UNIVERSIDADE DA BEIRA INTERIOR
Engenharia

Authentication and Identity Management for the EPOS Project

José Pedro da Paula Manteigueiro

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática
(2º ciclo de estudos)

Orientador: Prof. Doutor Paul Andrew Crocker
Co-orientador: Prof. Doutor Carlos Manuel Chorro Simões Barrico

Covilhã, setembro de 2020

Dedicatória

Dedico este trabalho aos meus pais

Agradecimentos

A concretização desta dissertação não seria possível sem o apoio de muitas pessoas, pelo que deixo aqui os meus agradecimentos. Agradeço o apoio incondicional da minha família, em especial dos meus pais e irmão, que sempre estiveram presentes durante este percurso. Agradeço à Margarida por todo o apoio, ajuda e motivação que me deu, fazendo-me dar o meu melhor. Agradeço aos meus amigos e colegas que sempre me apoiaram e animaram. Agradeço aos meus professores pelo conhecimento transmitido durante o meu percurso académico, em especial aos meus orientadores, Prof. Doutor Paul Crocker e Prof. Doutor Carlos Barrico que sempre estiveram disponíveis para me ajudar e esclarecer todas as dúvidas que surgiram, bem como para orientar o meu trabalho.

Muito obrigado a todos.

Resumo

A importância da autenticação e gestão de identidades, de que dependemos inconscientemente, aumenta com o crescimento do número de serviços online ao nosso dispor. No EPOS, devido à disponibilização e gestão de dados heterogéneos de várias entidades, que podem ser públicas ou privadas, a existência de um sistema de autenticação e gestão de identidades é também crucial, em que o controlo e identificação do acesso a estes dados é a chave. Numa fase de desenvolvimento dos serviços, estes módulos de autenticação e autorização podem ser diretamente implementados e é possível existir uma adaptação do software aos mesmos. No entanto, há serviços já existentes, cujas alterações implicam mudanças de grande escala e uma reformulação de todo o sistema, e como tal não é exequível fazer alterações diretas aos mesmos. Esta dissertação aborda o desenvolvimento de um sistema de autenticação e autorização seguro e interoperável, associado a uma correta gestão de identidades e um módulo de controlo, identificando os problemas encontrados e propondo soluções para os mesmos. Este desenvolvimento é aplicado num dos serviços do TCS GNSS Data and Products e servirá como caso de estudo. Embora os mecanismos de autenticação tenham melhorado continuamente ao longo dos anos, com a adição de vários fatores de autenticação, ainda não existe um método único e claro de como a autenticação deve ser feita. Novas ameaças estão sempre a surgir e os sistemas atuais precisam de se adaptar e melhorar, mantendo um equilíbrio entre segurança e usabilidade. O nosso objetivo é propor um sistema que possa aliar a segurança a uma boa experiência para o utilizador, e que possa ser utilizado não só nos serviços do TCS, mas também em outros serviços web que enfrentem problemas semelhantes.

Palavras-chave

Acesso, Autenticação, Autorização, Controlo, EPOS, Gestão de Identidade, GNSS, Segurança, Serviços Web

Resumo alargado

A importância da autenticação e gestão de identidades, de que dependemos inconscientemente, aumenta com o crescimento do número de serviços online ao nosso dispor. No EPOS, devido à disponibilização e gestão de dados heterogéneos de várias entidades, que podem ser públicas ou privadas, a existência de um sistema de autenticação e gestão de identidades é também crucial, em que o controlo e identificação do acesso a estes dados é a chave. Numa fase de desenvolvimento dos serviços, estes módulos de autenticação e autorização podem ser diretamente implementados e é possível existir uma adaptação do software aos mesmos. No entanto, há serviços já existentes, cujas alterações implicam mudanças de grande escala e uma reformulação de todo o sistema, e como tal não é exequível fazer alterações diretas aos mesmos. Esta dissertação aborda o desenvolvimento de um sistema de autenticação e autorização seguro e interoperável, associado a uma correta gestão de identidades e um módulo de controlo, identificando os problemas encontrados e propondo soluções para os mesmos. Este desenvolvimento é aplicado num dos serviços do TCS GNSS Data and Products e servirá como caso de estudo. Embora os mecanismos de autenticação tenham melhorado continuamente ao longo dos anos, com a adição de vários fatores de autenticação, ainda não existe um método único e claro de como a autenticação deve ser feita. Novas ameaças estão sempre a surgir e os sistemas atuais precisam de se adaptar e melhorar, mantendo um equilíbrio entre segurança e usabilidade. O nosso objetivo é propor um sistema que possa aliar a segurança a uma boa experiência para o utilizador, e que possa ser utilizado não só nos serviços do TCS, mas também em outros serviços web que enfrentem problemas semelhantes.

No primeiro capítulo, é introduzido ao leitor o que é a autenticação e gestão de identidades, bem como os problemas que existem na atualidade. É também introduzido o projeto EPOS, no qual incidirá o foco deste trabalho. Em seguida, são especificadas as dificuldades encontradas e quais as nossas propostas para as resolver. Por último, é apresentada a estrutura do documento.

No segundo capítulo, é feita uma revisão da literatura que existe sobre esta matéria. São apresentadas várias formas de autenticação e controlo de identidades, sendo especificados os protocolos atuais e enunciadas algumas das suas vantagens e desvantagens. Os métodos de identificação baseados no cartão de cidadão português são realçados. Introduzimos também os tipos fatores de autenticação existentes, dando a conhecer algumas das suas características. Por último, introduzimos um método de autenticação que julgamos que irá ganhar popularidade num futuro próximo.

No terceiro capítulo, é dado a conhecer ao leitor como foi feita a implementação de um sistema completo de autenticação, autorização, controlo e gestão de identidades num dos serviços do TCS GNSS Data and Products. Em maior detalhe, é apresentada a resolução dos problemas previamente enunciados e demonstrado o funcionamento do sistema atual. A resolução do problema de múltiplas contas para uma só entidade é destacado dado ser um problema que afeta bastantes serviços online. O funcionamento de uma autenticação e autorização via CLI é também especificado. Duas formas de assegurar uma autenticação multi fator no sistema são introduzidas. São ainda referidas outras implementações deste sistema num contexto diferente.

No quarto capítulo, é feita uma análise de usabilidade e segurança ao sistema implementado, com base nas formas de autenticação e métodos utilizados.

Por fim, no quinto capítulo, é dada uma conclusão sobre o trabalho realizado, bem como as contribuições que surgiram do mesmo, incluindo um artigo submetido para uma conferência, e as nossas opiniões sobre o que deve ser melhorado nos sistemas atuais e futuros.

Abstract

The increase in the number of online services emphasizes the value of authentication and identity management that we, even without realizing, depend on. In EPOS this authentication and identity management are also crucial, by dealing and being responsible for large amounts of heterogeneous data in multiple formats and from various providers, that can be public or private. Controlling and identify the access to this data is the key. For this purpose, it is necessary to create a system capable of authenticating, authorizing, and account the usage of these services. While services in a development phase can have authentication and authorization modules directly implemented in them, this is not an option for legacy services that cannot be modified. This thesis regards the issue of providing secure and interoperable authentication and authorization framework, associated with correct identity management and an accounting module, stating the difficulties faced and how to be addressed. These issues are approached by implementing the proposed methods in one of the GNSS Data and Products TCS services, that will serve as a study case. While authentication mechanisms have improved constantly over the years, with the addition of multiple authentication factors, there is still not a clear and defined way of how authentication should be done. New security threats are always showing up, and authentication systems need to adapt and improve while maintaining a balance between security and usability. Our goal is, therefore, to propose a system that can provide a good user experience allied to security, which can be used in the TCS services or other web services facing similar problems.

Keywords

Access, Accounting, Authentication, Authorization, EPOS, GNSS, Identity Management, Security, Web Services

Contents

List of Figures

List of Tables

Acronyms

AAA Authentication, Authorization, and Accounting

AAI Authentication and Authorization Infrastructure

AMA *Agência para a Modernização Administrativa*

API Application Programming Interface

C4G Collaboratory for Geosciences

CC *Cartão de Cidadão*

CLI Command Line Interface

CMD *Chave Móvel Digital*

CRL Certificate Revocation List

ECC European Citizen Card

eID Electronic Identification

EPOS European Plate Observing System

ERR Equal Error Rate

FAR False Acceptance Rate

FRR False Rejection Rate

GDPR General Data Protection Regulation

GLASS Geodetic Linking Advanced Software System

GNSS Global Navigation Satellite System

GUI Graphical User Interface

HMAC Hash-based Message Authentication Code

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

IAS Identification, Authentication, and Signature

ICS Integrated Core Services

IDM Identity Manager

IdP Identity Provider

IETF Internet Engineering Task Force

IP Internet Protocol

ISO International Organization for Standardization

JSON JavaScript Object Notation

JWT JSON Web Token

LDAP Lightweight Directory Access Protocol

MFA Multi-Factor Authentication

OCSP Online Certificate Status Protocol

OTP One Time Password

PIN Personal Identification Number

PKI Public Key Infrastructure

QR Quick Response

RFC Request For Comments

RINEX Receiver Independent Exchange

RP Relying Party

SAML Security Assertion Markup Language

SES Solid Earth Sciences

SFA Single-Factor Authentication

SHA Secure Hash Algorithm

SMS Short Message Service

SP Service Provider

SPA Single Page Application

SSO Single Sign-On

TCS Thematic Core Services

U2F Universal 2nd Factor

UBI *Universidade da Beira Interior*

URI Uniform Resource Identifier

URL Uniform Resource Locator

XML Extensible Markup Language

XRD Extensible Resource Descriptor

XRI Extensible Resource Identifier

Chapter 1

Introduction

1.1 Introduction

This thesis regards the problem of having a secure interoperable Authentication, Authorization, and Accounting (AAA) system for the European Plate Observing System (EPOS) project, that is capable of providing the best experience to the user, by proving to be quick, intuitive, and effortless, without disregarding the security and privacy components, and have an identity management of the users. Our proposal is the implementation of a complete authentication system that solves all the EPOS community needs regarding, not only authentication, but also authorization and accounting, on the web services that are currently being developed for the project. The work developed in this thesis can and should be used to improve other web services that face similar problems. The Global Navigation Satellite System (GNSS) Products Portal web service will be the main focus and case study of this thesis, since we are involved in its development. In order to have a better understanding of what kind of AAA system will better suit the needs of the project, a general idea of what type of users will be consuming our services is necessary. For a clarification of this matter, we will briefly introduce what the EPOS project is and who is it directed to.

The EPOS project is a long-term plan to facilitate integrated use of data, data products, and facilities from distributed research infrastructures, for Solid Earth Sciences (SES) in Europe. Bringing together a great diversity of researchers, experts, and infrastructures, with the goal of developing new concepts and tools for accurate, durable, and sustainable answers to societal questions, that are relevant to the environment and human welfare [1]. The huge volume of data handled demands a new integrated approach to a collaboration between researchers and scientists, that is now possible thanks to the advances in information technology. EPOS aims to link research communities, like earth scientists, through a multidisciplinary research platform, providing new ways to access data, quality assured metadata and tools for analysis, while also promoting the development of new data products and services [2]. Not only scientists and researchers, but also the governments, the private sector, and society in general should benefit from the work developed by EPOS. Seamless access to national and European data and services is expected to attract a variety of diverse users, that should be taken in consideration when developing web services and, specifically, when building an authentication system [3].

The EPOS work package 10 — GNSS Data & Products, handles the dissemination of file-based GNSS data and derived products, which is the work group where *Universidade da Beira Interior* (UBI) is integrated. Specifically, this team is in charge of the development of the GNSS Products Portal [4], a web service that disseminates GNSS data and data products of EPOS analysis centers, like daily and weekly GNSS time series, estimated velocities, and strain rate maps, for various regions in Europe. From the Graphical User Interface (GUI) of this service, it is possible to enable stations visualization over a world map, that allows users to perform actions over the

displayed stations and corresponding metadata, in order to retrieve the desired information [5]. While some of this data is publicly available, some data providers want to have an embargo on their data, restricting the access of it during a certain time and to a limited number of users. To satisfy this requirements, it is necessary to create an Authentication and Authorization Infrastructure (AAI) to authenticate users and verify their permissions.

In short, authentication is the process by which a user proves his identity before an entity. While there are multiple ways on how a user can prove his identity before a system, citing [6], all the approaches for human authentication rely on at least one of the following factors:

- Something you know — usually a Personal Identification Number (PIN) or a password. This is the most common way of authentication, but also most vulnerable since the user has to remember it, commonly making it simpler, or write it somewhere;
- Something you own — like a smart card, a pen drive, or another type of physical object, like a Yubikey. The downside is that the user has to carry it with them all the time;
- Something you are — like a fingerprint, an iris of the eye, or other biometric data;

It is possible that one factor is present more than once, as is the case with a password and PIN system. These factors may, or may not, be aggregated depending on the desired security level of the system. In order to increase the security level, a combination of different types of factors can be used, creating a Multi-Factor Authentication (MFA) system [7].

Nowadays, users are required to authenticate to an increasing number of services: e-mails, social media, bank accounts, online shops, and many others. This translates in a huge increase of the number of identities that a single user can have. For each service, the user has to prove their identity by performing an authentication. This process usually requires the user to present some kind of authenticator, in other words, something the user possesses and controls, that proves unequivocally their identity [8]. Currently, authentication can be done through multiple different ways like password, token based, certificate based, or even biometric [9], that rely on the factors that were previously introduced. In the hope of solving these multiple account problems, identity management systems have emerged with the main purpose of allowing users to manage their multiplicity of credentials. By giving the option of managing each identity individually, users are allowed to manage authentication, authorization, roles, and privileges for a given service, without being required to share their private credentials with the Service Providers (SPs) [10]. MFA systems are also on the rise, with additional security measures being required, like having to provide a short code, sent by Short Message Service (SMS) to a linked phone number, or a One Time Password (OTP), generated by a previously configured authenticator application, or even a physical token. In short, there is a great variety of techniques for performing authentication. Existing mechanisms will be discussed with more detail in section 2.

1.2 Problem Statement

EPOS integrates multiple services of the scientific community, which are organized by groups specialized in an area, to which we call Thematic Core Services (TCS). These TCSs include

areas like Geodesy, Seismology, Anthropogenic Hazards, and others, each developing multiple platforms and web services, to interact and manage the enormous quantity of data. There are no standards regarding programming languages, databases, or protocols imposed on these heterogeneous services, which makes it difficult when trying to specify and develop common functionalities across all the developed services. One of the most requested feature in EPOS, is the existence of a user account system that can work for every service, in which user attributes can be shared between these TCS, and users can use one account for all the EPOS services. There is also a requirement from the European Research Infrastructures to track data access and usage, and report statistics about the services being developed and implemented. The task would be simpler if all the TCS software followed a harmonized structure and used the same information technologies and protocols, so that each service could just delegate all the authentication and accounting process to a centralized identity provider the same way. Since this is not the case, an Identity Manager (IDM) was developed for EPOS, that should be used by every TCS, and to which the authentication process should be delegated. This IDM is based on Unity IDM [11] and uses `OAuth 2.0` tokens for authentication. The service is available at the EPOS IDM page [12] and offers a variety of login methods via Identity Providers (IdPs) such as `EGI Checkin`, `B2ACCESS`, `Google`, and also the ability of registering as a new user directly. The EPOS IDM is offered as a login option in the GNSS TCS, for the GNSS Products Portal [4], but does not, by itself, solve all the difficulties that we have identified.

The first problem we want to solve has to do with the registration of users on a service provided by a TCS. It is important to define where the registration will occur if the user wants to access a service provided by a TCS and does not own an account. While a registration in a central system, where the account can be accepted by every TCS service, would be the ideal solution, EPOS does not yet provide that functionality, mainly due to the fact that the majority of the services provided are not yet connected to the central EPOS AAI.

A use case for this problem is when a user visits for the first time a given TCS service and want to access a set of data. Now, the EPOS policy regarding data retrieval is that the user can freely access all the metadata present in the web services, but has to be authenticated in order to download the actual set of data, since these data retrievals need to be accounted. Being the user's first visit to this service, they will not have an account registered in the system, and need to create one. Whether they should be redirected to an external page, such as EPOS IDM, or create an account in the service they are accessing is what we pretend to solve.

If the registration is done locally, all the information is handled on the TCS side, and this information needs to be shared with the central EPOS IDM, so that a shared account in the EPOS ecosystem can be created. The attributes that the central system require need to be compliant with the information that the TCS gathers locally. However, nothing stops the TCS from gathering additional information to the one required by the EPOS IDM, and some TCSs might want that. In case of the registration being done in the EPOS AAI, the TCS service simply delegates the process, and expects to receive all the necessary information on their end, to grant the user full access to the service. But if the TCS requires additional attributes on their side, it is necessary to require them from the user after receiving the attributes from EPOS AAI, which may look like a second registration to the user. Also, the service may not integrate the EPOS AAI, and therefore, the local registration will not be shared with the central EPOS system,

staying restricted to that TCS services at best, which is far from ideal.

The second problem we want to solve is the authentication of users through the service GUI. In this use case, the user is already registered. The TCS can authenticate the user locally or delegate the authentication to the EPOS AAI, or any other IdP of their choice. If the TCS is delegating this procedure to the EPOS IDM, the user will be redirected to the EPOS IDM page, and once logged there, it will be redirected back to the TCS with the necessary `OAuth` tokens to ensure their identity, and the central system can record this session. If the authentication is done locally, or using any other third party IdP, it is necessary to create a way of ensuring that this information is transmitted to the EPOS IDM, so that they can account this login session. Even better would be that the session can be stored in the browser cookies, so that the user can access any other TCSs service, avoiding the necessity of a new login from the user, improving the quality of the process. It is also important that the authentication process can be done in a comfortable way to the user. While it is necessary to protect the service, and therefore the user's data, from malicious actions, mechanisms like intrusive `reCAPTCHA` should be avoided.

Another problem that we want to address is the authentication through Command Line Interface (CLI), since some services provided by EPOS TCSs are legacy services, that have been in a production state for years, with well-defined functionalities and that cannot be modified. Aside from that, there are services that are used by other software, or simply by user's scripts, through CLI calls. Both of these situations need to be considered when implementing an authentication system. If the services cannot be modified, it is not possible to implement a registration or authentication mechanism directly in the software. And if the access is not done through the GUI, it is necessary to create an alternative way to authenticate. The difficulty here is that many of these tools or scripts are done purely using CLI, such as a python, NodeJS, or bash shell scripts. In these cases, it is necessary to give the user mechanisms that deal with the complexity of adding and managing tokens to their scripts. One use case for this situation is the retrieval of daily processed data from GNSS stations, through the GNSS Products Portal Application Programming Interface (API). Following the EPOS policy, this retrieval of data will need to be authorized and accounted for. It is therefore necessary to create a procedure where the users can prove that they are who they say they are, confirm that they are authorized to access the requested data, and account information for these requests. In these cases, it is necessary to give the user mechanisms that deal with the complexity of adding and managing tokens to their scripts. In this thesis, we try to provide users a secure and comfortable way to authenticate that matches EPOS requirements.

One other issue is the level of assurance that these authentication methods have. While some of the biggest internet platforms are highly popular as IdPs for all kinds of web services, they do not offer a high level of assurance. It is very easy to create an account on one of these platforms, trying to impersonate someone else, in order to gain privileges in the service were the authentication process is ongoing. The high level of popularity of these IdPs brings the advantage of having a larger reach in total number of users, as a direct result of an easier registration and login process, so many web services allow authentication to be done through them. For instance, EPOS researchers are highly likable to have an account on one of these popular platforms, and are more inclined to select these as the session IdP over others that provide a higher level of assurance. The issue here is that there might be access limitation

on certain data sets, with restrictions of use and purpose. To control who access what, it is necessary to know with certainty the identity of the users, so that aspects such as researcher status are not forged. It is also not feasible to validate every user manually, so it is necessary that the system handling authentication can provide a high level of assurance over the identity of its users, at least for the more delicate operations. In this thesis, we propose a method that can bring a high level of assurance to the system.

The last issue we want to handle on this thesis is the problem of multiple accounts for the same identity. With the integration of numerous IdPs for authentication in a system this problem comes along, such as allowing registration through `Google`, `ORCID`, etc. Since users can have an account in more than one of these platforms, if they log in through a different IdP than their previous session, they will end up logged on a different account, since the system will not make a connection between IdPs. Whenever the (same) user logs on through a new IdP, they will look like a different user to the system, and if the system allows a direct registration from that, a new account will be created for them, thus ending up with multiple accounts for the same identity. Not only this is inefficient, because the user is registering a new account, repeating a process that was already concluded, it is also creating redundant data for the SP, the TCS, which translates in noise in statistics and data. Also, by forwarding the user to a brand-new account, they will lose all account related data, like preferences or history, which may lead to the user abandoning the service. We try to tackle this problem by merging this accounts upon registration.

1.3 Dissertation Structure

The main body of the dissertation is composed by MANY chapters, including the present one. The contents of the remaining chapters are organized as follows:

- In this chapter, chapter 1 — **Introduction** — the subject of authentication and some related key aspects are introduced to the reader. The authentication problems that we are facing at the EPOS project are then presented together with a short introduction to the EPOS project itself. Information about the actual state of online authentication is provided, and we define the problems that the exponential increase of existing services and available data are generating. The structure of this document is also provided;

- In chapter 2 — **Related Work** — an introduction of the analyzed existing authentication mechanisms is presented, providing the background of these and what requirements they fulfill. Complementing this information, a detailed description of how they function and their use cases is presented;

- In chapter 3 — **Implementation** — we apply the results of the analysis of the existing literature to implement a solution for the existing web services, creating an AAA infrastructure that suits the TCS needs. In particular in this section the work done and the problems solved are described;

- In chapter 4 — **Analysis** — we analyze the usability and security of the proposed and implemented schemes. What type of authentication factors are used in the implemented system and some statistics are provided;

- In chapter 5 — **Conclusions** — we summarize the results obtained and present the conclusions of this thesis, and what contributions were taken from this work. We finish by providing our opinions about future work and possible improvements in the current systems.

Chapter 2

Related Work

2.1 Introduction

This chapter discusses the existing AAI mechanisms, from the native password based system to the modern standards for authentication, introducing to the reader some characteristics, benefits, and cons of the most popular protocols and the rising alternatives. A brief background of the reasons behind each mechanism development will also be presented, since this is relevant for the work developed. Federated identity is referred as today's standard for authentication on the web, while new solutions that make use of biometrics and smart cards are also explored.

The chapter is structured as follows:

- Section 2.2 — **Password Authentication** — in this section the most common authentication method of the last decades is presented. We will also explain why this scheme is obsolete and should be abandoned, since it is no longer secure;
- Section 2.3 — **MFA** — in this section we do a detailed introduction to the concept of MFA, as well as presenting some literature about the existing schemes and types of factors;
- Section 2.4 — **Federated Identity** — in this section the federated identity technology will be presented and some existing and more popular protocols for federated identity will be introduced. A short review of the features of each protocol is given, as well as the most used workflows;
- Section 2.5 — **Autenticação.GOV** — an overview of the Portuguese Electronic Identification (eID)-centered system is given, as well as the benefits of using this secure and reliable solution. This authentication mechanism is becoming quite common in Portugal, with the government doing an effort to make it the standard authentication method for online public services. There are currently two different forms of authenticating using this method that will be explored in this section. While EPOS is a European system, this method is relevant for us, since this may become the future standard for European authentication, with the rise of eID cards.
- Section 2.6 — **Biometrics** — our vision is that, in the near future, authentication will be done solely through biometrics. With this in mind, we introduce this concept and explain if it fits our requirements. Some of the available, and already in use, mechanisms are explored and brief key aspects about biometrics are presented;
- Section 2.7 — **Continuous Authentication** — while biometrics should be the future of authentication, the procedures to authenticate an identity will move from an active authentication to a passive one, where no direct inputs of the user will be required. We present to the reader existing concepts for this type of authentication.

2.2 Password Authentication

In the primitive years of the World Wide Web, passwords were the logical and easy way to create a protection layer in order to secure private information. Using a set of credentials, like a username and password pair, that remained secret to the user, an authentication mechanism could be created, providing the first line of security against malicious attacks. While the problem of bad management of these private credentials by the users and services was a reality, it served for the time being, since nothing better existed. In the modern days, with so many more authentication schemes, which are better in every aspect, an authentication scheme that only relies on a password, which may not even get hashed before being stored in the database, is very far from secure, but still often used [13]. The findings of [14] report that, in 2010, 80% of the top 1000 most visited websites of the internet used a textual password for user authentication. Nowadays this is not enough to guarantee that our data is protected, being a totally outdated system, since malicious attacks are getting much more sophisticated each year. Passwords, as a Single-Factor Authentication (SFA) method, only provides client authentication, and fails at every other security property we might desire [15]. Systems that rely on SFA are suffering some significant pitfalls, since if it fails, users can no longer access the service, and there is no way to predict if the service has been breached.

Another problem is that password authentication relies too much on e-mails, which are everywhere. With over forty years of history behind them, they have become omnipresent, with billions of messages sent every day. Due to that, these are a very popular method for attackers with malicious intents, as is referred in [16]. The spam levels are increasing every year, with more than half of all e-mails received, on average, being categorized as spam. Malware and phishing are also quite common threats, since a malicious e-mail can be found in every 412 emails, proving that e-mails may be dangerous if not used with proper awareness. Meaning that they are not a secure way of transmitting sensitive data, which is another weak point in the password authentication scheme, since many of its functionalities rely on e-mails to transmit the information to the user. For instance, most websites using a password authentication scheme rely on a password recovery system to allow their users to recover their account access, if they ever lose it. More often than not, these recovery systems include sending a password reset link by e-mail. The security of the password recovery, and hence, the whole system is dependent on the e-mail being acted upon correctly. Now, these e-mails, which sole purpose is to recover the account access, are not designed to maximize security and can introduce vulnerabilities in the system. According to [17], services like `Microsoft` and `Facebook` suffer from e-mail design, which vary from having the e-mail blocked by the spam filter, poor instructions, lack of information on what to do if the password reset was not requested by the user, and so on. It is also pointed out that there is a clear need for more research in the area, as well as improved ways of performing a password recovery. But if instead of solving the problem, we try to avoid it in the first place, we will realize that the only reason that a password recovery system is necessary, is because users tend to lose their passwords, since having dozens of username and password pairs to remember is quite the challenge.

Before federated identity protocols, like `OAuth`, became popular, applications would ask the users for third-party application credentials. For example, a third-party e-mail application would request the user's `Gmail` username and password, in order to authenticate as the user on their `Gmail` account, and obtain the necessary data to populate their service [18]. This is

often called the password anti-pattern, where an application (A) gets full access to another application (B), by possessing the user's secret credentials and impersonating the user before application (B) [19]. The biggest issue with this is that there is no way to revoke the access that the application (A) just got, without changing the password, and therefore turning the set of secret credentials stored on the application (A) obsolete, but having implications on the application (B). Another issue is that there is no way to selectively grant permissions to the application (A), being an all or nothing situation, where it is impossible to control how much access it has to the application (B).

In a quest to solve most of these problems, modern authentication protocols have been developed over the last years, as we will see in the next sections, and we can expect a future *passwordless*.

2.3 Multi-Factor Authentication

One way to improve the security in SFA systems is to add different authentication factors to the system, creating an MFA system, where each factor should fight a different threat. The MFA systems provides a more secure, resilient, and robust access to the legitimate users of a system, making it harder for intruders to gain unauthorized access [20]. An example of an MFA system is the use of a chip and a PIN, present in many banking cards, where a user has to know the PIN and also be in the possession of the chip, present in the physical card. Another example is the addition of Time-based OTPs to the conventional password system, which is used in `Google Authenticator`. Push or approval systems, like the `Google` login verification implemented in most recent `Android` smartphones, also fall in this category. However, MFA brings some security concerns regarding user identity hijacking. The majority of these systems perform a control over the user identity in the login process, but does not validate this information again once the service is in continuous use. This flaw opens a backdoor for user impersonation through the existing session. It is important that MFA is presented to the user in a way that do not invite dangerous misconceptions [15] [21]. For instance, the user may be led to believe that, since it has MFA enable, weak factors can be used, like a weak password. Users may also believe that, since they are using more than one factor, it is safe to save their credentials on a compromised device and have it require the OTP every time, and assume that no harm will come from that. The benefits that MFA brings need to be weighed against the security damage that may result from behaviors that its usage might encourage.

These authentication factors can be divided into three main groups: the knowledge factors, the inherence factors, and the possession factors. So, an authentication mechanism that relies solely on a knowledge factor, like password authentication, is called a SFA. To the process where we combine more than one type of authentication factor, we call MFA [22]. A combination of multiple factors from different groups can enhance the security of a system [23].

Some examples of these MFA systems are the combination of the knowledge and possession factors, as described in [24] and [25], the combination of the knowledge and inherence factors, which can be found in [26] and [27], and, less common than the previous ones, the combination of inherence and possession factors, present in [28]. There is also the possibility of combining the three main groups, as [29] suggests, but this is rarely used in practice.

The security benefits that MFA systems bring are real and legitimate in many cases, but they may rarely be what users are expecting, and may not always be applicable to the individual use [15]. Any of the factors previously referred can be either stolen or compromised, but the concept is that different attacks, and therefore different attacking capabilities, are necessary to compromise the different factor types. An attacker that is in position to compromise the knowledge factor may not be in position to compromise the possession or inherence factor. Besides the three main authentication factors, there is another authentication factor that is winning popularity, the location factor. This factor answers the question of where the user is, by using his location to verify his identity. This can be done by using GNSSs, Internet Protocol (IP) addresses, cell tower identification, and others, and is used as an addition to systems that already use the other factors. For instance, after the user has successfully passed the other authentication factors, the system can check if the user is authorized to access from that location [22].

An authentication scheme is a module that implements a way for the user to authenticate itself. Each authentication factor has multiple possibilities of authentication schemes, that are based on different traits. Based on what we have found in the existing literature, we list some authentication schemes aggregated by their group and alphabetically ordered, in the table 2.1. The same literature says that in the last years there has been a general increase in the research of new authentication mechanisms using multiple factors, which can be translated into an increasing effort to popularize MFA, and with this, increase security over the Internet.

Group	Scheme
Inherence	Brainprints
	Brow recognition
	Cheek recognition
	Fingerprints
	Gait biometrics
	Hand gestures
	Heartbeats
	Iris recognition
	Keystroke biometrics
	Knuckleprint biometrics
	Lip recognition
	Palmprint biometrics
	Passthoughts
	Retina recognition
	Speaker recognition
	Textural features
	Touchstroke biometrics
Knowledge	Cognitive authentication
	Graphical passwords
	PINs
	Questions
	Textual passwords
Location	Cell tower identification
	GNSS location
	IP addresses
Possession	Identity card based
	Mobile based
	OTP tokens

Table 2.1: Some authentication schemes that can be found in the existing literature [23] [30].

The huge amount of different existing schemes creates the problem of choosing the right authentication scheme for a given operating environment [31]. This choice will determine the security performance of MFA, which determines part of the security of the entire system. Additionally, using multiple authentication schemes from the same authentication factor group to validate users is not reliable, since if one of the traits can be exploited, there is a big probability that others from the same group also can. One example of this situation is systems that require a password and a decryption key, both belonging to the knowledge factor group, like `ProtonMail`. In [31], we can find a good implementation of an authentication framework, that uses multiple authentication factors, in order to provide a trustworthy, resilient, and scalable solution for authentication, where the proposed model computes trust values for the different factors and uses them in real-time. This results in the decreasing chance of establishing recognizable patterns, and therefore no information is provided to attackers beforehand. However, MFA is not enough for today's security requirements, since authentication is not exclusive to users, since authentication system to system is a reality.

2.4 Federated Identity

The next required functionality was to create a way to have an authentication system without compromising any of the parts. The ability to delegate the authentication process to a third party was also required, since these diminish the number of different credentials that a user possesses and knows. The solution for these problems was federated identities [32]. These provided the benefits of the Single Sign-On (SSO), that will be explained in 2.4.1, while extending the reach of a user's credentials to include external resources [33].

Federated identity management systems separate entities that enroll users, and are able to identify them, from entities that rely on the result of the authentication process. They are a set of technologies and processes that allow the distribution of identity information to be done dynamically. This way, web applications can offer cross-domain SSO [34]. In such systems a user may be registered as a user of service A with a certain identity, and authenticate using this identity to a third party service B, as stated in [35]. This service B may allow authentication to be made through multiple other services like service A.

The federated model involves four logical components when a human is involved in the identity interaction, which can be found in [34], and are listed below:

- The User is who assumes a digital identity in order to interact with the web service;
- The User Agent is a software application, like a browser, that runs in some type of device, like a computer. The user interaction takes place through this agent, being an intermediary in the connection, who can passively allow an identity information flow or actively moderate it;
- The SP is a web service that delegates the authentication to a third party, receiving from it a set of attributes related to the user. Another name for the SP is Relying Party (RP), because it relies on external information;
- The IdP is a web service where the user logs in to and that stores attributes of common interest to provide the various SPs.

This architecture separates the identity information source from its usage, benefiting everyone. The user only needs to log in once and is then able to access multiple web services without revealing his credentials to them. The SP can offload most of the account management tasks and improve the accuracy of the users' information by having more up to date attributes. Regarding the IdP, they benefit with the received feedback and data from the usage of its own authentication services, allowing it to improve the authentication mechanisms and to create better and more innovative features, related to the user account [34]. While federated identities have the benefits that have just been pointed, there are several security, privacy, and architectural challenges that come with it. For example, some legacy systems may not work with the SSO functionality, simply because their application systems cannot be modified and are independent of other services. In this case, the login is done in the more traditional way, usually by filling forms [36].

As happens in our day-to-day day life, we use physical documents for many identification purposes without the document issuer's knowledge. For example, we may use our driver's license to prove our identity in a library. This kind of freedom is not possible to achieve through federated identity protocols. IdPs will always know in which RP the user is authenticating. However, it is important that each part avoids leaking unnecessary information [37].

2.4.1 Single Sign-On

With the gradual increasing number of services provided to users, SPs needed a way of unifying the authentication systems, in order to provide a better management and security, and keeping one interoperable single set of credentials that could be used across several internal services. SSO is a property of access control that provided a solution for this problem, and was widely adopted. These protocols with SSO enable companies to establish a federated environment in which the clients only need to sign in the service once, using their account identifier and password, to gain access to a connected system, which is accomplished by using the Lightweight Directory Access Protocol (LDAP). Some benefits of it include mitigating the risk of accessing third party websites, since the user password is not stored or managed externally, reduce the password fatigue from different combinations, reduce the time re-entering credentials for the same identity, and reduce costs related to credential management [38].

2.4.2 Existing Protocols

There are three major protocols for federated identity: `OpenID`, `Security Assertion Markup Language (SAML)`, and `OAuth` (now `OAuth` 2.0), with `OpenID Connect` being a layer over `OAuth` 2.0 [32]. Table 2.2 represents a small set of commonalities between these protocols, including also `InfoCard` from `Windows Cardspace`, which is another protocol less used. Later more detailed information about these protocols will be given. There are many more protocols, but we decided to not include them here. This is the case with `Shibboleth`, an AAI protocol based on `SAML`, that uses the concept of federated identity [9], but we decided to not include this protocol in the comparison since we already include `SAML`.

Feature	OpenID	SAML	InfoCard	OAuth 2.0
Enables Direct Interactions between IdPs and SPs	✓	✓		✓
Has a goal of consistent user interface	✓		✓	✓
Can self-assert attributes	✓		✓	✓
Uses Extensible Markup Language (XML) message formats		✓	✓	
Can use WS-* web services		✓	✓	✓
Enables user-centric identity	✓	✓	✓	✓

Table 2.2: Common traits in four federated identity protocols: OpenID, SAML, OAuth, and InfoCard [34].

While this is a small comparison that does not include every feature each protocol has, if we exclude InfoCard since it is Windows exclusive and already canceled project, we can conclude that OAuth 2.0 is the most complete of the set only regarding the presented aspects. While it has some flaws, OAuth 2.0, with OpenID Connect on top of it, is the most complete solution up to date, when considering federated identity protocols. In the next sections we will provide a deeper introduction to these three federated identity protocols.

2.4.3 Security Assertion Markup Language

Designed with trust and privacy in mind, the SAML framework is flexible and some of its components are used by other protocols like some OpenID extensions and InfoCard. Its core is composed by XML packets that contain the information about the entity and authentication [34]. In short, we can consider it a session cookie on the web browser that gives access to web applications [18]. An XML-based framework for exchange of identity and security information cross-domain is defined in the protocol, that uses the approach of expressing assertions about an entity that other applications, across system domain boundaries, can trust [39].

It is important to understand the methodology of this framework before we can present the workflow of it. SAML needs an IdP, an RP, which in this case it will be called an SP, and the protocols for authentication. In the case of legacy applications, the IdP is the combination of an identity authentication server and an SSO-Agent, as it is represented in the figure 2.1. In the most common SAML SSO systems, the user requests the access directly to the SP, represented in the figure 2.2. The protocol provides a broad range of solutions for: attribute exchange, IdP and SP initiated SSO, assigning a federated identifier to a user, connecting it to multiple accounts, and long-term identity management of it [34].

Since 2005, when the latest version of SAML was launched (SAML 2.0), a lot has changed. Modern web applications like Gmail, Facebook, or Twitter are Single Page Applications (SPAs), meaning they have a different behavior than the traditional web applications because they use asynchronous JavaScript and XML to communicate with the APIs. SAML's SSO does not excel in this situation [18].

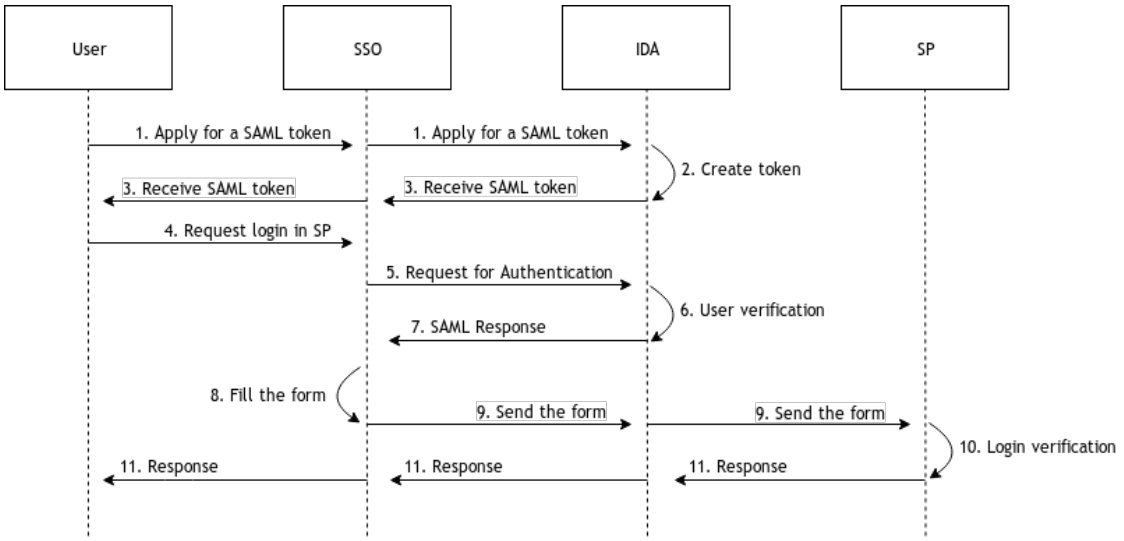

Figure 2.1: SAML legacy protocol flow [36].

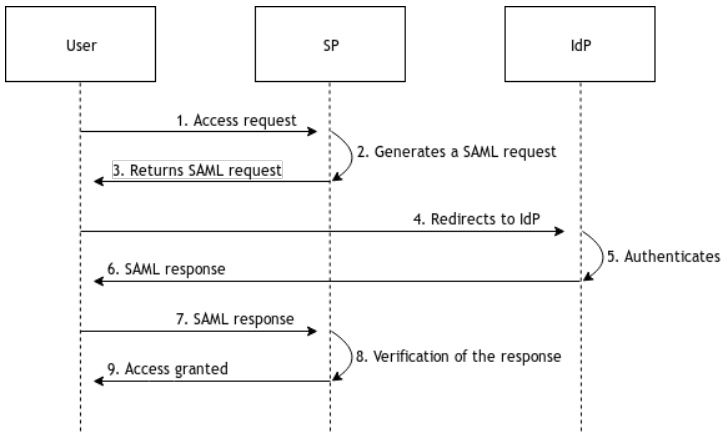

Figure 2.2: SAML most common protocol flow [40].

Below we describe with more detail how the SAML token exchange works, following the work flow presented in figure 2.2, to get a better idea of what is happening in the background of this authentication process.

1. The protocol initiates with the User, or the user's browser, sending a request to the SP;
2. The SP then responds with a SAML request to the browser;
3. After receiving this request from the SP, the browser redirects it to an IdP of choice;
4. The selected IdP parses this request, authenticates the user at their side, if the user isn't already authenticated, and generates a SAML response;
5. IdP sends the encoded SAML response to the browser;
6. Which is then redirected to the SP;
7. If the authentication response is successful, the SP grants the browser access to the protected resources.

In order to better understand what are the advantages of SAML, we listed below some benefits referred in [40]:

- The **standardization** takes away the common problems with specific architectures and implementations, allowing seamless interoperability between systems;
- Users have a **better experience**, since they can access multiple service providers with a single sign in, allowing for a better and faster experience. It also eliminates some password issues like the reset and recovery procedures;
- Being **security** one of the key aspects of web services needing authentication, SAML provides a single point of authentication which happens at a secure IdP, ensuring that the credentials do not leave a secured system;
- Since there is no need to maintain user information across multiple services, **reduces the costs for SPs**, and leaves the burden of authentication to the IdPs.

2.4.4 OAuth 2.0

The OAuth 2.0 Framework, maintained by the Internet Engineering Task Forces (IETFs) OAuth Working Group, is a protocol for establishing identity management standards across services, providing an alternative for sharing the usernames and passwords, with the purpose of avoiding attacks using that information [41]. Published as Request For Comments (RFC) 6749 [42] in October 2012, OAuth 2.0 Framework is now the industry-standard protocol for authorization, having replaced the original OAuth 1.0 protocol, that started being developed in 2006, and published as RFC 5849 [43] in April 2010. This framework, designed to solve privacy and access control issues related to large scale internet connected applications [44], enables a third party application to obtain limited access to a Hypertext Transfer Protocol (HTTP) service. This can be done either on behalf of a resource owner, by orchestrating an approval interaction between the resource owner and the HTTP service, or by allowing the third party application to obtain access on its own behalf [42]. The protocol also solves a number of important needs, which most API providers have, like delegating and revoking access, and reducing the password sharing between users and third-parties [45]. As referred in [44], a good example of this is when social networks ask for the user credentials to a third party application, like an e-mail provider, to obtain their contact list from there and check if those contacts are using the social network, in order to create a connection. From here a number of problems pops up:

- There is no fined grain access control, since the social network can do anything, like sending spam;
- There is no time limitation for the use of this access;
- The access can only be revoked with a change of credentials;
- Uses a user-centric security, with usernames and passwords, instead of machine to machine communication, through tokens for identification and access;

OAuth 2.0 solves all these problems of having an API communicating with IdPs, and the protocol is used by high demand APIs, like Google, which is a clear evidence that it can be used in a highly scalable system.

The interaction between the user with his browser, that we will consider the RP, and the IdP can be performed in four different flows, or grant types, that are listed beneath. Following the OAuth 2.0 Framework terminology, the IdPs are called authorization servers and resource servers, the RPs are called clients, and the users are called resource owners [46].

The OAuth 2.0 Framework specifies several grant types, each suitable for a different use case, as well as a framework for creating new grant types. In the list below we refer some of them. The first two being the most common grant types, and the latest two, that are also specified in the RFC 6749 [42], have been created posteriorly to the protocol release. The resource owner password grant and client credential grants are used for very specific situations.

- **Authorization Code Grant** - obtained using an authorization server as the intermediary between the client and resource owner. The authorization server authenticates the resource owner and obtains authorization. The resource owner's credentials are never shared with the client;
- **Implicit Grant** - a simplified authorization code, optimized for browser implementations using a scripting language;
- **Resource Owner Password Grant** - used by first-party clients to exchange a user's credentials for an access token. The credentials should only be used when there is a high degree of trust between the resource owner and the client;
- **Client Credentials Grant** - can be used as an authorization grant when the authorization scope is limited to the protected resources under the control of the client;
- **Extension Grant** - an extension mechanism to support the definition of additional grant types. Extension grant types may also define additional endpoints if needed. This can be used to provide a bridge between this protocol and other trusted frameworks [47];
- **Device Code** - this is an extension grant type. Is used by browser less or input-constrained devices, in order to exchange a previously obtained code for an access token [48];
- **Refresh Token** - a type of credentials that is used to obtain new access tokens when the current ones become invalid or expire.

OAuth 2.0 is one of the most widely deployed authorization and SSO protocols, and also serves as the foundation for the new SSO standard, OpenID Connect, that we will introduce next in section 2.4.6. This protocol is already in use and actively supported by big names of the industry, such as PayPal, Google, and Microsoft, among many others. In practice, OAuth 2.0 is often used for authentication as well, that is, a user can log in at an RP using his identity managed by an IdP (SSO). However, this is a misuse of the protocol, since it was explicitly designed for authorization. What the OpenID Connect layer does is to adapt the protocol for authentication. Authorization and SSO solutions have found widespread adoption on the web over the last years, with OAuth 2.0 being one of the most popular frameworks, used to turn those big names into IdPs [46]. Considering the broad industry support for OpenID Connect, the widespread adoption of the SSO technology in the next years seems likely.

The diagram in figure 2.3 allows a general look on how the different roles interact between each other. Beneath it there is an explanation, step by step, of how the protocol exchanges information [49].

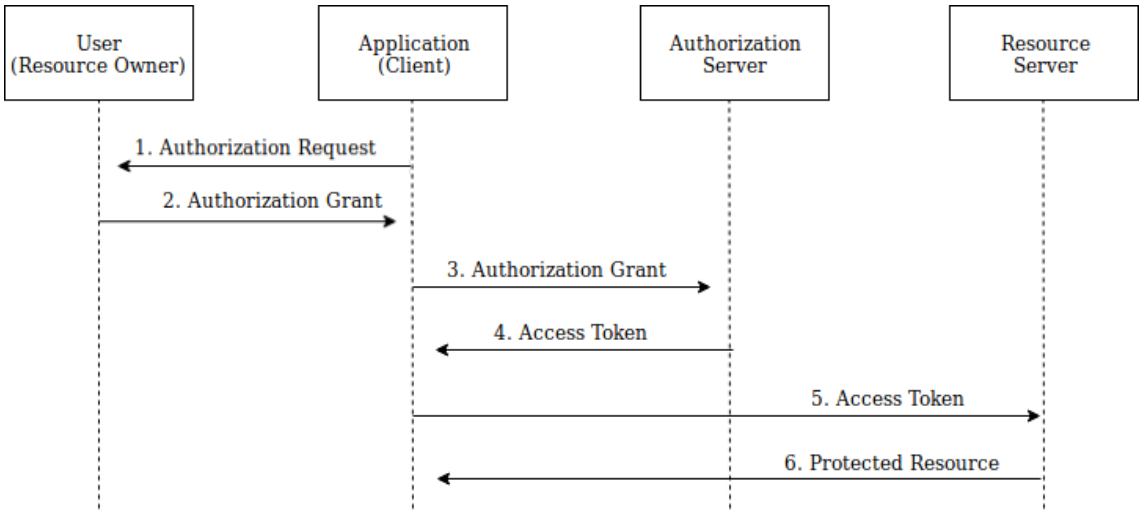

Figure 2.3: `OAuth` abstract protocol flow [49].

1. The application starts by requesting an authorization to access service resources from the user;

2. If the user authorizes the request, the application receives an authorization grant, otherwise the protocol ends here;

3. The application requests an access token from the authorization server by presenting the authentication of its own identity, and the authorization grant;

4. If the authorization grant is valid and the application identity is authenticated, the authorization server issues an access token to the application, ending the authorization process;

5. The application requests the resource from the resource server and presents the access token for authentication;

6. If the access token is valid, the resource server provides the protected data to the client.

However, there are still flaws in the protocol. For example, the top ten security vulnerabilities report of 2018, which can be found in [50], include a vulnerability related to `OAuth`, where, through Spring Expression Language Injection, it is possible to execute remote code. Another weakness of this protocol is that the payload of data that the IdP returns to the RP is not standardized. While it is usually a JavaScript Object Notation (JSON) payload, the information that is contained inside it varies from IdP to IdP. For instance, one IdP may use the `family_name` key to refer to the user's last name, while other may use the *surname* key. On a rare occasion, it is possible that the payload does not use the JSON format. This lack of standards makes it very difficult to create general plugins, because an adaptation to the chosen IdPs is always necessary.

2.4.5 OpenID

With the advances in user-centric and Uniform Resource Identifier (URI)-based identity systems in the last decade, it had become clear that a single specification would not be the solution to all problems. A better solution would be, as it happens with the internet layers, develop small, interoperable specifications, that could be implemented without dependencies, which would ultimately lead to a market adoption of these technologies [51]. OpenID 1.0 was originally created in 2005, and then evolved into a framework, because some companies shared the same vision and wanted it to become an umbrella under which multiple technologies could fit. The purpose was to have a framework that was flexible and adaptable, but yet simple enough to allow broad adoption. The diagram represented in figure 2.4, taken from [51], shows how different technologies can build upon one another.

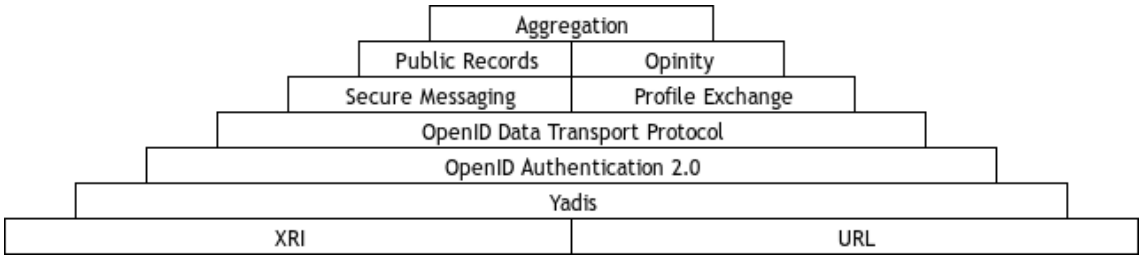

Figure 2.4: OpenID stack [51]

- Uniform Resource Locators (URLs) and Extensible Resource Identifiers (XRIs) are on the bottom, and act as the end user identifier;
- Above them, the Yadis layer, provides a simple service discovery using the Extensible Resource Descriptor (XRD) document format from OASIS;
- OpenID Authentication provides the SSO layer on which the higher level services depend;
- OpenID Data Transport Protocol provides a level of abstraction for the higher level services, that depend on trusted data exchange;
- On top of these, other identity-based services can be layered, depending on their needs.

If we consider the first four layers present in the figure 2.4, we have:

- **Identifiers** — a digital entity infrastructure where the end user has independent control over his personally-identifiable information sharing. There are two architectural approaches for this: address-based identity and card-based identity, which can be complementary to each other;
- **Discovery** — the discovery of the user's associated identity services, which is accomplished by the Yadis discovery protocol, if it is a URL, or the XRI resolution protocol, if it is a XRI. Both protocols share the XRDs format. The process is quite simple: the retrieval of an XRDs document that describes the services available for a particular URL or XRI. It also contains more information like the OpenID service endpoint;
- **Authentication** — OpenID Authentication is a service that allows users to prove they own a URL or i-name. The procedure is done through a communication between the RP and the user's OpenID service provider, the IdP. The use of URIs and XRIs as identifiers means that OpenID can be decentralized;

- **Data Transport** — the OpenID Data Transport protocol provides an abstract method for exchanging data between the IdPs and the RPs. With this protocol, the framework becomes extensible and capable of supporting multiple services that require trusted data sharing, like secure messaging;

Today, OpenID is an open standard for SSO, backed by big names in tech. It allows a user to be authenticated using an IdPs, where they can choose their preferred provider to log in to web services, that accept the OpenID authentication scheme [32]. It is reported that in the year 2009, there were already over one billion of OpenID-enabled user accounts and nine million websites using OpenID [52].

The OpenID Connect, run by OpenID Foundation, is a protocol to verify user identities and obtain user profile information. It builds upon OAuth 2.0, described in section 2.4.4, and provides clearly defined interfaces for user authentication and additional features, such as dynamic IdP discovery and RP registration, signing and encryption of messages, and log out [46]. Other features, listed on [53], consist in a more web and developer friendly software, simpler JSON-based claims model, and modular specifications so that only the necessary functionalities for an application are implemented.

This protocol makes it easier for developers to provide users with a usable and secure authentication mechanism, while avoiding the management of passwords. With this in mind, developers can shift their focus to the applications core functionalities in order to build more compelling applications [54]. For example, Google Sign-In is built over OpenID Connect.

OpenID Authentication allows a user to prove he controls an Identifier. This is done without the RP needing access to his credentials, like a password or any other kind of sensitive information. No central authority needs to approve or register RPs or OpenID Providers, making this a decentralized protocol. A user is free to choose which of the available OpenID Providers he wants to use, preserving the Identifier if he decides to switch to another OpenID Provider. The authentication scheme functions nicely with the ”AJAX”-style setups, meaning that there’s no need to leave the current web page to prove the Identity to an RP [55].

OpenID Authentication uses only standard HTTP(S) requests and responses, so it does not require any special capabilities of the User-Agent or other client software. OpenID is not tied to the use of cookies or any other specific mechanism of the relying party or OpenID Provider session management. Extensions to User-Agents can simplify the end user interaction, though are not required to utilize the protocol.

The exchange of profile information, or the exchange of other information not covered in this specification, can be addressed through additional service types built on top of this protocol to create a framework. OpenID Authentication is designed to provide a base service to enable portable, user-centric digital identity in a free and decentralized manner.

2.4.6 OpenID Connect

OpenID Connect, although it appears that it is an extension of the OpenID protocol, referred in section 2.4.5, they are not directly related. In reality, it is a subset of OpenID concepts made

to be compatible with the OAuth 2.0 framework, introduced in section 2.4.4. It was created by the OpenID team, that noticed the popularity of the OAuth 2.0 authorization protocol that was being wrongly used for authentication, so they came up with the OpenID Connect solution.

Is the leading standard for SSO and identity provision on the Internet. Simple JSON-based identity tokens, or JSON Web Tokens (JWTs), are delivered via OAuth 2.0 flows designed for the web.

2.5 Autenticação.gov

In order to solve the problem of providing an unequivocal identification of Portuguese citizens before online web services, the Portuguese public institute *Agência para a Modernização Administrativa* (AMA) created the *Autenticação.gov*, which will be denominated A.GOV in the remainder of this document. This entity is in charge of the authentication and authorization process and supply of personal attributes, that allow external entities to perform the identification of the citizen. It is responsible for the management of the multiple available attribute providers and has a narrow connection to the Public Key Infrastructure (PKI) of *Cartão de Cidadão* (CC), with the purpose of maintaining a high level of security and privacy in all the processes. It also allows the creation of credentials, that are interoperable to every public administration websites [56].

A.GOV is also the governmental service responsible for the authentication mechanisms of AMA — the CC and *Chave Móvel Digital* (CMD). These mechanisms are gaining more popularity over the years, as the statistics (found in appendix A.1) show. This is due to the fact that some Portuguese public services are starting to force the authentication of its citizens to be done through these mechanisms, since they provide a reliable and unequivocal identification of user's identity.

The principal objectives and functionalities of A.GOV, as listed in [56], are the following:

- Sectoral identification based on CC — it is based on the accreditation of the citizen during the CC emission that, together with the federated identity of the public administration interoperability platform, allows a secure identification of a citizen;
- Sectoral attributes availability — it is possible to obtain identifiers, like the tax number, using the interoperability platform;
- Simplification of the authentication process — the authentication process can be delegated to the A.GOV, being this the responsible for the certificate validation, obtaining qualified attributes, and to provide these to the requesting entity;
- Normalization of the authentication process — the authentication process security level and quality depends on the certificate that is used, or the CMD. The PKI is guaranteed through the Online Certificate Status Protocol (OCSP) validation of the certificates, whenever this is available. When it's not, the validation against Certificate Revocation List (CRL) is done;

- Full control over the information that is provided — the user has full control over the attributes that are provided. An explicit permission from the user is necessary to perform the information transmission. At any point the user can also cancel the process.

At the moment, the A.GOV service allows the authentication to be performed in two different ways:

- Using the CC — Chapter 2.5.1 — the primary method, which consists in using the identification document of Portuguese citizens, the eID physical card;
- Through the CMD — Chapter 2.5.2 — which is an alternative authentication method, using a mobile device, and that is lately being strongly advertised and gaining popularity.

2.5.1 Cartão de Cidadão

The Portuguese eID Card, referred as CC in this document, is a physical electronic identity card, valid in all national territory, that permit the owner to prove his identity, physically or digitally, to any public or private entity. It was created with the purpose of increasing the interoperability between entities, by unifying some Portuguese services merging and replacing, at the moment, five identification documents: taxation card, civil identification card, healthcare card, voting card and social security card [57]. The Portuguese constitution forbids the centralization of data in a database, so several identifiers can be found in the CC, allowing for a way of legally centralizing information. Another advantage is the ability to perform an online authentication using digital certificates, that may not be the ones present in CC, allowing other entities to delegate the authentication to the CC. Digital signatures are also possible, proving that Portugal is ahead in the digital age. The introduction of these functionalities allow a normalization of the electronic authentication act, and the device provides the required cryptographic means for secure access to e-government services portals [58].

The card itself is a Java smart card that follows the international norms and is officially recognized as an identification document, being in line with the European Citizen Card (ECC) guidelines and International Organization for Standardization (ISO) 7501 and 7810 standards [10]. The Portuguese government selected this device based on Identification, Authentication, and Signature (IAS), being the first in Europe to do so [58]. This infers that the work done with the CC can be applied to any other smart card that follows the same standards.

It allows performing strong authentication in both public and private electronic systems, being already used as an authentication token in diverse systems like national social security portal or the citizen's portal [59]. It supports biometric authentication, using fingerprint templates stored inside the microprocessor of the card, and using the `match on card` mechanism without a central biometric database, that is forbidden in Portugal as previously referred [58].

In [60] and [61], a federated identity provider using the CC is presented, where personal information and digital certificates are extracted from the card in order to assure associated web applications that the holders of the digital identities exist and are certified by a National State. In figure 2.5, based on a diagram present in [56], we can find a simplified use case of an authentication with A.GOV. The steps that occur during this process are enumerated and will be detailed later.

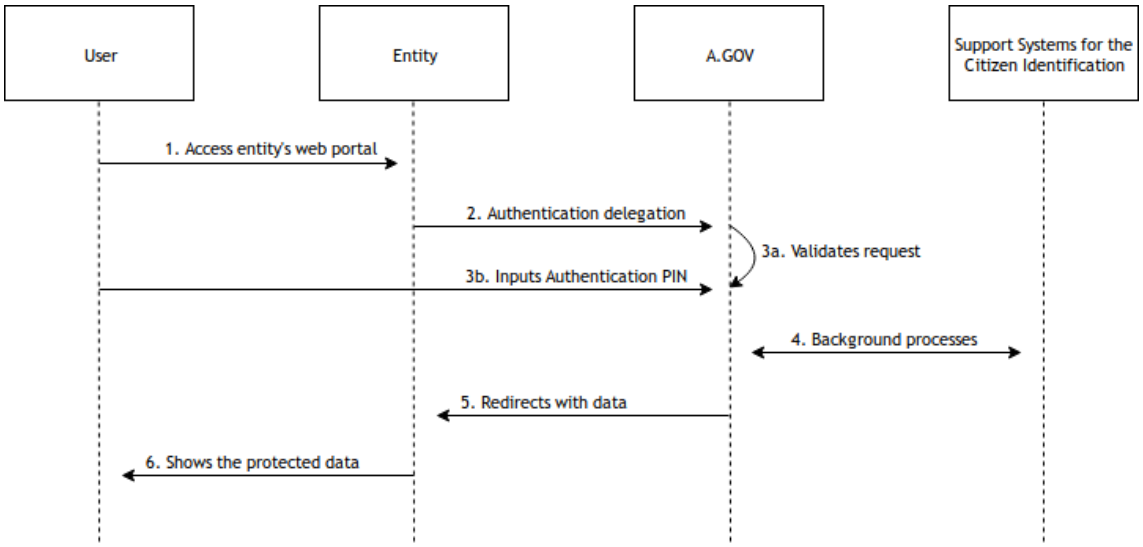

Figure 2.5: Authentication flow using the A.GOV [56].

The following setup, aside from the basics like an internet connection or a supported web browser, is needed for an authentication with the CC: a smart card reader, preferably an officially supported one, A.GOV plugins installed, and knowledge of the authentication PIN.

1. The Citizen, which is the client in this process, accesses a web portal of an entity, that requires an identification of the client;
2. This entity delegates the authentication process to A.GOV, redirecting the client there, together with an authentication request that is digitally signed;
3. A.GOV validates the received authentication request and requires the client to authenticate using the CC, by confirming with the authentication PIN;
4. In the background, two operations are done by A.GOV: the validation of the client credentials, using the CC PKI and the OCSP; and the retrieval of attributes that are requested by the entity, which is done using the interoperability platform;
5. The identification and attributes of the Citizen are then authenticated and digitally signed by A.GOV, redirecting the client back to the entity page. The entity should then validate these information;
6. Once the user is validated, the entity can show the protected data.

2.5.2 Chave Móvel Digital

It was recommended to the Portuguese government to implement a national program with the purpose of modernizing, simplifying and de-bureaucratizing the administrative processes. One of the main recommendations was the adoption and implementation of an electronic multi factor authentication mechanism, complementary to the CC, in order to increase the number of users of the online public services. The implementation of this mechanism is quite easy once the standard A.GOV authentication is implemented on the service, requiring only some small modifications on the data that is exchanged.

The CMD, defined in *Lei nº 37/2014, Diário da República. 1ª Série - N.º 121 - 26 de junho de 2014*, is an alternative authentication method with the main purpose of allowing a simple and secure authentication in public institution web services. Until August 2020, the total number of authentications through CMD is almost equal to the number of authentications through CC, as can be seen in the appendix A.2. However, we should acknowledge that, until the official presentation of the CMD, on November, 9th of 2017, there was barely any advertisement to it, and only a couple of public administration services allowed the authentication through the CMD. At the moment it is widely supported in Portuguese web services that offer public services, like the `Portal do Cidadão`. The increase of supported services and the higher advertisement are leading to an increase in the number o authentications using this method. It can also be used as an authentication method on other web services upon agreement with the AMA.

The CMD can be activated by any citizen after completing their sixteen birthday, as long they are not interdicted. It's permitted to associate the citizen identification number with a single phone number and a single e-mail. During the authentication process, the citizen should choose where they want to receive the temporary security code. In the case of foreign citizens, this association can be made with their passport number. This association data can only be used for authentication purposes. A request to activate the CMD has to be made either online, using the citizen card to authenticate on the web portal or by sending a letter with a generated password to a governmental entity or in person, either on the delivery moment of the citizen card or in the `Espaço do Cidadão`.

It implements a two-factor authentication, requiring a PIN, that is permanent but chosen and changeable by the citizen, and a temporary six digit security code, unique to each authentication, that is sent to the user either by an SMS, an e-mail or as a direct message on `Twitter`. The generation and distribution of the temporary security code is handled by AMA. Additionally, a digital certificate can be associated with the CMD, namely the existing one in the CC.

2.6 Biometrics

Biometrics is the science of analyzing physical or behavioral characteristics that are specific to each individual, in order to be able to authenticate their identity [62]. Physiological traits include models using hand, finger geometry, or those based on the verification of the face, that includes multiple categories like eye (iris or retina), brow, check, lip, textural features, and so forth. Behavioral techniques include those based on voice, signature, typing speed, and others. Although these characteristics are linked to the individual, their category is not immutable, since the method used to treat them may act differently, changing from one category to the other, as it happens with the more recent voice biometrics, that place voice in the physiological category [63]. Other literature also divides biometrics into two categories: soft biometrics and hard biometrics. Soft biometrics are the ones that can be changed easily, like the clothes color, the signature or hair color. These can be used to identify an individual in a short span of time, but can't be used as firm features like hard biometrics, which include the physiological traits referred before [64].

Biometric authentication is the process of comparing a set of recognizable and verifiable data from a person's characteristics to that person's biometric template, in order to determine

resemblance. What all biometrical approaches have in common is this digital template, created during the enrollment process, unique and specific to the person, and that needs to be stored in some way, like a database, so that it can be used for matching purposes later. This template is created by determining the identity of a person, through a biometric identification that aims to capture a template of one or more biometric traits from the person. This capture can be done by multiple ways, like using a profile photo, a sample of their voice, a template of their fingerprint, and others [62]. One factor in favor of biometric authentication security is that is based on *"who the user is"* and this biometric information is not easily spoofed or forged, since it is not something that can be lost or stolen, which would result in an identity theft [65]. Biometric techniques may ease many of the problems that exist on other authentication methods. They can confirm that the user is actually present, which does not happen with a password or token, and does not require the user to remember anything [63]. For example, while a user leaves their device unsupervised in the office, someone can access to it and find the passwords saved, which, in theory, should not happen with biometrics. According to [62], biometrics has quickly established itself as the most pertinent means of identifying and authenticating individuals in a quick and reliable way.

As it was previously referred, the CC also supports a biometric identification, through the use of fingerprinting. The more traditional ways of doing a fingerprint identification include saving the fingerprint in a central system, along with the identifying information, creating a template. Then, when the holder of the card scans their fingerprint, the data is sent to the server and a second template is created and compared to the original one. This is not the case with CC, since it uses a `match on card` method, where the data scanned by the reader goes inside the card, where the comparison happens, and the original template never leaves the card. Nowadays, this is the more conventional method, allowing to perform authentications in non-secured or only partially-secured environments, like verification terminals. This is the case when, for instance, the citizen goes to pick up a brand new CC from the representative entity, where they must perform a successful fingerprint verification in order to pick up the card. Since the data never leaves the physical card, a sensible transmission from inside it to the reader does not occur, which impossibilities the interception of data. But not every card has the `match on card` technology like the CC [66].

A decade ago it was expected that biometrics would become a significant component in authentication when: i) the prices of biometric sensors would continue to fall; ii) the underlying technology would improve; iii) the general public would become more aware of the strengths and limitations of biometrics [67]. All these happened and nowadays a great percentage of the public owns some kind of device with biometric authentication, like a smartphone. Smartphones tend to get outdated quickly because of the technological advances, so the majority of users has a model that is just a couple of years old. In the latest years, these devices started to include some kind of biometric authentication. For example, `Apple's iPhone` has the `Touch ID` mechanism, where the phone holder has their fingerprint analyzed and compared after pressing the main button of the device, in order to gain access over it. `Android` phones usually delegate the fingerprint detection to a dedicated sensor, most times on the back of the phone. More recent models are starting to popularize the fingerprint detection over the screen. Another biometric authentication being popularized on smartphones is the face detection, that uses the frontal camera of smartphones to perform a comparison with a previous setup face template of the owner. This technique has been constantly improving, and so are the frontal cameras

on smartphones, and the recognition accuracy is already high enough for a practical use. It is also starting to be used for commercial payments, which require a very high accuracy [68] [69]. Face recognition is associated with advantages such as non-intrusive data capture and low cost sensors, and it is being used in diverse applications like forensics and surveillance [70]. But there is a trade-off between security and ease of use. The 2D face recognition systems can be easily fooled with a photograph, clearly being unusable for this purpose. Virtual camera is a category of software that adds a layer between the physical camera and the operating system. While the purpose of this software is benign, like adding effects to photos, they pose a threat to the security of facial authentication. For device unlock, this is not such a big threat since it uses the system level camera and has more control over the whole system, but website authentication is susceptible to this attack [65]. Below this paragraph we present some challenges and difficulties of biometric identification, based in the information found on [71].

Total number of identities — the biometric identification needs to, potentially, be able to distinguish every individual on this planet. This gives a total of billions different identities, whom distinguishing factors may be very subtle. In order to solve this problem a highly complex model able to perform this distinction is required;

Intrapersonal variations — multiple samples of the same individual are necessary to create their identity. This is due to having to completely capture the variations of the intrapersonal characteristics. This type of variations, confined to the same individual, may be more noticeable than variations between different individuals (i.e. interpersonal variations);

Uniqueness — if a single biometric trait can uniquely identify an individual is not yet clear. Specifically, most behavioral biometrics are not effective for identification, and are solely used for verification purposes;

Consistency — since biometric traits are based on human characteristics, they might not be consistent, and therefore not suitable for identification purposes. Physiological biometric vary gradually with time and behavioral biometrics are also affected by socio-environmental factors;

Noise and distortion — biometric data that is collected outside a controlled environment, like in a real world application, are usually affected by noise and distortion, due to factors like noisy biometric sensors or others;

Preprocessing — in order to extract relevant biometric information from noisy input data requires significant preprocessing (i.e. extract an individual speech signal from a noisy background);

According to the results present in [72], where a study was performed to find the reasons of using or not using biometric authentication on smartphones, usability comes in the top, over privacy and trust, as the main factor for not using those. While it was believed that the fear of what companies could do with their biometric data would be the main issue, it turns out that the speed and reliability of these methods is what makes users give up and return to PIN or patterns. With the improvement of the hardware and algorithms, since the publication date of this study, results should largely differ and these methods, with no doubt, are better accepted. Although it is widely considered to be more secure than password or PIN authentication, it is rarely used in practice for device unlock or website login in smartphones. The conclusion of [65] is that

ease of use and security issues, like the previously referred 2D media attack and virtual camera attack, are the most relevant issues that prevent the prevalence of facial authentication.

Although the recent advances in technology have improved this aspect, the performance of a biometric recognition or verification is still deeply analyzed. Biometric authentication relies on statistical algorithms, and therefore cannot be fully reliable when used alone [62]. To evaluate the performance of such systems, generally, decision error rates are used [30]. These take in account criteria like False Acceptance Rate (FAR), which measures the likelihood of an unauthorized individual gains access, False Rejection Rate (FRR), which measures the likelihood of an authorized individual being rejected, and Equal Error Rate (ERR), that is when FAR and FRR met. Others factors, like speed and template size, are very relevant when the system scales, but for performance purposes we will only consider the decision error rates. The lower FAR and FRR values are, the better is the system. Also, the lower ERR is, the higher the accuracy of the system will be. Every biometric system allows the adjustment of the sensibility of these rates, with implications in security and usability. The formulas to calculate each of these values, with μ being the security level, are the following:

$$\text{FAR}(\mu) = \frac{\text{Total number of false successful authentication attempts}}{\text{Total number of authentication attempts}}$$

$$\text{FRR}(\mu) = \frac{\text{Total number of false unsuccessful authentication attempts}}{\text{Total number of authentication attempts}}$$

For instance, the possibility of having an iris validation mechanism in automated teller machines has already been researched [63], and might be a real possibility in the near future. In this case we are potentially talking about millions of templates, which is the entire world banking system users. In the negative side, there is a percentage of people that is unable to perform certain types of biometric verification [73]. Visually impaired users should not be required to perform an iris verification. Although this is a minority of users, they must be taken in consideration and alternatives must be provided. For some time now, there has been a use of several biometrics in combination, like the combination of the iris and fingerprints, which are called multi-modal biometrics. This has the purpose of reducing error rates considerably, but still depends on the quality of the acquisition tools and algorithms used [62]. Also, biometrics suffers from the fact that the matching algorithms cannot be compared like password hashes. This means that, in order to compare two templates, they need to be present at some point in a usable way in the system memory, like a plain text. These comparison checks must be carried out on trusted devices, like the CC, to not compromise the process.

As presented in [74], an ideal biometrics characteristics would be universal, unique, exclusive, permanent, indispensable, collectable, that can be stored digitally, precise, easy and efficient to record, and acceptable to contemporary social standards, which as of now, these objectives are unachievable by any current methods [63]. Speaker recognition measures the sound waves of the speaker. There are two types of speaker recognition: text dependent, where the speaker is required to say a determined phrase, and text independent, which is more flexible but harder to recognize. Keystroke recognition tries to capture the unique way that each person types. Through typing some features can be extracted like the keystroke, which includes the speed of pressing a key down and the speed of the subsequent release, the hold time of each keystroke, and the delay in between keystrokes.

As pointed before, there are multiple traits of biometric authentication, and each has a different usability, accuracy, and overall performance. In table 2.3, we included the decision error rates of some traits found in literature, ordered from worst to best by ERR. We should take in account that these values can largely vary for each biometric trait, depending on the features and classification models that were used by the model from where these rates were obtained.

Trait	FAR %	FRR %	ERR %
Textural features	1	20	10
Speaker recognition	2	10	6
Fingerprint	2	2	2
Keystroke	7	0.1	1.8
Retina scan	0.31	0.04	0.8

Table 2.3: Decision error rates for some biometric traits [30].

Other traits, like passthoughts and brainwaves, rely on electroencephalographs to authenticate individuals. Each individual's brain reacts differently to the same set of images. Using this information it's possible to identify an individual with total accuracy. However, these systems rely on some kind of devices that have electroencephalograph sensors, not as widely available and more invasive. These biometric systems have a great potential to be deployed where total accuracy is an absolute need.

2.7 Continuous Authentication

In a conventional authentication system [75], the user is only requested to provide his credentials once, which is at the start of the interaction with the system, to have access granted to him until the end of his session [64]. This is the case with most of the current systems, that do not actively or continuously verify the identity of its users. Once the initial authentication is done and the authorization granted, these systems use that approval until told otherwise or some time window has passed [76]. This is a situation that does not change with the implementation of an MFA system, because MFA only affects the login process, and does not offer any type of protection once the user is authenticated. The approach of making passwords increasingly more complex also does not address the root of the problem of authenticating proxies for users instead of authenticating the users themselves [76]. The solution for this may pass by using a continuous authentication system that can constantly verify that the current user is still the same initially authorized user [77].

Continuous authentication, and verification, is not a brand-new idea, since there has been continuous research on this subject for more than a decade. However, the interest in this field is growing over time due to the need of an increase security over the conventional one-time authentication systems [64]. The continuously verification of the users is done by attributing a numeric value — a rating, to the user, which is constantly updated according to the user's behavior. This analysis is done through algorithms based on user's traits, that compare the current behavior to the template one, and can be improved over time. Based on this rating, the system may deny the access at any point during its usage if a suspicion of the user's identity is raised, and prompt a validation mechanism to re-validate the user's authenticity [78].

The current continuous authentication mechanisms are good enough to collect and analyze, in real time, data from multiple sources of information, such as the various sensors present in smartphones, the time spent on an application, or other behavioral characteristics. Is then possible to assign specific restrictions that are based on the level of confidence of the continuous authentication. Some cases, like online banking applications, may need a high level of confidence, while others, like games, may not have such a high level of confidence requirement. This level of confidence may be adjusted depending on what the user does with the application, since applications that may seem harmless at first may escalate their need of security depending on the users actions within it, like the use of a credit card details to perform financial operations.

There are two categories of continuous authentication: passive and active. If the user has to provide a sample of his biometrics upon registering, this means the system is using active biometrics. Otherwise, when the sample collection takes place during the interaction with the systems, these are called passive biometrics. The passive continuous authentication system in [64] keep verifying the user's identity without interrupting his work, by using face recognition at its core. This is done because of two reasons: 1) since it is an inherence factor, it is something that belongs to the authorized user, and there is a very small chance of losing or forgetting this identification; and 2) the biometric information is the one with the highest chance of disappearing when the user leaves the terminal. Many other proposals for continuous authentication, both passive and active, can be found in the literature: Bae *et al.* proposed a real-time face detection and verification using hybrid-information extracted from face space and facial features [79]. Sim *et al.* proposed a continuous verification to protect user sessions using multimodal biometrics [80]. Kim *et al.* proposed a non-cooperative user authentication system in robot environments using semibiometric information [81]. And Fridman proposed an active authentication on mobile devices using multiple sensors of these devices together with applications [82].

Nonetheless, there are still multiple challenges that continuous authentication has yet to overcome, which in reality are challenges for biometric authentication. Some examples are the domain adaptation of sample collection, since most of the biometric feature registration is done in optimal conditions, like a well illuminated face, that not always reflects the real usage of face recognition, and the possibility of spoof attacks, that can occur with features like speech recognition.

Chapter 3

Implementation

3.1 Introduction

This chapter discusses the work that was done regarding the implementations of the AAA infrastructure in the case study of the GNSS Products Portal, and how the problems stated in section 1.2 were tackled.

The chapter is structured as follows:

- Section 3.2 — **GNSS Products Portal** — in this section we present the EPOS requirements for an AAA Infrastructure. We provide the authentication and identity management procedures that have been implemented, and end the section with the accounting module for user's actions;
- Section 3.3 — **Multiple Account Problem** — in this section we explain in more detail how the multiple account problem was solved by using an account aggregation;
- Section 3.4 — **EPOS AAAI** — in this section we show how the EPOS AAI works, including a use case of it with the GNSS Products Portal, and how can a user authenticate in this framework;
- Section 3.5 — **CLI Authentication** — in this section we detail why this is a necessary feature, how it is solved using tokens or a proxy, and how to obtain and use these tokens;
- Section 3.6 — **Multi Factor Authentication** — in this section the options for MFA in the GNSS Products Portal are presented, and we briefly detail how they work;
- Section 3.7 — **Other Implementations** — in this section we present other systems where the work done in this thesis was used.

3.2 GNSS Products Portal

The GNSS Products Portal is a web service that is currently being developed for the EPOS project, being a component of the software package that the TCS of GNSS Data and Products is providing in order to disseminate data — Geodetic Linking Advanced Software System (GLASS). Using the services provided by the GLASS API, the portal handles the dissemination of GNSS derived products, allowing researchers, scientists, and all the other interested users easy access to this type of Geodesic data, which is one of the EPOS project goals. However, the accounting of who is accessing this data (from where and which data) is one of the requirements that is necessary to fulfill. Therefore, and as referred in section 1.2, an AAA infrastructure is necessary in order to gain control over who is using the service and to enable the accounting of all this information.

Besides this EPOS requirement, some GNSS data providers have manifested their intention to have an embargo over their data, which will implicate the creation of an authorization mechanism for the data provided in the GNSS Products Portal. While the details of this embargo are not yet final, it is expected that data sets such as RINEX data files will have their access limited by date. This temporal embargo means that until a specific date, defined by the data owners, the access will be limited to a restrict group of users, until it becomes of open access for everyone. While the GLASS database is prepared for this situation, the portal lacked a way of managing the access from part of the users, and therefore was not able to restrict the access to only a specific group of users. This authorization is very important because services such as the GNSS Products Portal are responsible for the data that data providers make available. Usually, there are signed contracts with these data providers that only authorize specific purposes for the usage of this data, and unauthorized access or leaks of this data could pose a huge problem financial and legal.

What was decided is that there would be three options for an AAA infrastructure integration of EPOS with a TCS:

1. **No integration at all** — meaning that the web service would be completely open, without any possibility of control from EPOS and non-EPOS authenticated requests, and therefore unable to perform the desired accounting;
2. **Integration with EPOS AAI** — providing direct access for EPOS users, while taking advantage from the EPOS profile that will be shared between all the integrating services;
3. **Integration with the TCS own authentication mechanisms, or other third party IdP** — the TCS would be responsible for the user management, creating or implementing a system in their own way, with the possibility of integrating this IdP with the EPOS AAI, so that the user profile is still shared.

This EPOS user profile consists of a defined set of attributes and data related to an EPOS services user. Whilst this profile is still under development, it is possible to get an idea of what it will contain, as shown in table 3.1. The final version should also be fully compliant with the General Data Protection Regulation (GDPR), so that no complications arise from storing this information. It is important to decide what information should be present in the profile in order to provide a reasonable level of information without going against the regulation.

Attribute name	Required	Description
First name(s)	✓	Personal user data
Last name(s)	✓	Personal user data
Email	✓	For confirmation and association purposes
Institution		Can be confirmed through EduGAIN
Researcher status	✓	Yes/No
EPOS member	✓	Yes/No
Groups	✓	Special EPOS groups membership

Table 3.1: A draft version of the information that the EPOS user profile contains.

As a way of explaining how the ecosystem should work, the following policy was created: for the execution of a TCS web service, *AuthN* should be required, or, as an alternative, Integrated Core Services (ICS)'s API should create an unauthorized user token, indicating that the current

user is not authenticated. Also, the web services shall accept the EPOS method of *AuthN*, a token, and should not deny users that, coming from other web services, possess a valid EPOS AAI token. However, web services are allowed to secure their service in any way they feel appropriate, as long as it does not contradict the previous statement. Additionally, the services will need to be secured through Hypertext Transfer Protocol Secure (HTTPS), otherwise they are not eligible to use this mechanism, under the possibility of compromising the whole system. In this case, there will be no method of authentication provided for a web service that communicates through HTTP only. Also, this HTTP services will not be represented in accounting, and shall consider migrating to HTTPS. If a web service is fully open to the public, the authentication header can just be ignored. Since the GNSS Products Portal is already secured through HTTPS it was just necessary to adapt it, in order to accept every user that comes from other EPOS web services. The idea is to maximize the ease of use of the portal, by adopting multiple authentication mechanisms in order to attract a bigger variety of users.

3.2.1 Authentication and Identity Management

The GNSS Products Portal is a GUI built on top of REST web services. Since it was still in a development process, the implementation of an authentication system in the software was feasible. The decision was to create a local authentication system, that includes multiple IdPs and also integrates the EPOS IDM. These IdPs include `Google`, `ORCID`, EPOS IDM, and `A.GOV`. They are implemented using the `OAuth` protocol, and use different scopes for each IdP, depending on the attributes that are necessary. This variety of login options in the authentication process should make the registration of an account in the service more appealing to the user.

In order to improve the registration process, if the user does not own an account in the service and tries to login using one of the available IdP, an account is registered using the attributes shared by that IdP. This means that, if the user tries to log through `Google`, and they do not own an account on the service, the attributes shared in the `OAuth` protocol are used to register an account in the service. This registration is done locally on the service, but can later be associated to an EPOS profile, as it will be explained. To avoid the existence of multiple accounts for the same user, a mechanism of account aggregation was created, that will be introduced in section 3.3. There is also a mechanism to aggregate attributes from multiple sources. When the new IdP is associated, either by logging through them or by aggregating directly through the user profile as shown in figure 3.1, the attributes shared by the new IdP are compared to the existing ones. If the system detects a change in an attribute of the same type, such as the name, the user is questioned about which one they want to keep. Changes on the attributes on the side of the IdP are also reflected here. This allows to have the user's attributes up to date and according to their preference.

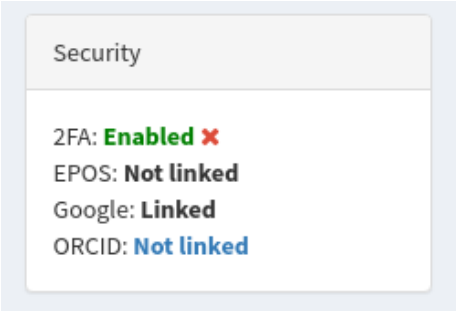

Figure 3.1: Association of new IdPs from the GNSS Products Portal user profile.

To assure the user's identity, two mechanisms were implemented. The first is the A.GOV authentication mechanisms, that provide a high level of assurance of the user's identity. But this method only works for users that have access to a Portuguese CC or the CMD. The second, available for anyone, is that MFA methods were implemented, which will be explained in section 3.6. While MFA does not provide a level of assurance over the user's identity as high as the A.GOV authentication, it does increase the level of assurance and adds another layer of security over the process. These methods should help to prevent the impersonation of the portal users. In figure 3.2 a graphical overview of the AAI is shown.

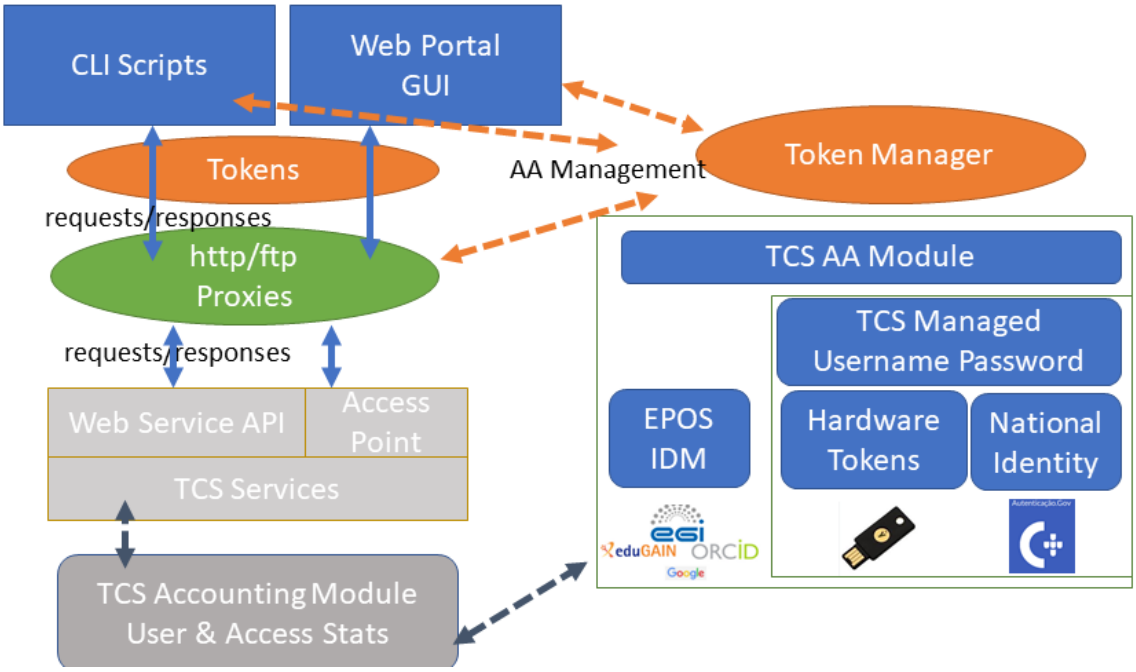

Figure 3.2: Overview of AAI in the GNSS Products Portal.

In figure 3.3, an activity diagram of the authentication process is presented. As previously referred, there are multiple ways for the users to authenticate themselves with a local password authentication and external IdPs, such as Google, ORCID, and A.GOV, as well as with the own EPOS IDM. These authentications are validated with the database, and allow a registration at that moment, if the user does not own an account already. The system always checks if, after identification, the user has an MFA enabled. If so access to sensitive data is only granted after a successful verification of that factor.

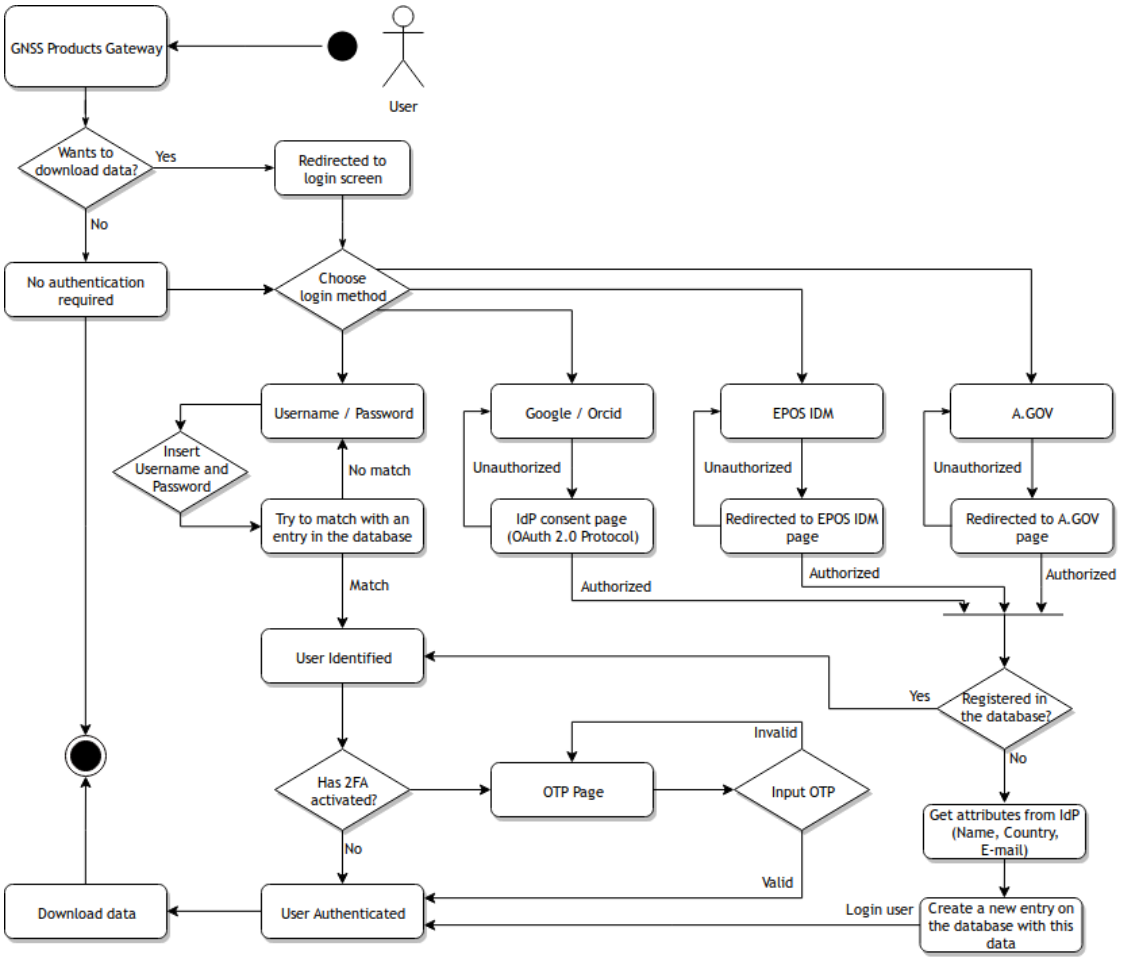

Figure 3.3: Activity diagram of the authentication at the GNSS Products Portal.

3.2.2 Accounting

Accounting is an important part of the system's workflow. The necessity of authenticating and identifying users comes from the requirement of knowing that the services are fulfilling their objectives. If a service is not attracting the expected users, then something has failed along the process and an identification of the issues is necessary. This accounting mechanisms help EPOS to identify which services need to be improved and where the focus should be.

For the GNSS Products Portal, statistics are retrieved from the user's actions. Since the web service allows the user to authenticate through other methods than EPOS IDM, these statistics need to be stored locally and later shared with EPOS AAI. Relevant statistics such as what type of data the user's are interested in the most, are stored in the database associated with their user's identity that can be used to match to an EPOS user profile, in order to provide a better experience. In case a user decides to delete their account in the service, all these statistics are anonymized, following strict rules to protect the user's privacy.

Other relevant statistics, such as what is the user's background (public sector, university, non-profit organizations, etc.), can be retrieved from the user's profile, if they decide to share that information. This helps to understand what purpose the data is going to have. Additional information, such as what countries access the services the most, can be retrieved from the

implemented Google Analytics. All this data statistics help to improve the services provided and fulfill the EPOS requirements.

3.3 Multiple Account Problem

One of the biggest problems we found during our research is that, while some services allow the user to authenticate using a variety of IdPs, like Google, Facebook, Twitter, and others, they not always log into the same account. This translates into multiple accounts for the same identity, where if the user authenticates using a different method than the one he previously used, he may end up in a brand-new account, with no linkage to the one he already owned.

This is a problem that affects multiple web services, some of which are quite popular. Depending on the web service, this problem consequences may differ. Sometimes the user is allowed to login using the IdP and identity in question, but ends up in a different, and possibly brand-new, account, instead of the pretended one, or the login may be simply denied. The results depend on the way that each web service handles the problem.

For a better understanding of this problem, we will present a use case that we have experienced in an external web service. Suppose we already have an account on the web service *A*, that we have previously registered through the manual registration form using the e-mail: *email@gmail.com*. We now try to use the Google's IdP to perform the login on our account, using the same e-mail address that we used to register in the service previously — *email@gmail.com*. Since the account was not initially registered using the Google's IdP mechanism, the connection between identities is not made and the system catalogs this entry as a new user, creating a brand-new account. We then end up with two accounts belonging to the same user, both using the same e-mail address, but different login methods.

Our vision is that, if a user links other third-party accounts to a service, such as an ORCID account, they should be able to use that external service as an IdP during the authentication process, as long as their identity is verified in both ends. This way, not only the multiple account problem is solved, but additional authentication options are provided to them. This issue is solved on the GNSS Products Portal by having a mechanism to aggregate these accounts. This mechanism tries to find the connections between different IdPs based on a unique primary attribute, such as the e-mail. For instance, if the user usually authenticates through EPOS IDM, but tries to authenticate with Google IdP using the e-mail associated with their EPOS profile, the system will make the connection and aggregate their account. In case a match exists, these accounts are linked and the user can use any of the IdPs to login into their account for the next time. Else, it is concluded that this is the first time the user authenticates in this service, and a brand-new account is registered. This provides a good solution to avoid multiple accounts for the same user or identity, and the difficulties involved with managing different settings and preferences from part of the users.

3.4 EPOS AAI

The EPOS AAI is an integrated service that serves three purposes for the EPOS community: authentication, authorization, and accounting. For authentication it is understood as the central EPOS service for maintaining user credentials, and that should ensure that the user is recognized in the EPOS community. For authorization, it is the source of authentication attributes, that were agreed by the EPOS community, that grant or deny access to specific resources and functions. For accounting, it will be the collector of the data usage by the users [83]. This service is based on `Unity-IDM`, providing a variety of relevant features to our problem, including registration in the IDM directly or through external IdPs, a profile information for the SPs, authentication based on `OAuth 2.0` tokens, and a federated authentication based on `OpenID Connect` scheme. The diagram in figure 3.4 presents an overview of how the authentication procedure works using the EPOS IDM as the selected IdP in GNSS Products Portal, followed by a use case.

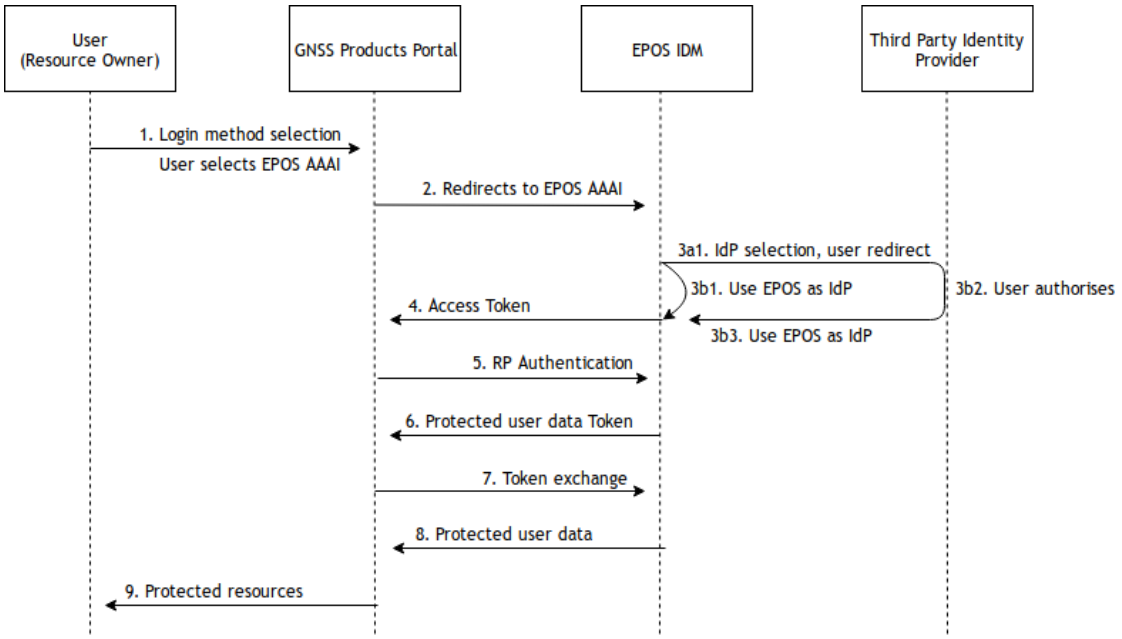

Figure 3.4: EPOS IDM authentication flow with the GNSS Products Portal

1. The user, which is the resource owner, accesses the GNSS Products Portal login page, wanting to retrieve some protected resources. They choose the EPOS AAI as their IdP, to authenticate themselves;

2. The user is then redirected to the EPOS AAI web page. At this point, if the user has already used the EPOS system as their IdP to GNSS Products Portal previously, and still has an active session in the EPOS system, all the procedures happen in background. In the user perspective, they do not even leave the GNSS Products Portal, jumping straight to point 9. Otherwise, they need to authenticate in EPOS IDM;

3. To authenticate before EPOS, the user may choose to use a third-party IdP, like `EGI Check-in`, going through steps 3a, 3b, and 3c presented in the figure, or authenticate using a previously registered account in EPOS;

4. After a successful login, an access token is sent to the GNSS Products Portal;

5. GNSS Products Portal sends the access token back, together with the secret credentials, redirect endpoint, and grant to EPOS IDM;
6. The EPOS IDM confirms that the GNSS Products Portal is the SP that it claims to be and sends a token that can be used to access the user attributes, depending on the grants that were exchanged previously;
7. The GNSS Products Portal now possesses a token that can be traded, during its lifetime, for the user's attributes. It is then sent to EPOS IDM in order to authenticate the user;
8. EPOS IDM returns then the user information that possesses, still according to the grants that were asked on the token request;
9. Now possessing the user information, GNSS Products Portal can authenticate the user and allow them to access the protected resources.

While the EPOS IDM is still in development, it already includes a wide variety of IdPs that can be used to authenticate inside the EPOS IDM. It is possible to use the `EGI Check-in` IdP to authenticate using federated credentials, which is the case of UBI, or another IdPs, like `Facebook` or `Linkedin`, that are not included in the GNSS Products Portal.

The exchange of messages is done by HTTPS requests, following the `OpenID Connect` protocol. A `GET` request is initially sent by the user's browser, upon selecting the EPOS IDM as the IdP, initiating the protocol. The rest of the protocol happens server side, where it cannot be intercepted by the user, using a `POST` request for obtaining the access token and a `GET` request for trading the token for the user information.

3.5 CLI Authentication

As referred before, the GNSS Products Portal is a GUI built on top of `REST` web service, GLASS Framework. While the portal serves as the main way of retrieving data products from this `REST` service, there is also an open API for when users want to interact directly with the framework, or use their own procedures, such as `Python`, `Bash` or `NodeJS` scripts. As it should be clear by now, this retrieval of data sets have to be authenticated and authorized, which means it is necessary to create an authentication procedure for CLI applications.

Using the account created in GNSS Products Portal, and having this account associated to an EPOS profile, it is possible to obtain an `OAuth` token in the user profile page of the portal, as shown in figure 3.5. This token, if valid, can then be passed in the headers of CLI requests to the API, using the attribute `Authorization` with the value `Token` followed by a space and the token. This token allows the API to identify who is requesting the data and check if they are authorized to do so, as well as account this request. If the token is not valid, the request will return a 401 status code. This CLI authentication can also be used for other TCSs in their legacy applications that can no longer be modified to include an authentication system directly in them. It is proposed that the requests pass through a proxy in the server, and authenticate with EPOS IDM, before interacting with APIs, as diagram in figure 3.6 shows.

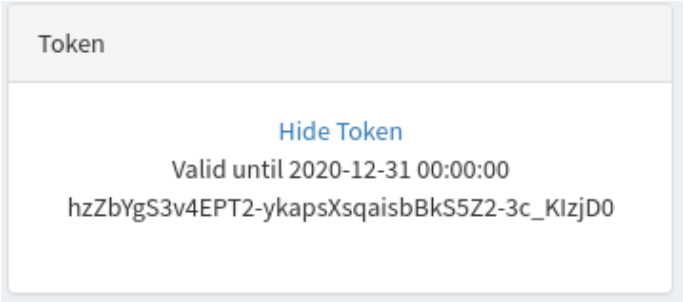

Figure 3.5: Example of an access token retrieval in GNSS Products Portal.

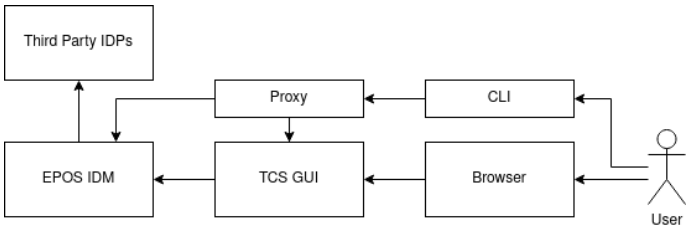

Figure 3.6: Diagram of TCS authentication through a proxy.

This token has an expiration date, so that it does not compromise security and to not have deprecated tokens active. When this token expires, the user should return to the GNSS Products Portal and require a new token, through the refresh button, that will show up in the same location where the valid token was, as presented in figure 3.7. This action starts the refresh token mechanism in the background using `OAuth` protocol. The GNSS Products Portal communicates with EPOS IDM and requests a new access token for this user.

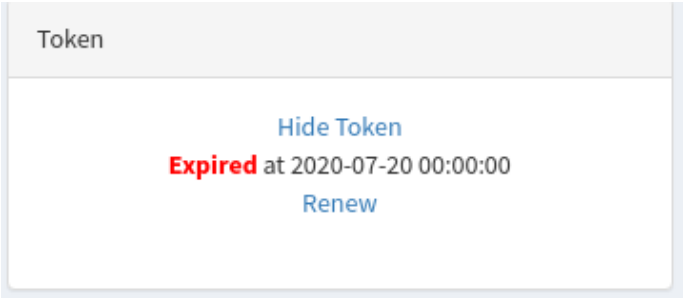

Figure 3.7: Example of an expired token in GNSS Products Portal.

3.6 Multi Factor Authentication

From reviewing current trends in authentication services we concluded that our services would benefit from using MFA. Adding a second authentication factor not only increases the security of the users, by reducing the possibility of unauthorized access to their account, but would also protect our services by denying increased privileges to possible attackers.

We already had the knowledge factor implemented with a password scheme authentication, which is still the most common scheme in everyday web services. By using an IdP (like `Google`) to authenticate, additional factors can be brought to the process. That would depend on which

security measures the user had active on their IdPs account, where an OTP or a biometric validation could be necessary to login, but it was also possible that every additional security measure was turned off. Another issue would be a hijack of a device where the user's IdP already had an active session, granting easy access to our services. The exception is the use of A.GOV for authentication, since this scheme often uses at least two authentication factors to conclude the process, independently of choosing CC or CMD as the method.

It was decided that the possession factor would be implemented, rather than the inherence factor, since it was more likely that the user would adopt this factor if given the choice between the two. However, it would be interesting to have the inherence factor implemented, and in section 5.4 we give our opinions about how this feature could be included.

Regarding this possession factor, two mechanisms have been implemented: firstly Google Authenticator, which is a time-based OTP, and Yubikey's OTP. The user may opt to enable only one of these and that will then be required as the second authentication step, after a correct setup of it. However, enabling MFA is not mandatory, and the user may keep it disabled.

3.6.1 Google Authenticator

The Google Authenticator works by possessing a mobile with an application capable of generating time-based OTPs installed. There are multiple applications for this purpose, not having to necessarily be Google's one. In the MFA enabling process it is necessary to store the secret key from where the OTPs derive, together with the current timestamp. The generated OTP then gets validated by the service where the authentication is ongoing.

For the user the process is quite simple. Firstly a Quick Response (QR) code is generated in the enabling process that contains information about the user's email, the service name, and a randomly generated sixteen alphanumerical characters secret key. In the setup page this QR code is shown to the user, so that it can be scanned using a QR code scanner application, like Authenticator. The secret key is also provided in plain text, if the user is not able to scan the QR code. After storing the secret, the user will then have offline access to a dynamic six digit PIN code that needs to be introduced during the login process.

This type of OTP is generated by hashing a combination of the secret with an epoch, and it is denominated Hash-based Message Authentication Code (HMAC)-based OTP. The hashing algorithm used in this implementation is HMAC-Secure Hash Algorithm (SHA)1. The validation of the OTP is done on the authentication server, which in this case is the GNSS Products Portal. This can be done since both the user and the service have the secret key in their possession. To prevent that the two ends use different epochs as input on hash generation, due to not being synchronized, the generated OTP is still valid even if the user is slightly ahead or behind the server. An example of the time-based OTP generation can be found in figure 3.8. This QR code is never shown again, and the secret cannot be retrieved after finishing the process.

Figure 3.8: Example of a QR code for the time-based OTP.

In order to improve the user's experience, and assure that the user has correctly setup the OTP, it is required that they test the generated OTP before enabling it for the next authentications. For this test, there is an input box under the presented QR code where they can test if the OTPs they are obtaining in the application are valid or not, as presented in figure 3.9. A successful validation is necessary before enabling the MFA.

Figure 3.9: Example of an OTP validation test.

No recovery keys are provided for this MFA. If the user loses access to the secret key, and therefore, loses access to the OTP generator, it will not be possible to login anymore. It is then necessary to contact the support and confirm their identity, in order to have the MFA disabled.

3.6.2 Yubikey OTP

The `Yubikey` MFA works differently from the `Google Authenticator`. This mechanism uses an hardware physical Universal 2nd Factor (U2F) token where the secret key is stored. The OTPs are generated inside the U2F token and transmitted into the user's device by connecting this token to the device, either through `USB`, `bluetooth` or `NFC`, and performing an action over the U2F token, such as pressing a button.

The OTP validation occurs in the `Yubico` servers, contrary to `Google Authenticator` that occurs locally. This means that the inserted OTP and token identifier are sent in a payload to `Yubico` and the authorization answer is returned to the web service. Since the process depends on a third-party, this authentication method can fail if the service can not reach

Yubico. However, the generated OTPs are much more secure than the ones generated by Google Authenticator. Another problem with this method is that not everyone own a U2F token, and might not have enough interest in getting one.

3.7 Other Implementations

3.7.1 EPOS

The work developed on this thesis, and applied to the GNSS Products Portal, was shared inside the GNSS TCS to be applied to other software of the group. This TCS provides all the necessary software for third parties to implement their own system, including the authentication framework developed in this thesis.

Regarding the EPOS community, this web service was one of the pilots implementing the EPOS AAI for authentication, and the work done on this service was shared with the community, including the suggestions on how to improve the EPOS AAI.

3.7.2 Collaboratory for Geosciences

The Portuguese Collaboratory for Geosciences (C4G) infrastructure is the EPOS project Portuguese representative. This distributed research infrastructure promotes the networking of researchers and the sharing of equipment, data, collections, and tools in SES. It compromises the disciplines of geology, hydrogeology, geochemistry, geodesy, geophysics, geomechanics, and geomathematics, with researchers of these disciplines actively participating [84].

For C4G services, an e-infrastructure will be implemented to harness the power of the distributed resources, easing effective collaboration, and allowing remote access to products and services [84]. This e-infrastructure is currently under development and one of its requirements is an AAI that allows the identification and verification of users identity.

The work done in this thesis will be used to improve the AAI system. Since C4G is a Portuguese infrastructure, the vast majority of users of this service are expected to be of Portuguese nationality, and therefore, expected to own a CC. It is also necessary that the service has a high level confidence authentication mechanism, since some resources available for distribution are of high value and great responsibility to the requester. The logical solution is to apply the A.GOV mechanisms here, since they provide a high level of trust and most users have access to it. Additionally, as it already happens with GNSS Products Portal, a considerable amount of the C4G users are researchers, and is expected that they own an ORCID account. Enabling the OAuth 2.0 login with ORCID as the IdP is also a valuable option for this service.

The work done in this thesis for the EPOS TCS was shared with C4G, so that it can be applied in their intranet and other services.

Chapter 4

Analysis

4.1 Introduction

This chapter analyzes the implemented AAA Infrastructure in the GNSS Products Portal, providing some analysis on the usability of the system and on the security of the mechanisms and data exchanges.

The chapter is structured as follows:

- Section 4.2 — **Usability** — in this section we present some of the usability analysis performed over the implemented authentication system and identity management in the GNSS Products Portal;
- Section 4.3 — **Security** — this section includes a description of some security measures that were implemented in the portal, and the analysis to the shared information with external parties.

4.2 Usability

In order to evaluate the usability of the developed AAI in the GNSS Products Portal, we decided to perform a series of tests that measured the speed, usability, and reliability of the different authentication methods that we make available in the GNSS Products Portal.

For a better visualization and understanding of this performance analysis, we decided to measure the average elapsed time that the authentication through that method takes, and how many redirects happen during that process. We consider a redirect to be a change of active page or window. For these tests we did not consider the MFA enabled on the GNSS Products Portal. The results are presented in table 4.1.

Authentication method	Average elapsed time (seconds)	Number of redirects
Password	9	1-2
Google's IdP	7[1]	1-3
ORCID's IdP	7[1]	1-3
EPOS IDM	10[1]	1-5
A.GOV	29[2]	4-6

Table 4.1: Usability of the authentication methods in the GNSS Products Portal.
[1] Will depend on if a session is already active or not.
[2] Will depend on which method is chosen.

In our opinion, the current system fits the user's needs and provides a good amount of authentication options, that do not compromise security or usability. The identity management,

with the up to date user attributes and attribute aggregation from multiple sources, improves the overall system. There was also a positive feedback for the implemented AAI in the GNSS Products Portal given by the EPOS User Feedback Group. From the statistics acquired, almost half of the users having their account associated with `Google`, one third of the users having their EPOS profile associated, and a smaller fraction having their accounts associated to `ORCID` and `A.GOV`. Regarding MFA, less than twenty percent of the users have activated this mechanism directly in the GNSS Products Portal.

4.3 Security

The communication between the TCS web services and third-parties, like the EPOS IDM, should be done through HTTPS which encrypts all the data exchange. By acting as *man-in-the-middle* between the GNSS Products Portal and external IdPs, we were able to perform an analysis on these communications using a network traffic analyzer, `Wireshark`, and using `TCPflow`. The analyzed traffic was encrypted as we expected, and no kind of useful data was obtainable from the payloads.

In table 4.2 it is possible to get an overview of how many authentication factors each method uses. The type of authentication factor is also specified with a check mark if it's always active or with a circle if it is optional. The total of different authentication factors is given in the last column.

Authentication method	I	K	L	P	Total
Password		✓		○	1-2
Google's IdP		✓	○	○	1-3
ORCID's IdP		✓		○	1-2
EPOS IDM		✓		○	1-2
A.GOV		✓		✓	2

Table 4.2: Overview of authentication factors per method in the GNSS Products Portal.

If the user has MFA enabled with `Google Authenticator`, in case the knowledge factor is compromised, he has the chance of one in a million to correctly guess the OTP and gain illicit access to the account. Since this is a time-based OTP, the risk of brute force is low, since the code is changing every minute. However, to diminish the possibility of a brute force attack, the OTP validation is delayed by a couple of seconds after some failed tries, with this delay increasing exponentially with each consecutive failed try, and the user is warned that someone may be trying to gain unauthorized access to their account. This delay also occurs after various incorrect insertions of e-mail/password credentials, as can be seen in figure 4.1.

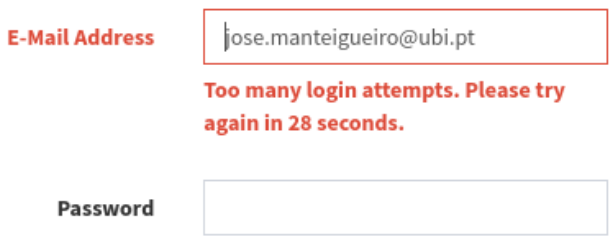

Figure 4.1: Delay after various incorrect e-mail/password tries in GNSS Products Portal.

Chapter 5

Conclusions and Future Work

5.1 Introduction

This chapter finishes the thesis with the conclusions of the work done, contributions provided and future work in the area.

The chapter is structured as follows:

- Section 5.2 — **Main Conclusions** — in this section we present some main conclusions over the work done through the thesis, such as the implementation of the AAI and the difficulties encountered;
- Section 5.3 — **Contributions** — in this section we refer the contributions that this thesis had in EPOS, C4G and the GNSS Data and Products TCS, as well as the article produced;
- Section 5.4 — **Future Work** — in this section we give our opinion about what can be done to improve authentication and identity management systems, as well as some remarks about the current status of authentication.

5.2 Main Conclusions

Through our review of the existing literature, we found a lot of work done in the field of authentication and identity management that use a variety of technologies and protocols. It is clear that this subject has constantly improved over the years, not only because it is a crucial part of the protection of resources of web services, but also because users are getting more aware of the risks that they are exposed to.

In our opinion, with the implementation of the procedures previously described, the authentication and authorization problems identified in the TCSs services have been addressed. In specific, in the case study reviewed, the implemented procedures solve the AAA problems and provide a good identity management of the users and their attributes. By having the registration available either in the portal or the EPOS IDM, the user is able to choose the option that they prefer. Authentication in the GUI can be done in a variety of ways, and the users are able to associate more IdPs for authentication, where the problem of having multiple accounts is solved. Authentication and Authorization for CLI applications is also tackled with the use of access tokens, so that the user can be identified and their actions accounted, and for legacy applications the usage of a proxy is recommended. However, there are still improvements that can be made, as it will be referred in section 5.4.

As for the implementation of these procedures, we found that there is a lack of documentation for some IdPs, creating great difficulties in the integration of them in web services, which is the case with the implementation of `A.GOV` authentication in other than `.NET` services. Regarding the OAuth protocol we conclude that there should be a standardization of the attributes shared by the IdPs, since this is an aspect that could be easily improved and the web services would highly benefit from it. The management of access tokens is not simple as we would like it to be, and it is an aspect that needs refinement. For biometric and continuous authentication, we conclude that there is already a good formal idea of what should be their purposes and procedures, but there is a lack of reliable and feasible implementations in real world environments.

5.3 Contributions

As referred in section 3.7, the work done in this thesis was shared with other services that faced the same problems. The `A.GOV` authentication method was used by C4G in order to improve the authentication mechanism in the services they provide. Inside the working group of GNSS Data and Products, the authentication system was shared in an effort to have a seamless and shared access to all the services provided. The introduction of authentication in the GLASS API, using the procedures described in section 3.5, provide an authentication method for most of the services of this TCS. The conclusions obtained from implementing EPOS IDM as an IdP for GNSS Products Portal was shared with the EPOS community, and the issues faced helped to improve the current system.

From the work done in this thesis resulted the article *Identity Management and Access Control for the GNSS Community within a European Research Infrastructure*, accepted for *COMPSAC 2020*, that can be found in appendix B.

5.4 Future Work

After the work done in this thesis, we leave here our comments to what can be improved or done as a future work.

Regarding EPOS IDM, we think that the creation of an intermediary IDM on the TCS side can be beneficial to provide more control over token management. This would enable the TCS to create, refresh, and delete the access tokens, providing a better management of user identities. Being able to decide the token's lifetime (e.g. one access only, one month, etc.) would provide an overall better access control.

As already referred before, biometric and continuous authentication need to become more common as authentication methods. Once these options are widely adopted, we can start implementing it on our authentication system. One of our suggestions is to use the version 3 of `Google Recaptcha`, which is already able to perform a silent and unobtrusive validation of the user, and adapt it to authentication, joining various other biometric factors like keystrokes. Another suggestion is to implement a global logout mechanism in `OAuth` protocol, that when the user performs a logout action, all the SP sessions authenticated through that IdP become invalid.

In general, a continuous improvement of the existing authentication schemes is necessary. New threats are always appearing, intentionally or not, and the information of authentication needs to be secured. While the existing mechanisms may be safe now, a continuous research in search for improvements must exist. In terms of usability, an improvement of speed and security is always welcomed.

Bibliography

[1] EPOS, "What is epos," 2020. [Online]. Available: https://www.epos-ip.org/about/what-epos 1

[2] EPOS, "Epos for scientists and researchers," 2020. [Online]. Available: https://www.epos-ip.org/who-benefits/epos-scientists-and-researchers 1

[3] EPOS, "Who benefits," 2020. [Online]. Available: https://www.epos-ip.org/who-benefits 1

[4] EPOS-WP10, "Epos - gnss products portal," 2020. [Online]. Available: https://gnssproducts.epos.ubi.pt/ 1, 3

[5] L. Féres, "Epos gnss thematic core service portal," 2019. [Online]. Available: http://gnss-epos.eu 2

[6] F. B. Schneider, "Something you know, have, or are," 2005. [Online]. Available: https://www.cs.cornell.edu/courses/cs513/2005fa/nnlauthpeople.html 2

[7] S. Furnell, I. Papadopoulos, and P. Dowland, "A long-term trial of alternative user authentication technologies," *Information Management & Computer Security*, vol. 12, no. 2, pp. 178–190, 2004. [Online]. Available: https://doi.org/10.1108/09685220410530816 2

[8] C. Jacomme and S. Kremer, "An extensive formal analysis of multi-factor authentication protocols," in *Proceedings of the 31st IEEE Computer Security Foundations Symposium (CSF'18)*. Oxford, UK: IEEE Computer Society Press, 7 2018, pp. 1–15. 2

[9] M. Kabiru Hamza, H. Abubakar, and M. Yusuf, "Identity and access management system: a web-based approach for an enterprise," *Path of Science*, vol. 4, pp. 2001–2011, 11 2018. 2, 12

[10] J. Gouveia, P. A. Crocker, S. M. d. Sousa, and R. Azevedo, "E-id authentication and uniform access to cloud storage service providers," in *2013 IEEE 5th International Conference on Cloud Computing Technology and Science*, vol. 1, Dec 2013, pp. 487–492. 2, 21

[11] Unity, "Unity idm," 2019. [Online]. Available: https://www.unity-idm.eu/ 3, 65

[12] A. C. AGH, "Epos - aaai," 2020. [Online]. Available: https://aaai.epos-eu.org/ 3

[13] F. Aloul, S. Zahidi, and W. El-Hajj, "Two factor authentication using mobile phones," in *2009 IEEE/ACS International Conference on Computer Systems and Applications*. IEEE, 2009, pp. 641–644. 8

[14] S. Preibusch and J. Bonneau, "The password game: negative externalities from weak password practices," in *International Conference on Decision and Game Theory for Security*. Springer, 2010, pp. 192–207. 8

[15] J. Goldberg, "What does "mfa" mean?" 2018. [Online]. Available: https://s3.amazonaws.com/com-agilebits-users/goldberg/MFA/mfa.pdf 8, 9, 10

[16] Yubico, "The 2020 State of Password and Authentication Security Behaviors Report," 2020. [Online]. Available: https://www.yubico.com/wp-content/uploads/2020/02/Ponemon-Infographic-2020-final.pdf 8

[17] F. Al Maqbali and C. J. Mitchell, "Email-based password recovery-risking or rescuing users?" in *2018 International Carnahan Conference on Security Technology (ICCST)*. IEEE, 2018, pp. 1–5. 8

[18] M. Raible, "What the heck is oauth?" 2017. [Online]. Available: https://developer.okta.com/blog/2017/06/21/what-the-heck-is-oauth 8, 13

[19] R. Paul, "Oauth and oauth wrap: defeating the password anti-pattern," 2010. [Online]. Available: https://arstechnica.com/information-technology/2010/01/oauth-and-oauth-wrap-defeating-the-password-anti-pattern 9

[20] A. Armando, R. Carbone, L. Compagna, J. Cuéllar, G. Pellegrino, and A. Sorniotti, "An authentication flaw in browser-based Single Sign-On protocols: Impact and remediations," *Computers and Security*, 2013. 9

[21] D. Wang and P. Wang, "Two birds with one stone: Two-factor authentication with security beyond conventional bound," *IEEE transactions on dependable and secure computing*, vol. 15, no. 4, pp. 708–722, 2018. 9

[22] D. Dasgupta, A. Roy, and A. Nag, "Authentication basics," in *Advances in User Authentication*. Springer, 2017, pp. 1–36. 9, 10

[23] I. Velásquez, A. Caro, and A. Rodríguez, "Authentication schemes and methods: a systematic literature review," *Information and Software Technology*, vol. 94, 09 2017. 9, 10

[24] W.-H. Yang and S.-P. Shieh, "Password authentication schemes with smart cards," *Computers & Security*, vol. 18, no. 8, pp. 727–733, 1999. 9

[25] G. Yang, D. S. Wong, H. Wang, and X. Deng, "Two-factor mutual authentication based on smart cards and passwords," *Journal of computer and system sciences*, vol. 74, no. 7, pp. 1160–1172, 2008. 9

[26] A. T. B. Jin, D. N. C. Ling, and A. Goh, "Biohashing: two factor authentication featuring fingerprint data and tokenised random number," *Pattern recognition*, vol. 37, no. 11, pp. 2245–2255, 2004. 9

[27] D. Wang, D. He, P. Wang, and C.-H. Chu, "Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 4, pp. 428–442, 2014. 9

[28] D. S. Bonalle and G. Salow, "Biometric safeguard method for use with a smartcard," Jan. 15 2008, uS Patent 7,318,550. 9

[29] F. Hao, R. Anderson, and J. Daugman, "Combining cryptography with biometrics effectively," University of Cambridge, Computer Laboratory, Tech. Rep., 2005. 9

[30] D. Dasgupta, A. Roy, and A. Nag, "Biometric authentication," in *Advances in User Authentication*. Springer, 2017, pp. 37–84. 10, 26, 27

[31] D. Dasgupta, A. Roy, and A. Nag, "Toward the design of adaptive selection strategies for multi-factor authentication," *Computers & Security*, vol. 63, 09 2016. 11

[32] S. Koussa, "Comparing the top 3 federated identity providers," 2018. [Online]. Available: https://www.softwaresecured.com/federated-identities-openid-vs-saml-vs-oauth 11, 12, 19

[33] S. Koussa, "Differentiating federated identities: Openid connect, saml v2.0, and oauth 2.0," 2017. [Online]. Available: https://www.softwaresecured.com/differentiating-federated-identities-openid-connect-saml-v2-0-and-oauth-2-0 11

[34] D. Reed and E. Maler, "The venn of identity: Options and issues in federated identity management," *IEEE Security & Privacy*, vol. 6, pp. 16–23, 03 2008. [Online]. Available: doi.ieeecomputersociety.org/10.1109/MSP.2008.50 11, 12, 13

[35] G. Danezis, J. Domingo-Ferrer, M. Hansen, J. Hoepman, D. L. Métayer, R. Tirtea, and S. Schiffner, "Privacy and data protection by design - from policy to engineering," *CoRR*, vol. abs/1501.03726, 2015. [Online]. Available: http://arxiv.org/abs/1501.03726 11

[36] F. Nie, F. Xu, and R. Qi, "SAML-based single sign-on for legacy system," in *IEEE International Conference on Automation and Logistics, ICAL*, 2012. 12, 14

[37] A. Pfitzmann and M. Hansen, "Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management-a consolidated proposal for terminology," *Version v0*, vol. 31, p. 15, 2008. 12

[38] U. of Guelph, "What are the benefits of single sign-on (sso)?" 2018. [Online]. Available: https://www.uoguelph.ca/ccs/security/internet/single-sign-sso/benefits 12

[39] N. Ragouzis, J. Hughes, R. Philpott, E. Maler, P. Madsen, and T. Scavo, "Security Assertion Markup Language (SAML) V2.0 Technical Overview (OASIS)," OASIS, Tech. Rep., 2007. 13

[40] P. Otemuyiwa, "How saml authentication works," 2016. [Online]. Available: https://auth0.com/blog/how-saml-authentication-works/ 14, 15

[41] B. Leiba, "OAuth web authorization protocol," *IEEE Internet Computing*, 2012. 15

[42] D. Hardt, "The OAuth 2.0 Authorization Framework," RFC 6749, Oct. 2012. [Online]. Available: https://rfc-editor.org/rfc/rfc6749.txt 15, 16

[43] E. Hammer-Lahav, "The OAuth 1.0 Protocol," RFC 5849, Apr. 2010. [Online]. Available: https://rfc-editor.org/rfc/rfc5849.txt 15

[44] P. Fremantle, B. Aziz, J. Kopecký, and P. Scott, "Federated identity and access management for the internet of things," in *2014 International Workshop on Secure Internet of Things*, Sep. 2014, pp. 10–17. 15

[45] T. Spencer, "Api security: Deep dive into oauth and openid connect," 2018. [Online]. Available: https://nordicapis.com/api-security-oauth-openid-connect-depth 15

[46] D. Fett, R. Küsters, and G. Schmitz, "A Comprehensive Formal Security Analysis of OAuth 2.0," in *A Comprehensive Formal Security Analysis of OAuth 2.0*, 2016. 16, 19

[47] E. Torroglosa-García, A. D. Pérez-Morales, P. Martinez-Julia, and D. R. Lopez, "Integration of the oauth and web service family security standards," *Computer networks*, vol. 57, no. 10, pp. 2233–2249, 2013. 16

[48] W. Denniss, J. Bradley, M. Jones, and H. Tschofenig, "draft-ietf-oauth-device-flow-09: Oauth 2.0 device flow for browserless and input constrained devices," 2018. [Online]. Available: https://self-issued.info/docs/draft-ietf-oauth-device-flow-09.html 16

[49] M. Anicas, "An introduction to oauth 2," 2014. [Online]. Available: https://www.digitalocean.com/community/tutorials/an-introduction-to-oauth-2 17

[50] M. Rogan, "Whitehat security: Top 10 application security vulnerabilities of 2018," 2019. [Online]. Available: https://www.whitehatsec.com/blog/whitehat-security-top-10-application-security-vulnerabilities-of-2018 17

[51] D. Recordon and D. Reed, "Openid 2.0: A platform for user-centric identity management," in *The second ACM workshop on Digital identity management (DIM 2006)*, 2006. 18

[52] B. Kissel, "Openid 2009 year in review," 2009. [Online]. Available: https://openid.net/2009/12/16/openid-2009-year-in-review 19

[53] D. Thibeau, "Openid's second act: Openid connect," 2011. [Online]. Available: https://openid.net/2011/05/20/openids-second-act-openid-connect 19

[54] Onelogin, "Openid connect single sign-on (sso)," 2018. [Online]. Available: https://www.onelogin.com/pages/openid-connect 19

[55] O. Foundation, "Openid authentication 2.0," 2007. [Online]. Available: https://openid.net/specs/openid-authentication-2_0.html 19

[56] Agência para a Modernização Administrativa, "Autenticação.gov - manual de integração," 2018. [Online]. Available: https://www.autenticacao.gov.pt/documentos 20, 21, 22

[57] Justiça.gov, "Cartão de cidadão," 2020. [Online]. Available: https://justica.gov.pt/Registos/Identificacao/Cartao-de-Cidadao 21

[58] Gemalto, "10 years of eid: Portugal's citizen card," 2019. [Online]. Available: https://www.gemalto.com/govt/customer-cases/portugal-id 21

[59] H. Gomes, J. P. Cunha, and A. Zúquete, "Authentication architecture for ehealth professionals," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, 2007, pp. 1583–1600. 21

[60] F. Pimenta, C. Teixeira, and J. S. Pinto, "Globalid: Federated identity provider associated with national citizen's card," in *5th Iberian Conference on Information Systems and Technologies*, June 2010, pp. 1–6. 21

[61] F. Pimenta, C. Teixeira, and J. S. Pinto, "Globalid - privacy concerns on a federated identity provider associated with the users' national citizen's card," in *2010 Third International Conference on Advances in Human-Oriented and Personalized Mechanisms, Technologies and Services*, Aug 2010, pp. 16–21. 21

[62] Gemalto, "Biometrics: authentication and identification," 2019. [Online]. Available: https://www.gemalto.com/govt/inspired/biometrics/portugal 23, 24, 26

[63] L. Coventry, A. De Angeli, and G. Johnson, "Usability and biometric verification at the atm interface," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '03. New York, NY, USA: ACM, 2003, pp. 153–160. [Online]. Available: http://doi.acm.org/10.1145/642611.642639 23, 24, 26

[64] P. Tsai, M. K. Khan, J. Pan, and B. Liao, "Interactive artificial bee colony supported passive continuous authentication system," *IEEE Systems Journal*, vol. 8, no. 2, pp. 395–405, June 2014. 23, 27, 28

[65] S. Chen, A. Pande, and P. Mohapatra, "Sensor-assisted facial recognition: An enhanced biometric authentication system for smartphones," in *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '14. New York, NY, USA: ACM, 2014, pp. 109–122. [Online]. Available: http://doi.acm.org/10.1145/2594368.2594373 24, 25

[66] Gemalto, "Biometrics with strict confidentiality | the portuguese experience," 2018. [Online]. Available: https://www.gemalto.com/govt/inspired/biometrics/portugal 24

[67] A. K. Jain, R. Bolle, and S. Pankanti, *Biometrics: personal identification in networked society*. Springer Science & Business Media, 2006, vol. 479. 24

[68] C. Burt, ""smile-to-pay" facial recognition system now at 300 locations in china," 2018. [Online]. Available: https://www.biometricupdate.com/201811/smile-to-pay-facial-recognition-system-now-at-300-locations-in-china 25

[69] Q. Anh Tran, "Finnish grocery retailing market assessment for the deployment of payment innovation: Case: Uniqul face recognition payment application," 2016. 25

[70] R. Ramachandra and C. Busch, "Presentation attack detection methods for face recognition systems: A comprehensive survey," *ACM Comput. Surv.*, vol. 50, no. 1, pp. 8:1–8:37, Mar. 2017. [Online]. Available: http://doi.acm.org/10.1145/3038924 25

[71] K. Sundararajan and D. L. Woodard, "Deep learning for biometrics: A survey," *ACM Comput. Surv.*, vol. 51, no. 3, pp. 65:1–65:34, May 2018. [Online]. Available: http://doi.acm.org/10.1145/3190618 25

[72] A. De Luca, A. Hang, E. von Zezschwitz, and H. Hussmann, "I feel like i'm taking selfies all day!: Towards understanding biometric authentication on smartphones," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI '15. New York, NY, USA: ACM, 2015, pp. 1411–1414. [Online]. Available: http://doi.acm.org/10.1145/2702123.2702141 25

[73] B. S. E. Ahmed and H. T. I. Elshoush, "Biometrics solutions for blind person authentication," in *2017 Intelligent Systems Conference (IntelliSys)*. IEEE, 2017, pp. 1053–1058. 26

[74] R. Clarke, "Human identification in information systems: Management challenges and public policy issues," *Information Technology & People*, vol. 7, no. 4, pp. 6–37, 1994. 26

[75] K. Niinuma, U. Park, and A. K. Jain, "Soft biometric traits for continuous user authentication," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 771–780, Dec 2010. 27

[76] R. P. Guidorizzi, "Security: Active authentication," *IT Professional*, vol. 15, no. 4, pp. 4–7, July 2013. 27

[77] P. R. S., U. Kandaswamy, K. Balakrishnan, and B. Soumiya, "Analysis on continuous authentication system using multimodal biometrics," *International Journal of Applied Engineering Research*, vol. 9, pp. 5082–5087, 01 2014. 27

[78] E. System, "Rethinking iam: Continuous authentication as a new security standard," 2019. [Online]. Available: https://www.ekransystem.com/en/blog/continuous-authentication 27

[79] H. Bae and S. Kim, "Real-time face detection and recognition using hybrid-information extracted from face space and facial features," *Image and Vision Computing*, vol. 23, no. 13, pp. 1181–1191, 2005. 28

[80] T. Sim, S. Zhang, R. Janakiraman, and S. Kumar, "Continuous verification using multimodal biometrics," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 4, pp. 687–700, 2007. 28

[81] D. Kim, J. Lee, H.-S. Yoon, and E.-Y. Cha, "A non-cooperative user authentication system in robot environments," *IEEE Transactions on Consumer electronics*, vol. 53, no. 2, pp. 804–811, 2007. 28

[82] L. Fridman, S. Weber, R. Greenstadt, and M. Kam, "Active authentication on mobile devices via stylometry, application usage, web browsing, and gps location," *IEEE Systems Journal*, vol. 11, no. 2, pp. 513–521, 2016. 28

[83] T. Szepieniec, "Guide to epos aaai for service providers," 2019. 35

[84] C4G, "About c4g," 2019. [Online]. Available: http://www.c4g-pt.eu/about/ 40

[85] Agência para a Modernização Administrativa, "Autenticação.gov," 2019. [Online]. Available: https://www.autenticacao.gov.pt 55, 56

[86] J. J. Garrett *et al.*, "Ajax: A new approach to web applications," 2005. 65

[87] T. Akiyama, T. Nishimura, K. Yamaji, M. Nakamura, and Y. Okabe, "Design and implementation of a functional extension framework for authn & authz federation infrastructure using web browser add-on," in *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*. IEEE, 2013, pp. 389–396. 65

[88] B. J. Blackburn and L. Holford-Strevens, *The Oxford companion to the year: An exploration of calendar customs and time-reckoning*. Oxford University Press Oxford, England, 1999. 65

[89] E. Parliament and C. of the European Union, "Regulation (eu) 2016/679 of the european parliament and of the council," 2019. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:32016R0679 65

[90] D. Odijk, N. Nadarajah, S. Zaminpardaz, and P. J. Teunissen, "Gps, galileo, qzss and irnss differential isbs: estimation and application," *GPS solutions*, vol. 21, no. 2, pp. 439–450, 2017. 65

[91] P. Davis, N. Sakimura, M. Lindelsee, and G. Wachob, "Extensible resource identifier (xri)," *Syntax*, vol. 2, p. 0, 2005. 65

[92] H. Haas and A. Brown, "Web services glossary," *W3C Working Group Note (11 February 2004)*, vol. 9, pp. 784–786, 2004. 65

[93] P. Teunissen and O. Montenbruck, *Springer handbook of global navigation satellite systems*. Springer, 2017. 65

Appendix A

Autenticação.GOV

Year	Number of authentications
2014	147796
2015	1001277
2016	1381474
2017	2148304
2018	3093017
2019	7491226
2020[1]	8001947
Total	23265041

Table A.1: Number of A.GOV authentications per year, until August 2020 [85].

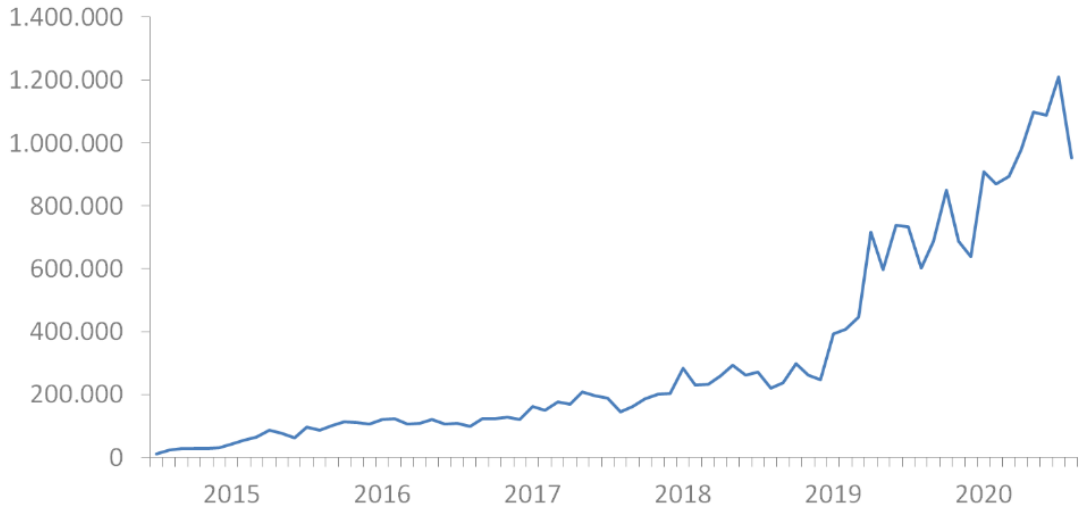

Figure A.1: Growth of the number of authentications using the A.GOV, until August 2020 [85].

Figure A.2: Type of authentication method for A.GOV authentications, until August 2020 [85].

Appendix B

Identity Management and Access Control for the GNSS Community within a European Research Infrastructure

Identity Management and Access Control for the GNSS Community within a European Research Infrastructure

1st José Manteigueiro
C4G - Collaboratory for Geosciences
University of Beira Interior
Covilhã, Portugal
jose.manteigueiro@c4g-pt.eu

2nd Paul Crocker
Institute of Telecommunications
University of Beira Interior
Covilhã, Portugal
crocker@di.ubi.pt

3rd Carlos Barrico
INESC Coimbra
University of Beira Interior
Covilhã, Portugal
cbarrico@di.ubi.pt

***Abstract*—Identity and access management systems aim to simplify authentication and authorization, however they do not solve all the problems that an application have related to issues such as the aggregation of user identities, token management and usage for user applications. This article describes the authentication and authorization problems related to service-oriented architectures that are built on web services, specifically, how systems may aggregate identities for authorization and statistical usage purposes, how they manage tokens for web services built for open systems, by using proxy servers, and how systems should handle user applications that need to manage identity tokens. We discuss the difficulties and implementation solution adopted for the Global Navigation Satellite System Community within the European Plate Observing System research infrastructure project. In particular the paper discusses the thematic core service users that are using multiple identities and accessing services using both web portals and command line interfaces that are built on accessing RESTful web services.**

***Index Terms*—GNSS, Identity, Authorization, WebServices, Management**

I. Introduction

The European Plate Observing System (EPOS) project is a long-term project to facilitate integrated use of data, data products, services and facilities from distributed research infrastructures of Solid Earth Sciences (SES) in Europe. Since it integrates multiple areas of SES, each with their own types of data and web services, it is difficult to standardize the necessary procedures to authenticate and authorize registered users when accessing data, and also to collect statistics on data access. Another problem is that, although accessing data through web portals is a fairly standard operation, involving some sort of login process, many users are interested in using a whole variety of command line tools to access and manipulate the data. In this scenario, obtaining and using the necessary access tokens is a non-trivial procedure. One should also add that these SES communities have many web services, legacy code, and well-defined interfaces and implementations that often do not mandate the use of any authorization or authentication, being completely open. Here we include File Transfer Protocol (FTP) servers with anonymous access, as well as web services over standard Hypertext Transfer Protocol (HTTP) protocols.

This document focuses on solving the problems of authorizing users to retrieve Global Navigation Satellite Systems (GNSS) data and products, not only via the GNSS web portals but also through Command Line Interface (CLI) applications, that make use of the communities RESTful Application Programming Interfaces (API). In order to authenticate through CLI, users need to change their procedures, so that they can include additional information in their requests. It is necessary to pass authentication data about themselves to a proxy on the server side, where they can be validated and their actions accounted and logged. The difficulty is obtaining, managing and using this data, in the format of tokens.

The problem of having multiple accounts for the same user or identity is also tackled in this document. This multiple identity problem is approached by implementing methods that try to find the connection between the various Identity Providers (IdP), and aggregate their attributes into a single account. The registration problem is solved by allowing users to register locally using the web portal, through a form or by a third party IdP, and later link their account to the EPOS central system. Identity assurance is also a problem. The more popular IdPs, from the big platforms (Google, Facebook, etc) are seen as more user-friendly, since it is highly likely that a user will already have an account created and logged in these platforms. However, they offer a lower level of assurance when compared to some other solutions, that require an additional effort from the user side. We try to combine these options in order to obtain a balance between assurance and ease of use.

The rest of this document is organized in the following way, we firstly give an overview of some general problems in identity management. We then discuss the problems of most concern to the GNSS community, followed by the proposed methodology to solve these problems. Afterwards, the details of our specific case study implementation are given, and the document finishes with a discussion, conclusion and future work.

II. Identity Management

A. Attribute sharing

Federated Identity Management (FIM) systems separate entities that enroll users, and are able to identify them, from entities that rely on the result of the authentication process [1]. They are a set of technologies and processes that allow the distribution of identity information to be done dynamically. When it follows the standards, as in [2], FIM facilitates the sharing of user's attributes and the process of authentication and authorization. There are three major federated identity protocols: OAuth 2.0, SAML, and OpenID Connect. Table I represents a small set of commonalities between these protocols that are relevant for our case study [3]. The addition of the single sign-on feature to FIM protocols also brings the benefit of only having to authenticate once in order to be logged into successive service providers within the federation [4].

TABLE I
FIM protocols properties

Property	Federated Identity Protocols		
	OAuth 2.0	*SAML*	*OpenID*
Enables Direct Interactions between IdP and SP	✓	✓	✓
Has a goal of consistent user interface	✓		✓
Can self-assert attributes	✓		✓
Enables user-centric identity	✓	✓	✓

While this FIM protocols enable security in attribute sharing between IdPs and Service Providers (SP), they still have flaws and need to be implemented with care, since they may handle sensitive information about the users [5].

B. Attribute aggregation model

Attribute aggregation, as in [6], is the combination of attributes of the same identity, provided from multiple sources, and which are then merged into the single identity itself. This aggregation is more often found in the context of multiple federated identities providing attributes to a single domain, since isolated domains do not have the necessity of exchanging attributes with other entities. SPs rely on IdPs to authenticate and provide user's identity attributes, and from there decide to grant or deny access to their resources. One limitation is that the SP can only retrieve attributes from one IdP within a single login session, and the ability of retrieving attributes from multiple IdPs would be very handy [7]. Chadwick et al. created an attribute aggregation layer for Windows Card Space [8], that gathers attributes from multiple IdPs, like a user presenting multiple cards in real life to assert multiple attributes. In [5], we can find another example of an attribute aggregation model that takes place at the SP level, where at each IdP there is a registration phase, with a consequential level of assurance registration, and an authentication phase, also with a consequential level of assurance choice. This system enables different levels of assurance for each session at the SP, where the selected level of assurance for the current session can never exceed the one selected during the registration phase, where otherwise it would mean that the session is asserting that the user's attributes are known to a higher level than when they were registered. Ferdous et al. [9], [10], concluded that there were no secure, usable and realistic models for attribute aggregation, and that more research and development was necessary in this area. They also discuss the possibility of two IdPs acting as IdP and SP to each other in a way to securely share attributes by delegating the authentication to another IdP and aggregating attributes in that way, adding another level of complexity to this process.

C. Authentication in the API

There is already a good amount of research in the area of authentication for Representational State Transfer (REST) applications, as in [11], that is the technology used for the backend of the GNSS Products Portal. Bradley et al. suggest this to be done by resorting to a JavaScript Object Notation (JSON) encapsulation and JSON web signatures, which offer integrity protection using symmetric and asymmetric cryptography [12]. In [13], we can find some other solutions for OAuth 2.0 authentication on a REST application.

III. Problem Statement

EPOS integrates multiple Thematic Core Services (TCS) from Scientific Communities, such as Geodesy, Seismology, Anthropogenic Hazards, etc. each developing their own platforms and web services. No standards regarding programming languages, databases, or protocols are imposed on these heterogeneous communities. This makes it difficult when trying to specify and develop common functionalities across all the developed services.

One of the most requested feature across the communities is the existence of a user account system that can work for every service in which user attributes can be shared between TCS, and users can use the same accountfor all the EPOS services. There is also a requirement from the European Research Infrastructures to track data access and usage, and report statistics about the services being developed and implemented. The task would be simpler if all the TCS software followed a harmonized structure and used the same information technologies and protocols, so that each service could just delegate all the authentication and accounting process to a centralized identity provider the same way.

Since this is not the case, an Identity Manager (IdM) was developed for EPOS, that should be used by every TCS, and to which the authentication process should be delegated. This IdM is based on Unity IdM[1] and uses OAuth 2.0 tokens for authentication. The service is available at the EPOS IdM page[2] and offers a variety of login methods via IdPs such as EGI Check-in, B2ACCESS, Google, and also the ability of registering as a new user directly. The EPOS IDM is offered as a login option in the GNSS TCS, for the GNSS Products

[1]https://www.unity-idm.eu/
[2]https://aaai.epos-eu.org/

Portal[3], but does not, by itself, solve all of the difficulties that we have identified.

a) : Registration

The problem starts on the first visit of a user to the TCS web service, which should be through the Graphical User Interface (GUI). This is one of the most common use cases, since it is expected that, when researching information about one of EPOS SES areas, the user comes across this web service, and wants to obtain data and products from it. The EPOS policy regarding data retrieval is that the user can freely access all the metadata present in the web services, but needs to authenticate to download the actual data, in order to be accounted. In this use case, this is the user's first visit to the web service, so they will need to register an account, provide their information, and, depending on the web service, be validated. There are two possible scenarios for registration:

- They register through the local TCS mechanism;
- They register using an IdP;

In the first case, all the information is handled on the TCS side, and this information needs to be shared with the central EPOS IdM, so that an account global to the EPOS ecosystem can be created. The attributes that the central system require need to be compliant with the information that the TCS gathers. The TCS can, however, gather more information than that required by the EPOS IdM, and some TCSs might want that. In the second case, the authentication process is delegated to the selected IdP, that sends the user's attributes to the TCS, for the account registration. If the TCS requires additional attributes, it needs to require them after receiving the IdP attributes. Some IdPs allow more broad attribute scopes than others, and this should be taken into consideration when offering the registration through IdPs.

A. Authentication by GUI

In the use case where the user is already registered and is accessing the GUI of a TCS, user authentication should be done by inserting their credentials in a login form. If the TCS is delegating this procedure to the EPOS IdM, the user will be redirected to the EPOS IdM page, and once logged there, it will be redirected back to the TCS with the necessary OAuth tokens to ensure their identity. If the authentication is done locally, or using any other third party IdP, it is necessary to create a way of ensuring that this information is transmitted to the EPOS IdM, so that they can account this login session. Even better would be that the cookies can be used in other TCS web services, avoiding the necessity of a new login from the user, improving the quality of the process.

B. Level of assurance

While some of the biggest internet platforms are highly popular as IdPs for all kinds of web services, they do not offer a high level of assurance. It is very easy to create an account on one of these platforms, trying to impersonate someone else, in order to gain privileges in the service were the authentication process is ongoing. The high level of popularity of these IdPs brings the advantage of having a larger reach in total number of users, as a direct result of an easier registration and login process, so many web services allow authentication to be done through them. For instance, EPOS researchers are highly likable to have an account on one of these popular platforms, and are more inclined to select these as the session IdP over others that provide a higher level of assurance.

C. Multiple identity problem

With the integration of multiple IdPs, the multiple identity problem comes along. Since users can have an account in more than one of these platforms, if they log in through a different IdP than their previous session, they will end up logged on a different account, since the system will not make a connection between IdPs. Whenever the (same) user logs on through a new IdP, they will look like a different user to the system, and a new account will be created for them, thus ending up with multiple accounts for the same identity. Not only this is inefficient, because the user is registering a new account, repeating a process that was already concluded, it is also creating redundant data for the SP, which translates in noise in statistics and data. Also, by forwarding the user to a brand-new account, they will lose all account related data, like preferences or history, which may lead to the user abandoning the service.

D. Authentication through CLI

Many TCS allow the retrieval of data through the CLI, since most of the TCS applications provide an API for this purpose. This allows users to create procedures that can facilitate the retrieval of specific types of data. One use case of this situation is the retrieval of daily processed data from GNSS stations, through the GNSS Products Portal API. Following the EPOS policy, this retrieval of data will need to be authorized and accounted for. It is therefore necessary to create a procedure where the users can prove that they are who they say they are, confirm that they are authorized to access the requested data, and account information for these requests. The difficulty here is that, unlike with the GUI approach where everything is handled automatically, many of these tools are purely done using CLI, such as a python, NodeJS, or bash shell scripts. In these cases, it is necessary to give the user mechanisms that deal with the complexity of adding and managing tokens to their scripts.

IV. Methodology

A. Proposed methodology

1) Identity assurance: To assure the user's identity two methods are proposed. The first, is to use a more secure or trusted IdP with a strong guarantee of the user's identity. The second method to increase the level of confidence in the user's identity is by implementing a Multi-Factor Authentication (MFA). While it does not guarantee such a high level of assurance as the first method, MFA adds another layer of security to the authentication process.

[3]https://gnssproducts.epos.ubi.pt

2) Attribute aggregation: To solve the problem of multiple attributes from various sources, we propose an attribute aggregation system. When a new IdP is connected, the shared attributes from this IdP are compared to the already existing ones. If the system detects a change in an attribute of the same kind, like the display name, the user is questioned which of the two they wants to keep. This allows to have the user's attributes up to date and according to their preference.

3) Multiple identities: We propose a verification procedure to solve the problem of multiple identities. Whenever a user logs in through a new IdP, a verification is made in the back end to check if there is already an account registered for that user. This is done with resource to a primary unique attribute, which we have chosen to be the email, since it is unique and it is also present in most of the user accounts in other IdPs.

4) CLI authentication: For applications that are still being developed or updated this validation can be implemented directly in the application, using a database to prove the validity of the token. For legacy software and services that were deliberately built without any authentication and authorization and that cannot be modified, adding the authentication procedure to the code is not feasible, so the token validation needs to occur outside the application. We propose a token validation procedure on the side of the EPOS IdM, using a proxy, as seen in the figure 1. This will allow that the CLI application communicates directly with EPOS IdM, and once authenticated, exchanges the received tokens with the TCS, obtaining the requested access. In this situation the use of a web server proxy, as in [14], is recommended. A possible configuration for a *NGINX* proxy would be to define a proxy pass to the EPOS IdM, making the user authenticate there, and using a *Lua* script to verify the HTTP response code. If the response was positive, the user would be authenticated and able to proceed. How the user obtains the token is also an important part of the process. This needs to be decided by each TCS. We suggest it is implemented as a functionality in the GUI for authenticated users, where they can enable and obtain the token for authentication in the CLI.

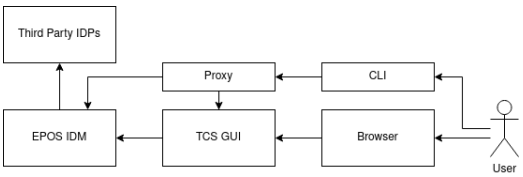

Fig. 1. User interaction with TCS web service by CLI and GUI.

V. Case Study

As the case study, we have chosen to study the GNSS Products Portal, from the EPOS GNSS community.

A. Assuring the user's identity

To assure the user's identity, the two proposed methods were implemented. Firstly, the Portuguese Electronic Identification (EId) authentication system, that allows Portuguese identification card holders to authenticate in authorized web services by the government was implemented. This authentication brings a high level of assurance because the user needs to authenticate with their official Portuguese EId credentials, either by using the short message service or using the EId card in a smart card reader. There are other authentication methods allowed by this EId authentication system that do not guarantee such a high assurance, like using the e-mail, but the assurance level is one of the attributes that is shared with the SP, so this can be taken in consideration when asserting the session privileges. Another shared attribute is the e-mail, that can be used to perform the account association, in case the user already owns an account on the service. The biggest problem with this authentication method is that it is only available for Portuguese EId card holders, and since EPOS is a European project, many of the users will not be able to use this method.

To complement this situation, the Time-based One Time Password (TOTP) MFA was implemented. Here the user has to scan a quick response code in their mobile device, using an appropriate application for that, and save a secret that will generate TOTPs in a defined time interval. Then, when authenticating into the system, the user needs to input the generated TOTP to validate their identity. In addition, the usage of MFA with a Yubikey is allowed, that can generate one-time passwords very intuitively. The MFA can be enabled or disabled on the user's profile, once authenticated. To increase the security of this mechanism, there is a limit to the number of consecutively unsuccessful tries.

B. Attribute aggregation from multiple IdPs

The proposed attribute aggregation mechanism was also implemented in this case study. This can be done as the developers already have the knowledge of what, and in what form, attributes are shared from the IdPs that have been implemented, since it is based on the OAuth 2.0 scope that was previously defined. If this attribute is of a new relevant type, it is asked to the user if they want to add this attribute to their profile. This is an efficient way of having the user attributes up to date and in concordance with their preferences.

C. Multiple identities

A verification procedure to solve the problem of multiple identities is implemented. Whenever a user logs in through a new IdP, a verification is made in the backend to check if there is already an account registered for that user. This is done with resource to a primary unique attribute, which in this case study is the email, due to its uniqueness and presence in most of the available IdPs. It is also allowed to perform a linkage of new IdPs to the account, once the user is already authenticated, where this primary attribute does not have to match between the SP and the IdP. Since the user is already logged in, it is assumed that the authentication process was correctly done previously, and the user is not impersonating anyone. From there the linkage between this account and a third-party account can be done, by performing the OAuth 2.0

protocol, and exchanging tokens and attributes with the IdP. After a successful linkage with the new IdP, it can be used as a login method for their next session. At any time, from the user profile, they can remove this link.

D. Implementation

Figure 2 shows the whole authentication process for the GNSS Products Portal. Multiple ways for the users to authenticate themselves are offered, that verify if they are already registered in the database or not, and proceed accordingly. The system always checks if, once identified, the user has a second factor authentication enabled. If so access to sensitive data is only granted after a successful validation of that second factor.

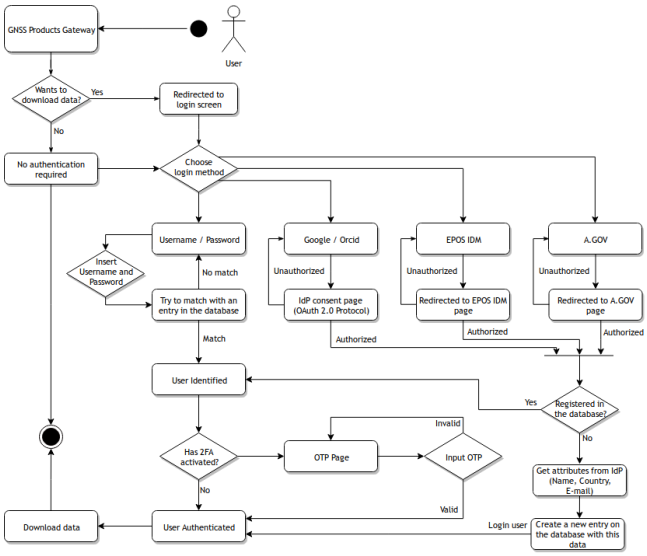

Fig. 2. Activity diagram of the authentication in the GNSS Products Portal.

E. Authentication for the CLI

Since the GNSS Products Portal is an application still in development, the user authentication can be implemented directly in the application source code, using a database to prove the validity of the token. When doing the HTTP request, the user should include in the request header a valid token, previously obtained through the GNSS Products Portal GUI, that will allow the service to identify who is doing this request. This token is obtained in the user's profile page of the GUI, and can be in two forms — an access token, which has an expiration date of 3600 seconds, and a refresh token, that has an unlimited lifetime, and can be used to acquire new access tokens. Due to limitations on the EPOS IdM side, the lifetime of an access token cannot be modified, which makes the access token unproductive in the use case of scripts. For this use cases, the refresh token will be necessary, alongside with the user's identifier as confirmation. The usage of the OAuth 2.0 refresh token functionality will have to be part of this proxy procedure, retrieving a new valid access token from EPOS IdM. If the refresh token is invalid, or the identifier does not match the given refresh token, the refresh token mechanism will fail, and so will the authentication through the proxy.

Figure 3 shows part of the user profile page in the GNSS Products Portal. On the left, the user can see and link, directly from there, other IdPs to the current account. On the right, the user can enable the token mechanism and obtain a valid token to use in CLI applications.

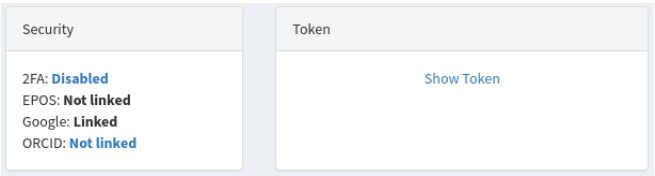

Fig. 3. User profile in GNSS Products Portal showing linked IdPs and token retrieval.

For the GNSS Products Portal the validation procedure has been added in the API. From the request header the token can be extracted and then validated with a database in the backend. This returns either the requested data or the unauthorized response code - 401. In order to correctly authenticate users, it will be necessary that they provide an identifier and the correspondent token.

Figure 4 illustrates the entire GNSS TCS architecture. The main components are the web services and other services, the authentication and authorization module, the user access interface, via CLI and web portals, a proxy web server for dealing with the token validation, and an accounting module for gathering statistical data.

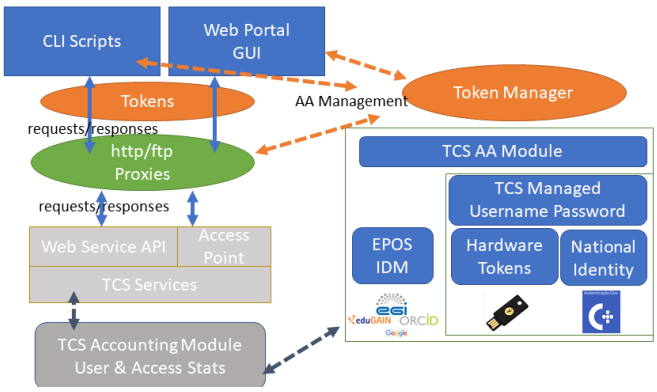

Fig. 4. The GNSS TCS Architecture.

VI. Discussion

With the implementation of the procedures previously described, the user authentication and authorization problems, that we have identified in the TCS's services, have been addressed. In specific, the case study's implementation solved most of the identified problems. However, there are still remaining challenges that can improve the user's overall usability. In particular, a guide on how to create, extract, and copy tokens to CLI applications, and to help users understanding

the token mechanism and why it is necessary. Also, in our opinion, creating an intermediary IdM on the side of the SP to enable more control over who has access to the data would be beneficial. Having the options to generate, grant, revoke, and refresh the tokens directly from the SP's side will allow the system to improve and extend the features that it provides. For instance, it is possible to create types of tokens that only grant access to certain sections of data, like limiting the access to only the selected data providers, or revoke tokens after a certain time to enforce users to re-authenticate and obtain valid tokens.

VII. Conclusions and future work

To conclude, we have discussed several problems in identity and access control related to service orientated architectures, that are built on web services and that make use of multiple third party IdPs. In particular, we have described some general solutions and describe how these were applied within the context of the GNSS community services within EPOS, a European Research Infrastructure.

As future work, a good documentation, along with visual examples like a video tutorial, can make it clearer how the authentication process should work, and what actions and procedures are necessary on the part of the user. Likewise, good documentation on how to implement these procedures in other software, and adding the tokens to common libraries like *wget*, *curl*, or *python*, can also increase TCS's developers chance of achieving good results. Deeper interaction by the SPs with the IdPs also seems desirable and needs more investigation is necessary in order to develop mechanisms that can provide additional functionalities, such as increased control over token lifetimes, without compromising security.

s

VIII. ACKNOWLEDGMENTS

This work is funded by FCT/MCTES through national funds and when applicable co-funded EU funds under the project UIDB/EEA/50008/2020 and also by the EPOS-IP European Union Horizon 2020 research and innovation program under grant agreement Nº 676564.

References

[1] Temoshok, D., Temoshok, D., and Abruzzi, C. "Developing Trust Frameworks to Support Identity Federations," US Department of Commerce, National Institute of Standards and Technology.

[2] Khattak, Z., Sulaiman, S., and Manan, J., "A study on threat model for federated identities in federated identity management system," 2010 International Symposium on Information Technology, Kuala Lumpur, 2010, pp. 618-623.

[3] Maler, E., and Reed, D., "The Venn of Identity: Options and Issues in Federated Identity Management," Security & Privacy, IEEE. 6. 16 - 23. 10.1109/MSP.2008.50.

[4] Qiang, W., and Konstantinov, A., "The Design and Implementation of Standards-Based Grid Single Sign-On Using Federated Identity," 2010 IEEE 12th International Conference on High Performance Computing and Communications (HPCC), Melbourne, VIC, 2010, pp. 458-464.

[5] Fett, D., Küsters, R., and Schmitz, G. (2016, October), "A comprehensive formal security analysis of OAuth 2.0," In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 1204-1215).

[6] Klingenstein, N., "Attribute Aggregation and Federated Identity," 26. 10.1109/SAINT-W.2007.29.

[7] Chadwick, D., and Inman, G., "Attribute Aggregation in Federated Identity Management," Computer. 42(5). 33 - 40. 10.1109/MC.2009.143.

[8] Chadwick, D., and Inman, G., "The Trusted Attribute Aggregation Service (TAAS) - Providing an Attribute Aggregation Layer for Federated Identity Management," 2013 International Conference on Availability, Reliability and Security, Regensburg, 2013, pp. 285-290.

[9] Ferdous, M., and Poet, R., "Analysing attribute aggregation models in federated identity management," In Proceedings of the 6th International Conference on Security of Information and Networks (SIN '13). Association for Computing Machinery, New York, NY, USA, 181–188.

[10] Ferdous, M., Norman, G., Jøsang, A., and Poet, R., "Mathematical modelling of trust issues in federated identity management," In IFIP International Conference on Trust Management (pp. 13-29). Springer, Cham.

[11] Iacono, L., and Nguyen, H., "Authentication Scheme for REST." International Conference on Future Network Systems and Security, 2015, pp. 113–128.

[12] Bradley, J., Tschofenig, H., and Richer, J., "A Method for Signing HTTP Requests for OAuth."

[13] Kanmani, K., and Smitha, P., "Survey on Restful Web Services Using Open Authorization (Oauth)," IOSR J. Comput. Eng, 15(4), 53-56.

[14] Szepieniec, T., "Guide to EPOS AAAI for Service Provider,"

Glossary

AJAX	A technique for creating better, faster, and more interactive web applications combining Hypertext Markup Language, JavaScript and XML [86];
AuthN	In an authentication system it refers to authentication, dealing with user identity [87];
Browser	The default application launched by the operating system to handle HTTP and HTTPS scheme URI content;
EGI Check-in	A proxy service that operates as a central hub to connect federated IdPs with EGI service providers;
Epoch	An epoch, for the purposes of chronology and periodization, is an instant in time chosen as the origin of a particular calendar era. The epoch serves as a reference point from which time is measured [88];
GDPR	An European regulation, approved by European Parliament and the council, that intends to protect the person regarding his process of personal data, and the free movement of such data [89];
GNSS	Is a general term describing any satellite constellation that provides positioning, navigation, and timing services on a global or regional basis [90]. These include GPS (United States), BeiDou (China), Galileo (Europe), GLONASS (Russia), IRNSS (India), and QZSS (Japan);
GLASS	An integrated software package, developed for the EPOS project, that is meant to be deployed in a GNSS infrastructure to manage data and metadata;
i-name	An informal term used to refer to a re-assignable XRI; more specifically, an XRI in which at least one sub-segment is re-assignable [91];
OASIS	The Organization for the Advancement of Structured Information Standards is a global nonprofit consortium that works on the development, convergence, and adoption of open standards for security, Internet of Things, energy, content technologies, emergency management, and other areas;
Proxy	An agent that relays a message between a requester agent and a provider agent, appearing to the Web service to be the requester [92];
REST	Representational state transfer is a software architectural style that defines a set of constraints to be used for creating web services;
RINEX data files	RINEX is a data interchange format for raw satellite navigation system data [93], from where timeseries, and others, can be extracted;
Unity IDM	A complete solution for identity, federation and inter-federation management. An extremely flexible authentication service [11].