:

**Business Continuity Planning**

Has the objective to counteract interruptions in the business activities and critical business processes from the consequences of major failures.

**System Access Control**

The system access control focuses on:

• Controlling information access.

• Preventing information systems from unauthorized access.

• Ensuring the protection of network services.

• Preventing unauthorized computer access.

• Detecting unauthorized activities.
System Development and Maintenance

The objectives regarding the system development maintenance are:

• Ensuring that the operational systems have built-in security.

• Preventing loss, change, or misuse of user data.

• Protecting the confidentiality, authenticity, and integrity of the information.

• Ensuring that IT projects and support activities are conducted in a secure manner.

• Maintaining the Security of the system, both of software and data.

Physical and Environmental Security

This section has as objectives:

• Preventing unauthorized access, damage, and interference to business premises and information.

• Preventing loss, damage, or corruption of assets and interruption to the business activities.

• Preventing compromise or theft of information and information processing facilities.

Compliance

The objectives around compliance are as follow:

• Avoiding breaches of any criminal or civil law, statutory, regulatory, or contractual obligations and also security requirements.

• Ensuring compliance of the systems with security policies and standards.

• Maximizing the effectiveness of and to minimize interference to, and from the system audit process.

Personnel Security

The personnel security has its objectives defined as:

• Reducing the risks of human error, theft, fraud, and misuse of the facilities.

• Ensuring users are aware of the threats on information security and are equipped to support the corporate security policy in the course of their normal work.

• Minimizing the damage and learning from security incidents and malfunctions.
Security Organization

Security Organization objectives are:

• Managing information security within the company.
• Maintaining the security of organizational information processing facilities and information assets accessed by third parties.
• Maintaining the security of information when the responsibility for information processing has been outsourced to another organization.

Computer and Network Management

The computer and network management have the following objectives:

• Ensuring the correct and secure operation of information processing facilities.
• Minimizing system failure risks.
• Protecting the integrity of both the software and the information.
• Maintaining the integrity and availability of information processing and communications.
• Ensuring the safeguarding of information in networks and the protection of the supporting infrastructures.
• Preventing damage to assets and interruptions to business activities.
• Preventing loss, modification, or misuse of information exchanged between organizations.

Asset Classification and Control

The asset classification and control focus on maintaining the appropriate protection of corporate assets and to ensure that information assets receive an appropriate level of protection.

Security Policy

Finally, it provides a management direction and support for information security.

2.4.2 Security Architecture and ADM

As it was already stated in this paper, the architecture development method is TOGAF’s framework core. It offers a set of steps, that can also be interpreted as guidelines, to successfully build an enterprise architecture through a proven methodology. These guidelines are comprised in eight different stages.

Firstly, security policies and security standards must be previously established, as they are a very important part of an enterprise requirement management process. As stated in the previous chapter,
security policies are at the highest level when regarding the existing corporate level, as they represent a
general view on the security goals, are goal-driven, and should not specify technology. After being set,
serves as a requirement for all architectural projects. On a further instance, the security standards are
viewed as a support for security policies, being dynamic and involving the technologies [3][4].

After all the requirements are well-established, we are presented with the ADM for security architec-
tures, the framework’s guidelines, being comprised by the following stages:

Preliminary Phase

It all starts with the preliminary phase, with the main focus to establish and document the business
rules and security policies needed as managing requirements. For the TOGAF-9 framework, ISO/IEC 1
7799:2005 serves as common grounds to the development and establishment of these security policies.
The constraints established in these policies must be communicated and implemented by a security
architect, or a team composed of multiple security architects, depending on the project's scope and/or
budget that is designated for the architectural project. All issues regarding conflicts between the security
considerations and the functional considerations must be addressed at this point [3][4].

This first stage considers as input the written security policies, all of the relevant statutes and applica-
cable jurisdictions and returns as output the list of applicable regulations, a list of applicable security
policies, the security team roster to be responsible for the architecture, the security assumptions, and
the boundary conditions [9].

<table>
<thead>
<tr>
<th>No.</th>
<th>Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4</td>
<td>Ensure information security supports legal and regulatory compliance requirements.</td>
</tr>
<tr>
<td>4.2</td>
<td>Assign information security responsibilities throughout the organization.</td>
</tr>
</tbody>
</table>

Table 2.2: Recommendations applicable to the Preliminary Phase [5].

Phase A – Security Architecture Vision

Prior to this phase, the security policies were established, and the security measures considered. In
this stage, the main goal of the security architecture roster is to offer a management support for those
security measures, therefore all the project related decisions regarding the security for the architecture,
not only are to be documented, but also should be updated in accords to security related aspects of the
project, identified by the management personnel and executives [3].

Due to the goals concerning time deliveries around new business functions and security policies,
there might be needed to define milestones regarding security-related management sign-offs.

In cooperation with management, the architectural roster needs to identify that this role is target-
ing the protection of the enterprise assets. Any applicable disaster recovery or business continuity
plans/requirements need to be very well understood and documented at this stage, along with all the
anticipated physical, business, and regulatory environments in which the system is about to be deployed into [3][4][5].

To sum it all up, in this security architecture vision phase, it is required as input the security policies, applicable jurisdictions, disaster recovery plans, and business continuity plans. As a direct result, the output offers the physical security environment statements, business security environment statements, regulatory environment statements, security policy cover letter signed, a list of architecture development checkpoints for sign-off, disaster recovery and business continuity plans, and system critical statements [9].

<table>
<thead>
<tr>
<th>Recommendations applicable to the Phase A</th>
</tr>
</thead>
<tbody>
<tr>
<td>No.</td>
</tr>
<tr>
<td>1.1</td>
</tr>
<tr>
<td>1.3</td>
</tr>
<tr>
<td>2.3</td>
</tr>
<tr>
<td>4.1</td>
</tr>
<tr>
<td>5.1</td>
</tr>
<tr>
<td>5.4</td>
</tr>
</tbody>
</table>

Table 2.3: Recommendations applicable to the Phase A [5]

Phase B – Business Architecture

At this point, the business architecture focal point recalls in identifying who are the legitimate actors about to become responsible for the interaction with the product, service, or process. These actors considered, not only are part of the administrative personnel, but also customers of the software applications and can be defined as system users. Identifying who these actors will be must be a very careful process, as many of the future decisions related to authorizations require a precise comprehension of the users intended and their capabilities and characteristic, being them administrators and operators of the system. Not only those with administrative roles should be focused, but also those not-to-be trusted users, such as Internet-based users (i.e., customers) [3][4].

There is also a need to document how these customers are classified as proper users of the system. They, who are considered an outside party related to the system, are already taken into equation from high-level scenarios developed in Phase A.

The correct application of the security measures may pose as a goal obstruction to both users and administrative personnel alike. This obstruction may give birth to shortcuts developed to overcome
these difficulties and be the origin of the loss of customers to a direct competitor. Henceforth, it’s an important step to determine how much of these shortcuts are acceptable while also regarding the security measures and provide balance between the security advantages and the business advantages [3].

Interconnecting systems beyond project control should also be identified and documented, as well as security forensic processes to properly implement the security measures [9].

Another consideration to be taken into account is the assets at risk upon a security breach and determining whether its security cost is justifiable both by the threats presented and the value of those assets[? ]?[9].

At this stage, it is identified as an input the applicable disaster recovery and business continuity plans, the business and regulatory security environment statements, and the applicable security policies and regulations. Then as an output, we have the forensic processes, new disaster recovery and business continuity requirements, validated business and regulatory environment statements, validated security policies and regulations, target security processes, baseline security processes, security actors, interconnecting systems, security tolerance for each class of security actor, asset list with values and owners, list of trust paths and availability impact statements, and a threat analysis matrix[9].

<table>
<thead>
<tr>
<th>No.</th>
<th>Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Develop information security strategy consistent with the business goals and responsibilities of the organization, with Board-level approval.</td>
</tr>
<tr>
<td>3.1</td>
<td>Conduct information security risk assessments in line with the enterprise risk assessment methodology.</td>
</tr>
<tr>
<td>3.2</td>
<td>Prioritize the treatment of risks and ensure the treatment is proportionate to the business impact.</td>
</tr>
<tr>
<td>4.2</td>
<td>Assign information security responsibilities throughout the organization.</td>
</tr>
<tr>
<td>4.3</td>
<td>Allocate responsibility for information security to match business roles.</td>
</tr>
<tr>
<td>4.4</td>
<td>Define information security responsibilities for external parties in engagement contract.</td>
</tr>
<tr>
<td>5.1</td>
<td>Implement information security controls to support service continuity.</td>
</tr>
<tr>
<td>5.3</td>
<td>Ensure the security of all organizations involved in the business value chain.</td>
</tr>
<tr>
<td>5.4</td>
<td>Consider employee interests in the design of security systems.</td>
</tr>
</tbody>
</table>

Table 2.4: Recommendations applicable to the Phase B [5].
Phase C – Information Systems Architecture

This chapter describes the information systems architectures for an architectural project and includes the development of Data and Application architectures. While preparing for a gap analysis, all of the architecture elements considered in the security services implementation must be determined [3][4].

It is mandatory to identify safe default actions and failure states. These safe default actions and failure modes must be established in order for the system known by the current state, business environment, applicable policies, and regulatory obligations [3].

Standards are viewed mostly for reducing costs, enhancing interoperability, and improving upon innovation. That being said, from a security perspective, by identifying and evaluating guidelines and standards such as standard protocols, standard object libraries and standard implementation, also ensures an error-free implementation. These errors are perceived as the system’s security vulnerabilities [3].

The presence of system failures or loss of functionality represent a serious value loss for the stakeholder. Therefore, if possible, these losses should be quantified and documented [3].

The previously determined business disaster and continuity plans should shelter the system from unwanted results. This solution may not always be possible, and if this is the case, a gap analysis is due to determine the gap and its cost.

Upon this phase, as an input, it’s considered the threat analysis, risk analysis, forensic processes, business policies and guidelines, DR and BCM requirements, and the interconnected systems. As a result, it’s expected the event log-level matrix and requirements, risk management strategy, data life cycle definitions, list of configurable system elements, baseline list of security-related elements of the system, new or augmented security-related elements of the systems, security use-case models, list of applicable security standards, validated interconnected system list, information classification report, function critically statements, revised disaster recovery and business continuity plans, and a refined threat analysis matrix.

<table>
<thead>
<tr>
<th>No.</th>
<th>Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5</td>
<td>Embed information security within the lifecycle of enterprise information systems.</td>
</tr>
<tr>
<td>2.6</td>
<td>Implement security based on transparent, trusted and proven solution.</td>
</tr>
<tr>
<td>3.1</td>
<td>Conduct information security risk assessments in line with the enterprise risk assessment methodology.</td>
</tr>
<tr>
<td>5.2</td>
<td>Ensure sensitive customer and community data is protected appropriately.</td>
</tr>
<tr>
<td>7.2</td>
<td>Review information security controls against national and international standards.</td>
</tr>
</tbody>
</table>

Table 2.5: Recommendations applicable to the Phase C [5].
Phase D – Technology Architecture

This phase, correspondent to the Technology Architecture, is responsible for the development of a technology architecture for the architectural project.

In security architecture, the current security-specific technology is assessed and measured for the enhancement of the existing objective, and assumptions related to the interconnecting systems beyond project control are revisited. There is also a need to identify and evaluate all of the applicable guidelines and standards [3][9].

As measurements, it's important to identify methods so that the resource consumption is regulated, as the system heavily relies on these, that can be consumed not only in anticipated cases but also on those not taken into account upon the system’s design. To achieve such feat, a method to measure the efficacy of the security measures can be engineered [3][9].

At this stage, as security inputs, it's considered the security-related elements and interconnected systems, the applicable security standards, actors and risk management strategy, validated security policies, and regulatory and trust requirements. After this phase, it's expected as security outputs the baseline list of security technologies, list of validated interconnected systems, list of selected security standards, security metrics and monitoring plan, user authorization policies, risk management plan, and user trust (clearance) requirements [9].

<table>
<thead>
<tr>
<th>No.</th>
<th>Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.6</td>
<td>Implement security based on transparent, trusted and proven solution.</td>
</tr>
<tr>
<td>4.3</td>
<td>Allocate responsibility for information security to match business roles.</td>
</tr>
<tr>
<td>5.1</td>
<td>Implement information security controls to support service continuity.</td>
</tr>
<tr>
<td>5.2</td>
<td>Ensure sensitive customer and community data is protected appropriately.</td>
</tr>
<tr>
<td>7.3</td>
<td>Implement systems and processes to identify and respond to malicious or unintended information security breaches.</td>
</tr>
</tbody>
</table>

Table 2.6: Recommendations applicable to the Phase D [5].

Phase E – Opportunities and Solutions

The opportunities and solutions phase describe the process of determining delivery vehicles (projects, programs, or portfolios) that effectively deliver the target architecture previously conceived.

During this phase, the risk profile of each of the proposed architectures is to be evaluated and then compared. In the event of having more than one architecture achieving the same business objectives, it’s preferable to consider the architecture that poses the lowest risk [3][5].
When the project's scope increases, the risk of over-complicating the security architecture also increases. This problem can be overcome while in the selection process through the identification of solutions proportional to the organization's risk [3][5].

<table>
<thead>
<tr>
<th>Recommendations applicable to the Phase E</th>
</tr>
</thead>
<tbody>
<tr>
<td>No.</td>
</tr>
<tr>
<td>3.1</td>
</tr>
<tr>
<td>3.2</td>
</tr>
<tr>
<td>7.4</td>
</tr>
</tbody>
</table>

Table 2.7: Recommendations applicable to the Phase E [5].

**Phase F – Migration Planning**

The migration planning falls upon the detailed specifications related to the architectural components to be implemented, and the scheduling of supporting tasks.

The current standard architecture will suffer an impact change, caused by the transitional architecture. By adopting a security perspective, this impact should be addressed with the purpose of ensuring that the security designed for the architecture is not compromised during its migration. In order to enforce a risk mitigation strategy, it's necessary to regularly test the efficacy of the security measures [5].

Other factor to have in consideration is the fact that architecture changes can have an impact on how an organization does business. Therefore, such changes should be communicated to the relevant stakeholders and/or affected parties to assure the efficient function of the business operations [5].

<table>
<thead>
<tr>
<th>Recommendations applicable to the Phase F</th>
</tr>
</thead>
<tbody>
<tr>
<td>No.</td>
</tr>
<tr>
<td>3.1</td>
</tr>
<tr>
<td>6.4</td>
</tr>
<tr>
<td>7.5</td>
</tr>
</tbody>
</table>

Table 2.8: Recommendations applicable to the Phase F [5].
Phase G – Implementation Governance

Upon the Implementation Governance phase, it’s provided an oversight of the architectural implementation. It’s responsible for providing architecture artifacts, design and code reviews, and designate the acceptance criteria for the successful implementation of the findings. It is also at this point where methods and procedures with the purpose of reviewing evidence, produced by the system, are defined. For this intent, it’s mandatory for trained personnel to guarantee the correct deployment, configuration, and operations of security-relevant subsystems and its components [5].

<table>
<thead>
<tr>
<th>No.</th>
<th>Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.2</td>
<td>Review information security controls against national and international standards.</td>
</tr>
<tr>
<td>7.4</td>
<td>Develop a feedback process to incorporate incident details into risk assessments and control selection as part of the systems lifecycle.</td>
</tr>
</tbody>
</table>

Table 2.9: Recommendations applicable to the Phase G [5].

Phase H – Architecture Change Management

This stage is responsible for determining if more architectural changes are needed. It assimilates the security related changes to the environment into the requirements for future enhancement [5].

<table>
<thead>
<tr>
<th>No.</th>
<th>Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2</td>
<td>Implement enterprise processes that support practical and timely solutions for information security.</td>
</tr>
<tr>
<td>2.5</td>
<td>Embed information security within the lifecycle of enterprise information systems.</td>
</tr>
<tr>
<td>6.1</td>
<td>Develop and maintain the information security policy to be practical and current.</td>
</tr>
<tr>
<td>6.4</td>
<td>Participate in informal and formalized information sharing networks.</td>
</tr>
<tr>
<td>7.2</td>
<td>Review information security controls against national and international standards.</td>
</tr>
<tr>
<td>7.3</td>
<td>Implement systems and processes to identify and respond to malicious or unintended information security breaches.</td>
</tr>
</tbody>
</table>

Table 2.10: Recommendations applicable to the Phase H [5].
2.5 Lifecycle

An architectural view represents an organization’s architecture through a graphical presentation perspective at any given moment. By being provided a blueprint, a set of architectural views according to a time-frame, it requires the definition of a Lifecycle associated with its architecture elements.

With a pre-designated color scheme to represent an artefact’s Lifecycle stage, the Lifecycle can be monitored at any given time by its element representation.

Blueprints are to be complemented by a time slider, able to offer a visualization support for the Lifecycle’s evolution in time, allowing architects to scan either previous representations (AS-WAS models, actual representation (AS-IS model), or predicted representations (TO-BE models).

The implementation of a Lifecycle visual approach of architectural elements, integrated with IT management processes, offer the following benefits:

- EA compliance with the organization’s dynamics, able to represent the impact of ongoing projects.
- EA is able to display a prediction status, through the consolidation of project results not yet completed.
- Representation of the organizational transformation through project deliverables.
- Maintenance of an updated AS-IS model and TO-BE model according to project status.
- Able to provide What-If analysis.

2.6 Overview of Existing Frameworks in Enterprise Architecture

2.6.1 Zachman Framework

Published by John Zachman in 1987, who is considered one of the pioneers regarding Enterprise Architecture. Due to the increased scope of design and levels of complexity of information systems implementation at the time, the application of some logical construct (or architecture) was required. This framework follows the basic principles regarding the classical architecture view by establishing a common vocabulary and perspectives with the purpose of describing complex enterprise systems. The Zachman Framework is comprised by six perspectives (or views): Planner, Owner, Designer, Builder, Subcontractor, and User. As a follow-up, the framework attends to six simple questions: what, how, where, who, when, why. It does not feature any guidance on sequence, process, or implementation, but rather guarantees all views are well established, assuring the system is complete regardless of the order in which they were incorporated. The Zachman Framework does not feature explicit compliance rules as this is not a standard when written by or for a professional organization. Notwithstanding, compliance can be achieved if it’s used in its entirety and all the relationship rules are followed [10].
2.6.2 Department of Defense Architecture Framework (DoDAF)

The Department of Defense Architecture Framework (DoDAF) features three sets of ‘views’: Operational, System, and Technical Standards. A new view, ‘All View’, improves the previous by providing a connection (link) between them by the means of a dictionary composed by terms and context, summary, or overview-level information. In this framework, descriptions of final products, guidance, and a set of rules are provided to ensure consistency [10].

2.6.3 Federal Enterprise Architecture Framework (FEAF)

The Federal Enterprise Architecture Framework was conceived by the US Federal Chief Information Officers (CIO) Council. It was originated by observing the industry trend of developing architectural frameworks to assist in the development of large, complex systems. FEAF has a major goal to organize and promote sharing of Federal information for the Federal Government, where the architectural segments are developed individually, based on a set of guidelines, where each segment is considered an enterprise on its own within the Federal Enterprise. This framework grants flexibility through the use of methods, work products, and tools [10].

2.6.4 Treasury Enterprise Architect Framework (TEAF)

Published by the Department of the Treasury, comprises a number of offices that function as individual enterprises. Henceforth, the architecture requires the map of interrelationships among the organizations so to manage its IT resources, with the aim of “facilitating integration, information sharing, and exploitation of common requirements across the department” [10].

2.6.5 Comparison of Enterprise Architecture Frameworks

When comparing EA Frameworks, the procedure may have some challenges. This is due to some of the studied frameworks having a specific scope and relate to a particular application or development methodology (i.e., object-oriented development or distributed systems). Moreover, these frameworks can be enterprise oriented or specific to the development of the IT system. Therefore, the comparison between the previous Enterprise Architecture Frameworks is conducted over both the views and their abstractions, as well as their System’s Development Life Cycle coverage. While performing a comparison study over the EA Frameworks, both views and abstractions are compared, with the views being more quantifiable [10].

Regarding this topic, Zachman’s views are determined as the most complete, in a sense that other stakeholder perspectives could be portrayed with its terminology. The Planner view represents the concepts for the final product and may include items for the relative size, shape, and intent of the final structure. The second view describes the owner and may be used to display the architect’s drawings where the owner agrees that the architect has fulfilled the desired representation. The designer view represents the architect’s final product, where both plans and resources are detailed, representing the
plan to be implemented. The fourth view, builder view, describes the medication over the architect’s final plans to detail how the construction will proceed. The subcontractor view shows drawings of parts or subsections of the plans, an ‘out-of-context’ specification. Lastly, the user view represents the final product. Even though this terminology might slightly differ among the different frameworks, the views can be represented this way [10].

As stated above, the abstractions comparison is less quantifiable than the views or perspectives. While most of the EA Frameworks reported can provide recommendations for each abstraction representation, they do not provide methods, procedures or deliverables required. Frameworks differ when starting to compare both the technological and physical aspects of the system. Some present detailed architectures while others do not. Regarding people’s abstraction, the Frameworks present the organizational relationships whereas timelines and justification for the system can be depicted in the Time/When and Motivation/Why abstractions are only found within Zachman’s Framework [10].

Other comparison relates to the Systems Development Life Cycle, and the framework’s ability to address this issue is important. These frameworks can be correlated to five phases used in SDLC models: Planning, Analysis, Design, Implementation and Maintenance. Even though it not specified how the system is to be developed, these frameworks provide guidance regarding the implementation in the SDLC [10].
### 2.7 Enterprise Security Architecture Tools

With the growth of information systems, namely, the Enterprise Architecture practice, there are numerous tools already developed in the market. These tools comprise enterprise modeling techniques developed into Enterprise Architecture Frameworks and support architects, managers, and CIOs in creating Enterprise Architecture products [11].

The usage of these tools provides advantages such as:

- Descriptions of architectural domains stored as part of the same model.
- Increasing consistency of the architecture.
- Automatic generation of views and visualizations.
- Work in teams.
- Description of enterprise architecture principles in the same model.

Even though the Enterprise Architecture practice is not focused on the usage of these tools, as they don’t solve problems on their own, they can assist the architects in doing so [11].

There are a few indicators for the need of an EA tool:

- While having a group of architects, consistency can become an issue.
- Having a federated environment, insight in others work is an issue.
- In strategic management and projects, different views are needed.
• When spending lots of time on synchronization, different independent models are hard to keep in sync.

• Delivering the requested insights in time.

• Having the need for analysis such as baseline/target comparison and impact of change.

Through a study over multiple architectural tools available, it was perceived what visualization type EA practitioners use, how these visualizations are configured, how they adapt both visualizations and information models [12].

The analysis is performed over the following management tools:

• ABACUS (Avolution)

• ADOit (BOC AG)

• ARIS (Software AG)

• BiZZdesign EA Tool Suite (BiZZdesign)

• Corporate Modeler Suite (Casewise)

• Enterprise Architect (SparxSystems Ltd)

• Envision VIP (Future Tech Systems)

• iteraplan (iterate GmbH)

• Layer8 (Layer8-Solutions GbR)

• leanIX (LeanIX GmbH)

• MEGA (MEGA International)

• planningIT (Software AG)

• PowerDesigner (Sybase/SAP AG)

• Process4.biz (process4.biz Softwareentwicklungs- und Vertriebs GmbH)

• QPR Enterprise Architect (QPR Software)

• Rational System Architect (IBM)

• SAMU Repository (Atoll Technologies Ltd)

• Txture (QELaB Business Services GmbH)
Visualization Type

One of the central concepts of this paper is the identification of how Enterprise Architecture's information models can be viewed while in a generic approach, able to display a wide range of unpredictable information. The management tools to support the architectural practice can be composed by multiple visualization's perspectives [12].

Configuration

The configuration required to bind information, and thus configure and deploy the visualizations can be performed in two ways, hard-coded or loosely coupled. While the first method can be applied to bind instances of specific classes and properties to represent on the visualization to deploy, the last method describes the application of filters over the existing binding schema [12].

Visual and Information Models Adaptation

The information models can be adapted during its run-time. While models can be adapted to the architectural context during its initial configuration it's still possible to create its meta-model. As such, these information models can still be restructured using predefined building-blocks if there's no organizational context prior to the visualization’s deployment and are complemented with modifications over their elements [12].
2.8 Matrices in EA Tools

By definition, a matrix is a structured graphical representation of information where data is organized by rows and columns.

In Enterprise Architecture, matrices are one of the main ways for presenting architectural information to stakeholders, as these provide an information rich representation of the architecture.

In more detail, both the Relationship Matrix and Gap Matrix are grids that allow the relationships between two sets of elements, or classes, to be visualized under a matrix layout format, where one set of elements is represented on the horizontal axis and the other on the vertical axis. Information displayed at the intersection of rows and columns represent data regarding the relationship between any given pair. Moreover, at the present state, visualizing is not enough as an efficient tool. Matrices also need to support interactivity in visualizations, enabling in-place interactions, including visual manipulation and analysis [12] [13].

An important aspect of an EA implementation is centered on the tools available to support architects by increasing their productivity. As such, the current capabilities of the tools available in the market today should be a study subject.

Based on the investigation conducted, it was perceived that the development through certain tools would result into an empowerment to its end-users (i.e., architects) by enabling model adaptation bindings as well as information modeling improvements.

These tools are commonly built as web-based solutions, providing architects with a platform able to
support interactivity in its represented visualizations through visual element manipulation and analysis. Animation usage within EA tools have been increasing with the passing of time as it’s no longer enough to simply display information. Meaning, data is to be manipulated by visual interactions provided by an EA platform and increase architect’s productivity.

Other topic of concern is the high demand of support in the customization of EA information models, as end-users require assistance for an adaptation of those models. And so, tools are also supposed to offer multiple approaches so to facilitate these models’ adaptations.

While providing a wide range of visualization covering multiple usages, some are required more often. And so, it can be concluded that the top five most requested visualization types, among the available EA tools, were as follow: cluster map, matrix, graph, timeline, and flow diagram. As it stands, these five types are implemented in almost every scenario (showing a usage percentage of 48%).

Moreover, some of the visualization types provided in by EA tools are shown to be more relevant regarding specific stakeholders, and thus can be seen more often, with the exception of the five already mentioned:

- Bubble Charts: mainly used to establish communication with management (covering Business as well as IT).
- BPMN: implemented to display information for business analysts and management.
- UML: important for enterprise and solution architects.
- ER Diagrams: favored by enterprise architects, solution architects, and business analysts.
- Dashboards: targeting stake holders and enterprise architects.
- EPC: used by business analysts.
- ArchiMate: used mainly by enterprise architects.
- Bar Charts, Radar Charts, and Pie Charts: popular among management, enterprise architects and IT managers.

As it's possible to infer from this investigative work, a matrix display tool in EA platforms is of the upmost importance to its wide popularity among architects. Being considered one of the five most sought-out visual representations regarding EA projects, should become a priority virtualization in every EA platform [12].

2.9 Background Conclusion

In order to implement and/or model an Enterprise Security Architecture project, from the previous chapters, it’s possible to identify that matrices visualization is highly demanded by enterprise architects, but also provides an essential visualization tool regarding the conception of architecture security projects.
As already stated, a security architecture aims to address both the requirements and risks inherited with a specific scenario, while determining which security controls should be applied and where.

At an architectural level, the major areas of concern when considering a security application on an Enterprise Architecture model suffer a huge drawback with the lack of a matrix visual artefact. This artefact is capable to facilitate the definition or attribution of relationships, and most important of all, visualize all of the relational dependencies between its represented objects. As observed, most of these areas can be covered with the integration of a visual artefact:

- Authentication: represented to cover both identities and systems.
- Authorization: relates the permissions of an entity over a system.
- Audit: displayed as a security policy and system relation.
- Assurance: representing the security attributes tests over a system, accordingly to a selected security policy.

Moreover, EA tools available in the market today still have not reached its full potential. Lifecycle representation should also be taken into consideration as it empowers organizations in the monitoring, via a visualization tool, of the model's evolution.
Chapter 3

Implementation

3.1 Description of the Solution and Motivation

As per concluded from an analysis of the present state of art, the matrix focal point has been highly demanded by enterprise architects these past years due to its information-oriented properties [12]. Through an analysis performed over a security context, most of its areas of concern evidence cases of study describing relations between multiple objects. Due to the matrix structure properties where data can be organized by columns and rows, this view stands as an efficient tool to address the concepts identified when designing a security architecture.

As it stands, the current state of the art offers a limited view over the matrix visual tool integration, not capable to provide enough versatility and complexity when considering the full extent of how these views can be represented or manipulated.

When contemplating the Atlas’ platform, the available visual representations did not offer enough support for the description of information security architectural models and its inherited challenges while addressing both risk analysis evaluations and describing its security controls.

The solution presented focus on the support over Atlas in the visualization of Security Architectures with the development of a matrix focal point. This matrix is designed as being generic, and thus independent of the data models, ensuring its reuse. This development gave birth to other challenges, such as how objects and relations can be represented within the matrix, and what configurations are required to deploy its view.

Moreover, a set of features were integrated to support the architectural practice, and thus allow architects not only to successfully establish and remove objects while interacting directly with this view, but also to be able to reconfigure and restructure the matrix’s data with the support of a new panel.

This solution also addresses how the lifecycle of the components present in the matrix can impact this view and successfully monitor its evolution through its represented time frames.
3.2 Methodology Applied for the Development

One important step regarding software development is the correct application of a development methodology, focused on the improvement of the quality and time delivery of the solution (product). This methodology, also known as Software Development Life Cycle (SDLC), translates into the process of splitting the software development work into various phases so to enhance design, product management, and project management. It may include the pre-definition of deliverables and artifacts created and completed by a project team to develop and/or maintain an application.

As for this project, the Agile Software Development methodology principles were followed. These are centered upon the idea of establishing an iterative development, the process of repeating and refining a cycle of working in development, where both the requirements and solutions evolve mainly through collaboration between teams or team elements. Agile methods or processes raise a disciplined project management process sustained by the iteration regarding the development and the continuous feedback provided to successively refine and deliver a software system faster, with an improved quality and predictability, and an increased capability to respond to change [14][15].

Within the Agile methodology it is possible to identify numerous methodologies and processes based on the principles previously described, two of which were actively applied: Feature-Driven Development and DevOps.

In software development, Feature-Driven Development (FDD) is described as an iterative and incremental process under the Agile method. This process blends together a considerable number of industry-recognized best practices, where these are driven from a client-valued point of view. The FDD is viewed as a model-driven short-iteration process designed to scale up from small development teams to larger projects. It consists of five basic activities or work stages [15]:

- Develop Overall Model.
- Build Feature List.
- Plan by Feature.
- Design by Feature.
- Build by Feature.

On the other hand, DevOps translates into a software development methodology aiming to bring development teams and information technology operatives together. It merges software development (Dev) with operations (Ops). The major goal is to establish communication between these teams so to build, test, and release software faster and with improved efficiency and speed [15][16].

3.3 Technologies

As with any development project, especially in the conception of web-based applications, its conception and integration comprise a multitude of technology implementations. The development of the matrix
visual tool is by no means different. In this chapter, technologies integrated to successfully deploy the matrix visual tool are detailed in full.

### 3.3.1 HTML and HTML5

The Hypertext Markup Language, commonly known as HTML, is defined as a standard markup language, describing the structure of a web page to display documents in a web browser while assisted by other programming technologies such as Cascading Style Sheets (CSS) and scripting languages like JavaScript.

Hypertext stands as a reference to the hyperlinks often found in an HTML page, while the Markup Language defines how tags are implemented to structure both the page layout and the elements within that page.

Even though the web was subjected to great changes over the years, HTML has always prevailed as a fundamental language used in web page development. While the complexity in website design has increased, becoming more advance and providing more interactive features, the HTML has been simplified, requiring the implementation of less code. This simplification over the modern HTML is due by relying on other technologies to format most of the elements present in a web page [17].

HTML5 is the latest major revision over the HTML standard succeeding its previous generations and has become a standard for structuring web pages and presenting them over the World Wide Web [18].

This new standard resolves in the incorporation of features such as video playback and drag-and-drop which were dependent of third-party browser plug-ins prior to HTML5 development.

Upon the incorporation of the HTML5 as the new standard, multiple new elements, attributes, and features were introduced to provide assistance in the development of modern websites. Some of the most prominent features are the following:

- **New Semantic Elements:** Such as header, footer, and section.
- **Forms 2.0:** HTML web forms were improved, introducing new attributes for the `input` tag.
- **Web Socket:** Next-generation bidirectional communication technology, targeting web applications.
- **Server-Sent Events:** Events flowing from the web server to web browsers.
- **Canvas:** Support of a two-dimensional drawing surface, programmable with JavaScript.
- **Audio and Video:** Audio and video can be embedded on web pages without relying on third-party plugins.
- **Geo-location:** Physical location sharing through web applications.
- **Micro-data:** Allows creating new vocabularies beyond HTML5, and so, extend web pages with custom semantics.
- **Drag and Drop:** Drag and drop items to a new location within the same web page.
3.3.2 Cascading Style Sheets

Cascading Style Sheets, also known by its CSS acronym, is often integrated to assist in formatting the layout of web pages. It is implemented to define text styles, table sizes, and other content present in web pages that could only be defined in a page's HTML.

CSS has been developed to assist web designers in the creation of the same layout presentation across several pages belonging to the same web site. Rather than defining each element within a page's HTML, the most featured styles need only to be defined once in the CSS document to be used on any page by referencing the CSS file. The application of Cascading Style Sheets also facilitates style changes across multiple pages.

Not only is CSS a useful tool in creating text styles, it also provides support in formatting other aspects regarding the web page's layout such as cell padding of table cells, the style, thickness, and color of a table's border, and the padding around images or other objects.

With the implementation of CSS, developers are given more control regarding web page's layout in comparison with HTML, which is why most of the modern web pages incorporate this technology [19].

3.3.3 JavaScript

JavaScript is one of the most popular programming languages today, frequently used in web development.

From day one, it has provided developers with means to add dynamic and interactive elements on their web sites. And even though it has a clear Java influence, the syntax itself is closer to C, and based on ECMAScript.

It's a client-side (i.e., front-end) scripting language, which means its source code is processed at the client's web browser and not at the server per usual. Therefore, functions written in JavaScript are available after the web page finishes its initial loading, not requiring further communication with the server. The code blocks produced in JavaScript can also compute error messages before any information is transmitted to the server.

Similarities to other back-end scripting languages exist as, much like PHP and ASP, JavaScript can be coded anywhere within the HTML of the web page with the minor difference of remaining fully visible on the web page's source [20].

JQuery

JQuery, an open-source feature-rich JavaScript library (or JavaScript Framework), allows developers to increase powerful functionalities to web sites, and stand among the most popular JavaScript libraries in web development.

For the JQuery library to be integrated, the web page's HTML needs only a reference to the JQuery JavaScript file, and while referenced, any of the JQuery functionalities are supported. Among many things, these features cover HTML document traversal and manipulation (such as text modifications,
form data processes, re-location of elements), event handling, and performing animations. Other im-
portant feature of JQuery resides in its compatibility with AJAX code and scripting languages to access
data previously stored in a Database. Also, AJAX coding becomes much simpler, as the library provides
an easy-to-use API, working with multiple browsers.

Since JQuery stands as a front-end framework, allows information/data to be updated in real-time,
not requiring the web page to be reloaded [21].

**AJAX**

Known by its acronym, AJAX, the Asynchronous JavaScript and XML is a technology designed to
assist developers to load data from the application’s server (back-end) and display it on the web page
without the need to refresh the browser (i.e., reloading the whole page).

The JQuery framework provides several methods incorporated with the AJAX functionality, and with
the application of these methods, it can request text, HTML, XML, or JSON from a remote server. All of
these procedures are integrated with the JQuery `ajax()` method, enabling asynchronous HTTP (AJAX)
requests to load the client’s selected HTML elements.

Other important feature present within an AJAX call is the execution of multiple JQuery methods
during its life cycle process, predicated by different events, or stages. Hence, an AJAX call can compute
other functions (callbacks) upon the competition of an AJAX request.

It is through the implementation of AJAX requests that communication with the back-end server is
established. This stands as a vital procedure in the retrieval of information stored within the database
[22][23].

**D3.JS**

D3 presents itself as a JavaScript library aiming at the manipulation of documents whilst focusing on
data, able to bring information to life with HTML, SVG, and CSS.

By prioritizing web standards, provides the full capabilities of effectiveness in contemporary browsers
with the combination of powerful visualization components and a data-driven approach to DOM manip-
ulation, while not committing to a proprietary framework.

The D3 framework is used to bind data to a Document Object Model (DOM) prior to the application
of data-driven transformations in the same document. While not providing every conceivable feature
present in the market today, it offers a highly efficient manipulation of the DOM based on data. This so-
lution avoids the usage of proprietary representations and affords additional flexibility. It’s an immensely
fast library, able to support large data sets and dynamic events for both interaction and animation.

Also, it encourages code re-utilization through a vast collection of official and community developed
modules. The combination between D3 and JQuery provides access to its dynamic properties, as these
can be defined as functions of data instead of simple constants, which can be extremely powerful.

This technology is responsible for computing data objects whilst providing support for the virtualiza-
tion in the development of the matrix artefact. Moreover, with the correct implementation of its proper-
ties, contributes to the integration of complex logistic problems such as the dynamic range of the matrix layout [24].

**Kendo UI**

Kendo UI for JQuery stands as a collection of JavaScript UI components for JQuery, Angular, React, and Vue. It translates into an extensive HTML5 user framework to develop interactive web sites and web applications, while being able to offer eye-catching, high-performance, and responsive applications through several UI widgets, data-visualization gadgets, front-end data source, and built-in MVVM (Model-View-ViewModel) library for enterprise-grade line-of-business applications capable of producing professional web sites regardless of the JavaScript framework of choice.

This framework delivers almost everything needed to design up-to-date applications while decreasing its time-to-market by integrating components to handle all the key functionalities for the UI, encouraging the development efforts to be focused on the proprietary features.

The Kendo UI stands as the basis for the conception of the customization panel available within the matrix artefact, through a vast composition of drop-down lists and cascading drop-down lists not only to offer, but also change the required configuration elements without the generation of a new matrix layout. These changes do not require any web page to reload, due to its JQuery inherited attributes [25].

### 3.4 Matrix Configuration

The adaptation of the blueprint’s configuration panel is a vital step to successfully deploy the new matrix artefact’s environment, where data is to be represented as a matrix visual layout, supported by a revisited customization panel.

During the first stages in development, an iterative process comprised by production and testing steps, was noted that the initially implemented full configuration did not offer enough versatility as a project management tool. It would also require higher learning curves for architects as this could be interpreted as a complex step. And so, by automate the matrix artefact configuration even further, the visualization tool is capable of offering a wider range of initial configuration setups.

The diversification presented by the multiple setups available allow architects to adapt the project configuration freely, as it can omit information while still capable of being deployed.

Therefore, the matrix artefact is enhanced to support three types of configurations: a fully detailed configuration, a partial configuration composed only by axes data, and finally, an empty configuration where no setup is required prior the matrix’s environment generation.

As for the detailed information, two data classes (each filtered by a query) are required for the matrix’s background visual representation and is responsible for the definition of each row and column element. These classes are structured as containers, in an organic representation, able to store its element’s information retriever from the axes’ queries computations.

Meanwhile, cells also require additional information in order for the matrix to be successfully deployed. As the matrix can represent relational data contained in intermediary objects, it’s imperative to
declare beforehand the class of the object capable of storing the information for the current matrix. For

cell content (i.e., relationship representation) to be visualized, the query responsible for its population

needs to be defined, able to filter content per parent object association at the intersection. Moreover,

in order to enable the addition and removal of relationships functionality by visual interaction events,

information related to addictive and destructive queries (capable of updating the database) are also

required.

3.4.1 Full Configuration

This solution focus on the deployment of the matrix’s visualization tool based on a full configuration

setup prior to its generation. Even though this type of configuration has the highest learning-curve,

due to the complexity inherited by the adaptation of the blueprint’s configuration panel, it still offers ad-

vantages that other types of configuration simply cannot compete with. The full extent application of this

configuration method can translate into a matrix fully deployed, ready for use, in a single step. Doing so,

also enables the possibility for the project to be recovered by simply launch the configuration saved.

3.4.2 Partial Configuration

The partial solution allows architects to power the matrix’s visual tool featuring only the axes infor-

mation, postponing the most complex configuration that can easily be manipulated with the support

integrated as the customization panel. Even though it still requires the identification of the relational

classes, all information necessary for the cell’s activation can be omitted.

3.4.3 Empty Configuration

The empty configuration setup requires only the indication of a new matrix and so, the only required

attribute is its unique identifier, the matrix’s name. As all of the remaining information can be obtained

from the variable manipulation provided by the customization panel, this option proves to be highly

valuable as the time spent to fulfill the matrix’s requirements for the graphical virtualization of data can

be completely omitted, as the customization panel that supports this artefact offers a much simpler

approach through a user-friendly interface as opposed to the complexity inherited by the adaptation of

the blueprint’s configuration panel.

3.4.4 Implementation

The blueprint content instance carries information regarding the configuration settings provided by

the Atlas platform and is represented as a data structure. To address the variety of configurations

available, a set of instructions are responsible for the inspection of this data structure and should be able

to evaluate the completeness of the data provided by the initial configuration. After which, the project’s

requirement variables are updated with the information available.
Once the data analysis routines are complete, if the information does not suffice the minimal requirements for the deployment, the client enters a standby state, composed only by the canvas’ environment, and waits for further custom information to be dynamically integrated through the customization panel.

On the other hand, if the configuration presents data defined, the system starts by the verification of the minimum requirements for the matrix background layout, comprised by the data concerning the elements to be displayed in the matrix’s rows and columns. If there’s no recollection of data concerning the details of the queries responsible for the cell’s definition, the intersections presented are presented visually even though no interaction is associated, and no relational information is displayed. At this stage, the system waits for further information.

Finally, if all of the setup information is available in the system, the matrix is deployed, powered by all its features.

3.5 Design of the Matrix Visualization Layer

At the beginning, prior to development, it’s imperative to reflect over the challenges presented, complemented with studies to comprehend the multiple technologies available and thus enable the implementation of a visual representation.

Even though, in Enterprise Architectures, there’s much more to a visualization layer than just the display of data under a specific format, the development of this layer establishes the basis of this project, where its complex interactions can be incremented. As such, the development process should prioritize the representation of the matrix visual interface.

Still considering studies conducted over the technologies, the matrix artefact is mostly developed in the JavaScript programming language, as this solution is to be fully integrated within a web-based application. Moreover, the layout development phase focus on the integration of the D3 JavaScript Library (framework) in order to achieve the goal of producing a dynamic and interactive data visualization tool, complemented with Scalable Vector Graphics (SVG), HTML5, and Cascading Style Sheets (CSS) standards.

The primary objective of the implementation of this visual data representation is to emulate the layout of an adjacency matrix. This concept, in both graph theory and computer science alike, reflects on the design of a square matrix to represent finite graphs where the inner elements of the matrix specify if pairs of vertices are adjacent in the graph. The squared matrix is then extended to cover scenarios, where both axes can be represented with different lengths.

Another key concept to consider prior to this early-build development stage is the Canvas element in HTML5. The Canvas defines an open drawable region, defined within the HTML code, to allow dynamic and scriptable rendering of 2D shapes with height and width attributes. JavaScript on the other hand, can access this region, through a set of implemented drawing functions, to generate graphics dynamically.

Therefore, this chapter regarding the development of the visualization layer covers two major topics: how to obtain data, and how this data is displayed.
Data Retrieval

After the initial configuration is set on the Atlas platform, a new matrix work-environment can be established as a new canvas is defined.

For all the intended purposes of data representation, a new data retrieval process is set in motion. As the major goal of an extended adjacency matrix is to replicate the behavior of a relationship matrix, able to display the relationships between objects from separate classes, the information regarding axes composition (i.e., a list of objects from each class) remains a top priority and is available by fetching the configuration setup.

This information, set within a blueprint configuration, can be obtained via a blueprint instance object. As such, it is required for the client to establish communications with the remote server and to request a data structure featuring the matrix settings, which may contain information related to the two classes to display. This data structure can also be provided with queries, regarding the two axes classes, to filter and retrieve objects from the database as per requested.

If the required axes data is not instantiated in the retrieved data structure, possible when provided with an empty configuration setup, the matrix artefact remains in a wait stage until this information is available through the customization panel.

The attributes present in the blueprint's configuration are retrieved by performing a JavaScript escapeBody tag procedure, and then converted into a JSON String via the JSON.stringify() method, storing its result in a new variable for further usage. This data is then sent to the server, through HTTP POST requests as per application of JQuery AJAX calls. The success callback function can then be executed with the results received as a server response, where a new data structure is defined to contain all the information required for the building the matrix.

The callback function computes the information related to the relationship between axes is obtained through the computation of queries, either provided by the blueprint configuration or via the customization panel selections. By feeding an object axes combination information to the query, the remote server retrieves an array representing the data of the inner elements (i.e., cells) of the matrix, at the position related to that object relation. In other words, the success of the computation of the query results in a list of objects related to the computed vertical and horizontal objects.

The definition of a matrix, in JavaScript, is represented by a 2D array. This is another important concept while conceiving its data structure.

For every inner-object association, the data structure comprises the following information:

- Vertical Object Name.
- Vertical Object ID.
- Vertical Object Class.
- Vertical Object Position in its axis array.
- Horizontal Object Name.
• Horizontal Object ID.
• Horizontal Object Class.
• Horizontal Object Position in its axis array.
• Inner Object Name.
• Inner Object ID.
• Inner Object Class.

The matrix is ready to be drawn when fed by this data structure.

Data Display

The matrix data virtualization is implemented, mainly, with an extensive usage of the D3 JavaScript framework and its available methods.

The 2D array, a matrix definition in the JavaScript programming language, is first generated by computing, as arguments, the length of both the vertical and horizontal axis (single arrays). By re-calling this process every time axes data is manipulated enables the creation of a dynamic matrix with an allocation of the required resources in accordance with the provided objects to display.

The cell content display process diverges with the relational type of matrix deployed. If an intermediary-object relational matrix is to be generated, a new structure defining the inner object has to be defined. This inner object, that can be referred as an intermediary relationship object, is defined with the following attributes:

• Object Name.
• Object ID.
• Object Class.
• Vertical Parent Object ID.
• Horizontal Parent Object ID.
When considering a direct relationship matrix, the relation is fetched by the computation of queries, able to filter data related to its relational parents’ information, direction of the relation and the property in which the represented relations are established.

The technology commonly recognized by its tag acronym SVG, Scalable Vector Graphics, is then integrated as means to implement vector-based graphics for the Web as 2D graphics in XML format.

Every element and attribute generated with the integration of SVG can be animated, which becomes a major asset to generate the much-needed interactions over the matrix data virtualization. Moreover, the graphical representations delivered not only possess higher quality standards at any resolution, but are also scalable, providing more flexibility as this quality is not lost as the layout is resized by the client.

While the canvas also draws 2D graphics with vanilla JavaScript, by implementing this XML solution, not only every element is available within the SVG DOM, but also event handlers can be attached to any given element. Furthermore, by remembering each shape drawn as an object, this solution provides additional support in their manipulation.

Initially, the SVG object is configured with its respective width, height, and margin attributes. The matrix background layout is then instantiated and appended to the blueprint's canvas HTML5 div, through the D3 selection method, a powerful data-driver transformation tool of the DOM.

At this stage, the scaled variables are defined, taking into consideration the background height and width attributes, along with the vertical and horizontal object lengths.

**Row and Column Representation**

After the matrix’s background display, a rectangle allocation of the matrix’s full size, the background is complemented by a virtualization of horizontal and vertical lines that together brings forth the representation of the matrix’s inner elements.

Previously, the SVG domain was configured to contain the vast range of possibilities across the axes through a data object containing the information regarding the matrix’s content, as well as the dimensions of the axes within the SVG graph.

The SVG language is also responsible for providing development options considering single SVG shape elements’ transformations, as well as for a group of elements. This transformation can be supported by a recently introduced SVG attribute named transform.

Other method to consider is a translation method, assembled to define elements position within the SVG domain, after the data structure inspection. Moreover, a ninety degrees rotation is applied over the object relation to the column representation. Doing so, it’s possible to append lines at these starter positions, and thus, successfully represent the matrix’s intersections.

Finally, text related to referred object's ID is appended on both axes to enable the matrix capable of identifying each row and column displayed.
Row and Column Representation

Cell configuration follows the same methodology applied in both rows and columns, transforming the content of the matrix’s intersections into a dynamic object with variable sizing, complemented with the addition of CSS attributes to structure is coloration layout, as well as its opacity, represented in the visual graph.

Representation wise, cells are composed by a color scheme to enable the identification of how many relations are present in the same intersection. This color system follows four cell stage patterns: no relations present, a single relation is represented, between two and ten relations contained, and more than ten relations at a single intersection.

Other concern regarding matrix’s cells is the support for interactions. Therefore, each cell is accompanied by event handlers designed for real-time user interactions, distinguished by mouse-clicks and mouse-overs.

The mouse-over interactions are responsible to deploy a tool-tip event, capable of fully describing a cell’s content, and thus, the relationships already established. The same event is also powered by a highlight feature able to display the relation’s parent objects within the matrix virtualization, applying a vibrant color change over the row and column identification for the selected position.

Meanwhile, mouse-click events are also determined, capable to deploy a new interaction window devised to add or remove relationships when in the presence of an intermediary object relational matrix. If this event type occurs over the simpler relational matrix visual representation, the mouse action recognizes the cell’s content, and directly establishes or removes the relation.

3.6 Implemented Features

This chapter focus on the implementation of a wide range of features as means to empower the matrix artefact with enough functionalities capable of increasing architect’s project management productivity, as opposed to merely become a stale virtualization tool.

By offering this increased complexity, it provides support to multiple interactions and animations directly from the graphical display. These interactions aim to aid enterprise architectures by minimizing the required time consumption associated with the implementation of architectures featuring a matrix visualization and thus, through the increased automation, prove to be an essential tool in its conception.

3.6.1 Matrix Modes and Interactions

The artefact conceived offers support for two ways of describing object relationships while still considering the relational matrix context. Besides the traditional unidirectional relational matrix, where cells are viewed as representations of relations from one axis element to its counterpart as per a selected direction over the virtualization, this relational data can also be stored within an intermediary object from a different data class. This newly devised relational concept allows architects to enclose more than one relation within the same data object.
By allowing relational data to be referenced by more than one object, it completely distances itself from the common approach applied over relational matrices.

This distinction of how relations are to be interpreted, have a major impact on how interactions should be handled. While always having the notion that automation through the application of events is to be exploited to minimize the number of steps and time consumed to achieve a determined goal, the user interface must adapt according to the problem’s essence.

As such, the interaction generated by the mouse-click action on the matrix’s cells responsible for the trigger of an event related to the creation or removal of relationships represented by the matrix can’t be the same. For this reason, matrix interactions are always dependent on the kind of matrix present.

### Direct Relations

![Direct Relationship Example](image)

Figure 3.2: Direct Relationship Example.

In a simple relational matrix, having previously set the direction of the relationship represented and considering the one relation limit per cell logic, the event handler starts by an inspection of the intersection’s content.

If there’s no relation information prior the interaction, the matrix interprets the action event as the creation of a new relationship and immediately establishes the relation, in accordance with the relation’s direction and property configured in previous steps.

On the other hand, if the cell already has an associated relation, the artefact recognizes the event as a relational removal, and so the information regarding the relationship is removed and the matrix updated.

In short, when interacting with a relational matrix, both events can be performed with just one action (mouse-click event), while the action event recognition is computed by the client while taking into consideration the matrix’s present state.

### Indirect Relations

The intermediary-object relational matrix differentiates itself from the simple relational matrix by allowing architects to store relational data within a new object. This object is also capable to contain multiple relationships established and, therefore, it requires a more complex interaction handler.

While the interactions are still generated by mouse-click event triggers, it now requires an integrated user-interface to present the multiple options available as each cell can contain more than one object
Figure 3.3: Indirect Relationship Example.

and thus it's not possible to infer if an action should be interpreted as a relation creation or removal. Therefore, the event deploys a new window to empower architects on the decision regarding which course of action is to be taken.

The addition of a new relationship can be customized, whether the information is to be stored in a new object and thus creating a new data object in the database, or within an object already in existence via a drop-down list selection. On the other hand, the removal process is computed by a simple object selection from a list containing only the objects present at the cell. Even though the relationship is removed, the object per-se cannot be deleted from the database, as it may contain more information associated.

Unlike in the direct relational matrix, these interactions are computed by hard-coded queries, able to be manipulated with the customization panel settings.

### 3.6.2 Customization Panel

Another feature to empower the matrix visualization tool is its integrated matrix-specific customization panel. This panel oversees the matrix’s structure content necessary to visually represent data in a matrix format and enable all its capabilities. At the same time also presents all of the available tools to set a new configuration and re-structure the matrix view in real-time without additional configuration. These options available, represented by structural decisions and the queries responsible for the new data setup, must be previously configured within the Atlas platform prior to the matrix deployment. Furthermore, the options available are filtered by taking into consideration previous decisions whilst in the same data definition process.

To achieve this feat, the panel responsible for the configuration changes empowers architects with multiple approaches towards the matrix layout customization. Most of these options are integrated as sets of drop-down lists, developed with the Kendo UI framework.

Moreover, generic tasks that do not require configuration step’s dependencies can be computed through already pre-defined buttons.

To ensure that all data manipulation processes are independent, once a course of action is completed (i.e., manipulate axis data, define new relationship’s representation, or the trigger of the incorporated
buttons), the matrix instantly deploys a newly updated matrix's visualization model.

**Axes Selection**

The customization panel's Axes Selection component provides a simple interface process new data to re-structure the matrix by either providing currently unknown information or modifying the present information to compose either axis' elements.

This selection is integrated in the web-based solution as two drop-down lists, one for its respective axis, each composed by a multi-step selection process.

The selection process starts with the definition of the object class related to the elements intended to represent on the axis and the storage of the information retrieved from this decision. By doing so, a new process dependency is enabled displayed as a cascading drop-down selection available in the same panel. The new selection pool references the kind of filter to apply over the previously defined object class. This filter is composed by two distinct paths: filter by Query and filter by Property.

The selection of either filter presents the architect with the final axes selection step, comprised by the decision regarding the query to apply over the previous data (with queries already defined within the Atlas platform) or over which property is the data filtered by.

While the query enables database's data requests for the retrieval of specific objects, the property filter retrieves all objects available that are related to other unspecified object through that property (i.e., selected property attribute is not empty).

Furthermore, the options regarding filter selections are presented as lists filtered beforehand. Meaning, both queries and properties are available at the customization panel take into consideration past decisions and thus only present the options capable to address the intended customization.

Finally, if this selection process only identifies one of the matrix's axis information at its conclusion (possible scenario resultant from the customization of an empty configuration), the web application's client is set on a waiting state until the definition of its complementary axis is completed.

**Cell Selection**

Another important step towards the matrix's deployment is its content definition. The Cell Selection process relates to the configuration of the matrix's content representation. The artefact provides its architects with two kinds of relation representations and thus the requirements for its definition diverge.

The first step required to display the relational representations is based on the definition of how these relationships are to be established. Therefore, architects must first provide information if the represented matrix is a simple relational matrix or an intermediary-object relational matrix. For this intent, the customization panel, again through an interactive selection feature, presents a decision related to matrix's mode to represent.

After the previous step the configuration of the cell's setup process diverges, and further decisions become based from the path already taken. Whereas the direct relational matrix requires the definition of the relationship's direction and its relational property, in contrast, the intermediary-object relational
matrix requires the indication of the queries responsible to populate the matrix, establish new relations and remove relationships currently available.

**Set All Button**

Unlike the previous configuration processes composed by the integration cascading drop-down lists, the set all option is represented as a single event button, as it does not require any further requirements.

The current matrix layout is interpreted by the client, able to verify its attribute data structure (i.e., classes represented, objects per class, type of matrix, and cell definitions), and thus establishing one-to-many relations. If in the presence of a relational matrix, these relations follow the direction provided in the configuration, whereas in an object matrix the relation is bi-directional.

**Clear All Button**

Another option displayed as a button. This function is responsible for clearing every relation displayed in the actual matrix according to its configurations, as it only clears the relations for the objects displayed.

### 3.6.3 Lifecycle

The lifecycle time slider present at the matrix’s user interface allows the monitoring of its content evolution over time.

The matrix is extended to cover both the lifecycle of the objects present as well as the lifecycle inherited by their relationships. As such, it requires the pre-assignment of a representation system, expanding the already existent scheme, through a manipulation of the cell's color opacity.

![Figure 3.4: Representation of the Lifecycle on the Visualization.](image)

If one object belonging to either a row or column is decommissioned, the entirety of said row or column relationships are pictured as also being decommissioned by an alteration of its color representation opacity. Again, there are two possible matrix representations able to display. While the previous logic applies to a simpler relational matrix, the intermediary-object relational matrix requires a slight
adjustment. Cell’s intermediary-object representations can be singularly decommissioned through this method, requiring at the very least one of these objects with the “Alive” status to represent a living cell.

Every time an object is added to the matrix visual representation, either by configuration or creation when establishing new relations, they’re added to the canvas data structure.

The client is provided with a listener with the purpose of handling the Lifecycle events, evaluating the state of every as alive or decommissioned. Every change over the canvas data structure (via matrix updates), trigger a new event handler responsible for removing the previous matrix virtualization, followed by the computations required for a new representation to be displayed.

3.6.4 Highlights

The highlight event allows objects composing the matrix to be highlighted after being selected in the respective highlight panel. The highlighted object is then displayed with a more vibrant color, able to easily be identified.

This feature proves to be a useful tool in finding singular objects, as the matrix can become more and more complex along with the increase of objects represented.

To achieve this visual representation, an event listener retrieves the input (i.e., object) provided by the Atlas’ highlight panel, and with the assistance of D3 framework’s methods, automatically apply an update regarding the text coloration of the respective object.

3.6.5 Row and Column Settings

Lastly, to complement the customization panel’s Set All and Clear All functions, the matrix’s artefact is also powered with the functionality to establish relationships by either row or column. This feature becomes available with the interaction over the elements representing a row or column.

The event handler then inspect all of the cells’ content represented in that row or column. If at least one relationship is represented, the client recognizes the action as a removal process, permanently removing the relational data presented. When no relational data is available, the handler identifies this process as a set event. Relationships are then established, following the one relation per element pair rule, taking into consideration the matrix’s current configuration.

3.7 Optimization

Software performance optimization relates to the ongoing process of modifying a software system with the purpose of increasing its productivity and thus capable of proving a more efficient and less time-consuming experience.

The performance optimization of a software tool prototype is also a key issue in the development process to achieve highly reliable and functional applications. By conducting a constant monitoring and analysis over a system, it empowers developers with enough knowledge to identify how it can be improved.
This process, when related to web applications, can be focused on the improvement of the system’s execution time, memory usage, disk space, bandwidth, and so-on. Moreover, it should adopt an architect's user experience perspective, and thus, capable to enhance its UI through minor adjustments [26].

3.7.1 Duplicate Code

Time and time again, throughout the coding development process, challenges can share similar operations. This repetitiveness present when solving partial problems can give birth to duplicate instructions among the system’s functions and should be discouraged.

The removal of duplicate code blocks not only reduces file sizes, but also enhances the system’s performance [26].

This problem can easily be overcoming by rendering the challenges with partial similar solutions into smaller problems. As this method is implemented, it becomes simpler to re-use already defined blocks of code and thus actively preventing repetition.

3.7.2 Adaptation of the SVG Size to the Canvas Display

Nowadays, a vast variety of display sizes is available in the market, ranging from laptops to monitors and their mobile counterparts. As the platform’s display measurements can vary, not only due to the hardware but also the browsers’ capability of being resized, its only logical that the matrix layout should also take advantage of this, maximizing space exploitation by inheriting its dynamic properties and adjust to the client's display settings.

The JavaScript is able to access the already dynamic canvas property values (i.e., width and height), as such, the SVG configuration size and margins can be manipulated, able to simulate the same dynamic effect.

3.7.3 Store Objects in the Canvas Data Structure

In order to successfully represent the lifecycle, objects need to be part of the canvas data structure to be caught by the listener and inform of their presence in the actual virtualization. When the matrix is provided with its axes data, objects are to be assigned to this structure.

Considering these axes references are obtained from a nested loop (i.e., a loop exists inside the body of another loop), a verification of its existence in the data structure is needed, as these repetitions may also affect the system’s performance drastically when computing a large pool of information.

Moreover, as the matrix offers support in the addition or creation of new objects through the available interactions concerning intermediary objects within cells, when establishing a new relationship, the object containing the relational data should also be added to the canvas data structure if it's unique (not already contained in the data structure).
3.7.4 Synchronization with Promises

When merging synchronous and asynchronous coding, such as the ever-present server-client communication, the need to establish a synchronization protocol have arisen.

As the server performs its computations, as per requested through asynchronous calls, the system must stand in an active wait state, as its synchronous counterpart could give birth to unwanted behavior if callback functions were to be executed prior to the server’s response.

In order to fulfill this need, JavaScript introduces the Promise object, representing either the competition or failure of an asynchronous instruction along with its resulting value.

Promises are defined by being a proxy for values that may not be known upon its creation. They are mainly used to handle results of asynchronous operations by allowing the integration of handlers associated with the asynchronous operation’s eventual success value or fault reason. It simulates the return of values as in synchronous methods, as instead of returning its final value, the asynchronous method returns a promise to bestow on that value in the future.

Vanilla JavaScript is engineered to continue the execution of other synchronous parts of the code that is able to compute, and thus not wait for the asynchronous code blocks’ competition beforehand. With the introduction of Promises, the execution of new code blocks can be postponed until an asynchronous request is completed.

This JavaScript method becomes of extremely high importance when requesting data to the remote server via queries. It becomes imperative to submit the client’s system to a wait state, until the data computed by the queries is provided to the client.

3.7.5 Execution-Time Improved with the Removal of Quadratic Complexity Algorithms

The implementation and integration of both algorithms and data structures are crucial to the system’s performance.

In order to corroborate over a system’s optimization, algorithms should be constant, logarithmic, or log-linear, as quadratic complexity algorithms usually tend to fail when scalable [26].

The number of complex algorithms can be highly reduced with the introduction of a state data structure, responsible to preserve data from previous requests or configurations. This data structure stores the variable values required for the actual matrix layout representation. As the matrix can be mutated through multiple customization settings, by having access to a globally defined data structure containing all of the matrix present preferences it is only required to spend computation resources over the actual changes. The application of this methodology discourages the allocation of unnecessary efforts, performing heavy-duty computations, that could have already been processed. In doing so, the client’s execution time for the matrix generation is drastically reduced.

Other important notion is that when performing massive operations and/or data transformations, such as updates, letting the database compute those requests tend to achieve faster execution times.

Some of the complex algorithms, namely searches throughout a vast list of object properties to
establish relationships, can also be simplified. These loops and subsequent methods to access the object data properties, and add or remove items from that data, can be replaced with an integration of hard-coded queries able to update property values within them.

A hard-coded query can be defined as the concatenation of multiple string values. The dynamic information for those queries, available within multiple variables stored in the state data structure, can also be interpreted as strings. With the concatenation of the queries' generic information and the selected data preferences (regarding operations, properties, relationship direction, and so on), a newly dynamic query is generated.

### 3.7.6 Documentation

One of the many factors to contribute to the success of a software project is the documentation. Software documentation can be described as an artifact with the main objective as to pass down information regarding the software system to which it belongs.

The document can also be viewed as a written description of a software system, holding an official status or authority, and as such, can be used as evidence or validation.

This methodical approach to documentation increases the confidence levels of the end deliverable. Moreover, it enhances and ensures the products’ success by improving its usability, marketability and support, proving to be a major factor for a projects’ success due to the clear communication of the key concepts and requirements.

A successful documentation relies on facilitating the access to information, providing a number of user entry points as well as minimizing the learning curve for new architects. With the assistance provided through these documents, support costs are predicted to be improved.

The creation of documentation resolves on a set of eight distinct processes, comprised by analysis, design, development, validation, production, manufacturing, delivery, and customer satisfaction. Moreover, there are also a set of rules or principles for writing a clear document:

- Should be written from the reader's perspective.
- Should avoid repetition.
- Should avoid unintentional ambiguity.
- Use standard organization.
- Record rationale.
- Keep it up to date.
- Review documentation.

To sum up, the software documentation translates into the creation of documents, which are to be used within the software development environment to establish communication to various stakeholders on the functions, operations, and events of a system. The documents specify the product at all levels

46
and also act as evidence of all the procedures and activities provided by the software, as such, needs to be updated, complete, consistent, and usable.
Chapter 4

Results

The evaluation methodology of the solution engineered follows an agile methodology of scheduled meetings with the Atlas DevOps team.

These bi-monthly meetings were focused on a detailed presentation of the solution's current state, complemented with an exchange of ideas (i.e., brainstorming) regarding how to overcome development issues and discuss future implementation features.

Lastly, at a final presentation, all of the matrix capabilities and multiple customization were exploited, with the matrix artefact being approved after an extensive evaluation over its complexity and was deemed ready for integration on the Atlas’ platform.

All in all, the complexity dept of the matrix developed still leaves room for improvement, as it’s possible to pinpoint situations that are not at its full potential.

The first issue being around the initial configuration step. This configuration process, based on a fully detailed setup of the blueprint’s settings adaptation, only considers one matrix type to deploy, the intermediary-object relational matrix. This problem can be interpreted as the inheritance of the configuration’s root container lack of options available to potentiate such a dynamic artefact as the matrix. To overcome this issue, a new matrix-specific container could be integrated, as it would enable the generation of a dynamic configuration panel based on a checkbox selection of the matrix type.

As far as cells are represented, the relational display could also provide additional support, aside from the already integrated toolbox. From a simplistic relational matrix point of view, cell’s rectangles representing an established relationship could also indicate the relation’s direction by superimposing an arrow over the relational cells. Moreover, in the presence of an intermediary-object relational matrix solution, applying the same method, relational representation could be enhanced to display the number of objects contained at the intersection.

In order to minimize the need to consult documentation, the matrix environment could provide one additional panel, capable to address the matrix current configuration setup and its most generic representation keys, namely, captions related to the cell’s color representation and its meaning. The integration of such panel would also allow for the color palette to be extended, able to display a different color for every different group of intermediary objects, generated as a dynamic color scheme functionality that
would update the matrix’s keys in real-time upon data manipulation.

Finally, an export functionality should be integrated to enable to capture the current visual representation of the matrix as a formatted JPEG (i.e., containing only the representation while discarding Atlas panels). This implementation would further empower this tool, improving its ability to share information between architects and stakeholders alike.

To address the facilitation regarding the description of the security concerns identified examples were made to address the visualization of authentication and authorization controls. For this purpose, demonstrations were made supported by the developed matrix after its integration as part of the Atlas solution and thus suffered minor changes to include other functionalities.

4.0.1 Example 1 - System Authentication

![Figure 4.1: Authentication System with the Matrix Visualization.](image)

This example represents how users can be authenticated on a multiple organizational system through a unique profile per system represented at its cells, and thus identify to which system each user is authenticated.

4.0.2 Example 2 - Access Control

![Figure 4.2: Access Control Visualization.](image)
Figure 4.3: Authorizations for User 1 at File G.

This example relates user authorizations over a file system, with three different levels of permissions: Read Only (blue), Read and Write (yellow) and Read and Write as the file owner. With this visualization, through a visual adaptation, authorizations levels are easily identified, whereas, by performing a visual interaction to identify its cell content, objects representing these permissions are available.
Chapter 5

Conclusions

This solution focuses on the support of the Atlas’ platform required for an information security architectural model.

After an extensive analysis of the state of art regarding Enterprise Architectures related to information security architectural models, as well as the Atlas’s platform to suffice the needs inherited by the complexity of those models, it was identified the lack of matrix visual artefact able to address the major areas of concern related to information security issues.

Moreover, limitations provided by the already available tools in the market were also pinpointed, as they do not offer support to the matrix’s elements lifecycle, nor the lifecycle associated to the represented relationships. Other key aspect to consider is the lack of representations able to store relational information on a new object external to the data virtualization.

The Atlas’ platform support was achieved by the implementation of a then unavailable data representation, the matrix visual representation, taking into consideration the visual faults identified in other tool’s matrix views. Furthermore, the visual representation is complemented by a wide range of interactions and animated manipulations, providing more versatility and productivity.

The visualization conceived also extends the current Enterprise Architecture tools, when considering an integration of the lifecycle listener to enable the monitoring of the matrix’s elements lifecycle and that of the represented element relationships. Furthermore, unlike any of the Enterprise Architecture tools studied, the developed matrix also considers element relationships with multiple objects, able to contain data regarding multiple relations at a single intersection.

As the visual tools often lacks versatility in a work environment, it’s empowered by features capable of dynamically manipulate the represented data, and thus eradicate the need to refresh the web page, new data created outside of the tool’s visual representation, and the requirements of a prior configuration.
Bibliography


