TÉCNICO LISBOA

Academic European E-identity Management Framework

Nuno Ricardo Jorge Mendes

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor: Carlos Nuno da Cruz Ribeiro

Examination Committee

Chairperson: Teresa Maria Sá Ferreira Vazão Vasques
Supervisor: Carlos Nuno da Cruz Ribeiro
Member of the Committee: André Ferreira Ferrão Couto e Vasconcelos

January 2021

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of Universidade de Lisboa.

Acknowledgements

Firstly, I want to thank my advisor, Professor Carlos Ribeiro, for giving me the opportunity to collaborate in such an ambitious project and for all the guidance and support he has provided.

Secondly, I want to thank my family for the support and encouragement provided through these 7 harsh years, particularly my mom for the incredibly difficult job of raising two teenage sons alone and to my late father as the role model I will always strive to be.

I would like to thank my colleagues of the Information Technology Department from the Rectory of University of Lisbon who gave me not only technical but also emotional support through this arduous time, mostly Tiago Felgueiras for his network knowledge, Pedro Pereira for his linux and Access Manager knowledge and Ruben Salgueiro for his windows knowledge. I would also like to thank the people who have worked with me on this project from other organizations, such as Anil Mamede from Qub-IT and the team at Agência para a Modernização Administrativa which made me consider thoroughly if my choice in career path was the correct one. I have only but gratitude of Anil for extending me his vast knowledge about Spring, Maven, Java and many more technologies.

I also want to thank my friends Artur, Eduardo, Lameiras e Portela who made my journey through IST much more painful than it had any right to be and also for giving me examples of what my path could look like and what I could become.

Finally, I want to thank Maria Casimiro, Ana Neves and Ella Saurén as the biggest motivators for getting this over with.

Lisbon, November 2020
Nuno Ricardo Mendes

Resumo

O regulamento electronic IDentification, Authentication and trust Services (eIDAS), também conhecido por European Union (EU) Regulation 910/2014, mudou drasticamente a visão dos Estados-membros da União Europeia quanto a um Mercado Único Digital entre os países da União Europeia. A integração entre a rede eIDAS e os fornecedores de serviços ainda está numa fase inicial mas já existem resultados satisfatórios.

A tese tem um projecto acompanhante fundado pela Comissão Europeia, o project eID for University (eID4U). Este projecto tem dois objectivos: o primeiro objectivo é desenvolver novas funcionalidades de acordo com a regulamentação eIDAS, para suportar novos atributos, extender o número de atributos pessoais de um utilizador e também criar um novo grupo de atributos para serem usados por universidades, os atributos académicos; o segundo objectivo é a implementação de serviçoes com utilizadores reais que utilizam a rede eIDAS.

Esta tese apresenta três serviços que usam a infrastrcutura do eIDAS; eRegistration, eLogin e eAccess; estes serviços que fazem parte do projecto eID4U foram planeados, desenvolvidos, implementados e instalados na Universidade de Lisboa. Os serviços utilizam a infraestrutura do eIDAS para autenticar e obter atributos dos utilizadores.

Palavras-Chave

eIDAS, Autenticação transfronteiriça, Gestão de Identidades, Identidade Electrónica (eID)

Abstract

The electronic IDentification, Authentication and trust Services (eIDAS) regulamentation, also known as EU Regulation 910/2014, has drastically changed how European Member States look at the prospect of a Digital Single Market in the European Union. While the eIDAS integration with Service Providers (SPs) is still in its infancy stage there have been great prospects for it's future.

The thesis Academic European E-identity Management Framework and the accompanying European project eID for University (eID4U) had two objectives: the first objective is to enhancing the initial specification of the regulamentation eIDAS to support new attributes, extending the personal attributes supported by the eIDAS Nodes and also creating a new set of attributes for academic purposes; the second is to support the practical implementation of eIDAS in real services.

This thesis presents three eIDAS-enabled services; eRegistration, eLogin and eAccess; developed during the project eID4U, these services have been designed, implemented and deployed at University of Lisbon (ULisboa). These services use the eIDAS infrastructure to authenticate and retrieve attributes of real users. They also provide solutions to typical problems encountered during the integration of eIDAS with legacy systems.

Keywords

eIDAS, Cross-border authentication, Identity Management, electronic identification (eID)

Contents

List of Figures

List of Tables

List of Acronyms

AG	Attribute Aggregator
AcP	Access Point
AEEMF	Academic European E-identity Management Framework
AM	NetIQ's Access Manager
AM	Access Management
AMA	Agência para a Modernização Administrativa
AP	Attribute Provider
AS	Authentication Service
BGCT	Bolsa de Gestão de Ciência e Tecnologia
CA	Certificate Authority
CAS	Central Authentication Service
CEF	Connecting Europe Facility
CSR	Certificate Signing Request
CMD	Chave Móvel Digital
DHCP	Dynamic Host Configuration Protocol
DSM	Digital Single Market
EC	European Comission
eID	Electronic Identity
eIDAS	electronic IDentification, Authentication and trust Services
eID4U	eID for University
EU	European Union
FA	Fornecedor de Autenticação
FCT	Fundação para a Ciência e Tecnologia
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
iAP	Interoperabilidade na Administração Pública
IdM	Identity Management
IDM	NetIQ's Identity Manager
IdP	Identity Provider
INEA	Innovation and Networks Executive Agency
IP	Internet Protocol Address

IST	Instituto Superior Técnico
JVM	Java Virtual Machine
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LoA	Level Of Assurance
FCUL	Faculdade de Ciências da Universidade de Lisboa
HTTP	Hypertext Transfer Protocol
HTML	Hypertext Markup Language
NAT	Network Address Translation
MDS	Minimum Data Set
MEEC	Electrical and Computer Engineering
MW	Middleware-Service
MS	Member State
OTP	One Time Password
PIN	personal identification number
PKI	Public Key Infrastructure
PKE	Public Key Encryption
SP	Service Provider
STORK	Secure idenTity acrOss boRders linKed
STORK 2.0	Secure idenTity acrOss boRders linKed 2.0
RADIUS	Remote Authentication Dial-In User Service
SAML	Security Assertion Markup Language
SAML v2.0	Security Assertion Markup Language version 2.0
SAX	Simple API for XML
SOAP	Simple Object Access Protocol
SP	Service Provider
SSID	Service Set Identifier
SSO	Single Sign-On
ULisboa	University of Lisbon
ULisboa	University of Lisbon
ULEP	ULisboa eIDAS Proxy
URL	Uniform Resource Locator
URI	Uniform Resource Identifier
VLAN	Virtual Local Area Network
VPN	Virtual Private Network
WLAN	Wireless Local Area Network
WSDL	Web Services Description Language

XML eXtensible Markup Language

XSD eXtensible Markup Language (XML) Schema

DNS Domain Name System

ERASMUS EuRopean Community Action Scheme for the Mobility of University Students

MW Middleware

1 Introduction

The master thesis "Academic European E-identity Management Framework" is the final project of Nuno Ricardo Jorge Mendes for the Electrical and Computer Engineering (MEEC) degree taught at Instituto Superior Técnico (IST). This master thesis serves as the culmination of the studies in MEEC and as a stepping stone for my future career path. The developments made during this thesis are a part of the project eID4U, under the supervision of Innovation and Networks Executive Agency (INEA), which builds upon the electronic IDentification, Authentication and trust Services (eIDAS) regulamentation.

1.1 Motivation

The Internet has changed the lives of billions of people around the world, the services and technologies offered by it are rapidly changing our world. Despite the lack of physical barriers there still exist digital barriers that prevent people to fully benefit from it.

Europe and the world is becoming increasingly aware of these barriers. To tackle them the EU and the European Comission (EC) have made intensive efforts to create a Digital Single Market (DSM)[1] between the Member States (MSs). In a DSM any citizen belonging to a MS would be able to access services in any other MS. One of the most important measures taken towards this effort was the EU Regulation 910/2014 on electronic identification and trusted services for electronic transactions in the DSM, also known as the eIDAS Regulation [2].

The project eID4U which includes this master thesis is a implementation and extension of the eIDAS regulamentation, which was drafted according to the conclusions of some identity european projects such as Secure idenTity acrOss boRders linKed (STORK)[3] and Secure idenTity acrOss boRders linKed 2.0 (STORK 2.0). The motivation for this project is aligned with the EC's intentions of creating and promoting a DSM.

With the pratical implementation of the eIDAS Regulation in certain services, such as the universities involved in eID4U, helping to mature the eIDAS environment, getting us one step closer to the goal of achieving a DSM in the EU.

The eIDAS infrastructure can be expanded, allowing a wider range of services and uses to operate under it's umbrella, such as the creation of ERASMUS applications using academic attributes outlined in the eID4U project.

1.2 Objectives

The goal of Academic European E-identity Management Framework (AEEMF) corresponds to the goals of the european project eID4U, which is divided in 3 different eIDAS-enabled services with different objectives.

In general terms eID4U wants to enhance the eIDAS regulamentation by adding new attributes (academic attributes) and also to support the practical implementation of the eIDAS Regulamentation by adding new services within it's scope, increasing it's range of services.

The eID4U project has 3 eIDAS-enabled services, these are: eRegistration, eLogin and eAccess. These new services will use the existing personal attributes and the newly implemented academic attributes[4].

The first service eRegistration will allow a student to create an ERASMUS application by authenticating him and retrieving his attributes using the eIDAS Network. By using the eIDAS Network the data retrieved will be given by either the student's government or his university, this method will improve the trustworthiness and reliability of the data used in the application process. The previous method, manual input of the data, is too prone to human error therefore this service will reduce the burden on the Universities the student applies to and it's respective ERASMUS offices.

As the name suggests, eLogin's goal is to allow the authentication of users at the universities' authentication systems, allowing access to the services provided by those universities. eLogin allows the user not only to authenticate using the eIDAS network but also using their own government's Electronic Identity (eID), providing another way for both foreign and national students to access the universities' resources.

Lastly, eAccess will grant Wireless Local Area Network (WLAN) access to users that are not part of the academic system and as such don't have access to eduROAM, i.e to government officials and business delegates during seminars and conferences. By using valid eIDAS credentials an user will be granted access to the WLAN giving the universities better control over who can access the network and better monitoring of the network resources compared to current methods, like a shared credential (username/password).

The eID4U project had these three goals, for which they defined the attributes necessary for cooperation between universities. My thesis has the same goals but the challenges and solutions are different, this is due to the underlying infrastructures the services must be built upon, that is the ULisboa's Identity infrastructure and the Portuguese Infrastructure for it's citizens.

1.3 Contributions

One of the contributions of this master's thesis is the development of a fully functional proof of concept for the eIDAS takeoff problem.The problem of connecting legacy authentication systems to the eIDAS Network, solved by connecting a legacy proxy between them. The legacy proxy developed, ULEP described in 4.5.3, allows the identity management system from ULisboa, NetIQ's Identity Manager (IDM) and AM, to connect to the eIDAS Network.

Improved the maturity of the eIDAS Network by developing services with real users, extending eIDAS use cases and functionality by defining new user attributes encompassed in the project eID4U and proposed changes to improve the connection from the portuguese eIDAS Node and the portuguese IdP, Fornecedor de Autenticação (FA). Integrated the academic system from ULisboa with iAP a system from Portugal's identity infrastructure to allow the request of academic attributes by the eIDAS Network.

Three services were created during this thesis: eRegistration, eLogin and eAccess. eRegsitration improves the ERASMUS application method by using the eIDAS Network to retrieve academic attributes from the user's own university. eLogin provides another option for ULisboa's users to login into ULisboa's SSO system and improved the usability of the login service. eAccess provided an option to obtain WLAN access when at the University of Lisbon (ULisboa).

1.4 Research History

The project eID4U was under the supervision of the European Comission and it was being managed by Antonio Lioy and Diana Berbecaru from Politecnico di Torino in Italy.

The master thesis was done under the supervision and orientation of professor Carlos Ribeiro one of the vice-rectors of the University of Lisbon (ULisboa). Both the project and the thesis were developed in the Information and Technology Department of the Rectory of the ULisboa, in the section Systems and Information, at the Administration of Systems unit.

During the development of the project I had the pleasure of working with Agência para a Modernização Administrativa (AMA), in charge of the Portuguese eID infrastructure and the implementation of the eIDAS Regulamentation in Portugal, Caixa Mágica, which worked in the first implementation of eIDAS, and Qub-It, responsible for ULisboa's academic system (Fenix).

This work has been supported by the Fundação para a Ciência e Tecnologia (FCT) through a grant of the type Bolsa de Gestão de Ciência e Tecnologia (BGCT) with the reference 09/BGCT/2018.

1.5 Outline

The document is structured in as follows: the first chapter is the Introduction, it presents a brief overview about the project; Chapter 2 deals with the most important theoretical concepts used in this thesis; Chapter 3 introduces the systems and implementations of the theoretical concepts from the previous chapter; Chapter 4 is the main chapter of this document presenting the work done and all of its specifications; The final chapter 5 concludes the document by presenting results and addressing the future work.

2 Concepts

2.1 Identity Management (IdM)

How do we know who we are talking to? How do we know who sent us a message? How do we know who posted this group photo on social media? How do I know who is trying to login to my site?

Identity Management is here to answer this simple question "who?", the answer being "an entity".

An entity is something; a specific individual, an organisation, a system, a machine, a group of people; something that can be identified. An entity is unique, as such it can be uniquely identified. However this "entity" is composed of several unique "identities"[5], an identity is the form the entity takes when undertaking a certain role. I am an entity with several identities, each identity performs a role, such as: as a student of Instituto Superior Técnico (IST), an employee of University of Lisbon (ULisboa), the author of this master thesis, a Portuguese citizen, or a member of an online community. Even though I am only one entity, myself, I am identified differently in each of these roles by: my student number, my employee number, my name, my citizen number or a pseudonym.

Identity Management is the field concerned with managing these entities and it's identities relying in two fundamental processes: the creation of an identity directly related to the entity, and the process of verifying said identity [6].

2.1.1 Identity

An identity is the form the entity takes when undertaking a certain role or the information about an entity that is sufficient to identify that entity in a certain context [7].

Therefore, to identify a subject a system will use an unique identity consisting of identifiers, credentials and attributes.

- Identifiers allow the system to uniquely distinguish the identity pertaining to the subject from other identities; these identifiers are unique attributes, usually: e-mail, username, phone number, pseudonym, uniquely created id.
- Credentials allow the system to verify a subject as the owner of a certain identity; the subject has to provide these credentials to the system during an authentication process for the system to validate it's ownership over the identity, examples include: passwords, digital certificates, fingerprints, SAML authentication assertions.

- Attributes are additional information about the subject pertaining to the system; these attributes can be created by the system or willfully provided by the subject, such as: age, name, gender or address for the latter and permissions or activity records for the former.

Identifiers can be created with attributes, by using a specific set of attributes to create an unique attribute, which identifies an identity within any set of them (e.g. using the attributes: full name, date of birth, gender, parent's names, nationality as one attribute).

2.1.2 Authentication

Authentication is the usage of a protocol by a system to check the identity of an entity, creating trust between the parties by validating the subject's claims of identity.

The subject must provide the unique identifier and the credential (authentication credentials) of the identity so the system can validate the subject's ownership of the identity. The credential is the data used in the authentication process. In the real world credentials usually also include the identifier, especially in the case of identification cards, such as: citizen cards, driver's licenses, employee identification cards. In the digital world there are a variety of technologies and methodologies to authenticate individuals. These methods include the use of passwords, personal identification numbers (PINs), digital certificates using a Public Key Infrastructure (PKI), smart cards, One Time Passwords (OTPs) and others [6].

The methods used to verify the identity of an entity are based in one, or several of these:

- Private Knowledge/ Shared Secret (password, PIN);
- Private Possession (key, magnetic card, smart card);
- Physical Characteristics (fingerprint, iris, facial structure);
- Behaviour Features (keyboard patterns, written patterns).

The use of several of these methods is called multi-factor authentication which provides much greater security and reliability, decreasing the likelihood of falling to methods which exploit the weaknesses of identity verification methods such as phishing, scamming, impersonation and identity theft.

2.2 Single Sign-On (SSO)

Nowadays one of the most serious issues related to personal security is the re-use and mismanagement of passwords. For each Service Provider (SP), described in chapter (2.4.2), one must have a set of authentication credentials (usually a username/password) to be able to access it's resources. To reduce the management burden of creating different passwords for each SP, a high percentage of users uses identical or similar passwords with the same username [8], which becomes a high security risk.

Single Sign-On (SSO) is a solution created to tackle this issue, it improves usability and convenience by only needing the user to authenticate once to obtain access to several SPs[9]. It can also improve personal security by using a stronger authentication method, like a smart-card or a two factor authentication system. While SSOs systems provide better personal security, if the information is leaked it has much higher negative impact because they contain more data pertaining to the user than a single SP does, unless the SP is Facebook [10].

A SSO system is a system where a user can obtain access to several SPs by authenticating at a single Identity Provider (IdP), described in chapter (2.4.1). This type of system can have several IdPs but the user only needs to authenticate in one of them. It is also possible to use proxies to relay the message of a SP to an IdP and the response from an IdP to a SP, proxies can act both as an IdP and SP and they are used when direct communication between the SP and IdP is not possible.

As an example, the University of Lisbon (ULisboa) has several IdPs that belong to different schools, a student must be part of at least one school so when he tries to access a SP in University of Lisbon's (ULisboa) system he can use his school's IdP to authenticate, even though he won't be able to authenticate in a school he is not a student in.

The entities which are part of the SSO system must have a trust relationship, this relationship is usually obtained by using a Public Key Infrastructure (PKI). Besides establishing the trust relationship a PKI allows the usage of secure communications between entities [9].

A SSO system is implemented based on an authentication protocol, such as Open Authorization (OAuth), Central Authentication Service (CAS), OpenID connect, Security Assertion Markup Language (SAML), Kerberos and others. Solutions based on these protocols have been experiencing a growth in usage. The solution used in ULisboa and in the eIDAS Network is based on the Security Assertion Markup Language version 2.0 (SAML v2.0) protocol.

2.3 Public Key Infrastructure (PKI)

One of the reasons to use a Public Key Infrastructure (PKI) is to create a secure method to exchange information between two systems by protecting against malicious attacks. To minimize or downright prevent these attacks the PKI has to provide the following security goals:

- **Confidentiality** - the concept that data is only available to the authorized entities;
- **Integrity** - a property of the data, it signifies if the data has been tampered with;
- **Identification** - or Entity Authentication, is the process where one entity (e.g. ULisboa) proves the identity of another entity (e.g. student);
- **Data Authenticity** - verification of the owner of a message and it's contents;

- **Non-repudiation** - a process preventing an entity from denying it's involvement in the execution of an action.

2.3.1 Public Key Encryption (PKE)

There are two major methods of using secrets to secure messages, symmetric and asymmetric. The symmetric method also known as Secret Key Encryption (SKE) relies on the two parties agreeing on one secret before commencing with the secure method of communication. The asymmetric method also known as Public Key Encryption (PKE) involves the use of a set of asymmetric keys called the public/private key pair.

Symmetric systems are not scalable and cannot provide all of the security goals, such as Identification and Non-repudiation because both parties use the same key to crypt and decrypt the message. For example in the case of a digital contract it would be possible to both sign the contract as the buyer and the seller.

Asymmetric systems scale much better, the parties do not need to agree on a secret before starting to communicate, each party has one asymmetric key pair and they can sign a message with their private key which can be verified by using the public key to check the signature. One of the most important factors of asymmetric keys is that one cannot obtain the private key by knowing the public key and vice versa but one of the keys can validate the other one.

A public key cryptosystem uses the concept that one key is used for encryption while the other one is used for decryption. If the private key is used to encrypt a message this proves the creator of the message. Encrypting using the public key is used when the sender only wants a certain entity to be able to decrypt that message. If Alice and Bob are exchanging messages using this method it's possible to create a message that can only be sent by Alice and can only be received by Bob, to create this message Alice has to first encrypt the message with her private key and then use Bob's public key to encrypt the message again, since Bob has access to both of the corresponding asymmetric keys he can decrypt the message to it's original form.

2.3.2 Digital Signatures

Digital Signatures are a method to prove the integrity and authenticity of data, it also provides Identification and Non-repudiation (only the signing entity could create that signature). The signing entity has a message and it uses their private key to calculate the digital signature of their message, with the asymmetric authentication mechanism of an asymmetric key pair, public/private key, each entity can verify the signature of the message using the public key [11].

2.3.3 Certificates

A valid certificate is a proof of the authenticity of a public key. When a asymmetric key pair is first created the certificate created is called a self-signed certificate this can be used if the parties establish a Direct Trust relationship, in 2.3.4, however this method just like SKE cannot scale at large because it would mean every single communication party had to send the self-signed certificate and it had to be trusted prior to the start of the secure communication.

Non self-signed Certificates are used to bind public keys to an entity by a trusted third-party. This is usually done by a Certificate Authority (CA) or one of the trusted appointees of a CA, if the party trusts the CA the authenticity of the public key will be verified.

The certificates used in this project are X.509 Certificates specified in the X.509 specification [12] of the telecommunication standardization sector of the International Telecommunication Union (ITU-T).

2.3.4 Trust Models

There are several models to establish trust between two parties, this establishment is necessary for the recipients to trust the authenticity of a public key.

Direct Trust is the most basic form of trust, it involves the recipient to trust the authenticity of the public key directly from the key owner or the key owner directly confirms its authenticity. While reliable, given the proper trust between both parties involved, it isn't useful at a large scale. Direct trust is used to initialize a relationship of trust to build other trust models for example the root certificate of a Certificate Authority (CA) is a self-signed certificate, so the recipient must trust the CA to trust it's root certificate.

Web of Trust was invented by Phil Zimmermann [13] which relies on either direct trust or in a recommendation system, several trusted members recommend trusting the aforementioned public key. It is important to note that only trusted members, members are users whom the recipient trusts to sign other keys, signatures are accounted in the recommendation system and not all users who are trusted by the recipient (recipient trusts their public key).

Hierarchical Trust solves an issue of Web of Trust, the lack of liability/responsibility of the entity that validates the authenticity of a public key. In this trust method the signers of public keys are liable for the authenticity of the public keys they sign, for this reason the trust of a public key is directly related to the trust of the Certificate Authority (CA). A CA is an authority that serves as the root certificate of a certificate chain, a finite sequence of signed certificates each certificate signed by the entity above in it's chain, this entity is also called a trust anchor. To note that the entity's certificate is only trusted when I also trust the CA that signed it's certificate (or the one at the root of it's certificate path).

Entities that are a part of different PKI cannot verify the validity of each's certificates so there are various methods to combine PKI:

- **Trusted Lists** - a list of trust anchors (CA);

- **Common Root** - a CA that signs each root of each PKI;
- **Cross-Certification** - each CA signs the other's public key;

2.4 Security Assertion Markup Language (SAML)

Security Assertion Markup Language (SAML) is an open standard for exchanging authentication and authorization data between entities. It is a XML-based markup language used to achieve SSO in a system. Both systems used in this thesis use this standard, the eIDAS Network is addressed in chapter 3.2 and University of Lisbon's (ULisboa) is in chapter 3.3

The standards SAML is built upon are:

- eXtensible Markup Language (XML);
- XML Schema (XSD);
- XML Signature;
- XML Encryption;
- Hypertext Transfer Protocol (HTTP);
- Simple Object Access Protocol (SOAP).

The three roles the SAML specification defines are: the principal (or subject) an entity whose identity can be authenticated, commonly a real user, the Identity Provider (IdP), and the Service Provider (SP). The subject aims to obtain access to a service or resource provided by a SP. The SP does not have the means to authenticate the subject so in order to authenticate it the SP triggers a SAML authentication process, detailed in chapter 2.4.5.

The SAML specifications are based around the notion of Assertions, protocols, bindings and profiles.

A SAML Assertion is an XML expression encoding one or more statements about a subject (typically a real user). There exist three types of SAML Assertions: Authentication, Attribute, and Authorization [14]. An authentication Assertion communicates that a subject was authenticated by the IdP at a certain time, while an Attribute Assertion provides attributes associated with the subject to the SP[14].

How and which Assertions are requested is defined by the SAML Protocols, which have their own XSD. The lower-level communication or messaging protocols (such as HTTP or SOAP) that the SAML protocols can be transported over are defined by Bindings. Lastly, SAML Profiles define the use of Assertions, protocols and bindings to achieve certain requirements[15].

The profile used in both eIDAS and ULisboa's implementation is the SP initiated HTTP POST-POST binding.

2.4.1 Identity Provider (IdP)

An Identity Provider (IdP) is an entity that manages and maintains the identity information of principals[16], as such it can be referred to as an Authentication Service (AS) because it provides authentication services to the SSO system it's included in.

A SAML IdP, also known as a SAML Authority, provides an authentication service by verifying the identity of a subject and issuing a SAML Authentication Response with the corresponding Authentication Assertion, usually alongside an Attribute Assertion, within a SAML SSO system.

2.4.2 Service Provider (SP)

A SAML Service Provider (SP), also known as SAML Relying Party, is an entity that provides services, either to a principal or to other system entities[16]. An SP in a SSO system does not have an authentication mechanism to verify identities of principals so it must request the verification of the subjects to an IdP by dispatching a SAML Authentication Request.

The SP can completely rely on the Identity Provider's (IdP) Authentication Assertion embedded in the SAML Authentication Response or it can request attributes which will be delivered in an Attribute Assertion. It can use these attributes to decide if the subject can have access to the services requested, e.g role-based access control, this can also be accomplished using an Authorization Assertion.

2.4.3 Attribute Provider (AP)

An Attribute Provider (AP) is an entity that stores and manages the attributes of a principal. A SAML AP, also known as a SAML Attribute Authority, is one that can provide the attributes related to the principals through the SAML standard.

While most IdPs are APs the opposite is not true. An IdP can issue Authorization Assertions and Attribute Assertions, since it needs to have the principal's attributes to verify his identity, an AP cannot provide Authentication Assertions because it does not have a mechanism to verify a principal thus must rely on an Authentication Assertion of an IdP to authenticate the subject. The AP receives the identity of the subject and issues an Attribute Assertion with the requested attributes pertaining to.

Attribute Aggregator (AG)

An Attribute Aggregator (AG) can be described as an AP, however an AG does not store or manage the attributes of principals, it's an entity that acts as a proxy for several APs. It receives an Authentication Assertion with the identity of the principal and a Request with an attribute set. An AG can retrieve attributes of the same principal from different APs, e.g an University's AG requesting the courses a student has been enrolled in from each faculty. An AG can choose which APs he issues the request to, depending on the original requested attribute set, as can be seen in the portuguese platform iAP, described in chapter 3.3.2.

2.4.4 SAML Metadata

SAML Metadata is the configuration necessary to negotiate agreements between entities and establish trust relationships in a SAML SSO system, defined by an XML Schema (XSD). It describes the set of identifiers, bindings, endpoints, certificates, keys, cryptographic capabilities, security policies supported by each principal[17].

SAML metadata is organized around an extensible collection of roles an entity can offer, each role is described by an element. The element that describes one single SAML entity is <EntityDescriptor>, the entity may act in many different roles (e.g IdP and AP) to support multiple SAML profiles. The root element of an entity, <EntityDescriptor>, must have an entityID, an unique identifier, and one or more elements extended from the abstract element <RoleDescriptor>, that describe a role it can perform[17].

The elements that extend <RoleDescriptor> are:

- <IDPSSODescriptor> for Identity Providers (IdPs);
- <SPSSODescriptor> for Service Providers (SPs);
- <AttributeAuthorityDescriptor> for Attribute Providers (APs);
- <AuthnAuthorityDescriptor>;
- <PDPDescriptor>.

The most common optional elements used within the root element are: <ds:Signature>, <Extensions>, <Organization>, and <ContactPerson>. These elements either provide more security, versatility, or information.

The trust relationship created between two SAML entities relies on a Public Key Infrastructure (PKI), the information about one cryptographic key is present in the element <ds:KeyInfo>. This element can be present in the <ds:Signature> element of the <EntityDescriptor> or in <KeyDescriptor> elements. Each <RoleDescriptor>, IdP or SP, contains one or more <KeyDescriptor> elements, these contain the information about the key and it's intended use (signing or encryption).

The element <IDPSSODescriptor> contains the elements and attributes which are crucial to perform the function of an IdP.The most important of these elements is <SingleSignOnService> which describes the supported profiles and the endpoints for them and <saml2:Attribute> the attributes supported by the IdP.

The most important elements of <SPSSODescriptor> is the <AssertionConsumerService> which describes the supported profiles and the endpoints for them.

2.4.5 Service Provider (SP) Initiated POST-POST Binding

Service Provider (SP) Initiated POST-POST Binding is one of the most common used bindings, it starts when a subject is trying to access protected resources on a SP, the access to these resources requires the authentication and thus authorization of the user. The SP requests the authentication of the principal to an IdP so that it may provide SAML Assertion(s) in order to validate whether they have access rights to the resource[15]

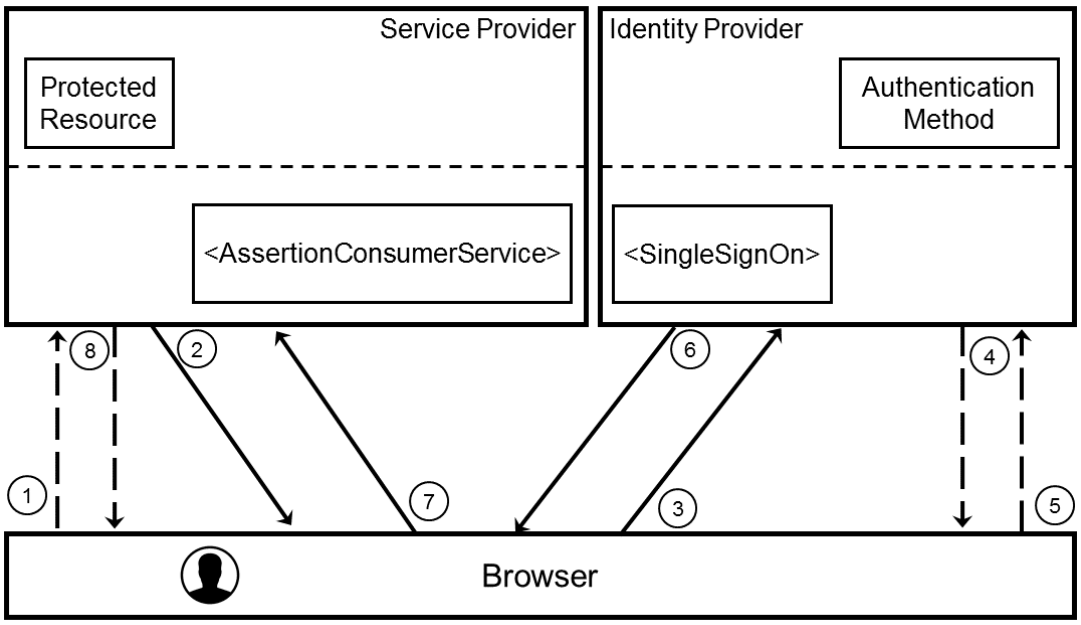

Figure 2.1: SAML Authentication Process using POST->POST Profile

The process is as follows:

1. The principal attempts to access a resource on the Service Provider (SP), it doesn't provide the resource because he doesn't have an authenticated session, the SP initiates the SAML authentication process;
2. The SP sends a Hypertext Markup Language (HTML) form back to the browser. The HTML form has a SAML Authentication Request encoded in base 64, with the root element being <AuthnRequest>, and a Relay State. The Authentication Request defines which authentication; Authorization or attributes are required. The Relay State is an opaque identifier used to verify the integrity of the message.
3. The browser issues a HTTP POST containing the SAML Authentication Request and the Relay State to the Identity Provider's (IdP) Single Sign-On (SSO) endpoint;
4. If the subject has not been authenticated by the IdP, the principal will be challenged to provide valid credentials;
5. The subject provides valid credentials, authenticating at the IdP;

6. A HTML form is created and sent back to the browser with the SAML Authentication Response (and SAML Assertions requested), encoded once again in base 64, and the Relay State. The SAML Authentication Response must be digitally signed. Typically the HTML form is automatically triggered by the IdP which will result in a HTTP POST;

7. The browser issues a HTTP POST containing the SAML Authentication Response and the Relay State to the Service Provider's (SP) AssertionConsumerService endpoint;

8. The SP validates the signature and Assertions in the SAML Response and sends a HTTP redirect to the browser so the subject can access the target resource.

The SAML specification only requires the Authentication Response to be be signed however its recommend and usual for the Authentication Request to also be signed, so the IdP must validate the Request's signature before proceeding. If both entities support and higher security and it is configured the SAML Assertions sent within the Response can be encrypted.

2.5 Captive Portal

A Captive Portal is a web page found in public-access networks, it is used to grant access to an user. Captive portals are typically used in airports, hotels and coffee shops. They are used instead of open WiFi hotspots and one-password-for-all approaches because it allows a more detailed monitorization of the network and it's users, in the case of hotels by only allowing paying customers access to the network.

When a user tries to access a web page using a network with a Captive Portal he is immediately redirect to the Captive Portal's web page where an authentication method is required. After the user successfully authenticates he will be granted access to broader network resources which include Internet access.

While Captive Portals are usually used to obtain WiFi access by connecting to a WLAN, in the example shown in 2.2 the user's computer is physically connected to the network using a cable. The biggest difference is the existence of an Access Point (AcP), acting as a virtual cable, which the user would be connected to wirelessly.

When the user first accesses the network (either by connecting to the AcP or by physically connecting an ethernet cable) he will be assigned an Internet Protocol Address (IP) in a Local Area Network (LAN). The gateway network is responsible for routing the traffic from all computers in the internal network to external networks, in a network with a Captive Portal enabled the gateway will not allow an user to access external resources without being authenticated.

Figure 2.2 depicts the flow a user must complete before he is granted broader access to network resources. When a user tries to access an external resource without being authenticated (1) the gateway will block the request and redirect the user to the Captive Portal (2). The Captive Portal is a web page that

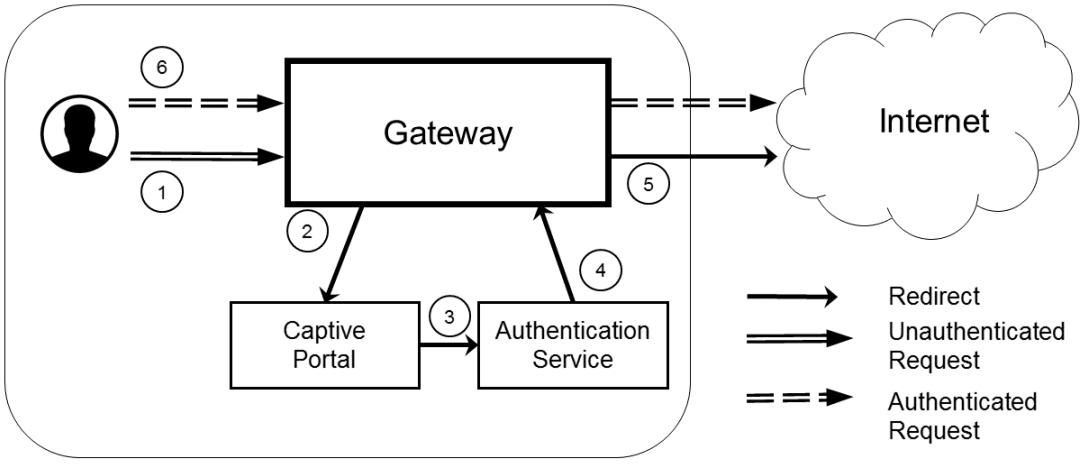

Figure 2.2: Example of a System with an integrated Captive Portal and Authentication Service (AS)

will provide a method for the user to authenticate, either the Captive Portal can act as an Authentication Service (AS), i.e by having a database with credentials, or it allows the redirection to another AS (3). An AS can be a Lightweight Directory Access Protocol (LDAP) server, a Remote Authentication Dial-In User Service (RADIUS) server, a Kerberos server or a SAML IdP, the important thing about an AS is that it has the mechanisms to verify the identity of an user. After the user completes the authentication process the AS will redirect to the gateway (4), the gateway will grant the user access to external resources and will do one final redirect to the user's original requested Uniform Resource Locator (URL) (5).

After the successful authentication of the user, subsequent requests (6) will not be redirected to the Captive Portal or the AS by the gateway.

3 State of the Art

3.1 ULisboa Identity Approach

The objective of the Identity Management system at ULisboa is to be able to integrate several systems by using only one identity from an user. This identity should be obtainable from any of the organizations belonging to ULisboa and should grant the user access to ULisboa's services.

3.1.1 Identity Model

University of Lisbon (ULisboa)'s main identity management system is NetIQ's Identity Manager (IDM), this product includes two applications: iManager and eDirectory. eDirectory is a Lightweight Directory Access Protocol (LDAP) directory it organizes it's objects into a tree. iManager is a web administration console that provides access to eDirectory's obejcts so an administrator can manage and configure them.

An user in ULisboa can have many identities, for example I am both a student at Instituto Superior Técnico (IST) and, during the development of this thesis, an employee at Rectory of ULisboa. This is possible because the university is composed of several schools, each school has it's own identity for it's users. The identity of an user in a certain organization for a certain role is a "profile" in IDM, such as a student and professor profile, the combination of all the profiles of an user in a school makes up the user's identity in that school. An user can have a student profile at IST and a professor profile at Faculdade de Ciências da Universidade de Lisboa (FCUL), or both profiles at the same school (i.e. a doctorate student). The combination of all identities from all schools makes up the master identity which is managed by ULisboa's IDM.

There is a very important distinction between the master identity and the "profiles" (or partial identities), the master identity does not have any specific information about an user like their academic record, the master identity is more of an aggregator of all of the different profiles of an user with the specific profiles containing the information of that user in that organization. For example my master identity does not have my student number (75656), neither does my profile from the Rectory but my profile in IST knows my student number.

To attach a profile to an already existing identity the IDM has to match the user's profile to the user's identity, this is accomplished by comparing attributes using one of the following matching rules:

- **ULBI** - using the national citizen's number of the user;
- **Given Name + Surname + ULBirthData** - using personal information of the user.

The first rule is used for national citizens and the second rule is used for non-portuguese citizens. The matching of an user's master identity with a it's identity from a school uses the same matching rules.

3.1.2 Authentication Model

After the proper creation of the master identity in the IDM the user should be able to authenticate in any Identity Provider (IdP) from ULisboa's schools as long as he has the proper credentials. Any school that has an identity management system also has an IdP, such as Instituto Superior Técnico (IST), Faculdade de Ciências da Universidade de Lisboa (FCUL), Faculdade de Motricidade Humana (FMH) or Instituto Superior de Agronomia (ISA).

Each profile, the identity of an user in one organization, and the master identity has an unique identifier. Most schools and ULisboa have opted to use the email generated by each school as an user's unique identifier. The unique identifiers are: for a IST stundent an email ending in "@tecnico.ulisboa.pt", for a FCUL student an email ending in "@fcul.ulisboa.pt" and the user's master identity identifier is an email ending in "@campus.ul.pt" or "@edu.ulisboa.pt". When the master identity is populated with a profile the unique identifier from that profile is set in an attribute in the user's identity object in eDirectory,

The authentication of an user is managed by NetIQ's Access Manager (AM) which is directly connected to the IDM's eDirectory. AM connects to the Service Providers (SPs) from ULisboa and the IdP from each school using SAML, particularly the SP initiated POST-POST Binding protocol see 2.4.5. Figure 3.1 describes an authentication process for an ULisboa user.

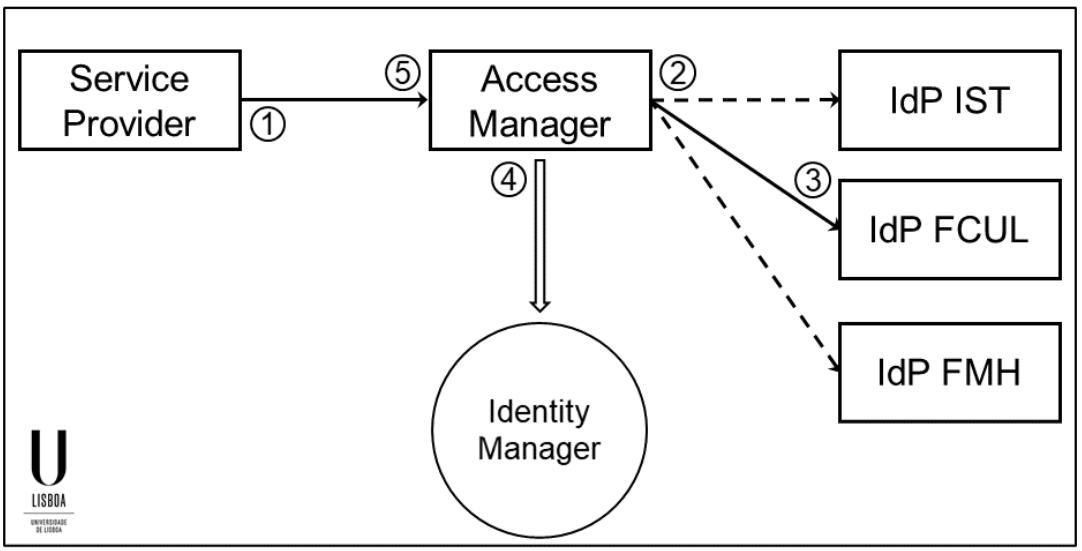

Figure 3.1: ULisboa Authentication Process

The process for the authentication of an user in one of ULisboa's services is the following:

1. The SP sends an Authentication Request to AM;
2. The user chooses a school IdP to authenticate in (i.e. FCUL), figure 3.2, and then AM sends an Authentication Request to that IdP;

Figure 3.2: NetIQ's Access Manager (AM) authentication page

3. At the IdP the user uses his credentials and is authenticated, the IdP sends an Authentication Response with the unique identifier of the user to Access Management (AM);
4. AM receives the Authentication Response with the user's unique identifier for that school, extracts the attributes of the user and connects to the eDirectory from the IDM. If the attribute received corresponds to one of the school's unique identifiers belonging to a master identity the user is authenticated;
5. AM sends an Authentication Response with the unique identifier of the master identity and some personal data to the SP.

3.2 electronic IDentification, Authentication and trust Services (eIDAS)

The electronic IDentification, Authentication and trust Services (eIDAS) Regulation, also known as the EU Regulation 910/2014, enables each of the Member States (MSs) to recognize the electronic identities of citizens from other EU MS, as long as that MS has notified it's Electronic Identity (eID)

scheme. The eIDAS Regulamentation was inspired by previous European projects, particularly Secure idenTity acrOss boRders linKed (STORK) [18] and it's successor STORK 2.0 [19] from which it brought many insights, some regarding usability and user privacy. With the recognition of other MS eID's mobility across EU will improve for European citizens and they will also have access to more services in a more secure way.

One of the biggest issues with the implementation of the regulation is the take-off, the Connecting Europe Facility (CEF) program has incentives for the use of nationally issued eIDs for cross-border authentication and also the integration of eIDAS in existing services, however the lack of early adopters means services are less likely to integrate with eIDAS while the lack of services using it means users are less likely to adopt it, after the take-off is complete both users and services will continue to grow exponentially.

3.2.1 eIDAS Infrastructure

The eIDAS Infrastructure is composed of proxies called eIDAS Nodes. There is a single eIDAS Node per country, except for Germany and Austria, and every MS is responsible for their own eIDAS Node [20]. Due to Germany and Austria's legislation eIDAS must support two different authentication models.

In the proxy model each eIDAS Node is composed of two modules the eIDAS Service and the eIDAS Connector. The eIDAS Connector connects to the national SPs, this module is the same for all eIDAS Nodes in the eIDAS Network. The eIDAS Service connects to the national infrastructure of the MS, therefore this module has to be altered due to the different specifications of the implemented eID schemes, for example Portugal has only one IdP while Italy has several IdPs so they had to create an IdP Proxy see 3.2.2.

The middleware model relies on a country-specific Middleware (MW) present in every eIDAS Node. Instead of using the eIDAS Connector - eIDAS Service connection Germany's citizens use the MW present in the eIDAS Connector which interacts with a citizen's eID token (smart card) providing the authentication necessary for the SP.

In order to provide cross-border authentication, eIDAS Nodes communicate via the eIDAS communication protocol [21], this protocol is based on SAML [22].

3.2.2 eIDAS Authentication Flow

In this chapter it will be shown an example of an authentication process using the eIDAS Network. In this example an Italian Citizen wants access to a protected resource from a Portuguese Service, so this Italian user has to authenticate at an Italian IdP. For reasons of clarity the example the SP is just a web service with an eIDAS login option. Figure 3.3 depicts the example flow of an Authentication Request.

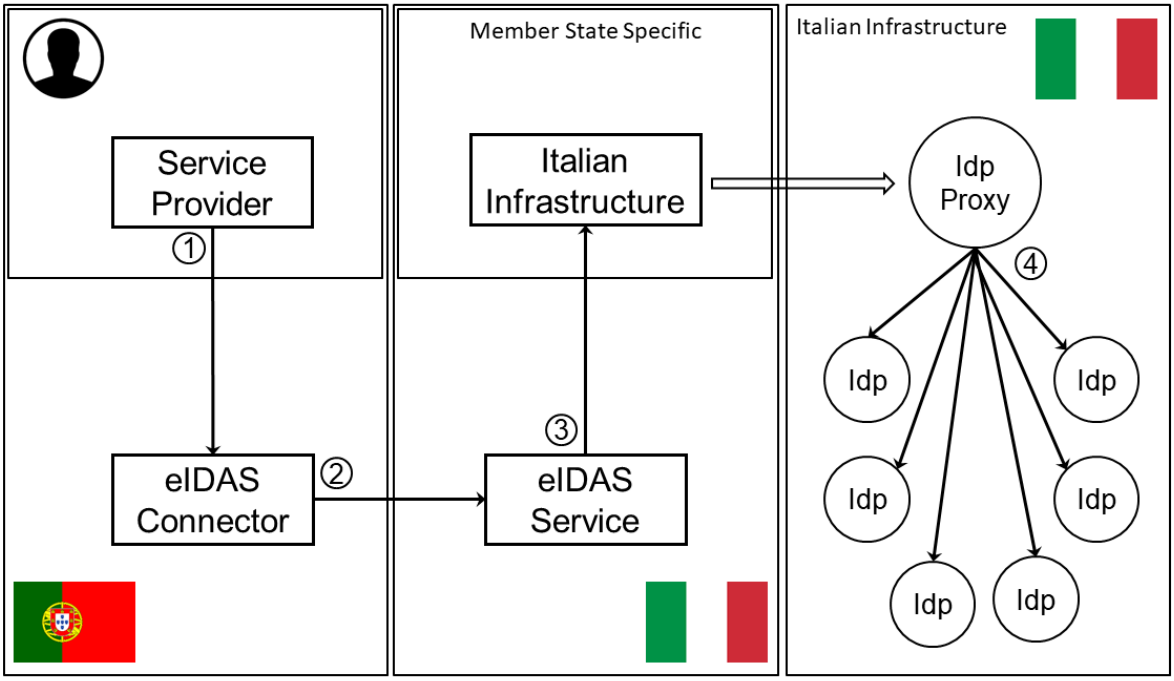

Figure 3.3: eIDAS Authentication Flow

1. The Service Provider (SP) creates a eIDAS compliant SAML Authentication Request sends it to the Portuguese eIDAS Node, requesting the Minimum Data Set (MDS);
2. The Portuguese eIDAS Node sends a SAML Authentication Request to the Italian eIDAS Node;
3. The Italian eIDAS Node sends a SAML Authentication Request to the Italian Infrastructure;
4. Due to Italian's eID infrastructure another step is required, the Authentication Request from step 3 is received by an IdP Proxy, this proxy allows the Italian citizen to choose from a variety of IdPs available to him. The proxy sends an Authentication Request to the IdP selected.
5. The user is now at the IdP where he can use his credentials to authenticate.

The steps 1,2,3 must use the SP Initiated POST-POST Binding of the SAML specification, described in chapter 2.4.5, this is because the 1st step needs to also send a "CountryCode" attribute in the POST message, the "CountryCode" determines the eIDAS Node the eIDAS Service will send the Authentication Request to. The fourth step is MS specific but in the case of Italy it is also a SP Initiated POST-POST Binding started by their IdP Proxy, in the case of Portugal there is no step 4, eIDAS connects directly to the only national IdP.

During this process 4 Authentication Requests were created, if any of the Authentication Responses related to each of these requests is incorrect or throws an error the entire process ends in failure and the user cannot login to the SP. The process after successful authentication at one of the Italian IdPs is the following:

4. The Italian IdP creates a SAML Authentication Response and sends it to the Italian IdP proxy;

3. The IdP Proxy evaluates the response and then with the attributes received creates a eIDAS compliant SAML Authentication Response to send to the Italian eIDAS Node;

2. The Italian eIDAS Node checks the response, transforming the attributes received as they need, and then creates an encrypted SAML Authentication Response for the Portuguese eIDAS Node;

1. The Portuguese eIDAS Node evaluates the response but does not evaluate the attributes received and then sends the final SAML Authentication Response to the SP.

0. Finally the SP evaluates both the response and the attributes retrieved. The SP uses the attributes requested to verify if the user already has an account and if he has permission to access the requested resource.

3.2.3 Attributes

The eIDAS specification defines a set of minimum attributes for a natural person, representative natural person, legal person and representative legal person. Due to the nature of this thesis only the natural person attributes are relevant even though each Member State (MS) must recognize all Minimum Data Sets (MDSs). This set of attributes is defined as the Minimum Data Set (MDS), used by SPs to authenticate users. Table 3.1 presents the natural person MDS.

Friendly Name	Uniform Resource Identifier (URI)
FirstName	http://eidas.europa.eu/attributes/naturalperson/CurrentFamilyName
FamilyName	http://eidas.europa.eu/attributes/naturalperson/CurrentGivenName
DateOfBirth	http://eidas.europa.eu/attributes/naturalperson/DateOfBirth
PersonIdentifier	http://eidas.europa.eu/attributes/naturalperson/PersonIdentifier

Table 3.1: eIDAS Minimum Data Set (MDS)

The MDS has one unique identifier, PersonIdentifier, this attribute can be used to uniquely identify one eIDAS subject however it might not be possible to use this identifier to link to a natural person. Each MS defines how their PersonIdentifier attribute will be created, in some cases[23] one person might have several PersonIdentifier attributes, however one PersonIdentifier attribute is associated with only one person. In these cases, the SPs can use a combination of other attributes to verify the identity of a user.

Due to the infant stage of the eIDAS regulation adoption, MSs have only fully integrated their authentication infrastructure, namely the IdP, while most MS have not integrated SPs and their national APs. Therefore sector specific attributes (from academia, eHealth, eJustice) are still not supported and thus services connected to the eIDAS infrastructure cannot rely on sector specific attributes[4]. The lack of support for new attributes is the biggest limiting factor in the use cases of the eIDAS infrastructure.

3.3 Portuguese Infrastructure

The portuguese infrastructure is divided between two systems that connect with one another, the portuguese Identity Provider (IdP) Fornecedor de Autenticação (FA) and the portuguese Attribute Aggregator (AG) Interoperabilidade na Administração Pública (iAP). The Portuguese Infrastructure can be seen in figure 3.4.

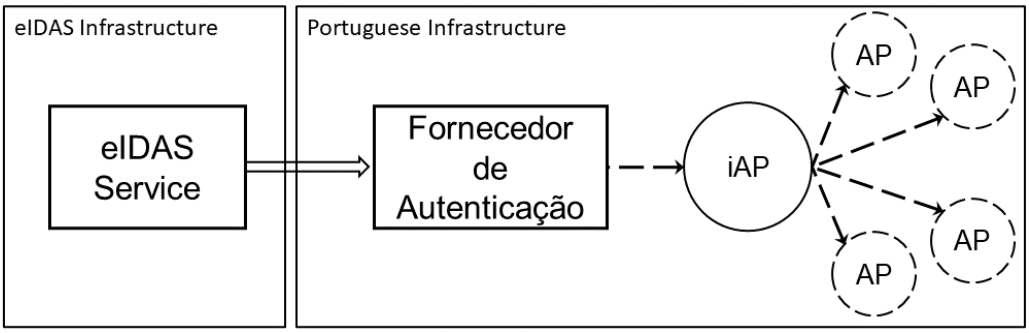

Figure 3.4: Portuguese Infrastructure

3.3.1 Fornecedor de Autenticação (FA)

Fornecedor de Autenticação (FA) is the national portuguese Identity Provider (IdP), which authenticates by using the Portuguese Electronic Identity (eID). It can authenticate a portuguese citizen using several authentication methods [24]. These methods are present in FA's login page in the following order:

- **Citizen Card** - the most secure authentication method based on the physical cryptographic token that is the Portuguese Citizen Card, users must use both the card and a PIN;
- **Chave Móvel Digital (CMD)** - An authentication method associated with a mobile phone number and/or an email to: citizen identification number (NIC) for a portuguese citizen; passport number for a foreign citizen;
- **Simple Authentication** - User/Password token;
- **Social Media** - Such as Facebook, Twitter or E-mail.

FA is also responsible for managing and providing several types of attributes, if these attributes are not present in the chip of the citizen card these attributes will be requested to the appropriate APs through iAP, explained in chapter 3.3.2.

Every attribute has a trust level, exactly like in eIDAS, the attributes can only be provided if the authentication method has the same or a higher trust level. There are 4 trust levels, from 1 to 4, 4 is

the highest [24]. When a request asks for several attributes the global level of trust must be equal to the highest level of each attribute. The trust levels are based on how secure the authentication method must be for these attributes to be retrieved [24]. The trust levels for each authentication method are:

- **Citizen Card** - 4;
- **Chave Móvel Digital (CMD)** - 3 (level 2 if it's through email or twitter);
- **Simple Authentication** - 1;
- **Social Media** - 1.

The model for authentication and attribute exchange with Service Providers (SPs) is based on the standard Security Assertion Markup Language version 2.0 (SAML v2.0)0 [24], the same standard used in ULisboa and eIDAS.

3.3.2 Interoperabilidade na Administração Pública (iAP)

In order for the Fornecedor de Autenticação (FA) to operate with several different types of systems an intermediary application was developed by AMA, Interoperabilidade na Administração Pública (iAP). This application is a proxy that translates and maps attributes from different systems for FA, just like eIDAS does from the national IdPs to the SPs however the method of to accomplish the goal is different. The Attribute Aggregator (AG), iAP, gathers attributes from Attribute Providers (APs) by relaying FA's authentication, using the authentication attributes from the user these APs can obtain the requested attributes matching that user.

To be connected to the iAP an AP must follow certain requirements [25]:

- **Infra-structure**
  - Virtual Private Network (VPN) ipSec between both parties involved;
  - Secure connection using Hypertext Transfer Protocol Secure (HTTPS);
  - Contacts of teams responsible.
- **WebServices**
  - Represented via Web Services Description Language (WSDL) 1.1 (`http://www.w3.org/TR/wsdl`);
  - Binding SOAP 1.1 or 1.2;
  - Async implementation;
  - Correlation of messages using WS-Addressing v1.0

The description regarding the web service of the AP is provided to iAP in the form of a WSDL. The requests of attributes are done using SOAP over HTTPS, SOAP takes care of the message transmission while WS-Adressing defines the headers which defines the endpoints and also a way to correlate asynchronous messages[25].

The relevant attributes of WS-Addressing are:

- **<MessageID>** - Unique Identifier of a message;
- **<RelatesTo>** - Identifies the original message with it's <MessageID>, used to correlate messages asyncronically;
- **<ReplyTo>** - Endpoint for the web service to send the response message;
- **<To>** - Destination endpoint;

The process to reply to messages is the following:

1. iAP sends a request of attributes through a message with <MessageID>;
2. Attribute Provider (AP) receives message, sends an acknowledgement and starts processing the request;
3. Attribute Provider (AP) generates a response, the field <RelatesTo> has the value of <MessageID>;
4. iAP receives the response and sends an acknowledgment.

One of iAP's limitations is that it provides only one value for each attribute type and each iAP type is mapped to one FA attribute. Each iAP attribute is mapped to only one AP so each FA attribute can only be retrieved by one AP. It is not possible to retrieve the same attribute from two different APs, this is an issue in the eID4U project because a student can attend several universities at once and should be able to choose which university he wants to retrieve his information from.

3.4 Zeroshell Linux Router

Zeroshell is an open-source Linux based distribution [26], it aims to provide network services such as router and firewall appliances. Zeroshell is completely administrable through a web interface, while it's still possible to configure it through the shell since its linux based. Zeroshell has been released under GPLv2 and was created by Italian Fulvio Ricciardi. It is available as Live CD, CompactFlash images, and VMware virtual machines.

Zeroshell has a wide set of features, that allows it to perform the duties of a router and a firewall. It also has a special set of features related to the Captive Portal implementation, those features are:

- Captive Portal to authenticate users;
- DHCP server to dynamically assign IPs to clients;
- Shibboleth SAML v2.0 Service Provider (SP) integrated with the Captive Portal;
- Dynamic IdP whitelist;

Zeroshell has the option to perform routing or bridging, in the case of bridging which is what was used the user is on a Virtual Local Area Network (VLAN), when he joins a WLAN through an Access Point (AcP), if Zeroshell acts as the router and default gateway, only routing the traffic from the user if he is already authenticated. In this configuration it is convenient to use the DHCP, so the user can obtain an IP in the VLAN, and Domain Name System (DNS) features provided by Zeroshell.

3.4.1 Zeroshell Captive Portal

The Captive Portal of Zeroshell can use different Authentication Services (ASs), such as an internal LDAP directory, a Kerberos 5 realm, a RADIUS server or a SAML v2.0 IdP (using the Shibboleth Service Provider (SP)). However I will only use the Shibboleth SAML v2.0 SP because it is the only one capable of authenticating using the eIDAS infrastructure which is composed of the eIDAS nodes, SAML v2.0 Identity Providers (IdPs), and Attribute Providers (APs).

Several different Captive Portal applications allow the use of a SAML v2.0 SP to initiate the authentication process, however that is not enough to allow authentication using the eIDAS network because most applications will only allow one redirect or they will have a static whitelist which the administrator must configure. What Zeroshell provides that other applications do not is an automatic dynamic whitelist, referred to as an auto-discovery of the Identity Provider's (IdP) URL. Zeroshell obtains the IdPs' URLs not by the metadata of the service but by interpreting the Service Provider's (SP) redirects, this in turn also adds the national eIDAS nodes and IdPs of each participating country to the whitelist.

In chapter 2.5 I explain how a captive portal works and show a basic example of a Captive Portal with an AS. Figure 3.5 depicts the graphic representation of Zeroshell's Captive Portal using a SAML v2.0 authentication mechanism. In SAML terms an Authentication Service (AS) is referred to as an Identity Provider (IdP), the service that can authenticate an user, and the service that makes the authentication request is referred to as an SP, Zeroshell provides an incorporated Service Provider (SP) that can make authentication requests to IdPs that are not part of the internal network.

The major difference between Zeroshell's authentication process and the basic example described in 2.5 is the addition of the external AS which is a SAML v2.0 IdP. So the flow of the authentication process is similar in the first three stages (1), (2), and (3) with the particularity that the captive portal and the SP are integrated with Zeroshell, while in the basic example the captive portal and the AS are separate entities.

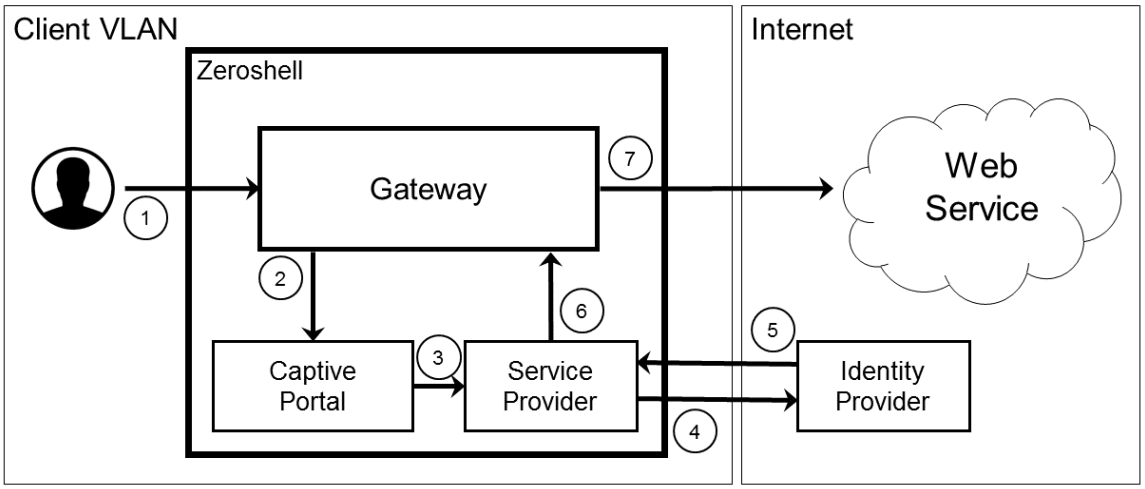

Figure 3.5: Authentication Flow in a Zeroshell system using it's Shibboleth Service Provider and an external Identity Provider

The client will make a request to access an external web service which will be promptly blocked (1) and redirected by the gateway to the captive portal (2) which in turn will redirect to the SP that will act as a proxy to obtain authentication at the AS. Depending on the setup of the captive portal and the SP these redirects might not be visible by the user, Zeroshell has the option to automatically redirect without showing the captive portal's web page. The communication between the SP and the IdP is based on the SSO principle and the SAML standards explained in chapters 2.2 and 2.4 respectively. Based on these standards the integrated Shibboleth SP will send a SAML Authentication Request and redirect to the IdP (4) and after successful authentication of the user he will be redirect to the SP (5) with a SAML Authentication Response containing the data of the user, this data will be validated in the SP and then it will redirect to the gateway with the authentication status (6). If the process is successful the gateway grants access to the user and redirects him to the original URL of the external web service (7).

3.4.2 Zeroshell Limitations

Even though Zeroshell was chosen because of it's integrated SAML v2.0 SP, the implementation has some limitations that had to be mitigated in order to connect Zeroshell with eIDAS infrastructure.

The Shibboleth SP installed is the version 2.4.3 while the latest Shibboleth version, at the time of writing, is 3.0.4[27]. Zeroshell's SAML module is composed by the Shibboleth SP and four libraries:

- log4shib version 1.0.4;
- opensaml version 2.4.3;
- xml-security-c version 1.6.1;
- xmltooling version 1.4.2;

Due to the libraries used, particularly xml-security-c, some encryption algorithms such as "gcm" and some SAML v2.0 features are not provided. Without these missing features it is not possible to connect to IdPs with higher standards of security or the production and pre-production eIDAS Connectors of each country.

A possible solution to the outdated libraries and outdated Shibboleth SP would be to update the module with the latest versions of the libraries, or versions that support the needed signing algorithms, encryption algorithms, and SAML v2.0 features. However the upgrade to the latest versions is not feasible due to a lack of documentation and because it would not solve the other fundamental issue.

The other fundamental limitation of Zeroshell's SAML v2.0 SP implementation is the inability to provide the metadata, there is no way to obtain the Service Provider's (SP) metadata dynamically by an IdP. Without the metadata of the SP the IdP will not be able to verify the signature of the Authentication Request and it won't be able to decrypt it either because it does not have access to the signing or the encryption certificates of the SP. It also cannot obtain the Identity Provider's (IdP) metadata dynamically, so the metadata of the IdP must be imported to Zeroshell as a xml file. The IdP must also import the metadata of the Zeroshell SP, so the SP metadata file must be imported by our country partner into the eIDAS Connector everytime the metadata changes, which is not an option.

4 eID4U

With the application of the electronic IDentification, Authentication and trust Services (eIDAS) Regulation, also known as the EU Regulation 910/2014 for electronic transactions in the internal market, each European Union (EU) Member State (MS) is obligated to recognize the electronic identities of other EU MSs. This should allow a user from any MS to access a service in any MS using the electronic authentication mechanism of their country. The eIDAS Regulation was based on the knowledge acquired from the successful Electronic Identity (eID) large scale projects such as Secure idenTity acrOss boRders linKed (STORK) and Secure idenTity acrOss boRders linKed 2.0 (STORK 2.0).

The eID for University (eID4U) is a project that builds upon the eIDAS Regulamentation, enhancing and adding to its services. The goal of the project is to implement three eIDAS-enabled academic e-services. eRegistration, eLogin and eAccess[4]. To implement these new services a set of new attributes (personal and academic attributes) need to be created and implemented.

The chapter is structure as follows: Section 4.2 describes what is a eID4U Node; Section 4.3 shows the portuguese eIDAS Node and the effort required to update it to an eID4U Node; Section 4.4 gives a detailed overview about the service eRegistration; Section 4.5 gives a detailed overview about the eLogin service; Section 4.5.3 goes into detail about ULisboa eIDAS Proxy (ULEP) an application used in eLogin and eAccess; Section 4.6 gives a detailed overview about the service eAccess.

4.1 Portuguese Architecture

There are two situations possible when using eIDAS, the incoming process where an user can try to access a system and use the eIDAS Network to authenticate and retrieve his attributes and the outgoing model where eIDAS requests the information about an user which the system has to provide.

Users will encounter two situations when using the eIDAS Network: incoming process and outgoing process. The incoming process is when the user wants access to a service and uses the eIDAS Network to authenticate and provide attributes. The outgoing process is the process of an AP providing the attributes requested during an eIDAS authentication process.

The example in figure 4.1 represents how a Portuguese citizen would use the eIDAS Network to obtain academic attributes.

To start an outgoing process a Portuguese citizen wants to retrieve attributes from it's Portuguese University so he accesses an Italian Service and uses it to connect to the eIDAS Network:

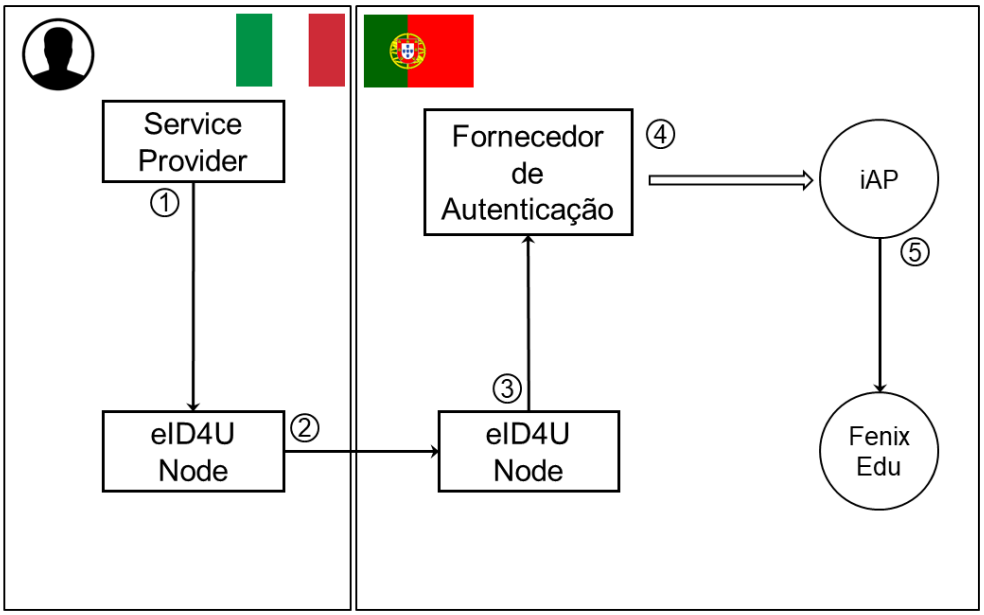

Figure 4.1: Outgoing eIDAS Model

1. To authenticate the user the SP sends an Authentication Request to the eIDAS Node of it's country, requesting eID4U attributes (academic attributes);
2. The Italian eIDAS Node sends an Authentication Request to the Portuguese eIDAS Node;
3. The Portuguese eIDAS Node sends an Authentication Request to the Portuguese IdP, Fornecedor de Autenticação (FA).
4. At the Portuguese IdP, FA, the user has to authenticate or the process will fail. After a successful authentication FA communicates with iAP because it cannot provide all attributes to eIDAS.
5. The iAP sends a message requesting the missing attributes to the AP capable of providing academic attributes ULisboa's academic system, FenixEdu.

The outgoing process is as follows:

5. FenixEdu retrieves the academic attributes and responds to the message from iAP;
4. iAP sends the attributes received to FA;
3. FA joins the authentication attributes and the academic attributes and responds to with an Authentication Response to the Portuguese eIDAS Node.
2. The Portuguese eIDAS Node sends an Authentication Response to the Italian eIDAS Node, with all the attributes provided.
1. The Italian eIDAS Node sends an Authentication Response to the SP giving the user's authentication. The SP grants access to the service the user requested.

The next example, shown in figure 4.2 shows an incoming process where a foreign user wants to get access to a Portuguese service, in this case FenixEdu the academic system of ULisboa.

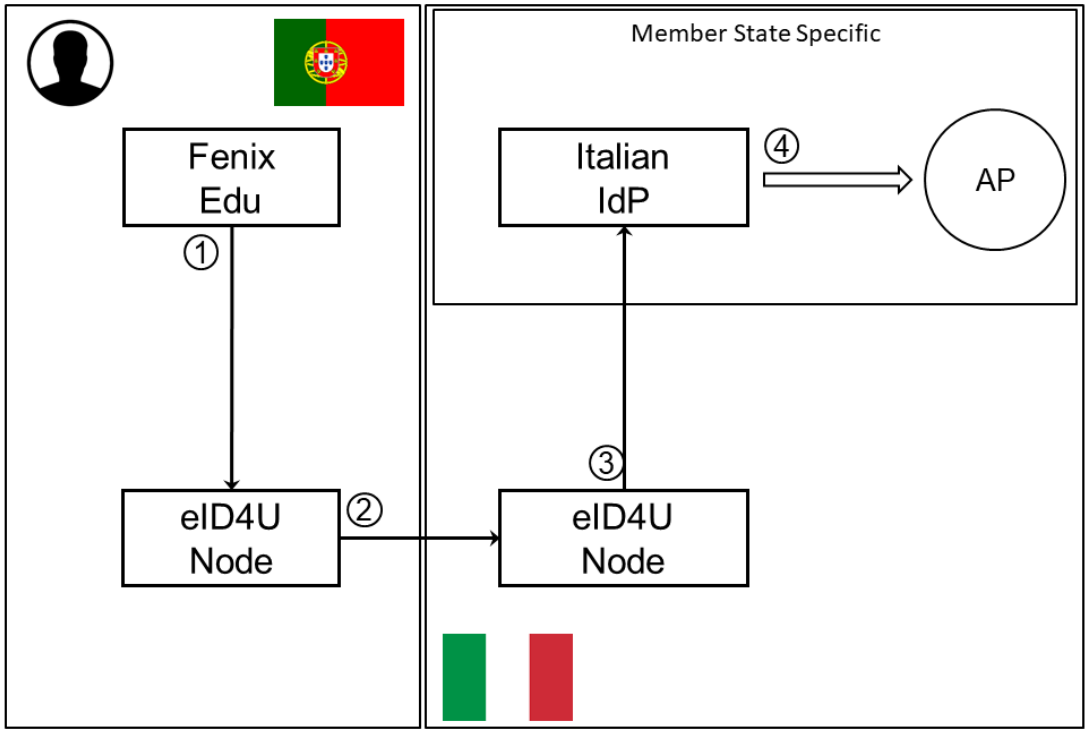

Figure 4.2: Incoming eIDAS Model

1. To authenticate the user FenixEdu sends an Authentication Request to the Portuguese eIDAS Node, requesting eID4U attributes (academic attributes);
2. The Portuguese eIDAS Node sends an Authentication Request to the Italian eIDAS Node;
3. The Italian eIDAS Node sends an Authentication Request to the Italian IdP.
4. The Italian citizen authenticates at the IdP and then he has to retrieve the academic attributes from an Attribute Provider (AP). This step is Member State (MS) specific, it's different for every MS and it depends on the MS eID infrastructure.

After the Request is received the opposite flow must happen where the Authentication Response is sent to each of the entities:

4. The AP sends the academic attributes to the IdP;
3. the Italian IdP sends the academic attributes and the personal attributes to the Italian eIDAS Node via a SAML Authentication Response;
2. The Italian eIDAS Node sends an Authentication Response to the Portuguese eIDAS Node, with all the attributes provided.

1. The Portuguese eIDAS Node sends an Authentication Response to FenixEdu with the personal attributes for authentication and the academic attributes.

4.2 eID4U Node

An eID4U Node is an eIDAS Node which supports the attributes declared in the eID4U project, an improvement of the original eIDAS specification with the objective of allowing new and more diversified services to work with the eIDAS Network. The eID4U and eIDAS nodes can work with each other but only to exchange non eID4U attributes. This chapter shows the different attributes an eIDAS Node needs to support in order to be classified as an eID4U Node and also how to deploy your own Node.

4.2.1 eID4U attributes

The extension of the set of attributes supported by eIDAS Nodes was one of the main goals of this project, in order to enable new services to use the eIDAS Network and to improve the variety of services that can use eIDAS. In this project a standard set of attributes was defined, due to the purpose of each attribute they were divided into two categories:

- **Personal Attributes** - Contain personal and identification information about a person, e.g. name, surname, date of birth and others. These attributes have a broader range of uses, a sector such as eHealth or eCommerce could take advantage of a higher variety of personal attributes.
- **Academic Attributes** - Contain information regarding the academic career of a student. These attributes are specific to the academic sector and it's services. These attributes contain information pertaining to both current and past studies, like the home institution, home institution's country and level of studies.

These categories are defined due to their usability in sector-specific services, personal attributes can be used by any service while academic attributes are only used in academic services. The systems responsible for providing each set of attributes are: the Identity Providers (IdPs) responsible for the personal attributes of the citizens while the universities of the students are responsible for providing the academic attributes.

- The names of the Personal Attributes declared in eID4U is present in table 4.1.
- The names of the Academic Attributes declared in eID4U is present in table 4.2.
- The format of the Personal Attributes declared in eID4U is present in table 4.3.
- The format of the Academic Attributes declared in eID4U is present in table 4.4.

To conclude this chapter table 4.5 has the final status of each attribute declared in eID4U.

eIDAS Friendly Name	eIDAS Attribute Name
Personal Identifier	http://eidas.europa.eu/attributes/naturalperson/PersonIdentifier
Family Name	http://eidas.europa.eu/attributes/naturalperson/CurrentFamilyName
First Name	http://eidas.europa.eu/attributes/naturalperson/CurrentGivenName
Gender	http://eidas.europa.eu/attributes/naturalperson/Gender
ID type	http://eidas.europa.eu/attributes/sectorspecific/eid4u/naturalperson/id/Type
ID number	http://eidas.europa.eu/attributes/sectorspecific/eid4u/naturalperson/id/Number
ID issued by	http://eidas.europa.eu/attributes/sectorspecific/eid4u/naturalperson/id/Issuer
ID expiration date	http://eidas.europa.eu/attributes/sectorspecific/eid4u/naturalperson/id/ExpiryDate
EU health card ID	http://eidas.europa.eu/attributes/sectorspecific/eid4u/naturalperson/EhicId
Nationality	http://eidas.europa.eu/attributes/sectorspecific/eid4u/naturalperson/Nationality
Citizenship	http://eidas.europa.eu/attributes/sectorspecific/eid4u/naturalperson/Citizenship
Marital State	http://eidas.europa.eu/attributes/sectorspecific/eid4u/naturalperson/MaritalState
Country of birth	http://eidas.europa.eu/attributes/sectorspecific/eid4u/naturalperson/CountryOfBirth
City of birth	http://eidas.europa.eu/attributes/naturalperson/PlaceOfBirth
Date of birth	http://eidas.europa.eu/attributes/naturalperson/DateOfBirth
Permanent address	http://eidas.europa.eu/attributes/naturalperson/CurrentAddress
Current photo	http://eidas.europa.eu/attributes/sectorspecific/eid4u/naturalperson/CurrentPhoto
Tax Reference	http://eidas.europa.eu/attributes/naturalperson/TaxReference

Table 4.1: Personal Attributes Names

eIDAS Friendly Name	eIDAS Attribute Name
Temporary address	http://eidas.europa.eu/attributes/sectorspecific/eid4u/naturalperson/TemporaryAddress
Email	http://eidas.europa.eu/attributes/sectorspecific/eid4u/naturalperson/Email
Phone Number	http://eidas.europa.eu/attributes/sectorspecific/eid4u/naturalperson/Phone
Home Institution Name	http://eidas.europa.eu/attributes/sectorspecific/eid4u/studies/homeinstitution/Name
Home Institution Identifier	http://eidas.europa.eu/attributes/sectorspecific/eid4u/studies/homeinstitution/Identifier
Home Institution Country	http://eidas.europa.eu/attributes/sectorspecific/eid4u/studies/homeinsitution/Country
Home Institution address	http://eidas.europa.eu/attributes/sectorspecific/eid4u/studies/homeinstitution/Address
Current level of Study	http://eidas.europa.eu/attributes/sectorspecific/eid4u/studies/CurrentLevelOfStudy
Current field of Study	http://eidas.europa.eu/attributes/sectorspecific/eid4u/studies/FieldOfStudy
Current degree name	http://eidas.europa.eu/attributes/sectorspecific/eid4u/studies/CurrentDegree
Degree	http://eidas.europa.eu/attributes/sectorspecific/eid4u/studies/Degree
Degree Awarding Institution	http://eidas.europa.eu/attributes/sectorspecific/eid4u/studies/DegreeAwardingInstitution
Year of graduation	http://eidas.europa.eu/attributes/sectorspecific/eid4u/studies/GraduationYear
Degree Country	http://eidas.europa.eu/attributes/sectorspecific/eid4u/studies/DegreeCountry
Level of language proficiency	http://eidas.europa.eu/attributes/sectorspecific/eid4u/studies/LanguageProficiency
Language certificates	http://eidas.europa.eu/attributes/sectorspecific/eid4u/studies/LanguageCertificates

Table 4.2: Academic Attributes Names

eIDAS Friendly Name	eIDAS Attribute Name
Personal Identifier	<OriginCountryCode> / <DestinationCountryCode> / <Unique Identifier>
Family Name	Text
First Name	Text
Gender	"M" or "F" or "X"
ID type	"National Identity Card" or "Passport"
ID number	Number
ID issued by	Text
ID expiration date	yyyy-mm-dd
EU health card ID	Number
Nationality	ISO 3166-1 alpha-2 code
Citizenship	ISO 3166-1 alpha-2 code
Marital State	"Single" or "Married" or "Divorced" or "Widowed" or "Separated" or "Civil Union"
Country of birth	ISO 3166-1 alpha-2 code
City of birth	Text
Date of birth	yyyy-mm-dd
Permanent address	PostalAddress as per CORA ISA Vocabulary v0.3
Current photo	Base64 document and file name
Tax Reference	TIN<CountryCode>- <Number>

Table 4.3: Personal Attributes Format

eIDAS Friendly Name	eIDAS Attribute Name
Temporary address	PostalAddress as per CORA ISA Vocabulary v0.3
Email	Email
Phone Number	Phone
Home Institution Name	Text
Home Institution Identifier	Number
Home Institution Country	ISO 3166-1 alpha-2 code
Home Institution address	PostalAddress as per CORA ISA Vocabulary v0.3
Current level of Study	Text
Current field of Study	Text
Current degree name	Text
Degree	Text
Degree Awarding Institution	Text
Year of graduation	yyyy
Degree Country	ISO 3166-1 alpha-2 code
Level of language proficiency	Europass LanguageLevelType structure
Language certificates	Base64 document and file name

Table 4.4: Academic Attributes Format

eIDAS Friendly Name	Status of integration
Person Identifier	Successfully integrated
Family Name	Successfully integrated
First Name	Successfully integrated
Gender	Successfully integrated
Id Type	Successfully integrated
Id Number	Successfully integrated
Id Issuer	Successfully integrated
Id Expiry Date	Successfully integrated
EU health card ID	Not available in autenticacao.gov.pt
Nationality	Successfully integrated
Citizenship	Successfully integrated
Marital State	Not available in autenticacao.gov.pt
Country Of Birth	Not available in autenticacao.gov.pt
Place Of Birth	Not available in autenticacao.gov.pt
Date Of Birth	Successfully integrated
Current Address	Successfully integrated
Current Photo	Not available in autenticacao.gov.pt
Tax Reference	Successfully integrated
Temporary Address	Not available in FenixEdu
Email	Successfully integrated
Phone	Successfully integrated
Home Institution Name	Successfully integrated
Home Institution Identifier	Successfully integrated
Home Institution Country	Successfully integrated
Home Institution Address	Successfully integrated
Current level of Study	Successfully integrated
Current field of Study	Successfully integrated
Current degree name	Successfully integrated
Degree	Successfully integrated
Degree Awarding Institution	Successfully integrated
Year of graduation	Successfully integrated
Degree Country	Successfully integrated
Level of language proficiency	Not available in FenixEdu
Language certificates	Not available in FenixEdu

Table 4.5: Integration Status of eID4U attribute

4.3 Portuguese eIDAS Node

Each eidas node has a Member State (MS) specific part that needs to be developed in order to connect the eIDAS Service to the IdP from that MS (or IdP equivalent if the MS doesn't use a federated identity model), this development was made by AMA for the portuguese eIDAS Node. During this development the mapping of attributes is defined between the eIDAS attributes and the MS specific attributes. As other Nodes, the portuguese Node must also be configured and managed, an effort provided by the national agency Agência para a Modernização Administrativa (AMA).

At the portuguese IdP, Fornecedor de Autenticação (FA), the user can authenticate using different methods, however only two are relevant for eIDAS due to the trust level required, 3.3.1:

- **Cartão de Cidadão** - Authentication via smart card, the most secure method available that has access to all FA attributes;
- **Chave Móvel Digital (CMD)** - A One Time Password (OTP) method that sends a code to a phone number linked to the citizen, less secure than the smart card and as such cannot provide all attributes to a SP.

One of the current limitations from this implementation is the impossibility of providing attributes from a foreign student studying at ULisboa because the user has to be successfully authenticated at the portuguese IdP before it requests the attributes from the AG and therefore the AP. One must have been born in Portugal to authenticate at FA or he must have obtained citizenship.

One of the foremost goals of eID4U was to upgrade the portuguese eIDAS Node to an eID4U Node, a Node that supports the specified attributes from eID4U. This section is divided with the previous objective in mind the first two sub-chapters, "Notified Attributes" 4.3.1 and "Consent in the Portuguese eIDAS Node" 4.3.2, explain the state and development done in the original eIDAS Regulamentation with some improvements of the eIDAS attributes already notified. The last two sub-chapters, "Personal Attributes" 4.3.3 and "Academic Attributes" 4.3.4, deal with the new attributes implemented in the scope of the project eID4U.

4.3.1 Notified Attributes

As described in chapter 3.2.3, the eIDAS Nodes must suppport a Minimum Data Set (MDS) of attributes, the portuguese institution responsible for the portuguese eIDAS Node, Agência para a Modernização Administrativa (AMA), has notified the respective committee of the personal attributes in table 4.6:

The portuguese eIDAS Node has already notified 3 new attributes besides the mandatory MDS, however these three new attributes were not implemented and one of the original MDS was misconfigured. The table 4.7 contains the eIDAS attributes, as per their original implementation.

Friendly Name	Uniform Resource Identifier (URI)
FirstName	http://eidas.europa.eu/attributes/naturalperson/CurrentFamilyName
FamilyName	http://eidas.europa.eu/attributes/naturalperson/CurrentGivenName
DateOfBirth	http://eidas.europa.eu/attributes/naturalperson/DateOfBirth
PersonIdentifier	http://eidas.europa.eu/attributes/naturalperson/PersonIdentifier
Gender	http://eidas.europa.eu/attributes/naturalperson/Gender
PlaceOfBirth	http://eidas.europa.eu/attributes/naturalperson/PlaceOfBirth
CurrentAddress	http://eidas.europa.eu/attributes/naturalperson/CurrentAddress

Table 4.6: Portuguese eIDAS notified attributes

eIDAS Friendly Name	FA Attribute Name	Status
PersonIdentifier	http://interop.gov.pt/MDC/Cidadao/NIF	Misconfiguration
FamilyName	http://interop.gov.pt/MDC/Cidadao/NomeApelido	Successfully implemented
FirstName	http://interop.gov.pt/MDC/Cidadao/NomeProprio	Successfully implemented
DateOfBirth	http://interop.gov.pt/MDC/Cidadao/DataNascimento	Successfully implemented
PlaceOfBirth	X	Not Implemented
CurrentAddress	X	Not Implemented
Gender	http://interop.gov.pt/MDC/Cidadao/Sexo	Not Implemented

Table 4.7: Portuguese eIDAS notified attributes mapping

Since some of the previously notified attributes were not implemented or were not working as intended, an evaluation of the current state was necessary, after the evaluation a proposal was created with an implementation or fix for each of the misconfigured attributes. The proposal was accepted and integrated with the Portuguese eIDAS Node, the changes proposed were:

- **PersonIdentifier** - <Citizen Country>/<Destination Country>/<Unique Identifier> - In the documentation used <Unique Identifier> was the citizen number "NIC", however the attribute used was the tax identification number of the citizen "NIF". Now the PersonIdentifier is created with the correct attribute, personal citizen number "NIF";
- **Gender** - Implementation uses the attribute "Sexo" retrieved from FA, portuguese IdP does not have a concept of gender, only of sex - Attribute has to be translated to the correct eIDAS format;
- **PlaceOfBirth** - Attribute unavailable in FA, there is no concept of place of birth in the portuguese infrastructure;
- **CurrentAddress** - The attribute CurrentAddress must be converted to an attribute called PhysicalAddress with mandatory and optional fields. To fill the fields from this attribute eIDAS had to retrieve many attributes from the portuguese IdP to compound into CurrentAddress' fields, in table 4.8 is detailed the FA attributes requested and the affiliated fields.

4.3.2 Consent in the Portuguese eIDAS Node

In the original eIDAS authentication flow 3.2.2 to comply with european laws and national laws the eIDAS Service of the destination country asks the user for consent about which attributes he allows to be given to the SP and to the eIDAS Network.

eIDAS Address Field	FA Attribute Name
Unit First Line	http://interop.gov.pt/MDC/Cidadao/Distrito
Unit Second Line	http://interop.gov.pt/MDC/Cidadao/Concelho
Street Name and Building Number	http://interop.gov.pt/MDC/Cidadao/DesignacaoDaVia
Street Name and Building Number	http://interop.gov.pt/MDC/Cidadao/NumeroPorta
Street Name and Building Number	http://interop.gov.pt/MDC/Cidadao/AbrTipoDeVia
Street Name and Building Number	http://interop.gov.pt/MDC/Cidadao/Andar
Street Name and Building Number	http://interop.gov.pt/MDC/Cidadao/Lado
Cv Address Area	http://interop.gov.pt/MDC/Cidadao/Localidade
Postal Code	http://interop.gov.pt/MDC/Cidadao/CodigoPostal4
Postal Code	http://interop.gov.pt/MDC/Cidadao/CodigoPostal3
Post Name	http://interop.gov.pt/MDC/Cidadao/LocalidadePostal

Table 4.8: CurrentAddress field mapping

In the case of the portuguese eIDAS Node the consent is skipped in the eIDAS Service, because the portuguese IdP, Fornecedor de Autenticação (FA), already has a mechanism to ask the user for consent. This is an upgrade to the original eIDAS Flow because it skips one web page the user has to interact with, improving the flow of the authentication process and also its usability. This implementation has been a part of the portuguese eIDAS since AMA notified CEF about the portuguese infrastructure.

To convey the consent obtained at the portuguese IdP to the eIDAS Node a attribute was created, so eIDAS can validate if the user consents to the attributes all received from the portuguese IdP. Only the attributes consented at the IdP are sent to the eIDAS Service, so any attribute not consented in this page will not be sent to the eIDAS Service and therefore it doesn't reach the SP who requested it. However the attributes from the MDS are always requested and are mandatory therefore if the user can't uncheck the option to consent to these attributes, by successfully authenticating in the IdP he gives the consent for the MDS to be shared, shown in figure 4.3.

Before proceeding with the authentication process and attribute translation after receiving a SAML Response from the IdP, eIDAS will first verify the presence of the attribute **PassarConsentimento**, translated to "convey consent", aborting the authentication process when the attribute is false. After this basic check the portuguese eIDAS Service continues with it's job of mapping the portuguese attributes to it's eIDAS counterpart.

eIDAS Friendly Name	Uniform Resource Identifier (URI)
PassarConsentimento	http://interop.gov.pt/MDC/FA/PassarConsentimento

Table 4.9: PassarConsentimento Attribute

4.3.3 Personal Attributes

Personal attributes are new attributes specified during the eID4U project which can be retrieved directly from the portuguese IdP, Fornecedor de Autenticação (FA). FA either stores these attributes or retrieves them from the national identity card. In the table 4.10 the eIDAS requested attribute with it's portuguese counterpart, eIDAS Service is responsible for mapping the requested eIDAS attributes into

Figure 4.3: Consent Page in Fornecedor de Autenticação (FA)

attributes specified in FA's name-space.

eIDAS Friendly Name	FA Mapped Attribute Name
IdType	http://interop.gov.pt/MDC/Cidadao/TipoDocumento
IdNumber	http://interop.gov.pt/MDC/Cidadao/NIC
IdIssuer	http://interop.gov.pt/MDC/Cidadao/EntidadeEmissora
IdExpiryDate	http://interop.gov.pt/MDC/Cidadao/DataValidade
Nationality	http://interop.gov.pt/MDC/Cidadao/Nacionalidade
Citizenship	http://interop.gov.pt/MDC/Cidadao/Nacionalidade
MaritalState	http://interop.gov.pt/MDC/Cidadao/EstadoCivil
CountryOfBirth	http://interop.gov.pt/MDC/Cidadao/Naturalidade
CurrentPhoto	http://interop.gov.pt/MDC/Cidadao/Foto
TaxReference	http://interop.gov.pt/MDC/Cidadao/NIF

Table 4.10: Personal Attribute Mapping

The Portuguese Service application of eIDAS is not only responsible for mapping the attributes from the eIDAS specification to the portuguese names but it is also responsible for translating the values of these attributes to something eIDAS partners can recognize. For example, the attribute TaxReference follows a certain format "TIN - <TaxNumber>" to create this attribute eIDAS takes the value obtained from FA which is a number, the tax identification number of the citizen "NIF", and creates the eIDAS value with it. By using the portuguese attribute retrieved eIDAS creates the attribute TaxReference with the value "TIN - <NIF>", following the eIDAS specification. The following list shows which attributes need to be translated:

- **IdType** - Must translate to eIDAS values, such as "Cartão de Cidadão" to "National Identity Card";
- **IdNumber** - No translation needed (FA sends the NIC when authenticating with the CMD method;
- **IdIssuer** - No translation needed;
- **IdExpiryDate** - Change date format from dd-mm-yyyy to yyyy-mm-dd;
- **Nationality** - Change country code ISO 3166-1 alpha-3 to ISO 3166-1 alpha-2;
- **Citizenship** - Change country code ISO 3166-1 alpha-3 to ISO 3166-1 alpha-2;
- **MaritalState** - Not Available in portuguese IdP;
- **CountryOfBirth** - Change country code ISO 3166-1 alpha-3 to ISO 3166-1 alpha-2;
- **CurrentPhoto** - Not Available in portuguese IdP;
- **TaxReference** - Portuguese TaxNumber is appended to eIDAS format;

FA implements a level of assurance system just like eIDAS so some personal attributes are not available with all authentication methods. The authentication using a citizen card is the most powerful authentication method at FA's disposal, so every attribute can be present, however when using the CMD method only a few attributes can be obtained, table 4.11 shows the different attributes each method can provide.

eIDAS Friendly Name	CC Authentication	CMD Authentication
IdType	X	
IdNumber	X	X
IdIssuer	X	
IdExpiryDate	X	X
Nationality	X	
Citizenship	X	
MaritalState		
CountryOfBirth	X	
CurrentPhoto		
TaxReference	X	X

Table 4.11: Personal Attribute Availability

4.3.4 Academic Attributes

Unlike the Personal Attributes, the academic attributes are not directly provided by FA instead the portuguese IdP uses an Attribute Aggregator (AG), Interoperabilidade na Administração Pública (iAP). FA constructs the Response on a per attribute basis, it only asks iAP for the academic attributes, iAP has an internal mapping for each attribute linking it to a certain Attribute Provider (AP). To obtain an attribute like a citizen's driver license iAP will make a request to IMT, the government organization responsible for that data, unfortunately it is impossible for the iAP to let the user choose from where it can return one attribute, for example asking for the medical records of a patient to several hospitals.

To accommodate the lack of this feature (feature that was "in the works") a set of temporary attributes was created to be used by eIDAS, with the prefix ULisboa before the name of the attribute. The FA attributes mapped from the eIDAS attributes are shown in table 4.12.

eIDAS Friendly Name	**FA Mapped Attribute Name**
Email	http://interop.gov.pt/eID4U/ULisboa/Email
Phone Number	http://interop.gov.pt/eID4U/ULisboa/Phone
Home Institution Name	http://interop.gov.pt/eID4U/homeinstitution/ULisboa/Name
Home Institution Erasmus Code	http://interop.gov.pt/eID4U/homeinstitution/ULisboa/Identifier
Home Institution Country	http://interop.gov.pt/eID4U/homeinstitution/ULisboa/Country
Home Institution address	http://interop.gov.pt/eID4U/homeinstitution/ULisboa/Address
Current level of Study	http://interop.gov.pt/eID4U/ULisboa/CurrentLevelOfStudy
Current field of Study	http://interop.gov.pt/eID4U/ULisboa/FieldOfStudy
Current degree name	http://interop.gov.pt/eID4U/ULisboa/CurrentDegree
Degree	http://interop.gov.pt/eID4U/ULisboa/Degree
Degree Awarding Institution	http://interop.gov.pt/eID4U/ULisboa/DegreeAwardingInstitution
Year of graduation	http://interop.gov.pt/eID4U/ULisboa/GraduationYear
Degree Country	http://interop.gov.pt/eID4U/ULisboa/DegreeCountry
Level of language proficiency	http://interop.gov.pt/eID4U/ULisboa/LanguageProficiency
Language certificates	http://interop.gov.pt/eID4U/ULisboa/LanguageCertificates

Table 4.12: Academic Attribute Mapping

All academic attributes are available independently from the citizen's authentication method in FA and also eIDAS Service does not have to translate any academic attribute because the AP, the academic system FenixEdu from ULisboa, has developed it's webservices in consideration with the eIDAS specification and the attribute does not suffer any transformation till he arrives at the eIDAS Service. While translation is not needed the attributes must still be mapped from the portuguese name-space to the eIDAS name-space.

4.4 eRegistration

The eRegistration service aims to register or enroll foreign students, particularirly ERASMUS students, in ULisboa using their national Electronic Identity (eID) retrieved using the eIDAS Network. I will be using the new attributes declared in the eID4U project to fill ERASMUS application forms. ERASMUS registration will give access to the academic platform of ULisboa for the accepted foreign students.

Before delving into the functional specifications of the eRegistration service, the actual process of ERASMUS registration is briefly explained, along with the information requested and the actions taken by each participant in each phase of the process. The ERASMUS registration process predates the eIDAS infrastructure and therefore it is independent from it, however the process can be improved using the eID4U's new attributes, refactoring the process.

4.4.1 ERASMUS Registration

The terms used in the ERASMUS Registration process:

- **Home University** - The university where the student is registered and actively enrolled in and the university that will grant the qualification of the complete course to the student;
- **Host University** - The foreign university with an agreement with the home university to receive their students and provide academic resources.

When a foreign student wishes to register in the University of Lisbon (ULisboa) under the ERASMUS program, first he must submit an application in the corresponding FenixEdu platform, with the approval of it's home university. After following these steps the student will have submitted the application:

1. **Create an account and profile in the school's FenixEdu platform**

   To start with the student must have an account and profile in the school's FenixEdu platform, since he is a foreign student he has to start by creating one. The account holds the credentials to access the FenixEdu platform, the profile holds the personal information of the subject registered in FenixEdu, may he be a student, applicant, teacher or employee of a specific school's service.

   To create the Account and Profile the student must provide the following data:

   - Given and Family names;
   - Identification document type and number. The type may be a Passport, a Citizen Identity Card or another document;
   - Date of birth;
   - Gender;
   - Email and contact phone;
   - Password for the account.

   After the creation of the account, the FenixEdu provides the credentials for the student to login and access the Application Portal, the service to create an ERASMUS application.

2. **Create and submit an ERASMUS Application**

   After creating an account, the student logs into the FenixEdu platform and accesses the Application Portal page. In this page he is presented with the open ERASMUS applications (e.g Licentiate, Master or PhD Program). Each open application has a deadline for the student to submit an application.

   When a student selects the ERASMUS Incoming Program, an application workflow process will be created. In this process the student fills their personal and academic information and attaches the requested documents. Some of the requested attributes are the Home University, the area and level of studies and the degree enrolled at Home University.

   After submission the ERASMUS application cannot be changed by the student.

3. **Validation and acceptance by the ULisboa school services**

   After the deadline the International Relations Office validates all submitted applications, they check if:

   (a) The Home University has a bilateral agreement with ULisboa in the area and level of studies selected.

   (b) The student was nominated by his Home University to spend his studies in ULisboa.

   If the application complies with all requirements, then it is accepted and the student can register in ULisboa.

4. **Registration in ULisboa**

   At this stage the Academic Services register the student and enroll him in the corresponding units of the degree, declared in the Learning Agreement.

To summarize, to create an ERASMUS application the student has to fills a form, with this date FenixEdu creates an account for this student, then he submits the ERASMUS application which is reviewed by the International Relations Office and after review the Academic Office starts the ERASMUS registration process.

4.4.2 Functional Requirements

The requirements for the creation of an ERASMUS application are:

1. **Personal and Academic attributes required to create a FenixEdu Account and ERASMUS application**

   In order to create an ERASMUS application, FenixEdu validates if the student has the following academic attributes:

   (a) **HomeInstitutionName** - The name of the Home Institution the student is attending;

   (b) **HomeInstitutionIdentifier** - The ERASMUS code of the student's Home Institution;

   (c) **CurrentLevelOfStudy** - The current level of study that the student is attending in Home Institution;

   (d) **FieldOfStudy** - The field of the degree the student is attending in Home Institution;

   (e) **CurrentDegree** - The name of the degree the student is attending in Home Institution.

   If the student is attending at his Home University then the above attributes should be returned with meaningful values. If one of the above attributes is missing, the student is informed of the missing data and asked to complete it in the form. The application process will be created and the International Relations Office will validate the requirements stated above.

For the creation of a FenixEdu Account, the following attributes declared as mandatory, the Minimum Data Set (MDS), must be returned with meaningful values:

- **CurrentFamilyName** - The family names of the student.
- **CurrentGivenName** - The given names of the student.
- **DateOfBirth** - The date of birth of the student.

If one of these attributes is not returned in the eIDAS authentication, the application process is aborted and the student is informed about the missing data.

The following optional attributes are also used for the FenixEdu Account creation. If one of the following attributes is not returned, the student must fill them before the account and application process is created:

(a) **IdType** - The ID document type of the student;

(b) **IdNumber** - The ID document number of the student;

(c) **Gender** - The gender of the student;

(d) **Email** - The email of the student;

The identification type and number filled by the student are checked against an existing profile with the same identification number. If a profile exists with the same identification number, the Account Profile and ERASMUS Application is not created and an error is triggered. This validation is necessary to prevent the hijack of an existing Account Profile.

The student provides this data in the ERASMUS application process.

2. **Use an existing FenixEdu Account and Profile**

Before the creation of a FenixEdu Account, it is necessary to check if a profile already exists, in order to prevent duplication of accounts. The procedure to check account duplication is:

(a) If more than one profile is found with the same IdNumber and IdType it is reported as an error and the student is informed to contact the Support Team.

If one profile is found with same IdType and IdNumber, with same DateOfBirth, CurrentGivenName, CurrentFamilyName and Gender, then this profile is used to retrieve or create the ERASMUS application process.

(b) If a profile is not found with the IdType and IdNumber, a profile is searched with the same PersonIdentifier, CurrentGivenName, CurrentFamilyName and DateOfBirth. This search is used when a student logs into an Electronic Identity (eID) scheme which does not return IdType and IdNumber, but the student has previously submitted an application with eIDAS, and the system associates the PersonIdentifier with his account. If one profile is found, then this profile is used to retrieve or create the ERASMUS application process. If more than one profile is found having the same PersonIdentifier, CurrentGivenName, CurrentFamilyName

and DateOfBirth, it is reported as an error and the student is informed to contact the Support Team.

(c) If a profile is not found then a new Account and Profile is created in FenixEdu.

3. **Check if an ERASMUS application for the same user exists**

   In order to not duplicate an ERASMUS application process, it is checked if a process already exists for the academic year and period in which the student will start his studies.

4.4.3 FenixEdu Entity Model

The following entities encompass the FenixEdu entity model, shown in figure 4.4, relevant in the ERASMUS Registration process:

- **Person** - Represents a profile of a subject registered in FenixEdu, may he be a student, teacher, applicant or employee. This entity contains the personal data of the subject.
- **User** - Represents an account with credentials that give access to the FenixEdu system.
- **AcademicCandidacy** - An application holding data relevant for the program, for example ERASMUS, Master or PhD.
- **Student** - This entity represents a student, it is composed of the registrations of units belonging to that student.
- **Registration** - The student record in which the student is enrolled to units.

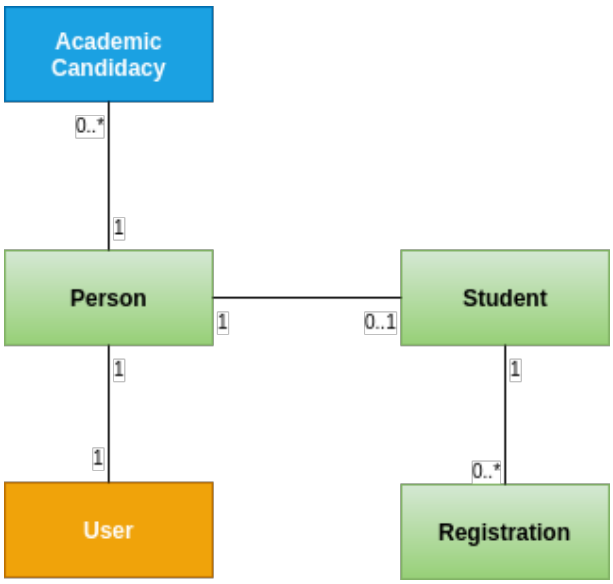

Figure 4.4: FenixEdu Academic Data Model

The following personal attributes are converted and registered in the Person entity:

1. The attribute **PersonIdentifier** is not converted and is used to validate the profile, this profile contains the attributes **IdNumber** or **CurrentGivenName**, **CurrentFamilyName** and **DateOfBirth**. It is stored during the application creation with eIDAS authentication.
2. The attributes **CurrentGivenName** and **CurrentFamilyName** are concatenated to create an attribute representing the subject's full name.
3. The values accepted by **Gender** are "M" and "F". If Gender is "X" then it is considered as missing and is not filled automatically in the application form so it must be fill in by the student.
4. The **IdType** is converted to the enum type **IdDocumentType**. If **IdType** value is "National Identity Card" the **Citizenship**, or in absence **Nationality**, is checked if it refers to Portugal. If the student is portuguese, the **IdType** is converted to IdDocumentType.IDENTITY_CARD, when the student is not portuguese **IdType** is converted to IdDocumentType.NATIVE_COUNTRY_IDENTITY_CARD or IdDocumentType.PASSPORT.
5. The attribute **IdNumber** is registered without conversion.
6. The attribute **IdIssuer** is registered without conversion.
7. The attribute **IdExpiryDate** is converted as datetime type.
8. The attribute **Nationality** is converted as an ISO 3166-1 alpha-2 type.
9. The Attribute **MaritalState** is converted as MaritalStatus enum type.
10. The attribute **CountryOfBirth** is converted as an ISO 3166-1 alpha-2 type.
11. The attribute **PlaceOfBirth** is registered without conversion.
12. The attribute **DateOfBirth** is converted as datetime type.
13. The attribute **CurrentAddress** is converted to PhysicalAddress with following rules:
    (a) eIDAS Address Locator property is registered as the Address Street name. In the case of a missing value it uses the eIDAS Address FullCvaddress property;
    (b) eIDAS Address PostCode property is registered as the Address Zip Code;
    (c) eIDAS Address PostName property is registered as the Address Area.
14. The attribute **CurrentPhoto** is converted to a document of type Photo, and attached to the ERASMUS application process, which is represented by the **AcademicCandidacy** workflow process.
15. The attribute **TaxIdentificationNumber** is split between Fiscal Country and Fiscal Number. Both the values are registered in the Person class.

The following academic attributes, relating to the current studies are registered in the provenance **Qualification** entity which holds the information about the student's current degree:

1. **HomeInstitutionName** - The name of the Home Institution the student is attending;
2. **HomeInstitutionIdentifier** - The ERASMUS code of the student's Home Institution;
3. **CurrentLevelOfStudy** - The current level of study of the degree from Home Institution;
4. **FieldOfStudy** - The field of study of the degree from Home Institution;
5. **CurrentDegree** - The name of the degree from Home Institution;

The following academic attributes, relating to the previous studies are registered in the highest **Qualification** entity:

1. **Degree** - The ISCED code of the level of studies describing the qualification.
2. **DegreeAwardingInstitution** - The name of the institution in which the student graduated.
3. **GraduationYear** - The year when the student graduated.
4. **DegreeCountry** - The country of the institution where the student graduated.

For the Language Proficiency it was created the **LanguageProficiency** entity to hold information of the attribute LanguageProficiency, which is an XML document. To retrieve the student's language proficiency information a Simple API for XML (SAX) parser was used.

4.4.4 Integration of ERASMUS with eIDAS

Enabling ERASMUS registration via eIDAS Network, is a usability improvement of the Account and ERASMUS application creation, with the single action of authenticating in the eIDAS Network. The process is initiated by selecting the ERASMUS application in FenixEdu, figure 4.5.

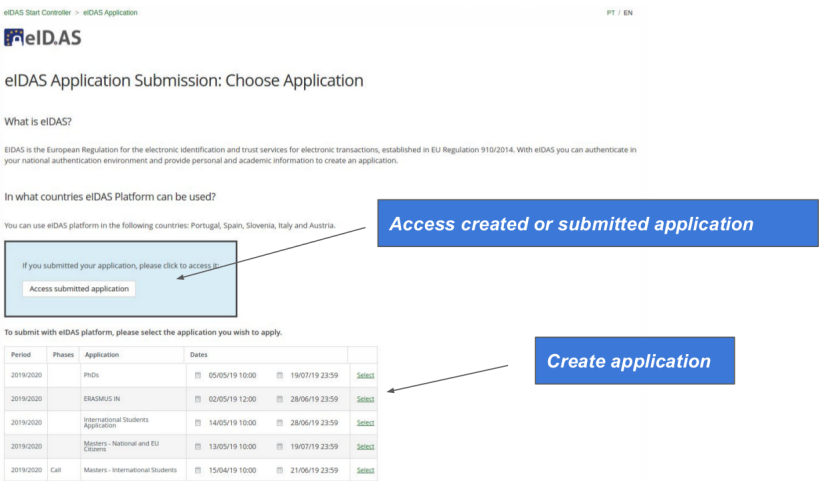

Figure 4.5: eIDAS Application Landing Page

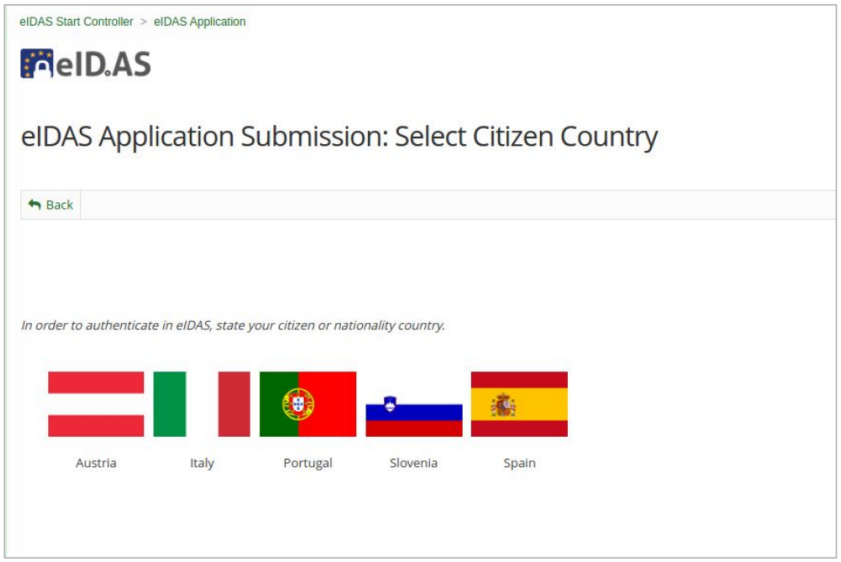

Figure 4.6: Select Country to authenticate in eIDAS Platform

After choosing the degree the student wishes to apply to, FenixEdu redirects the user to a page where he can choose one of the supported eID4U countries to start the authentication process.

After selecting the ERASMUS application and their citizen country, the student is requested to authenticate in the eIDAS Network. FenixEdu requests both personal and academic attributes, described in chapter 4.4.2. Before the student fills his application FenixEdu can check if the student is qualified to submit an ERASMUS application to that degree by analyzing the returned eIDAS academic attributes.

If the required eIDAS attributes are present, FenixEdu creates an Account and an ERASMUS Application by attempting to fill the student's personal and academic information and attaching the necessary documents. Then ERASMUS registration continues the previously explained process, where the student fills the remaining required data. When the student has provided all requested data correctly, he submits the application process for validation and acceptance.

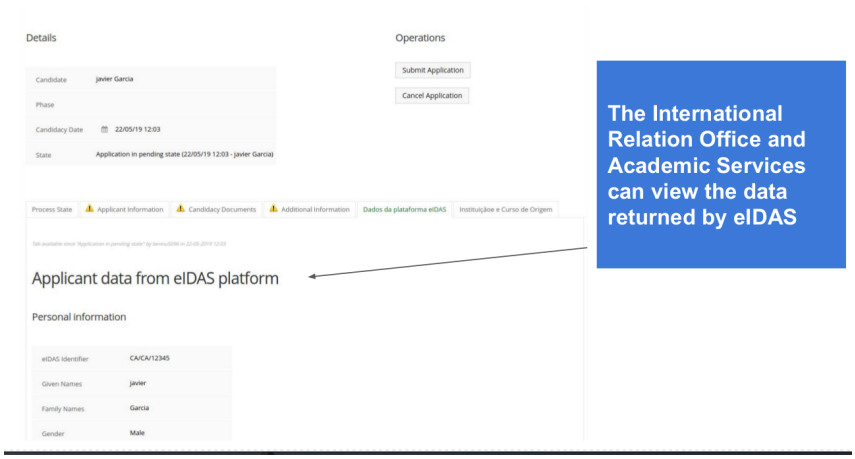

Figure 4.7: Access submitted application with eIDAS authentication

Although FenixEdu can do a simple validation based on the attributes returned in the authentication

process using the eIDAS Network, the International Relations Office and Academic Services must proceed with the validation of certain requirements, but with more assurance on the information provided by eIDAS than provided by the student.

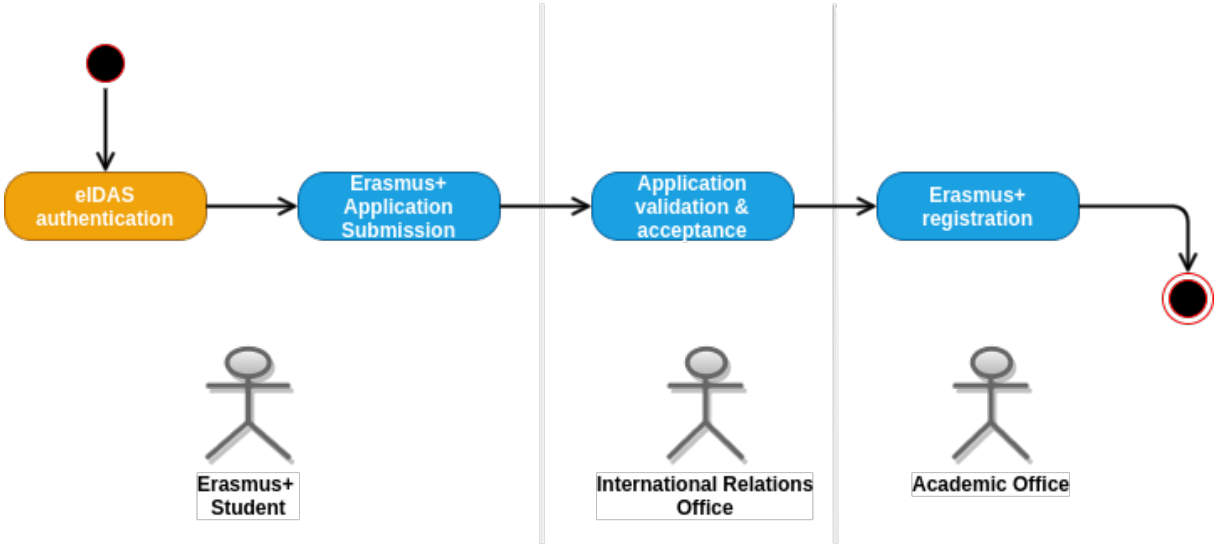

Figure 4.8: ERASMUS Application with eIDAS Activity Diagram

The ERASMUS registration process, integrated with eIDAS Network, is depicted in the following high level activity diagram (Fig 4.8)

4.4.5 Integration of FenixEdu in eIDAS as a Service Provider (SP)

This section will address FenixEdu's role as a SP in the eIDAS infrastructure. In this role FenixEdu will connect to the the Portuguese eIDAS Proxy Connector with the objective of retrieving attributes and authenticating users, so they can apply to the ERASMUS program.

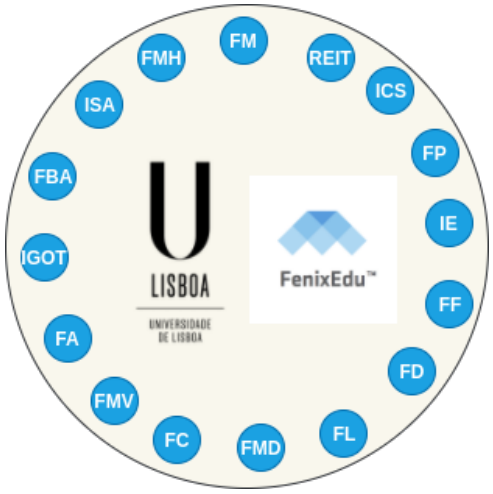

Figure 4.9: ULisboa schools that use the FenixEdu platform

Each ULisboa School has its own FenixEdu platform instance, which at the time of writing this report is 16 instances, identified by their own domain and accessible publicly by their URL.

The integration code to connect with eIDAS node is based on the **Demo-SP**, since FenixEdu also runs in a Java Virtual Machine (JVM) and is programmed in Java. Here FenixEdu acts as a SP, connecting to the eIDAS Portuguese Connector. First tested using the ULisboa eID4U machine and then changed the endpoint to the pre-production PT eIDAS node, in this node all 16 FenixEdu instances had to be white-listed, however due to the instances sharing SP properties Agência para a Modernização Administrativa (AMA) only had to import one set of Metadata, Sign and Encryption certificates.

4.4.6 Integration of FenixEdu in eIDAS as an Attribute Provider (AP)

This section will address FenixEdu's role as an AP in the eIDAS infrastructure. In this role FenixEdu will connect to the the Portuguese Infrastructure, specifically to Interoperabilidade na Administração Pública (iAP), with the objective of publishing the academic attributes not present in the Portuguese Infrastructure to eIDAS, the academic attributes. By publishing the attributes of portuguese citizens to eIDAS these students can apply for an ERASMUS program in other compatible european universities.

As previously mentioned FenixEdu is not a single system but a group of instances, one for each school in the University of Lisbon (ULisboa). The use of instances for each school poses a problem, the iAP does not know which instance to query to obtain the data of the student, since the data of the student might be scattered between the instances, like when a student is enrolled in several schools of ULisboa.

The chosen solution was to name one of the FenixEdu instances as the master instance, FenixEdu-REIT, and connect it to the iAP. The master instance receives the request from the iAP and conveys it to all other instances. The request contain the Portuguese Citizen Card Number to identity the respective student, along with the requested academic attributes. FenixEdu-REIT will query each slave instance including it's own instance for the academic attributes using a WebService. The workflow is detailed in the figure 4.10.

For a Portuguese student to apply, via eRegistration, to an ERASMUS program in an Italian University he must follow this sequence of steps:

1. Portuguese student navigates to the Italian eRegistration page, by following the designated steps at the Italian academic service. The academic service, also known as a SP, will send a SAML Authentication Request asking for personal and academic attributes to the Italian eIDAS Connector:
2. the Italian eIDAS Connector will send a SAML Authentication Request and redirect the student to the Portuguese eIDAS Service, the Authentication Request will request the same attributes with no mapping or translation.
3. Now on Portuguese ground, the eIDAS Service will send one more SAML Authentication Request and redirect to Fornecedor de Autenticação (FA) the portuguese IdP, asking for the personal and academic attributes.

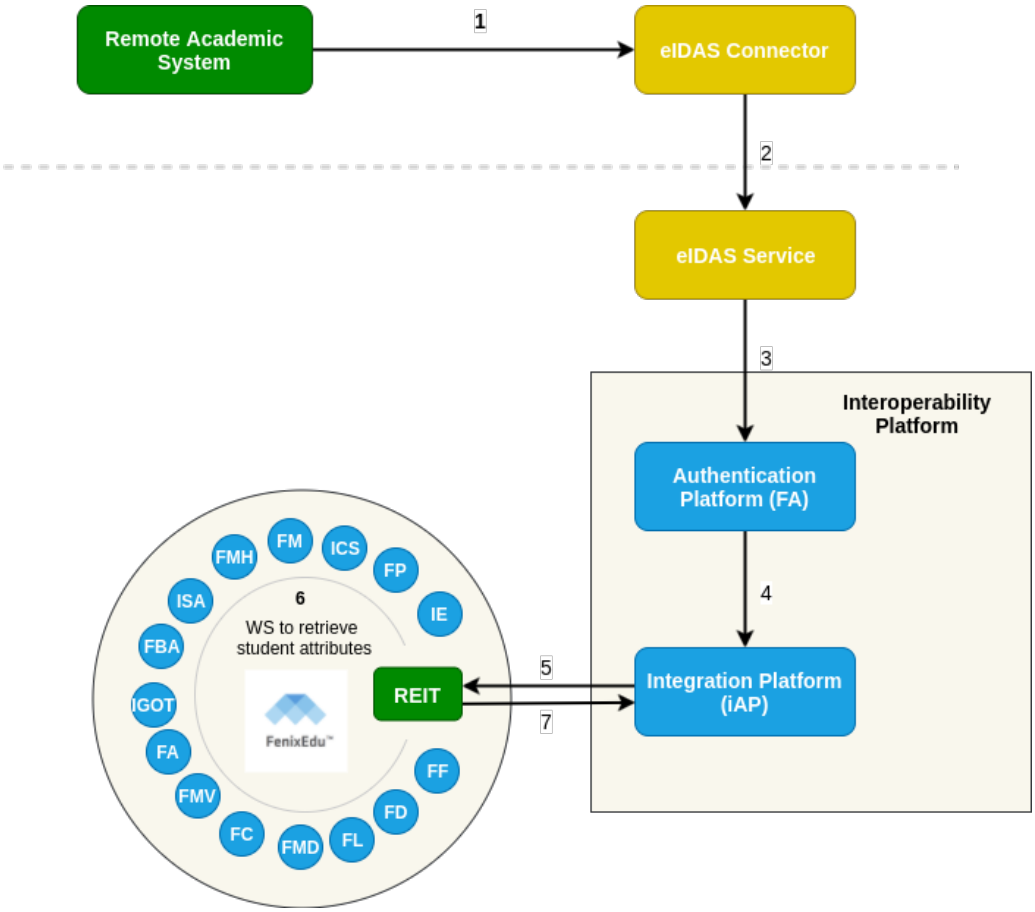

Figure 4.10: eIDAS academic attributes Request Flow

4. FA authenticates the user, asks for consent for each attribute requested and fetches the personal attributes from an internal database and request the academic attributes from iAP, through a Web-Service.

5. Interoperabilidade na Administração Pública (iAP) is an Attribute Aggregator (AG), it checks the attributes requested and creates a request for the appropiate AP, in this case ULisboa. The iAP formulates an asynchronous request via SOAP to the master FenixEdu instance, FenixEdu-REIT.

6. FenixEdu-REIT queries each FenixEdu system from each school for the academic data of the student and also for the consent to share their data with eIDAS and other entities. A student can have data from several schools, in that case the data is evaluated so the attributes only send the most relevant data, so the following precedence is applied:

   (a) The group of academic attributes in which the student is currently enrolled in a degree has more precedence.

   (b) If the student is enrolled in more than one degree, the one with higher academic level has more precedence.

   (c) If the student is not enrolled in any degree, the degree with higher academic level takes precedence.

7. Fenix-REIT aggregates the data and responds via SOAP to the iAP.

8. iAP returns the academic attributes to FA.

9. FA aggregates the personal attributes from FA and the academic attributes retrieved by iAP. The portuguese IdP creates a SAML Authentication Response with all the attributes requested and sends it to the Portuguese eIDAS Service.

10. The Portuguese eIDAS Service receives the Response and sends theirs to the Italian eIDAS Service, with the personal and academic attributes delivered by the portuguese IdP.

11. Finally the Italian University, the SP, receives the SAML Authentication Response and the student can continue with their ERASMUS application process.

The protocol used to exchange academic attributes between FenixEdu-REIT and Interoperabilidade na Administração Pública (iAP) is accomplished with WS-Addressing, which allows for the exchange of asynchronous messages between both services. The asynchronous messages are necessary to free up compute resources in iAP system as soon as possible.

1. To lookup academic attributes associated with a citizen's Electronic Identity (eID), iAP sends the following message to FenixEdu-REIT:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:wsrm="http://schemas.xmlsoap.org/ws/2005/02/rm">
  <soap:Header>
    <wsa:Action>ConsultaInfoAtributosAcademicos</wsa:Action>
    <wsa:MessageID>251d9408-1711-4faa-943c-ef492d906183
    </wsa:MessageID>
    <wsa:ReplyTo>
    <wsa:Address>http://www.w3.org/2005/08/addressing/anonymous
    </wsa:Address>
    </wsa:ReplyTo>
    <wsa:To>
    https://fenix-qua.reitoria.ulisboa.pt/web-services/iap-active-registration
    </wsa:To>
  </soap:Header>
  <soap:Body>
    <int:consultaInfoAtributosAcademicos
      xmlns:fen="http://fenixedu.ulisboa.pt/"
      xmlns:ind="http://autenticacao.cartaodecidadao.pt/servicos/2010/01"
      xmlns:int="http://integration.services.eidas.module.qubEdu.qubit.com/">
      <fen:ConsultaInfoAtributosAcademicos>
        <FAObterAtributos_1>
          <identificadorCidadao>XXXXXXXXX</identificadorCidadao>
          <pedidoAtributos>
            <atributos>
            <Atributo
              xmlns="http://autenticacao.cartaodecidadao.pt/servicos/2010/01"
              Nome="http://interop.gov.pt/eID4U/ULisboa/FieldOfStudy" />
            <Atributo
              xmlns="http://autenticacao.cartaodecidadao.pt/servicos/2010/01"
```

```
            Nome="http://interop.gov.pt/eID4U/ULisboa/GraduationYear" />
          </atributos>
        </pedidoAtributos>
        <DataHora>2019-09-19T15:06:34.6162337Z</DataHora>
        <nomeCidadao>XXXX XXXXXX</nomeCidadao>
        <numeroPedido>251d9408-1711-4faa-943c-ef492d906183</numeroPedido>
        <prestadorServicosRequerente>ULisboa</prestadorServicosRequerente>
      </FAObterAtributos_1>
    </fen:ConsultaInfoAtributosAcademicos>
  </int:consultaInfoAtributosAcademicos>
 </soap:Body>
</soap:Envelope>
```

FenixEdu-REIT responds with: HTTP code 200 - Accepted without any response; and then begins the attribute retrieval process.

2. FenixEdu-REIT uses the Citizen Card Number as the Unique Identifier for the user and request all the data pertaining to said user to other FenixEdu systems. After retrieving and aggregating all the necessary information of the user FenixEdu responds to iAP with the following message:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
 <SOAP-ENV:Header>
  <MessageID xmlns="http://www.w3.org/2005/08/addressing">
  uuid:37b9fadb-c80d-4d30-9724-94abddfbf2ad
  </MessageID>
  <Action xmlns="http://www.w3.org/2005/08/addressing">
  RespostaConsultaInfoAtributosAcademicos</Action>
  <RelatesTo xmlns="http://www.w3.org/2005/08/addressing">
  251d9408-1711-4faa-943c-ef492d906183
  </RelatesTo>
  <To xmlns="http://www.w3.org/2005/08/addressing">
  http://172.31.201.82/ulisboa/
  </To>
 </SOAP-ENV:Header>
 <S:Body>
  <ns2:ConsultaInfoAtributosAcademicos
   xmlns:ns2="http://autenticacao.cartaodecidadao.pt/servicos/2010/01">
   <NumeroPedido>251d9408-1711-4faa-943c-ef492d906183</NumeroPedido>
   <DataHora>2019-09-19T16:06:35.196+01:00</DataHora>
   <Atributos>
    <Atributo Nome="http://interop.gov.pt/eID4U/ULisboa/FieldOfStudy"
     Resultado="Disponivel">0913</Atributo>
    <Atributo Nome="http://interop.gov.pt/eID4U/ULisboa/GraduationYear"
     Resultado="Disponivel">2011</Atributo>
   </Atributos>
  </ns2:ConsultaInfoAtributosAcademicos>
 </S:Body>
</S:Envelope>
```

The tag RelatesTo contains the key of the original message sent by iAP in MessageID, and is used by iAP to correlate the attributes in the response with the request and therefore the user that request related to.

4.5 eLogin

ULisboa has a Single Sign-On (SSO) system, described in chapter 3.1, which allows it's users to authenticate with several different IdPs to gain access to several services managed by ULisboa, one of these is `www.sas.ulisboa.pt`. The purpose of the service eLogin is to allow ULisboa's users to authenticate with the eIDAS Network: european citizens should be able to authenticate using the eID from their country and portuguese citizens should also be able to authenticate with the portuguese national IdP, Fornecedor de Autenticação (FA).

The most challenging issue for this service is the lack of a practicable unique identifier in eIDAS. An unique identifier has been defined in eIDAS, while this identifier may be unique it is not immutable and always the same (a citizen may have two different eIDAS personal identifiers in two different IdPs, either from different MS or due to the architectural design of the identity infrastructure of the MS) and it may change with time. So the personal identifier retrieved from eIDAS is related to only one individual but one individual might have many different personal identifiers. Without a personal identifier the identification of an user is done using attribute matching, the requirements are present in chapter 4.5.1.

4.5.1 Functional Requirements

The functional requirements are the requirements necessary for an user to be authenticated in the ULisboa Identity Management (IdM) system, NetIQ's Identity Manager (IDM). The requirements are different for Portuguese citizens and European citizens in order to obtain a higher degree of fidelity in the authentication process of Portuguese citizens. The two types of authentication processes are defined below:

1. **European Citizens**

   In order to identify an european citizen in NetIQ's Identity Manager (IDM) an user match has to be found this is accomplished using a set of attributes that must match with a set of attributes from the internal LDAP database because there isn't a definitive unique identifier. The unique identifier in the eIDAS Minimum Data Set (MDS) is the personIdentifier, while unique this attribute can change for the same person, in a certain amount of time, so even if this attribute was in the IDM it wouldn't be reliable.

   So without the reliability of an unique identifier the set of attributes necessary to reasonably match a european citizen are present in table 4.13.

Friendly Name	IdM Attribute
Full Name	fullName
Country of Citizenship	ULCountryCode
Date Of Birth	birthDate

Table 4.13: European Attribute Matching Set

The attribute ULCountryCode was an attribute created specifically for eLogin, ULisboa's IDM already has an attribute for the birth country of an user, ULCountry, however this attribute is initialized with the user's input. Due to this lack of foresight the list of countries is not normalized, with some countries having different spellings, typos or just plain wrong spellings e.g Brazil, Brasil and Brasíl or a more relevant example Austria, Aústria and Áustria. With the normalization of the ULCountry attribute another attribute, ULCountryCode, was created which reflected the ISO 3166-1 alpha-2 country codes of the values in ULCountry.

2. **Portuguese Citizens**

   In order to identify a portuguese citizen in NetIQ's Identity Manager (IDM) an user match has to be found but only with one attribute, the portuguese citizenship number. Since the portuguese citizenship number is an unique identifier no other attributes are required for a successful authentication but in the case of a mismatched attribute the option to use the european attribute matching set is available.

   In the IDM the portuguese citizenship number is stored in the attribute **ULBI**. The matching attribute set when the user is a portuguese citizen is presented in table 4.14

Friendly Name	IdM Attribute
Portuguese Citizenship ID	ULBI

Table 4.14: National Attribute Matching Set

NetIQ's Access Manager (AM) directly connects to the IDM and provides it with the attributes needed for authentication. The AM creates the attributes required by the IDM with the attributes in table 4.15. Access Management (AM) will translate the AM attribute set to the attributes in the IDM set.

Friendly Name	AM Attribute
Full Name	FullName
Citizenship Country	CountryCode
Date Of Birth	DateOfBirth
Citizenship ID	IdNumber

Table 4.15: Access Manager Attribute Set

These attributes are the attributes that must be provided by the solution that will connect to AM, this solution should act as an IdP by delivering the attributes requested in 4.15 in a SAML Response. AM is responsible for the matching expression of the attributes, this will influence the attribute set used by the IDM.

An unique match is required, only one unique user in the LDAP database must have the same attribute values for the keys of the attribute set used, either the National or the European Attribute Matching Set.

4.5.2 Challenges

To connect NetIQ's Access Manager (AM) and the portuguese eIDAS Service there are a challenges that need to be solved. These challenges can be divided into three categories: eIDAS SAML requirements, attribute mapping and country selection.

In order for a SP to send a valid SAML Authentication Request to the eIDAS Service the request needs to meet the eIDAS SAML requirements, these are:

(a) **Provider Name** - this optional attribute of a SAML Authentication Request is mandatory for eIDAS, so the final IdP knows which SP is requesting the user's attributes.

(b) **Signing** - Signing of an Authentication Request is mandatory. Signature validation of the Authentication Response is optional, *Digest Algorithms* must be supported to be able to validate a signature;

(c) **Encryption** - Encryption of an Authentication Request is optional, but the decryption of the Authentication Response is mandatory since the eIDAS Response is encrypted.

(d) **Attributes Requested** - Mandatory for all Authentication Requests, in a *Extension*, the attributes requested must be in the eIDAS namespace, defined in eIDAS or eID4U projects. Only the minimum required attributes for authentication should be requested.

(e) **Level Of Assurance (LoA)** - similar to trust levels from chapter 3.3.1, a mandatory SAML tag must be created *RequestedAuthnContext* containing the LoA. LoA determines the assurance provided by the authentication method used, the SP decides the minimum LoA accepted thus deciding which authentication methods can be used. For example when a high LoA is requested by the SP only an authentication method on the same level as the Portuguese Citizen Card in FA can be used to authenticate that user;

Attribute Mapping is an issue present at both ends of the eIDAS Network, it is only normal for different systems to use different names and sets of attributes. The challenge here is to map and transform the attributes received from eIDAS into the attributes IDM can recognize, present in chapter 4.5.1, and the opposite map IDM's attributes to the eIDAS attributes when making a request.

The eIDAS Network is composed of several eIDAS Nodes from several European countries, the user must select the country he will authenticate in but this selection is not provided by the eIDAS Nodes. It is the responsibility of the SP to provide eIDAS with the information of the user select country. So the last challenge is to be able to send the parameter *CountryCode* and the SAML Authentication Request at the same time in the HTTP POST message to eIDAS, the SAML protocol used is explained in chapter 2.1.

One of the biggest issues with the eIDAS Network is the connection of a new SP to the eIDAS Nodes, this project is no exception. It is not possible to directly connect NetIQ's Access Manager (AM)

to an eIDAS Node, the only challenges AM can directly resolve are the Authentication Request signing and the Level of Assurance. To solve the challenges present in this section a proxy was created, ULisboa eIDAS Proxy (ULEP).

4.5.3 ULisboa eIDAS Proxy

ULisboa eIDAS Proxy (ULEP) is a Legacy Proxy developed to connect AM and the Portuguese eIDAS Node. It was created to solve the usual issues legacy SPs have when connecting to an eIDAS Node, with the proper configuration this solution can be used with other legacy SPs.

Due to it's role as a Proxy, ULEP is composed of three modules all developed using Java, the SP module that creates and receives messages from the eIDAS Node, the IdP module that creates and receives messages from the legacy SP, in this case AM, and the *CountryCode* resolver.

The SP module is based on the **Demo-SP** provided by the eIDAS package. There are two objectives the SP module must achieve: the first objective is to create the HTTP POST message sent to eIDAS that consists of the SAML Authentication Request, meeting the eIDAS SAML requirements, and also the parameter *CountryCode*; the second objective is to validate and decrypt the encrypted eIDAS SAML Authentication Response and extract the user's attributes so they can be used in the IdP module.

The *CountryCode* resolver is a page that allows the user to choose the destination eIDAS Node. AM cannot send a SAML Authentication Request and a POST parameter (*CountryCode*) at the same time, so a page was created in AM, figure 4.11, where the user can choose his destination country, AM sends the parameter to the *CountryCode* resolver, ULEP then stores the *CountryCode* in the user's browser session and then redirects the browser to AM's SAML creation page.

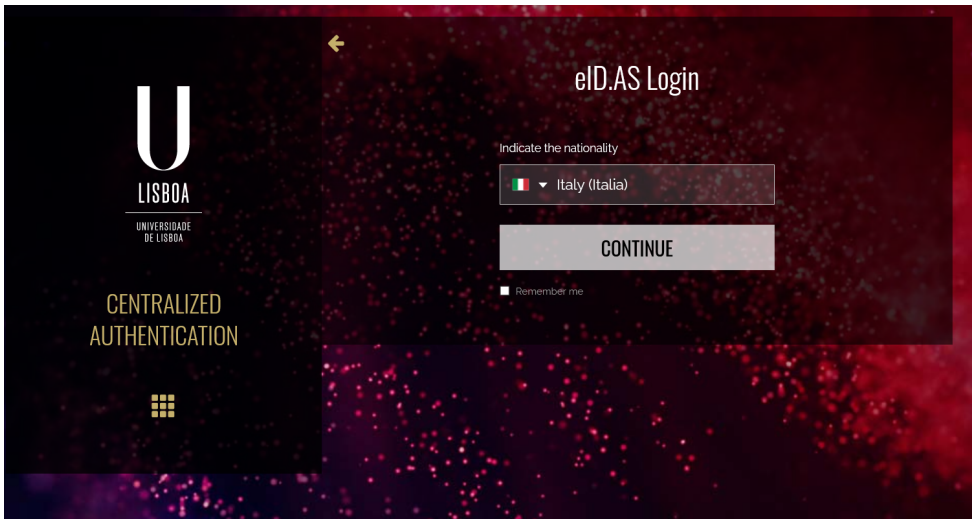

Figure 4.11: AM Country Selection page

The IdP module uses the SAML library OpenSaml v2 to create the SAML messages. The objectives of the IdP module are: to receive the Authentication Requests from AM and forward the important information to the SP module; to translate and map the attributes received from the eIDAS Network into

the attribute set specified in the functional requirements, table 4.15; create an Authentication Response to send to AM with the translated attributes.

4.5.4 eLogin Authentication

Figure 4.12 depicts an example of eIDAS Authentication if a SP from ULisboa (not connect to our SSO system, IDM and AM) could connect directly with the Portuguese eIDAS Node.

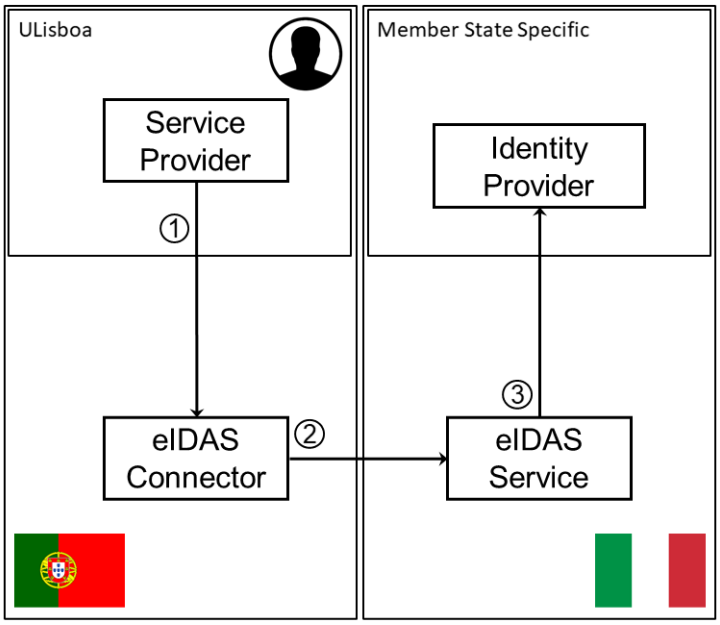

Figure 4.12: eIDAS Authentication Model

1. SP sends an Authentication Request to the Portuguese eIDAS Node, the Portuguese eIDAS Connector receives it;
2. the Portuguese eIDAS Node sends an Authentication Request to the Italian eIDAS Service;
3. the Italian eIDAS Service sends an Authentication Request to the Italian IdP.
4. After the authentication of the user the IdP sends an Authentication Response to the Italian eIDAS Service.
5. The Italian eIDAS Service sends an Authentication Response to the Portuguese eIDAS Connector;
6. the Portuguese eIDAS Connector sends an Authentication Response to the SP.
7. the SP gives access to the user upon the successful authentication.

However most of ULisboa's services area already integrated with ULisboa's SSO, the authentication model for it is explained in chapter 3.1. For a user to get access to one of ULisboa's services he needs to be authenticated in the IDM, the application AM is responsible for contacting the SPs and checking the user's authentication at the IdM.

Figure 4.13 depicts the flow of the Authentication Requests of a user trying to access ULisboa's services and using the eIDAS Network as his means of authentication.

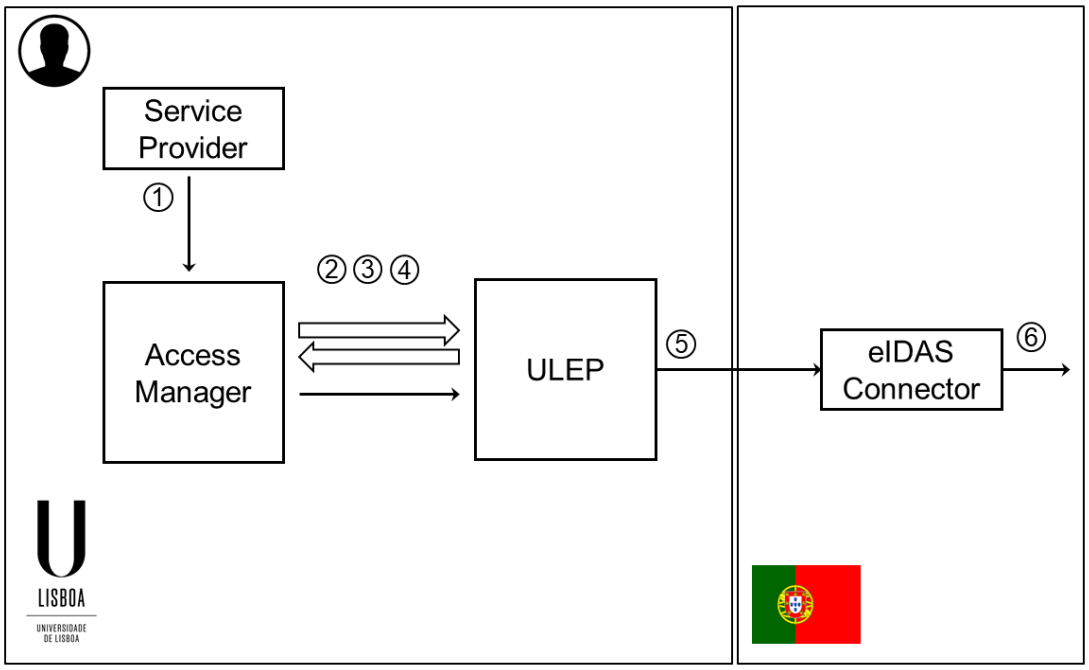

Figure 4.13: eLogin Authentication Request Flow

An user is trying to access a service connected to ULisboa's SSO system:

1. The SP sends an Authentication Request to AM;
2. In the landing page of AM the user can choose one of several IdPs to authenticate with, one of them being eIDAS, after choosing the eIDAS option a new page is shown to the user where he can select which country he will authenticate with. When the country is selected by the user AM redirects to ULEP with the *CountryCode*;

Figure 4.14: ULisboa SSO page

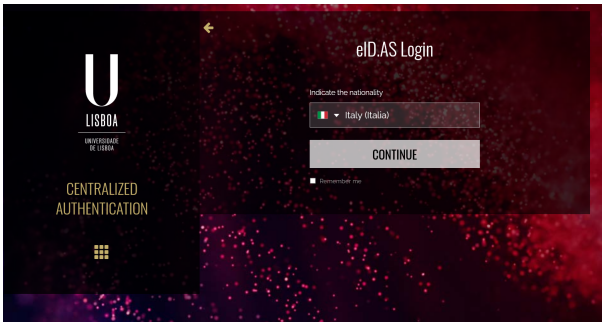

Figure 4.15: eIDAS Page

3. ULEP stores the *CountryCode* in a session and promptly redirects to AM, this step is not shown to the user.

4. Now AM creates an Authentication Request and sends it to ULEP;
5. ULEP creates a POST message to the Portuguese eIDAS Connector with the *CountryCode* from the session and a SAML Authentication Request with the attributes necessary for login and the requirements necessary for eIDAS.
6. The Portuguese eIDAS Connector sends an Authentication Request to the respective eIDAS Node which is dictated by the *CountryCode*.

After the user authenticates at one of the IdP supported in the eIDAS Network the Portuguese eIDAS Connector will receive an Authentication Response related to the previously sent Authentication Request, depicted in figure 4.16.

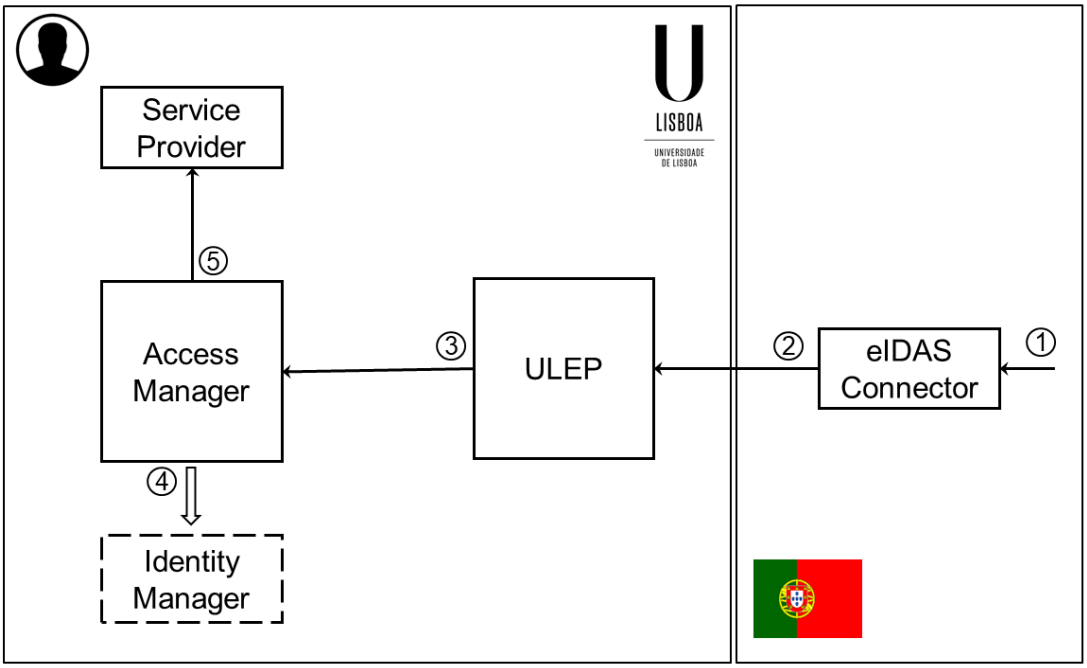

Figure 4.16: eLogin Authentication Response Flow

1. The Portuguese eIDAS Connector receives an encrypted Authentication Response by another eIDAS Node;
2. The Portuguese eIDAS Connector sends an encrypted Authentication Response to ULEP;
3. ULEP receives an encrypted Authetication Response, it decrypts the Response and extracts the eIDAS attributes, table 4.16. It takes the eIDAS attributes and translates them into AM attributes, table 4.15, i.e joining the attributes "FirstName" and "FamilyName" to create the attribute "FullName" or changing the format of "DateOfBirth". Then it creates an Authentication Response with the AM attribute set and sends it to AM.
4. Depending on the type of authentication, national or european (dictated by *CountryCode*), AM translates the attributes received from ULEP to IDM attributes and connects to IDM's eDirectory to verify the identity of the user.

5. If a match is found it sends an Authentication Response to the SP with the unique identifier of the master identity of the user and some personal data.

Friendly Name	eIDAS Attribute
First Name	FirstName
Family Name	FamilyName
Citizenship Country	CountryCode
Date of Birth	DateOfBirth
Citizenship ID	IdNumber

Table 4.16: eLogin eIDAS Attribute Set

4.6 eAccess

The EU through the CEF has funded several programs to improve the network and network access of the users in it's member states, notably, EduRoam has been a huge success allowing foreign and domestic students/academic staff to obtain WLAN access through their own university credentials by using the RADIUS protocol.

At it's core eAccess plans to solve a WLAN access problem, it is planned to be used during events organized by universities (such as project meetings, open conferences and seminars). In these events, there are several ways to provide WLAN access, each method with it's own issues. If the event's participants are strictly part of the academic system WLAN access can be exclusively provided by EduRoam, if there exist participants that are not a part of the academic system (corporation delegations or public officials) the most common solution is to set up a temporary valid credential (username/password) that can be used by the participants, either one shared credential that can be used by any participant or one credential for each of them. Other solutions are either more complex and require an even bigger management burden or are so simple that they can pose a security risk (e.g. open network access). eAccess aims to provide WLAN access to all participants even if they are not part of the academic system by verifying the identity of a participant through the eIDAS network, this way it's possible to control who gets access to the WiFi network based on the credentials received.

While eAccess is currently being implemented in 5 different universities that are part of the eID4U project with the goal of being used in said university's events the implementations are not specific to the academic system, which means it can be implemented for other, more general, uses in the future such as hotel's WiFi access (e.g retrieving the passport's number which the guest has checked in with), conferences, seminars and many others.

To accomplish this goal, Zeroshell will be used as a gateway, DHCP server, Captive Portal, and SAML v2.0 SP. ULEP will be used as an application that can relay and transform the Authentication Requests and Authentication Responses between Zeroshell's SP and the portuguese eIDAS Node (Connector). The authentication provider will be the eIDAS Network.

This chapter is divided in four different sections. The first section explains how the client will be inserted into the network managed by Zeroshell. In the second and third chapter the SAML authentication process is explained, the second chapter gives a more general view while the third chapter goes in depth about the complexity of the Authentication Service (AS) used. The fourth chapter details the configuration used in the applications.

4.6.1 Zeroshell Network

As previously mentioned eAccess is a method to provide WiFi access, for the user to begin the authentication process he will have to connect to the internal network of Zeroshell.

One of the main requirements of eAccess is that it must share the physical infrastructure with other network access means already in place at ULisboa, like EDUROAM. It needs to share the AcP and controller with other networks so the way to isolate eAccess was to configure the AcP to announce several wi-fi beacons, each associated with it's VLAN. For an user to use the eAccess service he needs to be in the VLAN where Zeroshell acts as the gateway so they can use Zeroshell's Captive Portal to authenticate and obtain access to broader network resources. An user in the same network as Zeroshell a VLAN was created with the subnet of 192.168.1.0/24, which means there are 255 IPs possible in this network, in the range 192.168.1.254 to 192.168.1.0. Zeroshell is the gateway for this VLAN and it has the IP address of 192.168.1.1. Zeroshell has two network interfaces, the internal network interface, the interface where this VLAN is connected to, and the external network interface that Zeroshell uses to communicate with the Internet (using Network Address Translation (NAT)).

The client will connect to the VLAN through an Service Set Identifier (SSID) that is propagated by an Wireless Access Point (AcP). An IP address in a certain range defined by Zeroshell's Dynamic Host Configuration Protocol (DHCP) server will be given to the client, the DHCP server configuration can be seen in the figure 4.17

4.6.2 eAccess Authentication Flow

When the user obtains its IP address from the DHCP server he becomes a part of the VLAN however he still needs to be authenticated in order to obtain access to broader network resources.

In the following figure 4.18 is the flow of the authentication process implemented. The biggest difference between the eAccess implementation and the flow already explained in chapter 3.4 is the exchange of a standard SAML v2.0 IdP with a complex Authentication Service (AS).

The authentication process can be started with two different methods, either the user is redirected automatically as soon as he connects to the network or the user is redirected only after trying to access an external web service. In either case, the user will be instantly redirected to the Authentication Service (AS), the other redirects will be opaque to the user because Zeroshell provides a feature to automatically

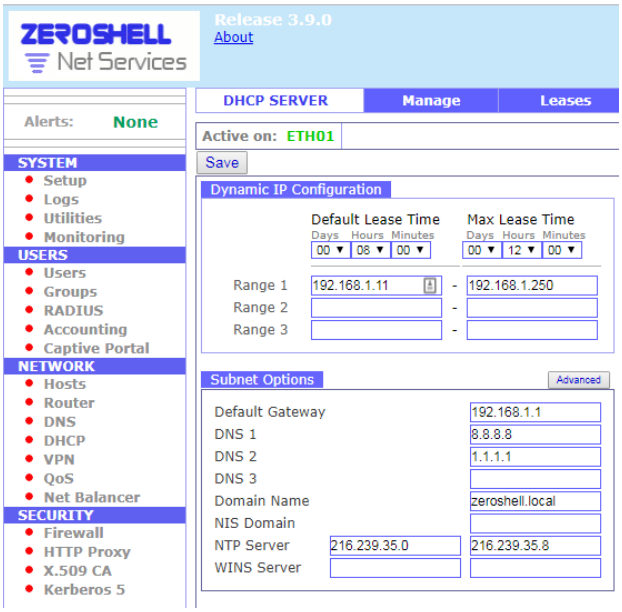

Figure 4.17: Zeroshell's Dynamic Host Configuration Protocol (DHCP) server configuration

start the SAML v2.0 authentication process, instead of showing the Captive Portal page. So the steps (1), (2), (3), and (4) are opaque to the user, as soon as the authentication process starts he will be redirected to the AS. The composition of the AS can be seen in figure 4.19.

After the authentication process inside the AS is completed the sp will receive a SAML Authentication Response and the user will be redirected once again to the SP (5) where the identity of the user will be verified (e.g. part of a list of users) by passing the attributes retrieved during the authentication process through an authorization filter, if the authentication is a success the user is redirected to the gateway (6) and finally to the requested web service (7). The steps (5), (6), and (7) are opaque to the user, as soon as the authentication is completed at the AS the next page the user will see, pending successful authentication, is the web service requested in step (1).

4.6.3 eAccess Authentication Service

The eAccess Authentication Service (AS) is composed of four or more SAML entities and yet all of these entities are part of the external network, the user under normal circumstances should not have access to them, however one of Zeroshell's features is a dynamic and automatically updated whitelist. This whitelist works by interpreting the redirects issued by each of the AS individual entities, so the only entity that needs to be added to the whitelist is the first entity the SP sends the Authentication Request to, this entity is the ULisboa eIDAS Proxy (ULEP).

There are two reasons for Zeroshell's SP to not directly connect to the eIDAS Connector. First the Shibboleth SP cannot decrypt the Authentication Response Assertion the eIDAS Connector sends because the encryption libraries are not supported, so it cannot obtain the attributes of the user and the status of the authentication. Secondly the SP cannot read the Connector's metadata and the Connector

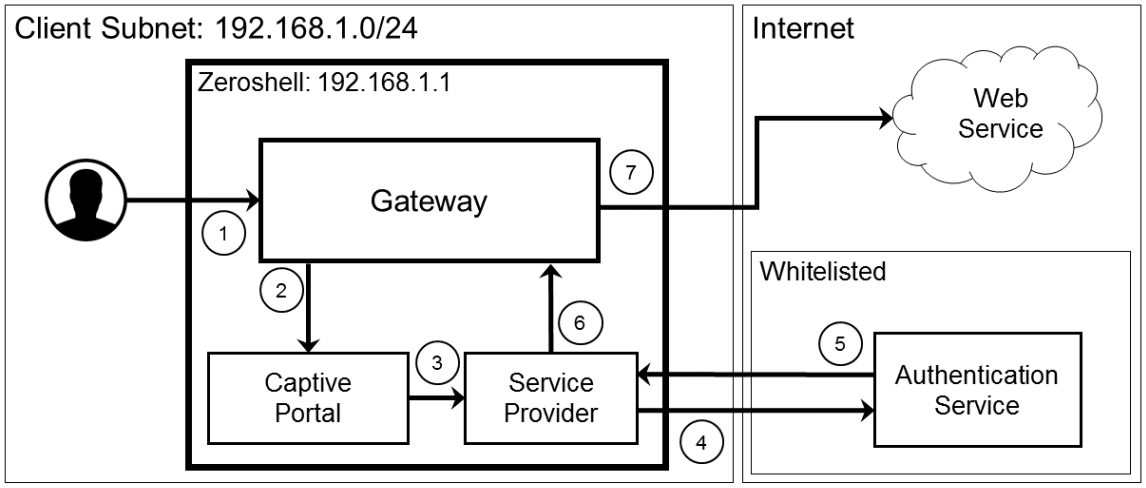

Figure 4.18: eAccess Authentication flow using Zeroshell's Captive Portal with external SAML authentication

cannot read the Service Provider's (SP), which means they cannot send SAML messages between each other. ULEP was designed and developed to solve both of these issues, first it's possible to integrate the metadata of the SP in a local repository and it is much easier to do the same in the SP, second the module that creates and sends the Authentication Requests was based on the demo SP from the eID4U project so it can support the latest, most secure algorithms. The application ULEP like the eIDAS Nodes act as a proxy between the real SP that requests the authentication of an user and the IdP that provides that authentication, by transforming the SAML messages to the standard of the receiver of those messages.

In the figure 4.19 we can see the flow of the authentication process inside the AS. The first entity of the AS will receive the request sent by the SP (4), due to previous mentioned limitations of the SP the application that receives this SAML Request (ULEP) will have to transform it in order to be in compliance to the eIDAS standards, this includes providing a web page where the user chooses his country so the eIDAS Connector knows which eIDAS Service it needs to redirect to. In step (A) ULEP will send an Authentication Request with the attributes in table 4.17, the attribute set required to perform a successful authentication, to the portuguese eIDAS Connector which will promptly redirect the user to the eIDAS Service corresponding to the country the user selected (B). In the eIDAS Service the user will be asked for consent about the attributes that were requested in the Authentication Request made to the eIDAS Connector and then he will be redirected to the national IdP of the country (C) (simplified, different countries have different eID systems, mentioned in chapter 3.2).

The IdP will authenticate the user and will trigger a chain of responses to the previous requests, starting with step (D), it will send an Authentication Response with the attributes in table 4.17, the SAML Assertion will be encrypted by the eIDAS Service and the user will be redirect to the eIDAS Connector (E). Finally ULEP will receive the SAML Response from the eIDAS Connector (F), with an encrypted SAML Assertion. Before ULEP responds to Zeroshell's Shibboleth it decrypts the Assertion so the Shibboleth can retrieve the attributes of the user, triggering the final Authentication Response (5) to the

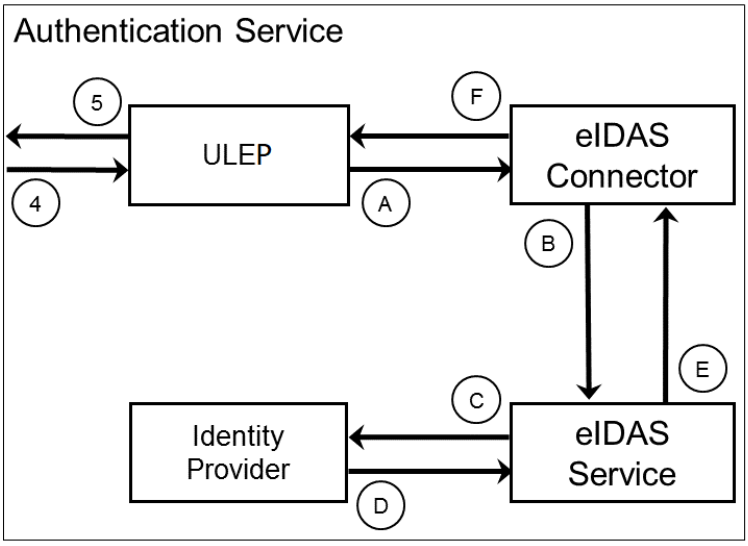

Figure 4.19: eAccess' AS individual components and authentication flow

SP.

4.6.4 Configuration

To improve user experience it was decided the captive portal would not be shown to the user, so the first web page the user sees is ULisboa eIDAS Proxy's (ULEP) country selection page. To obtain this goal we have to configure in Zeroshell's web administration console to automatically redirect the user to the SP instead of waiting for user input.

The process described in chapters 4.6.2 and 4.6.3 can be achieve by configuring the systems that take a part in it in the following ways.

We can access the Shibboleth's SP configuration files through the web administration console and it's in-built text editor. The file shibboleth2.xml has the main configuration of the SP. The main configuration file contains the SAML Authentication Request format, the Service Provider's (SP) metadata parameters (such as the entityID and the AssertionConsumerService), and the connection to the other necessary files.

The keys and certificates used are self-signed in the Service Provider's (SP) case but in the case of ULEP they abide by University of Lisbon's (ULisboa) standard, so a Certificate Signing Request (CSR) is made to Digicert to obtain the certificates that will be used by the proxy.

Both the SP and ULEP create a local metadata repository with the metadata of the other application, since it's not possible to obtain it automatically. To allow the exchange of messages ULEP has to be added to the whitelist of the SP, the rest of the hops in the authentication process will be added automatically by Zeroshell.

To authenticate the user the SP needs to verify the attributes received in it's SAML Response, the

Friendly Name	Uniform Resource Identifier (URI)
First Name	http://eidas.europa.eu/attributes/naturalperson/CurrentFamilyName
Family Name	http://eidas.europa.eu/attributes/naturalperson/CurrentGivenName
Date Of Birth	http://eidas.europa.eu/attributes/naturalperson/DateOfBirth
Person Identifier	http://eidas.europa.eu/attributes/naturalperson/PersonIdentifier

Table 4.17: Minimum Attribute Set for eAccess Authentication

file attribute-mapping.xml has all of the attributes used in this authentication process, the attributes in this file are the ones presented in table 4.17, which contains an attribute's friendly name and it's Uniform Resource Identifier (URI). The set of attributes is the same as the Minimum Data Set (MDS) of the eIDAS Regulamentation except the Current Address, since the SP has no need to verify the address of the user to authenticate him, while the Personal Identifier cannot be used to authenticate an user it is still requested because it's mandatory. So the requested attributes are Personal Identifier, First Name, Last Name, and Birth Date but the attribute Personal Identifier is not used in the authorization filter because there is no way to relate this attribute to the person in question previously. These four attributes are also declared in ULEP and they are the only attributes requested by it to the eidas node. The SP can filter the users that can have access to the network by checking the attributes received in the SAML Response with the rules set in the authorization filter. For example by creating a list of users (using their full name and birth date) that are allowed access and then comparing the users' attributes to the ones received in the SAML Response.

Finally the only alteration needed in the eIDAS Node is to include ULisboa eIDAS Proxy's (ULEP) publicly available metadata in it's SP whitelist.

5 Conclusion

5.1 Discussion

This thesis has provided some maturity to the eIDAS environment, especially on the Portuguese eIDAS Node. Some flaws with the previous implementation of the eIDAS Node in Portugal were discovered and a proposal was made to fix them, which was accepted, these flaws had to do with the translation and mapping of attributes from the Portuguese IdP, Fornecedor de Autenticação (FA), and the eIDAS Service module of the Portuguese eIDAS Node.

The Portuguese eIDAS and 4 others (Italy, Spain, Slovenia and Austria) have also added two different sets of attributes to their supported attributes, the personal attribute set and the sector-specific academic attribute set. The support of these attributes in an eIDAS Node implies the proper retrieval of these attributes from the Member State (MS) Electronic Identity (eID) scheme, which was also implemented in this project. The support of new attributes increases the possible use cases for SP connected to eIDAS, maturing the eIDAS environment and also increasing the incentive for new SP to adopt eIDAS as either an authentication mechanism or a way to retrieve attributes for it's users.

Once more the maturity of the eIDAS environment has been increased by the creation of 3 different eIDAS-enabled services in eRegistration, eLogin and eAccess. The implementation of these services can serve as an example to the community in how to integrate with eIDAS and also the benefits of using eIDAS over other authentication systems. These services have also contributed to an influx of real users into the eIDAS environment.

One of the biggest challenges of eIDAS is it's takeoff, by that I mean the difficulty for the project to gain wide adoption, this issue is shown by the fact that few users know of eIDAS therefore they don't know how to use the system or what is is for. This issue is exacerbated by the lack of usage of the eID systems of some MSs by it's citizens, in recent years the services that use national authentication and the users that use these systems have increased which helps eIDAS because eIDAS relies on the eID infrastructure of each MS and so it relies on citizens knowing how to use their own MS eID.

Despite the lack of real users using the eIDAS Network the most concerning issue for the takeoff challenge is the lack of adoption of the eIDAS system by SPs. The lack of adoption is caused by two main reasons: the lack of real users that can use the eIDAS system but primarily the difficulty of integrating legacy systems with the eIDAS Network due to it's high security requirements.

In this thesis I have presented one possible solution to integrate legacy SPs with the eIDAS Network,

this solution is to create a legacy proxy that can adhere to eIDAS security requirements and can also communicate with the legacy system, this legacy proxy has been connected to two different legacy systems. ULEP is the manifestation of this idea for ULisboa it adheres to all of eIDAS requirements and connects to the legacy systems of ULisboa, NetIQ's Access Manager (AM). This solution can be configured to work with other systems besides ULisboa since it is a eIDAS compatible legacy proxy.

The service eRegistration has also shown the possible utility of adding new attributes to the eIDAS Network, with a wider variety of attributes eIDAS can be used in a wider variety of services. eRegistration has also paved the way for more Portuguese Attribute Providers (APs) to connect to the Portuguese infrastructure through Interoperabilidade na Administração Pública (iAP). By integrating more APs into the Portuguese infrastructure more attributes are available to be used by Portuguese services connected to FA or by foreign services connected to the eIDAS Network. The lessons learned during the development of this service can serve as stepping stones for the integration of other types of attributes into the eIDAS Node for example e-Health attributes already available in the Portuguese infrastructure and these lessons can also help with the AP connection to iAP.

Finally these services have improved the services of ULisboa: eRegistration has improved the ERASMUS application process for foreign users also decreasing errors in the process and decreasing the need of validation resources at the International Relations Office; eLogin has improved the authentication system of ULisboa by providing a new IdP choice for the user, both for national users and foreign users (it has also resulted in the normalization of data in ULisboa's identity management system); eAccess has improved wi-fi access in ULisboa by adding a new network to the university's Access Point (AcP).

5.2 Threat Model

To analyze the possible vulnerabilities of the systems I used the STRIDE model, this model is used to analyze threats to a system or a group of systems, the threats are a violation of a security property of a system, the threats and the property these threats attack are:

- **Spoofing** - Authenticity;
- **Tampering** - Integrity;
- **Repudiation** - Non-repudiability;
- **Information disclosure** - Confidentiality;
- **Denial of service** - Availability;
- **Elevation of privilege** - Authorization;

Both eIDAS and the SAML standard have methods to combat these threats are to ensure the systems possess the related property. The methods to prevent these attacks are:

- **Spoofing** - The use of a PKI and properly signed public certificates prevents an enemy from spoofing a trusted entity.

  In ULEP, AM, eIDAS and FenixEdu every Authentication Request and Authentication Response is signed, these signatures are created using the private key of the entity and can be verified using the public key of said entity, these public keys are extract from the metadata of an entity or received confidentially;

- **Tampering** - To prevent tampering from a third party every Authentication Request and Authentication Response is signed, the signature of the message is created using a digest algorithm on the message itself, if the message is tampered the digest algorithm will create a different hash and the validation of the signature will fail;

- **Repudiation** - The signature itself is proof of the creator of the message. Only the holder of the private key can sign a message in a way the public key will be able to validate the signature;

- **Information disclosure** - The method to prevent a third party to obtain the information exchanged between entities is to encrypt the messages. The eIDAS Node encrypt the messages between each other so a third party cannot obtain any information from their communication, however legacy systems such as AM don't have the functionally to encrypt and decrypt messages, or the encryption algorithm is too weak, so the information can be obtained when the messages are not encrypt. The cases where a third party can obtain information about an user is in the SP-AM connection, AM-ULEP connection and between the eIDAS Service of a MS and it's IdP (like the case in Portugal, because either FA does not support encryption or they have choosen not to use it);

- **Denial of service** - Distributed Denial of Service is one of the biggest concerns to these systems, ULisboa has measures in place to protected against these types of attacks but a suficiently large attack can bring the availability of the services down for real users. The measures employeed by the eIDAS Nodes and the national IdPs are at the discretion of it's MS;

- **Elevation of privilege** - Not applicable. However eIDAS has the Level Of Assurance (LoA) which is a method to minimize the trust in attributes received from less secure authentication methods;

5.3 Future Work

Due to the level of maturity of the eIDAS standard there is still a lot to improve. From the development efforts required by third parties (SP) to implemented the standard and also to develop new use cases for the users themselves. ULEP can be developed into a proper solution that can be used to connect legacy systems to the eIDAS Network.

In the portuguese infrastructure it is impossible to authenticate foreign users and therefore it's impossible for a foreign citizen to obtain their academic attributes from a portuguese university by using the eIDAS Network, this could be achieved by allowing foreign citizens to create a CMD using their eIDAS

authentication instead of only being able to create one with their portuguese citizen card, this would allow an user to authenticate and exchange a few basic personal attributes but it would also allow the user to retrieve academic attributes through iAP.

A process of development has already started in the iAP to give the option to choose where the iAP retrieves one attribute from, this is especially important for portuguese healthcare attributes (obtain the same attribute from different hospitals) but it is also necessary for academic attributes if the student is from anywhere besides the University of Lisbon (ULisboa). Related to this point to create a more thorough academic infrastructure in Portugal other universities can connect to iAP to be able to provide academic attributes through their own academic systems.

It could also be interesting to create an academic portuguese infrastructure were Portuguese universities could exchange attributes electronically between universities by authenticating in FA and extracting those attributes from the university through iAP, this academic infrastructure could use the academic attributes of this project as a starting point and then develop a standard for portuguese academic attributes.

To better extract the benefits of creating a common standard for the european union a high level assimilation needs to be achieved, a good example is the mobile phone banking standard implemented by SIBS in Portugal MB WAY or Paym in the UK. To achieve a high level of assimilation a effort of transferring legacy systems or adding eIDAS functionality to existing systems is necessary e.g. make NetIQ's Access Manager (AM) eIDAS compliant.

5.4 Final Thoughts

Complex Identity Management (IdM) systems are needed to solve identity validation issues on a university, national or european scale. These complex systems have to thread a fine line between usability and giving too many options to users. In order to support various independent systems with different implementations standards have to be created, validated and implemented by all parties involved.

This project developed a possible inclusion to the eIDAS standard by implementing new attributes, personal and academic, these attributes would allow the independent services to extract more value from the eIDAS protocol being implemented little by little all across europe. The creation of new use cases for eIDAS compliant services is a stimulant for new organizations to adopt the eIDAS standard, maturing the environment itself. Standard work better the more ubiquitous they are, just like the usb battery charger standard has completely improved the landscape for the users.

The objective to improve the eIDAS standard and also the improvement of the national Electronic Identity (eID) infrastructure was complete with also the implementation of three services as a proof of concept using the proposed changes to the standard.

References

[1] E. Commission, “A Digital Single Market Strategy for Europe,” https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52015DC0192, 2015.

[2] E. Union, “Regulation (eu) no 910/2014 of the european parliament and of the council of 23 july 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing directive 1999/93/ec,” in *ISSE 2010 Securing Electronic Business Processes*. European Union, 2014.

[3] H. Leitold and B. Zwattendorfer, “Stork: architecture, implementation and pilots,” in *ISSE 2010 Securing Electronic Business Processes*. Springer, 2011, pp. 131–142.

[4] D. Berbecaru and A. Lioy, “On integration of academic attributes in the eidas infrastructure to support cross-border services,” in *2018 22nd International Conference on System Theory, Control and Computing (ICSTCC)*. IEEE, 2018, pp. 691–696.

[5] E. Bertino and K. Takahashi, *Identity management: Concepts, technologies, and systems*. Artech House, 2010.

[6] T. J. Smedinghoff, “Introduction to online identity management,” 2008.

[7] M. Harrop, “Identity management,” in *The Cottingham Group, ETSI Security Workshop*, 2009.

[8] A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang, “The tangled web of password reuse.” in *NDSS*, vol. 14, 2014, pp. 23–26.

[9] A. Pashalidis and C. J. Mitchell, “A taxonomy of single sign-on systems,” in *Australasian Conference on Information Security and Privacy*. Springer, 2003, pp. 249–264.

[10] A. Thusoo, Z. Shao, S. Anthony, D. Borthakur, N. Jain, J. Sen Sarma, R. Murthy, and H. Liu, “Data warehousing and analytics infrastructure at facebook,” in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 2010, pp. 1013–1020.

[11] J. A. Buchmann, E. Karatsiolis, and A. Wiesmaier, *Introduction to public key infrastructures*. Springer Science & Business Media, 2013.

[12] “Recommendation x.509 itu-t, information technology – open systems interconnection – the directory: public-key and attribute certificate frameworks,” Aug 2005.

[13] P. Zimmermann, “The international pgp home page, http://www.pgpi.org/,” PGP, 1992.

[14] OASIS, “Saml assertions,” http://saml.xml.org/assertions, accessed: 2019-09.

[15] S. OASIS, “Security assertion markup language (saml) 2.0 technical overview, working draft,” 2005.

[16] J. Hodges, E. Maler, N. Ragouzis, E. J. Hughes, P. Madsen, E. I. Reid, P. Austel, M. Hondo, M. McIntosh, T. Nadalin *et al.*, “Glossary for the oasis security assertion markup language (saml) v2. 0,” 2004.

[17] S. Cantor, J. Moreh, R. Philpott, and E. Maler, “Metadata for the oasis security assertion markup language (saml) v2. 0,” *OASIS Standard (March 2005)*, 2004.

[18] C. Ribeiro, H. Leitold, S. Esposito, and D. Mitzam, “Stork: a real, heterogeneous, large-scale eid management system,” *International Journal of Information Security*, vol. 17, no. 5, pp. 569–585, 2018.

[19] BSI, “Secure identity across borders linked (stork) 2.0 project (2012–2015) . available online: https://ec.europa.eu/digital-single-market/en/content/stork-take-your-e-identity-you-everywhere-eu.”

[20] D. Berbecaru, A. Lioy, and C. Cameroni, “Electronic identification for universities: Building cross-border services based on the eidas infrastructure,” *Information*, vol. 10, no. 6, p. 210, 2019.

[21] “eidas saml message format, version 1.1. available online:https://ec.europa.eu/cefdigital/wiki/download/attachments/80183964/eIDAS%20Message%20Format_v1.1-2.pdf?version=1&modificationDate=1497252919575&api=v2.”

[22] “Hughes, j.; cantor, s.; hodges, j.; hirsch, f.; mishra, p.; philpott, r.; maler, e.profiles for the oasis security assertion markup language (saml) v2.0. oasis standard. march 2005. available online: http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf.”

[23] BSI, “Technical guideline tr-03110-1 advanced securitymechanisms for machine readable travel documentspart 1 v 2.20,” https://www.bsi.bund.de/EN/Publications/TechnicalGuidelines/TR03110/BSITR03110.htm, 2015.

[24] A. para a Modernização Administrativa, “Fornecedor de autenticaÇÃo da administraÇÃo pÚblica portuguesa,” AMA.

[25] ——, “Requisitos técnicos para utilização da plataforma de interoperabilidade da administração pública,” AMA, 2009.

[26] F. Ricciardi, “Zeroshell linux distribtion,” https://zeroshell.org/, accessed: 2019-07.

[27] “Shibboleth service provider releases,” https://wiki.shibboleth.net/confluence/display/SP3/ReleaseNotes, accessed: 2019-09.

6 Annex

6.1 ULisboa's eID4U Node

The eID4U specifications must be developed and tested in a testing environment so the changes are approved to be used in the national eIDAS nodes. For that purpose a virtual machine was created in ULisboa running a eID4U Node, this machine was created in the public domain `eidas.ulisboa.pt`. The eIDAS node installed in the domain is a copy and reconfiguration of the eID4U source project available at the Github repository `https://github.com/eID4U/eIDAS-node`. The repository encapsulates 4 applications:

- **Demo-SP** a dummy Service Provider (SP) which provides a Graphical User Interface (GUI) to facilitate the manual testing of the nodes. It provides the choice of destination eIDAS node, source eIDAS node and attributes requested in the SAML Authentication Request;
- **eIDAS Connector** a proxy connected to the SP, it receives it's SAML Requests from the SP and forwards them to the eIDAS Service of the destination node. After receiving a SAML Response from the destination eIDAS Service it forwards the SAML Response to the SP;
- **eIDAS Service** a proxy connected to the IdP, it receives a SAML Request from the source eIDAS Node's eIDAS Connector and forwards it to the MS authentication infrastructe, namely the IdP. After receiving a SAML Response from the IdP it forwards it to the source's eIDAS Node eIDAS Connector.
- **Demo-IdP** a dummy Identity Provider (IdP) that provides a GUI where the user can choose which attributes and its respective values sent in the SAML Authentication Response.

The eIDAS applications are installed in a virtual machine with a CentOS 7.5 Operating System, running on an ApacheTomcat 7.0.76 Java Web Container. To get the eIDAS package up and running we had to follow the following straightforward procedure:

(A) Build the executables;

(B) Deploy the applications in Tomcat with the default configuration;

(C) Configure environment variables;

(D) Create private keys, certificates and configure the keystore;

(E) Configure each application;

(F) Install the certificates of the other eIDAS Nodes.

6.1.1 Build the executables

First we must clone the repository `https://github.com/eID4U/eIDAS-node` to obtain the eIDAS source code. After, follow the installation guide "eIDAS-Node Installation and Configuration Guide Version 1.4.3", available in the eIDAS Node integration package at the European Commission site. The eIDAS source code is divided into modules such as the SP, the IdP or the Common module.

As described in the installation manual, to run the executables on a Tomcat container we have to run the command below in the main folder, EIDAS-Parent:

```
mvn clean install -P tomcat
```

The command generates 3 .war files in the modules EIDAS-Node, EIDAS-SP and EIDAS-IdP. The generated SP war file contains the Demo-SP application, the IDP file contains the Demo-IdP and the EIDAS-Node file contains both the eIDAS Connector and the eIDAS Service. These files were then transferred to the eIDAS virtual machine.

6.1.2 Deploy the applications in Tomcat with the default configuration

We need to start by installing Apache Tomcat Java web server in the eIDAS machine and then copying the 3 war files to /usr/share/tomcat/webapps so tomcat can recognize the eIDAS applications.

Due to a conflict in libraries, you have to copy the libraries provided in "AdditionalFiles/endorsed", from the eID4U source code, to Tomcat's library repository the *lib* folder in /usr/share/tomcat/lib

```
resolver-2.9.1.jar | serializer.jar | xalan-2.7.2.jar
xercesImpl-2.11.0.jar | xml-apis-1.4.01.jar
```

After adding the libraries to the server, the eIDAS configuration files were copied to the server, to the Tomcat directory because the user running tomcat already has permissions to read that directory.

6.1.3 Configure environment variable

Each of the eIDAS applications has a folder with configuration files, in order for each application to access the proper folder the administrator must set up the eIDAS environment variables.

There are 4 folders in the eIDAS configuration:

- eIDAS folder - parent folder;
- Keystore folder;
- Specific folder;
- IdP folder;
- SP folder.

Three environment variables need to be created, this was done in /etc/tomcat/tomcat.conf, in order for the variables to be set everytime Tomcat execute.

- EIDAS_CONFIG_REPOSITORY - Pointing to the main eIDAS folder;
- IDP_CONFIG_REPOSITORY - Pointing to the IdP folder;
- SP_CONFIG_REPOSITORY - Pointing to the SP folder.

The expected configuration hierarchy is the eIDAS folder being the parent and the other three folder being it's sub-folder, this hierarchy must not be changed because the eIDAS applications expect the Keystore folder to be on the same level as the IdP and SP folders and below the eIDAS folder. The Keystore folder is very important because it contains the keystore file which has all the trusted certificates and also the singing and encryption keys to be used by each module.

6.1.4 Create private keys, certificates and configure the keystore

The default keystore comes with a default private key and self-signed public certificate, however to use the eIDAS modules with foreign partners (to our organization) a public certificate signed by a trusted third party, a Certificate Authority (CA), must be obtained or these partners would have to import our self signed certificate into their trust stores.

In an initial phase we opted for the self-signed certificates and then we transitioned to the CA signed public certificate. In a first instance we used one pair of private/public keys and one keystore for each application, since all 4 applications would be running on the same server we could use only use pair of public/private keys and also one keystore, this adds the benefit of all of our applications trusting eachother due to them using the same certificate.

While most SAML applications such as the IdPs at ULisboa and FA use 2048 bit private keys, eIDAS requires the private key to be at least 4096 bits. Also the key needs to use the signature algorithm SHA256withRSA or a superior one, at the moment eIDAS doesn't support a stronger algorithm.

6.1.5 Configure each application

With the keystore configure all we need is to correctly define the keys and certificates properties in our configuration files, these configuration files are in the folder the environment variables point towards. The values related to the keys and certificates are in the files *EncryptModule* and *SignModule*, the key data must be put for each application. In these files we changed the data of the certificates used,such as the path of the keystores, the passwords, the serial numbers, and the issuers of the keys.

Then we must configure the attributes supported in each application and then configure *eidas.xml* with the data of our foreign partners and of our own application (return addresses, whitelisted applications and whitelisted metadata).

6.1.6 Install the certificates of the other eIDAS Nodes

In the final step we exchanged our Metadata URL and the public certificates with the other MS. For most members we received only one certificate that verifies the identity of both the eIDAS Connector and the eIDAS Service. So we import the certificate to our application's keystore and create a trust relationship between our eIDAS Node and theirs. In the case of Italy we received two different certificates, one for the eIDAS Connector and another for the eIDAS Service, in this case we just added both to the same keystore.

In order for the connections to be available when using the Demo-SP application we had to change the *eidas.xml* file updating the whitelist with the Metadata URLs provided by the MSs. Also we had to add each eIDAS Service endpoint to the *eidas.xml* file. To be able to select the new citizen countries defined in *eidas.xml* we need to add a property for each new eIDAS Node we added. In the file *sp.properties* the property is created so it is available to be selected in the Demo-SP application.

6.2 ULisboa eIDAS Proxy (ULEP)

ULisboa eIDAS Proxy (ULEP) is a Legacy Proxy developed to connect AM and the Portuguese eIDAS Node. It is running in a tomcat instance on a

For that purpose a virtual machine was created in ULisboa running a eID4U Node, . ULEP is installed in a virtual machine with a CentOS 7.5 Operating System, running on an ApacheTomcat 7.0.76 Java Web Container. This machine is accessible in the public domain `https://eidas.ulisboa.pt/ULEP/`.

The project has the name IdPBroker in ULisboa's git.

6.2.1 Metadata

Both of ULEP's metadata is available at `https://eidas.ulisboa.pt/ULEP/`.

The IdP module metadata is available at `https://eidas.ulisboa.pt/ULEP/IdPmetadata`.

The SP module metadata is abailable at `https://eidas.ulisboa.pt/ULEP/SPmetadata`.

6.2.2 New Version Release

ULEP was developed using the Spring Boot Framework using maven to build the project. To build a deployable version of the code you need to add two libraries not available online to your mvn folder, usually located in C:/Users/<username>/.m2/repository. These libraries are in the folder from the project src/main/resources/mvn-repository, copy the two sub-folders to the mvn folder. After adding the new libraries update the project with the new libraries, in eclipse right-click and select the option Maven - Update Project.

To compile the project into a WAR file run the command: mvn clean package, then copy the generated ULEP.war file from the /target folder into the server and into the /usr/share/tomcat/webapps/ folder replacing the old one. Restart the tomcat server to run the new version of ULEP.

6.2.3 Configuration

ULEP uses the same configuration system as the eIDAS node, it contains a keystore folder, IdP folder and SP folder. The keystore folder contains the keystore file with the keys and certificates used in the metadata generation, message signing and response certificate validation. The IdP folder contains all the information from the IdP sub-application of ULEP used in the connection with ULisboa's AM while the SP folder contains all the information from the SP sub-application of ULEP used to connect to the portuguese eIDAS Service and eIDAS Network.

Just like in the eIDAS application, chapter 6.1, these folders are set up by environment variables, tomcat can load environment variables when they are set in /etc/tomcat/tomcat.conf. The name of these variables are dictated by the file ulepEnvironmentContext.xml at the time of building the application. Two variables were created ULEP_IDP and ULEP_SP corresponding to the IdP and SP folder respectively. The keystore folder must always be in the same directory of the SP and IdP folder.

Tomcat has to have permission to read the configuration folders, so the easiest way to set it up is to create a configuration folder inside tomcat, /usr/share/tomcat, with the three necessary folders inside. The file readme_text in /resources/configuration folder has more information.

The files idp.properties and sp.properties from each respective folder have the properties used in each sub-application. The SP properties file has information about the ULEP SP sub-application (endpoint URLs and metadata information) and also which eIDAS environment to connect to: production,

pre-production, ulisboa env. The IdP properties file has information about the ULEP IdP sub-application (endpoint URLs, metadata information, keystore information) but it also controls the test mode and which AM environment it's connected to: production or development.

6.2.4 Generating new Public Certificate

Certificates expire, so we have to generate a new public certificate from our private/public key pair in order for AM and eIDAS Node to trust our signature. The private key is to be kept a secret, it must not be shared with anyone. I recommend using *keystore explorer* to have a GUI interface to manage the keystore.

The following steps describe how to create a new public certificate while using the same private/public key pair:

- Open Keystore, eidas_ulisboa_pt.jks, from \src\main\resources\configuration\keystore and unlock public/private key pair with it's password;
- Generate a Certificate Signing Request (CSR) from it's key pair;
- Have a Certificate Authority (CA) issue a signed public certificate, ULisboa uses Digicert;
- Import the CA signed certificate into the key pair, with option "Import CA Reply";
- Export certificate chain from key pair;
- Send the certificate exported to the responsible teams of the AM and Portuguese eIDAS Node, sistemas@suporte.ulisboa.pt and AMA.

If you want to use a new key pair you must create a key with preferably 4096 bits and supporting SHA256withRSA or higher for it's signature algorithm. After generating the new key pair follow the previously mentioned steps.

Now we have to update ULEP's configuration files so the application can generate the metadata and sign with the new certificate created. The files that need to be update are:

- the keystore file, eidas_ulisboa_pt.jks, needs to be imported to ULEP;
- in SP folder EncryptModule and SignModule files;
- in IdP folder EncryptModule, SignModule and idp.properties files.

With the public certificate generated and the configuration files updated we reload tomcat with *service restart tomcat* and the metadata generation and message signing will start using the new certificate. More information can be found in readme_keystore.txt in keystore folder.

To obtain the new metadata we can access eidas.ulisboa.pt/ULEP/ and select the IdP metadata or SP metadata.

6.2.5 Importing Certificates

Certificates expire, so we have to periodically import the new certificates from AM and eIDAS into ULEP's keystore. The process to add a new certificate to ULEP is the same, independent from certificate type and the entity it belongs to.

It's recommended to import the certificate on an user's machine and then copy it to the ULEP machine, I used the program Keystore Explorer (`https://keystore-explorer.org/`) to import certificates and manage the keystore due to it's GUI.

To import a **Trusted Certificate** into ULEP's keystore one must follow these steps:

1. Obtain the main **Trusted Certificate** from the source (.p7b, .pem, .cer) and all trusted certificates from it's certificate path;
2. Locate Keystore, in git project it's in \src\main\resources\configuration\keystore, and open it using it's password;
3. Locate the file readme_keystore.txt in the same folder;
4. Import the **Trusted Certificate**, and the certificates in it's path, into the ULEP Keystore;
5. Stop the ULEP application or stop tomcat on the ULEP machine by running the command "service tomcat stop";
6. Copy the Keystore into the keystore configuration folder, details in chapter 6.2.3, from the ULEP machine. The original configured folder is in /usr/share/tomcat/configuration/keystore;
7. Start tomcat or ULEP again by running the command "service tomcat start" (If step 5 was skipped, restart instead).