University of Camerino

School of Science and Technology

Master of Science in Computer Science (L-18)

Moodle integration for Single Sign On authentication: Case Study elearning.unicam.it in IDEM

Candidate
Giacomo Nalli

Supervisor
Prof. Fausto Marcantoni

Dott. Marco Maccari

Academic year 2016/2017

Index

Introduction

The steady increase of the network has allowed the development of online services, which have become increasingly become a relevant aspect of our daily life. The networking services can be divided into two different categories: those with free access and those with controlled access.
Every day students, researchers and teachers take advantage of controlled-access contents and services in different ways.
In any context where it is necessary to manage numerous IT resources, such as access credentials, portals, domains, services, users, it is necessary to effectively coordinate the different identities that use them.
In fact, in these systems, it is possible to notice the problems related to the password proliferation, the use of services from an inter-organizational perspective and the creation of an infrastructure, which is able to overcome these difficulties.
Furthermore, it does not still exist a precise mechanism capable of identifying both the subject and the service through which people communicate: an infrastructure aimed at the management of the identities has always been lacking. Consequently, many companies and organizations have tried over time to develop isolated, partials and often-incompatible solutions to overcome the problems connected with the identification.
For this reason, there is an urgent need to introduce a federative element, which will coordinate these identities with its related services. As a result, the service will no longer be limited to a single organization; it will rather cover all organizations.
In the current work, firstly, an analysis of the used technologies and protocols will be carried out, and then a federated infrastructure/network will be presented.
It is obvious that universities are part of such organizations and for this reasons they require Identity and access management.
An identity and access management system includes all components, procedures and technologies that ensure the correct management of both identity and access.
Through a centralized IAM system, in which each individual has only one digital identity, all IT resources relate to the digital identity to manage network access.
In this way, it is possible to gain several advantages: each user will always have the same access credentials; consequently, it is not necessary to remember the passwords used for each resource people want to use.

There is also an improvement from an administrative perspective as users are managed through a single infrastructure. The IT resources refer to such infrastructure in order to get access, reducing the whole management process of users.

In this moment, the digital identities at the University of Camerino are kept separate according to their role within the university. In this way, if a person fulfils more roles (for instance being a student or a PhD student/ researcher), he/she can have more digital identities (a digital identity as a student and another one as a PhD student). Moreover, it exists an access management, which enables people to get different access keys according to the resource people want to use.

The University of Camerino has recently tried to standardize access to different Unicam resources, by using the Ldap internal authentication procedure connected with Microsoft Active Directory, or by implementing other services like Radius. It has provided authentication services limited to just students, researchers and Unicam internal personnel.

However, there are platforms that have not yet implemented a Single Sign-On (SSO) authentication system; they rather create new key access depending on the resource users want to use.

Therefore, this situation is quite different from that created through a centralized IAM.

Another resource, which still has not a Single Sign-On authentication service, is the Unicam e-learning platform.

The aim of this study is to analyze and to create an identity and access management system for the e-leaning platform (based on Moodle) of the University of Camerino enabling people to have a single digital identity and to employ the same credentials used to access other Unicam resources, which require an SSO authentication. In order to create such system, different solutions to access the Moodle platform were considered. In particular, internal and federated solutions were taken into account: using the SSO authentication system, analyzing different methods, depicting the pros and cons of the different configurations according to needs, the differences between an internal and a federated configuring authentication and finally the development of both a federated identity and an access management system.

In the following chapters, the IAM and the federated IAM system will be defined by analyzing the concept of the digital identity, of the identity lifecycle and of access to an IT resource. Therefore, a description of the tools that can be used to effectively implement an IAM server (for access and identity management) will be given. The focus will be on access management in general, in particular on the tools that can be used to manage accesses and not on how users' identities are created and kept within the IAM server.

After this introduction with some general concepts, the focus turns to the University of Camerino, in particular by describing the current situation of access to the Moodle platform, highlighting its limitations and problems. After this analysis, a study will be carried out in order to solve such problems and to introduce some advantages connected to access management of the e-learning platform.

Chapter 1

System for the Digital Identity and Access Management

Identity and Access Management

One of the main problems connected with the supply of specific IT resources, such as web resources, access to the Internet, e-mail, where authentication and authorization are required, is the recognition of the user and the attribution of the related permissions. The problems are managed by an identity and an access management system, which is defined as a framework of policies and tools aimed at managing information (i.e. users' identities) and at controlling access to resources.

1.1 History

The acronym for Identity and Access Management is characterized by precise terms, whose meaning has changed over time. The first two terms, Identity and Access, refer to the personally identifiable information. These concepts have been developed for the first time by the Organisation for Economic Co-operation and Development[1] (OECD), an organization that aims at both financial integration and cooperation and by the National Institute of Standards and Technology[2] (NIST), which is an American governmental agency that manages technology, and creates guidelines for the protection of personal information.
The term Management is instead used in different contexts with different meanings. In this case, it refers to the management of objects, attributes and elements that represent an identity. In particular, these objects are called OID[3] object identifiers that aim at mapping attributes related to an identity.

1.2 The definition of an IAM system

The Identity and Access Management (IAM) is a framework that favors the management of electronic or digital identities. The framework includes organizational policies for digital identity management and the technologies needed to support identity management.[1]

[1] Retrieved from http://www.oecd.org
[2] Retrieved from http://www.nist.gov
[3] Object Identifier

The growing need of companies or institutions to assign an authorization to a user is a common problem since authentication and authorization are required when it comes to specific information resources (web resources, Internet access, computer workstations, e-mails, etc.). An Identity and Access Management system can be defined as the whole process (implementation of appropriate policies and use of technological tools) aimed at managing information about user's identities and at controlling access to IT resources.

Authentication as well as authorization to network resources can be developed in two ways:

1. Each resource independently manages the authentication and authorization of the offered services;
2. Access management is centralized.

Hybrid cases may find place as some resources may have centralized access management and others may manage accesses on their own. The presented case can be clearly traced back to the first point since there is not a real centralization of accesses, there is rather a series of subsets that manage authentication and authorization in an autonomous way.

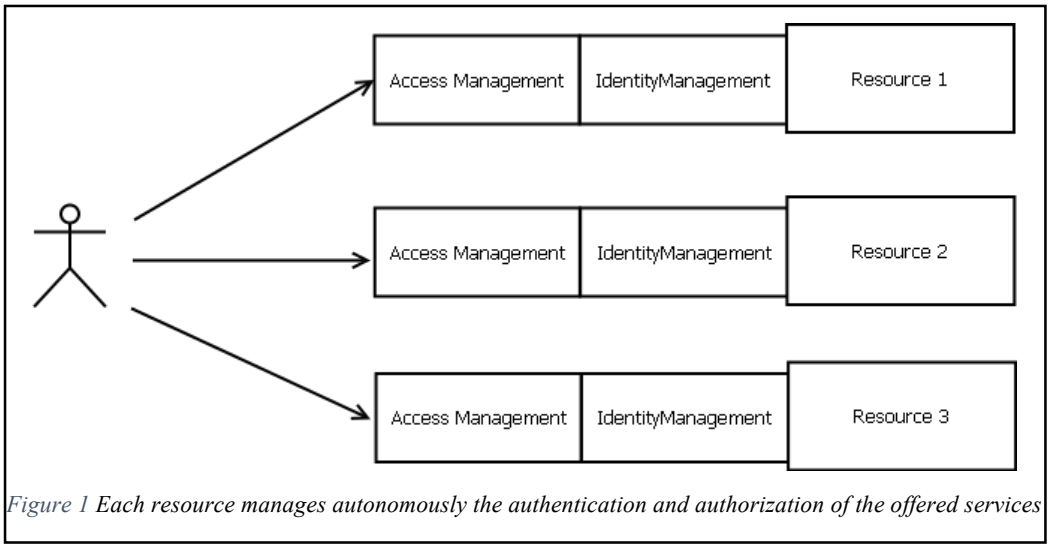

Figure 1 Each resource manages autonomously the authentication and authorization of the offered services

This is the most used model of Identity Management. Each service, as shown in Figure 1, has its own independent users and each user receives a separate credential for each service he/she accesses. This approach simplifies for the Service Provider the Identity Management, but this approach presents huge problems connected with the usability since users are not allowed to increase the used services.

Here the relationships of trust involved in this model:

- the Service Provider guarantees user privacy;
- the Service Provider implements the registration procedure and the appropriate authentication mechanisms;

- the customer responsibly manages the credentials received by the Service Provider.

The first method presents different problems when considering the resource manager's point of view:

- the information regarding the identity information is repeated for each resource;
- the increase of a new resource involves the increase of a whole access infrastructure/ framework/platform?!;
- the repetition of information on many systems means that security has to be managed in many points, for this reason there is less control over security.

From the user's point of view:

- each user has different login credentials according to the resource he/ she wants to access;
- Since the user has to remember many passwords, it can easily happen that the user chooses simple passwords or writes them down on slips of paper, which implies less security for his/her data.

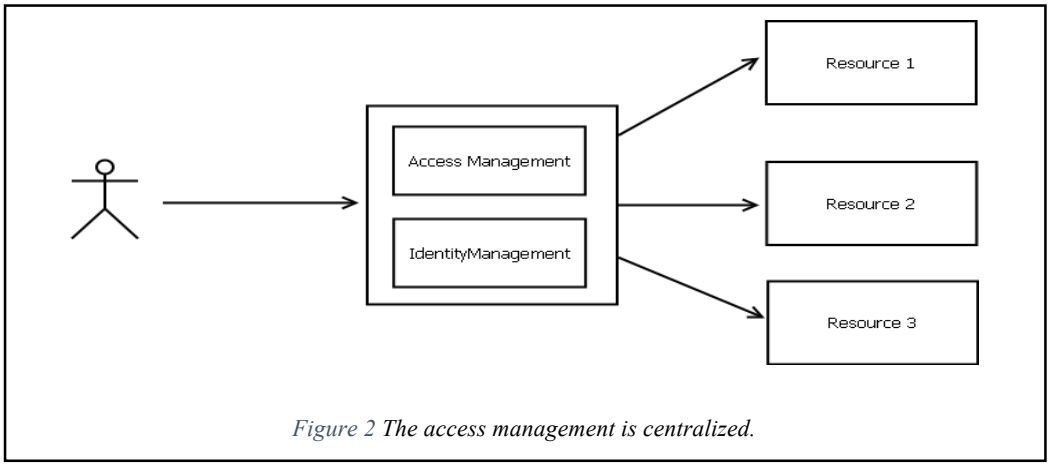

Figure 2 The access management is centralized.

The second method, based on a centralized access, appears to be the solution to all problems connected to the first method. In fact, the users' administration is carried out only in the centralized infrastructure, solving all problems associated with the service provider. Furthermore, users are kept in a single infrastructure, thus they have a single key access avoiding in this way the proliferation of passwords. However, considerable problems have to be faced, which are common of centralized systems: performance and reliability. In fact, it is essential to assess the number of the requests the system will receive, once the whole management of access will be transferred to a single infrastructure, in order to reduce them if needed and to be able to process such requests within a reasonable space of time. Another relevant aspect is reliability, this means ensuring that the infrastructure is always present even in case of accidental failures or a computer system compromise incident.

The centralized system, thanks to its numerous advantages, is substituting the traditional system based on the autonomous management of both identities and accesses. A problem can be identified in this transition phase the consolidation of the digital identities. It often happens that information on users come from different sources, for this reason, it is essential to provide a series of mechanisms that collects all information in a single structure.

Here the following steps to be considered in order to consolidate the identities:

- Analysis of existing databases in order to select the authoritative ones;
- which kind of information should be kept, maintained and possibly added;
- consolidation (a person can be present in more sources) in order to create a digital identity;
- keep the single database updated.

There are different advantages that can be obtained through the system after the process of the identity's consolidation:

- greater flexibility in adding a new resource or in changing access rights to a specific group of resources;
- single point for data modification;
- consolidated logging allows the application of rules concerning privacy, the conservation of auditing data, the creation of reports, the security control in an effective way;
- solution to the deactivation problem when it comes to the management of identities in disjointed systems;
- decrease in the amount of passwords/credentials to be known by the user;
- a more effective organization as access rights can be quickly modified (based on the user roles);
- by ensuring that the user is "the person he/she says to be", the level of confidentiality is increased by the institution.

The final goal of the IAM system is to increase the productivity, to simplify the usage of the computer system for the users, to increase the security level, to reduce the costs connected with the management of users, their identities, their attributes and credentials. In few words, the aim is to provide a tool that supports the process that rules access of users to the resources, the authorization allocation, as well as the management of the process and the monitoring of the activities, following the security policies.

1.3 Control of access

Another key aspect of the network infrastructure is the full control of access. In this way, it is possible to control who accesses the network and what the user can do, which services can be actually used once he/she is logged in. The AAA term intends for Authentication, Authorization and Accounting, which is a framework through which access on the network can be configured.[2]

Once obtained access to the resource, it is reported the authentication and authorization operations performed by the IT system, which allows users to use a resource, made available by the system.

It is expected that users, in order to access a resource, identify themselves with the resource operator, who registers the users and provides them with the credentials. Once users obtain the credentials, they can access the resource.

1.3.1 Authentication

Through the authentication, the user identity is verified. Infect, access to shared and protected resources can be either denied or allowed. In fact, through the authentication system, a user is associated with his/her digital identity to be found on the system. This process checks the real identity of the user; once a person's identity is confirmed, the system must ensure that the user is the person he/she says to be. For this reason, the system requires an evidence of the user's identity. It is necessary to keep this information hidden and stored accurately. Authentication techniques can range from simple login with username and password to more complex and strong mechanisms such as tokens, digital certificates or biometric public key systems. An IAM solution must therefore be independent from the authentication mechanism in order to be adaptable to any specific technological reality.

1.3.2 Authorization

The next step to the user authentication is authorization. This process allows users to access resources if they have the rights to do it. During the authorization phase, the user's digital identity is thus evaluated by applying the appropriate rules (allowed, restricted or prevented access to the resource).

This process verifies which kind of resources and data the user can access. This certainly takes place through the guarantee on access control, based on permissions previously given to the user. Through this system, it is therefore possible to establish the operations that can be allowed

or blocked to the user.

1.3.3 Accounting

The accounting module is the third and the last module of the AAA framework, whose task it not to authorize or to deny access to a user or a service. Its task is to keep track of their activities. Its activity is similar to that of the accountant who keeps track of the exchanged data, of the nature of the service provided, of the time a user remains connected to the service, and the place from which he/she is connected. This data can be used to present to the users the invoice, which will be in line with the amount of service he/she has benefited from. The Accounting operation, which is used in real time, provides to the service managers some relevant information such as the number of connected users and the analysis of anomalies in the provision of services. There is also the possibility, for the accounting module, in particular in some implementations, to store the information collected on a specific database.

Chapter 2

Federated Systems

2.1 The Federation

The term federation refers to a set of organizations, bodies or service providers that decide to create relationships of trust between them in order to exchange information on people's identities. Users who belong to a particular organization often asks to access services from other organizations, which are part of a common federation. In order to allow this process, organizations must select and share within the federation these mechanisms for the information exchange on users and for the management of access to resources that are wanted to be shared within a federation.

In order to do this, it is required the usage of standard technologies and agreements that allows the Service Providers (SP) to consider and accept the user identifiers managed by another set of Providers, called Identity Providers (IDP), as valid.

This community of providers (SP and IDP) is identified with the term federation. The main task of the federation is to manage the relationships between the IDPs and the SPs, which are "federated" with each other; this approach implicitly implements Single Sign On (SSO), this means the possibility for a user to authenticate with any of the federation providers and, thereafter, to access the services of all other providers.

A Federation is therefore an agreement between organizations, which is based on a mutual trust in the exchanged information (through the sharing of resources, services or applications), according to the predefined rules and through a certified and a secure (AAI[4]) infrastructure. [3]

The AAI is a federated infrastructure that allows a user, who belongs to an organization (part of a federation), to log in and to access the services offered by other organizations within the same federation, by always using the same login credentials.

The introduction of common policies is required in order to manage the relationships of trust between the various participants. A member of a federation represents any organization, (for instance university or research institution) only if it has signed the contract with the Federation. Members agree on the legal aspect, policies and technologies to be adopted.

Moreover, all federations provide a particular service called WAYF, which allows users to choose their own Membership Organization. WAYF (Where Are You From) is the central and

[4] Authentication and Authorization Infrastructure

the single service for the whole federation, whose role is to redirect the user to the membership IDP, once requested by the Federation's Service Providers. The IDP contains the digital identity of the user enabling the user to be properly authenticated and to let him/her taking advantage of the chosen service.

With the term federated IAM, it is meant the management of identities and of their access within federation, which is no longer limited to a single organization but rather to a group of organizations.

The Figure 3 presents a federation without an AAI: a user has a different access key for each service that requires authentication, and each single organization manages the authentication to its services even for external users. For this reason, each organization is forced to manage the users' identity of other organizations, apart from its own users' identity.

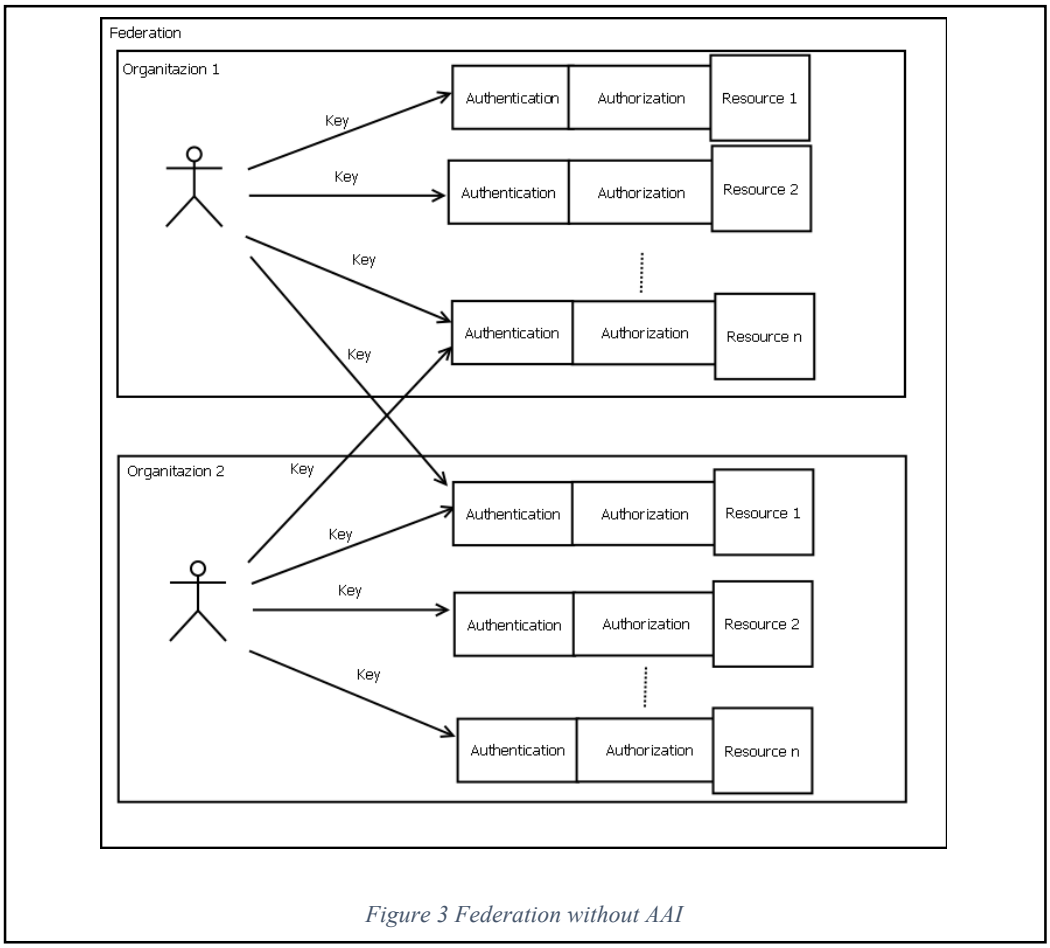

Figure 3 Federation without AAI

As shown in Figure 4, with a federated AAI, the authentication process is managed by the AAI, in this way each user has a unique access key which is valid for each service. If a user needs access to a service of an external organization, the authentication process is managed by the AAI, which collects the necessary information from the user's organization to which he/she belongs.

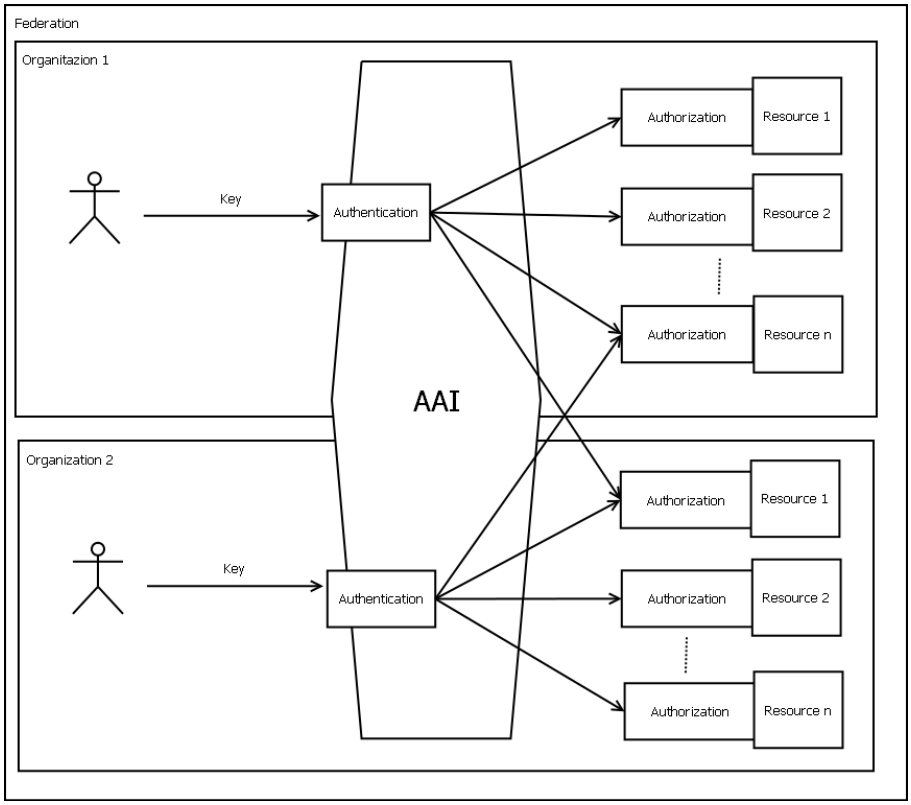

Figure 4 Federated AAI

Through a federated AAI, it is possible to gain some relevant advantages:

- through the IDP, users authenticate themselves only once and have access to different resources made available by the federation;
- a careful control is carried out regarding the release of information concerning the user;
- access is based on standards and open source products;
- new resources and new users can be quickly added;
- the attributes regulate the access right and not the credentials or the IP address;
- the attributes rule the access rights and not the credentials or the IP addresses;
- the resource federation improves users' management and reduces redundancy of credentials; this increases efficiency and quality.

2.1.1 Tasks of the federation

The main task of the federation is to keep the available resources updated for the users. In fact, the role of the federation is mainly organizational as it is coordinator of the organizations. In specific terms, it is responsible of all documentation that each member must have:

- definition of attributes;

- the membership form;
- the cancellation form;
- legal documents regarding the processing of personal data;
- the updated list of members with their related resources;
- the technical documents with minimum requirements to enter the federation.

It is relevant to create a management and a technical staff which coordinates this system in order to establish an efficient federation.

2.2 IDEM Federation

In the last years, national federations have been founded in over fifteen European countries, including Great Britain, Spain and Switzerland, as well as in the United States, Canada, New Zealand and Australia. When considering the Italian University, GARR[5] promotes the IDEM[6] project with the aim of creating an authentication infrastructure and a federated authorization for access to services. Many universities and research centers participate in this project, which gives the possibility to researchers, teachers and students to take advantage of the same standard system of access management.

The project, which has been started in January 2008, has different aims:

1. The non-theoretical operational demonstration of the technical and organizational feasibility and of the usefulness and usability of the GARR Federation;
2. The strengthening of the authentication and authorization systems of GARR bodies by sharing experience, adopted solutions and development plans. This is a necessary condition to increase security without affecting the flexibility in managing access to online services and the experience development in the context of federated authentication;
3. The increase of the user's awareness in using online services;
4. The encouragement to service providers in order to promote recognition criteria of users based on identification systems - logical and non-physical users. In this way, the management of access systems to some services (for example online catalogs) will be more efficient;
5. The promotion of identities' mutual recognition within the research network in the Italian context;
6. The creation of encouragement in order to promote the creation of services available to the

[5] Retrieved from https://www.garr.it/it/

[6] Retrieved from https://www.idem.garr.it/

GARR community, including single bodies or research institutions;

7. The arrangement of agreements and possible future enlargement of the initiative in national and European contexts;
8. The development of federated authentication;
9. The improvement in terms of efficiency in the use of services in areas with high or growing mobility of users.

A technical committee in charge of the management of projects, including representatives of some universities and research institutes, participates to this initiative.[4] Within the IDEM Federation, GARR acts as a coordinator, provides the central infrastructure, services and signs the membership agreements. It also provides the Federation with technical support for the implementation of an Identity and Service Provider, the related open source implementations, the WAYF service (used to identify the IDP of origin) the management of the federation map defined as metadata.

The organizations belonging to the GARR community can participate in the federation, in particular universities, research centers, and other organizations interested in sharing their services. To participate in the federation, it is necessary to register at least one Identity Provider and a management and identity's verification service or a Service Provider, which is a resource accessible on the network. In order to be able to adopt the federated system of access, institutions need to undertake different activities (in compliance with the specifications proposed by the Federation) as:

1. to define the needs to be met by managing access and to assess one's ability to manage identities through an institutional audit;
2. to develop directory systems (LDAP) to perform the identity management within the institution;
3. to choose an appropriate authentication system;
4. to implement the identity management system, the Identity Provider (IDP);
5. to join a federation (for example IDEM federation of GARR);
6. to provide training courses for staff, manual for users and related support.
7. to preserve the logs to associate a user with an authentication session;
8. the willingness to provide information regarding the accreditation system and the adopted users' managements.

For GARR organizations:

1. Implementation of a Shibboleth[7] Identity Provider;
2. Preservation of logs to associate a user with an authentication session;
3. Willingness to provide information on the accreditation system and user management adopted.

2.3 SPID

SPID[8], the Public Digital Identity System is an Italian Authentication and Authorization system which allows public administrations and private individuals to access their services on the web. Today SPID is primarily designed for the PA's specific purposes. It was firstly conceived for national borders, but then for European ambitions. In fact, the Regulation of the Digital Identity System issued by the Digital Italy Agency (AgID[9]) establishes a mutual recognition of electronic identities and before the implementation, the system had to overcome the European examination in order to be sure that there were not any clear obstacles in this sense. The European directive eIDAS[10] issued by the European Union and in force since 17 September 2014, defines the conditions for the mutual recognition in the field of electronic identification and for the common rules for electronic signatures, web authentication and related trust services for the electronic transactions. The Italian Identity Providers in SPID must be compatible with this regulation and must be notified in Europe.

3.4.1 How it works:

Citizens and businesses can access services with a unique digital identity - the SPID identity - that allows access from any device. The SPID identity can be obtained through a request to an accredited identity provider (Digital Identity Manager). Each user can freely choose the preferred identity manager from those accredited (and thus authorized) by the Digital Italy Agency (AgID). The Authentication through SPID is divided into three levels of credential security, depending on the service type.

[7] Retrieved from https://www.shibboleth.net/

[8] Public Digital Identity System, retrieved from: https://www.spid.gov.it/

[9] Digital Italy Agency, retrieved from http://www.agid.gov.it/

[10] Retrieved from http://www.agid.gov.it/agenda-digitale/infrastrutture-architetture/il-regolamento-ue-ndeg-9102014-eidas

There are different roles in the SPID system:

- **Identity provider**: provides the access credentials to the system (digital identities) and manages the processes connected to the user authentication.
- **Service provider**: provides digital services accessible through login with SPID credentials.
- **Attribute provider** (manager of qualified attributes): provides attributes that qualify users (states, roles, titles, charges), aimed at the usage of services.

AgID, in agreement with the privacy agent, has defined the technical rules for the adoption of the SPID system. Furthermore, it manages the accreditation procedures of the digital identity managers, and performs controls on the activities of the identity providers.

2.3.2 SPID and IDEM

IDEM and SPID represent two different ways to solve the problem of digital identity but they are not the ultimate solution. The SPID system has been introduced to be a strong digital identity aimed primarily at authenticating the identity of citizens in a secure way, so that they can use digital identity in every kind of transaction and public act. IDEM is not designed for commercial applications, conversely the rules of the IDEM federation are designed for academic application.

For this community, specific attributes are needed, for example non-personal attributes but characterizing the training and the role of the user within a University. [5]

Differences

Accreditation authority	**SPID**	**IDEM**
Idp,Sp	Open group for both private and public entities	Restricted group of public and private entities belonging to the Research and Education (R & E) sector
What Idp do	Management of the registration services and making credentials and tools available in order to access to the network. In particular, for both citizens and businesses on behalf of public administrations.	Management of the registration service and provision of credentials and tools in order to access the network by considering the R & E community (students, former students, teachers, researchers, employees or affiliates).
Active users	3 million (2016)	4 million (2016)
Expected users	60 million (the whole Italian nation)	8-10 million (all students and all graduated students)

Identificaion	• Visual identification with presentation of identity card • Visual identification remote access with presentation of identity card • IT identification through digital personal identity • IT identification through SPID identity • IT identification through qualified electronic signature or digital signature	• Visual identification with presentation of identity card

Chart 1 Differences between IDEM and SPID

Attributes

Id	**SPID**	**IDEM**
Identification number	spidCode (=<cod_IdP><nr. univoco>)	ePPN (=<stringa_univoca>@<domain>) ePTID o SAML2name_id
Name	Name	givenName
Family Name	familyName	sn
Fiscal Number	fiscalNumber (=TINIT-<Fiscal Number>)	schacPersonalUniqueID (=urn:schac:personalUniqueID:IT:CF:[CF<CodFis>]

Chart 2 Attributes IDEM and SPID

IDEM and SPID show relevant differences and for this reason, there is a need of some intermediation systems. These are needed, in particular, if the University and Research community aim to take advantage of these services at both national and international level. Although SPID identities are based on the same technologies , they are not automatically interoperable with any other system that meets European standards. Some problems arise for the existing service providers since they should be configured in order to be compatible. This is caused by the fact that SPID has established technical rules for SPs. Consequently, the

reconfiguration will be an extra expense and will lead to a decrease in the number of foreign SPs that will ask for a direct integration with SPID. For this reason, there is an urgent need to find an intermediation between the two different worlds.

Common points:

The desire for simplification (very user has only one identity) is a common point between IDEM and SPID. Thus, it makes sense to think about a researcher that uses his/her SPID identity to access services related to his/ her research work.
IDEM and SPID have the following characteristics:
1. Same standard SAML 2; this means that they both use Shibboleth
2. Differences just between attributes that can be compatible and integrated.
In order to integrate the two services, IDEM could integrate its role as a qualified "Attribute Authority", an authority able of managing roles and other additional characteristics of a specific identity.

There are two advantages of such approach: on the one hand, the digital identity of the user is preserved since, once his/ her identity has been verified, his/her data will not be on the network; on the other hand, operations are speed up in order to verify the user's role and access rights.
In the future, the SPID identity will become the single digital identity of the citizen, also within the organization to which the user belongs, in order to avoid managing multiple identities.
However, an R & ID service like that of IDEM, which manages roles and specific information connected with research and training, cannot be replaced by another service. The IDEM service is, in fact, already fully integrated into Italian and European regulations.
If these two services are implemented (Authentication through SPID and Enrollment through IDEM), it will be given the opportunity to users to access services with greater simplicity and better management of both hardware and human resources for service providers.

Chapter 3

Digital Identity

3.1 Definition of digital identity

The digital identity is a set of information stored in an IT system that represents an entity that accesses the resources granted by the IT system. [6]
A digital identity consists of the following characteristics:

- **Personal data**: information that describes the associated entity and identifies it. This information can be modified, but regardless of changes that occur within the entity. Personal data may be surname, first name and date of birth.
- **Attributes**: this is a set of information on the entity related to the organization, which can be divided into:
    - Institutional roles
    - Tasks
    - Attributes for access (privileges)
    - Group membership
- **Login credentials**: for instance username and password, or a digital certificate. These are used to access to computer resources, they can be modified by the user (for example, changing the password) or in particular situations (for example the loss of such log on credentials).

The main topics regarding the Identity Management are:

- Authentication: not only through the use of username and password, but also through the use of certificates;
- Confidentiality: the ability of the system to prevent third parties from intercepting and exploiting data which are received or transmitted;
- Authorization: to limit the access to private information or to allow access to specific services;
- Integrity of data: in order to be sure that nobody intercepts the exchanged data, it could be relevant to know whether someone has modified them during the transmission phase;
- Proof of the source: it is possible to carry out specific transactions in order to demonstrate that the data have been actually sent. This is possible when the data is sent with a digital signature.

- Non repudiation: to provide an evidence of a successful shipment or a receipt of incoming data on the network;
- User provisioning: allocation and revocation of privileges and user's profiles;
- Format and interoperability: digital identities must be able to communicate through a single communication protocol;
- Single Sign On: a specialized system that allows the user to authenticate just once and to access all the IT resources to which he/ she is enabled;
- Directory services: folders reserved for different type kind of users.

From the definition of the word identity, it can be stated that it is not excluded that the user can have more than one digital identity within the same information system, since the term "entity" is quite generic and a user can represent different entities. From the user's point of view, the non-uniqueness includes multiple accounts, consequently more credentials. However, this means that the system, would badly manage information regarding users in terms of data consistency, for example, changes made to the identity should be propagated to all other identities of that user.

It is therefore essential to find a group of attributes that can uniquely identify each user, in order to always refer to that user through the same attributes, thus creating a unique digital identity for each single user.

3.2 General structure

From the Figure 5 the typical structure of a federated Identity and Access Management infrastructure is presented. In this case, users who belong to a specific organization are authenticated through the identity provider (IdP) and can access resources that are made available thanks to SP providers.

Within this basic structure, organizations are part of the federation, whose task is the coordination of different resources within organizations.

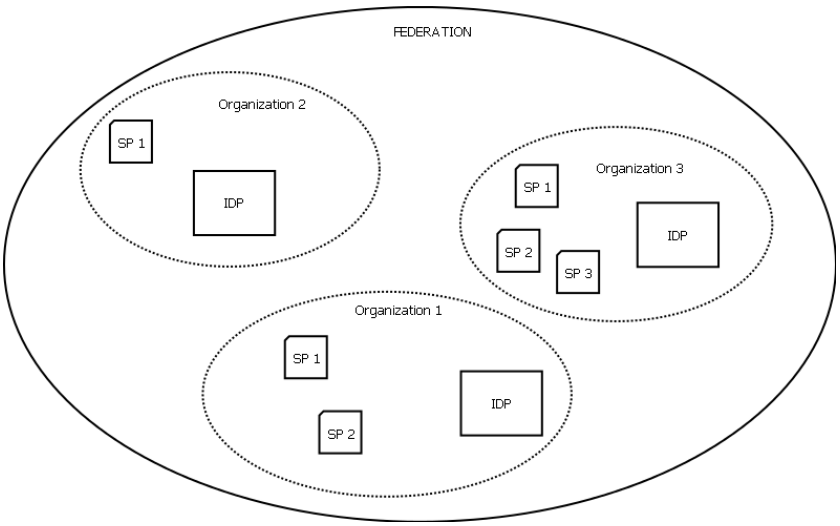

Figure 5 IAM general structure

3.2.1 Digital Identity within the federation

In this scenario, the federation is not the most significant element; it is rather the access and management system of the identities, which provides the SSO service.

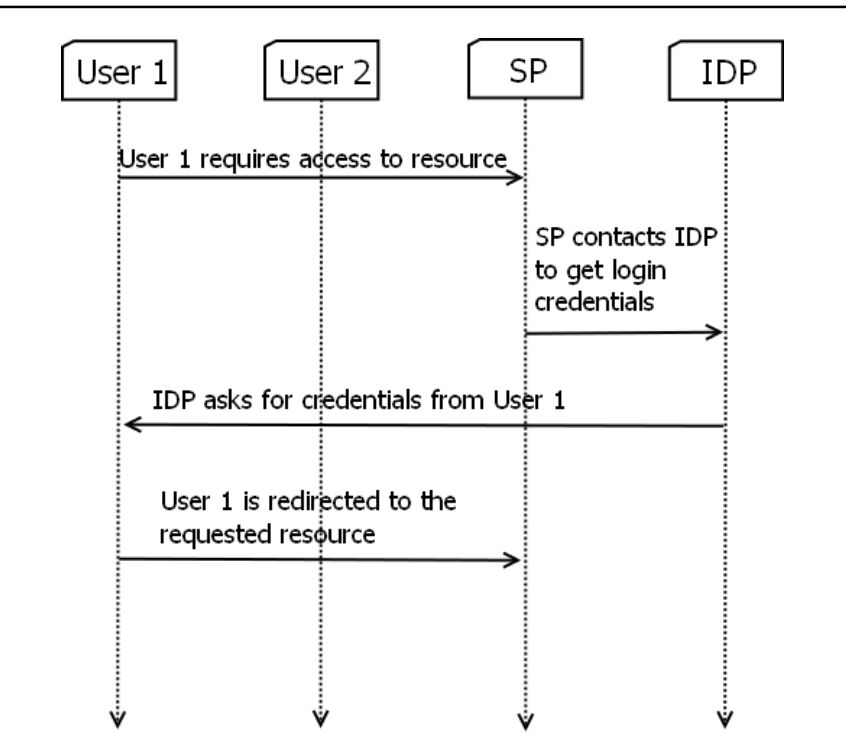

Figure 6 SSO scenario with SP and IDP

In this context, the flow of data is similar to that shown in Figure 6 described in the following points:

1. The user requests access to a resource within the SP;

2. The SP contacts the IdP to obtain access credentials;

3. The IdP requires the credentials to the user;

4. Once the authentication is completed, the user is redirected to the resource he/she has previously requested.

There are further cases in which the user accesses a resource of a different organization. In this scenario, the concept of federation is of great importance since, through this binding element, the user is able to access a resource of a different organization. The flow of data has a different trend compared to the case described above.

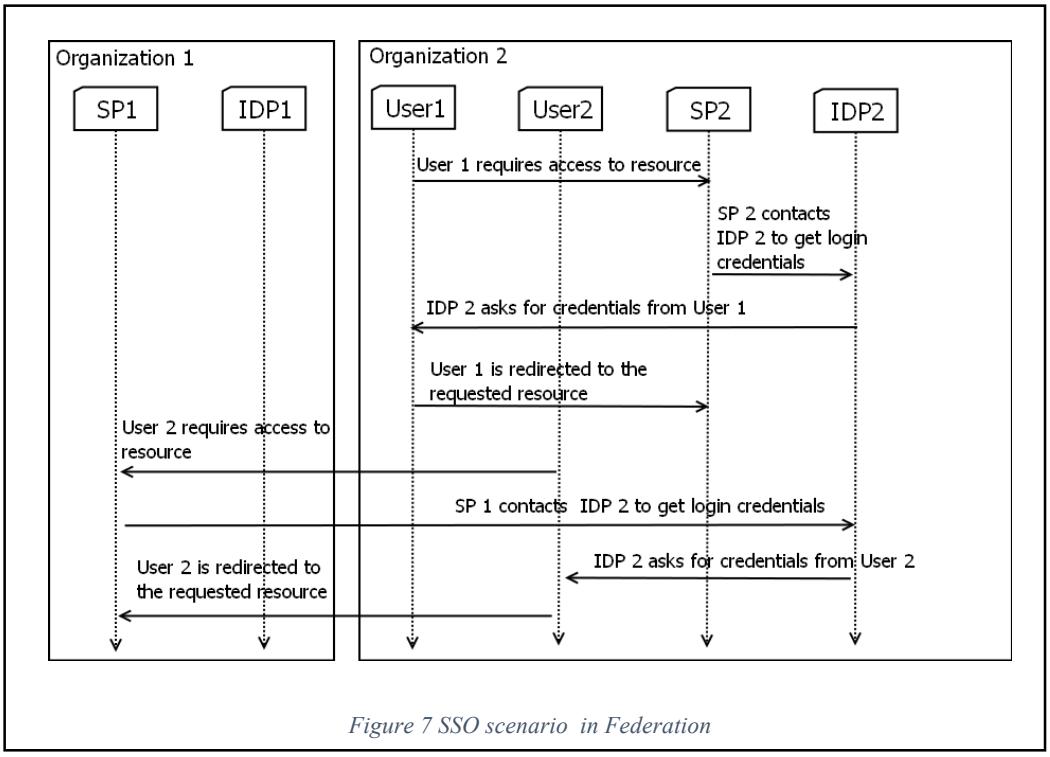

Figure 7 SSO scenario in Federation

Here the steps shown in Figure 7:

- The user requests access to a resource within the SP;
- The SP contacts the IdP to obtain access credentials;
- the IdP requires the credentials to the user;
- after authentication, the user is redirected to the resource he/she had previously requested;
- the user requests a resource localized within a SP other than the user's organization;
- SP contacts the user's IDP.

At this point, the federation plays a relevant role since it has to keep the list of Idp and SP (available within the federation) updated, while the IdP requires authentication credentials. Once the user is accepted, the user is able to access the resource within a different organization.

3.2.2 How it works

The identity does not remain unchanged once logged on the system (like in the University System); it is rather affected by changes because of some operations presented below:

- creation of new users and assignment of credentials;
- change of credentials;
- attribute changes due to promotions, transfers or general changes in roles;
- delete account.

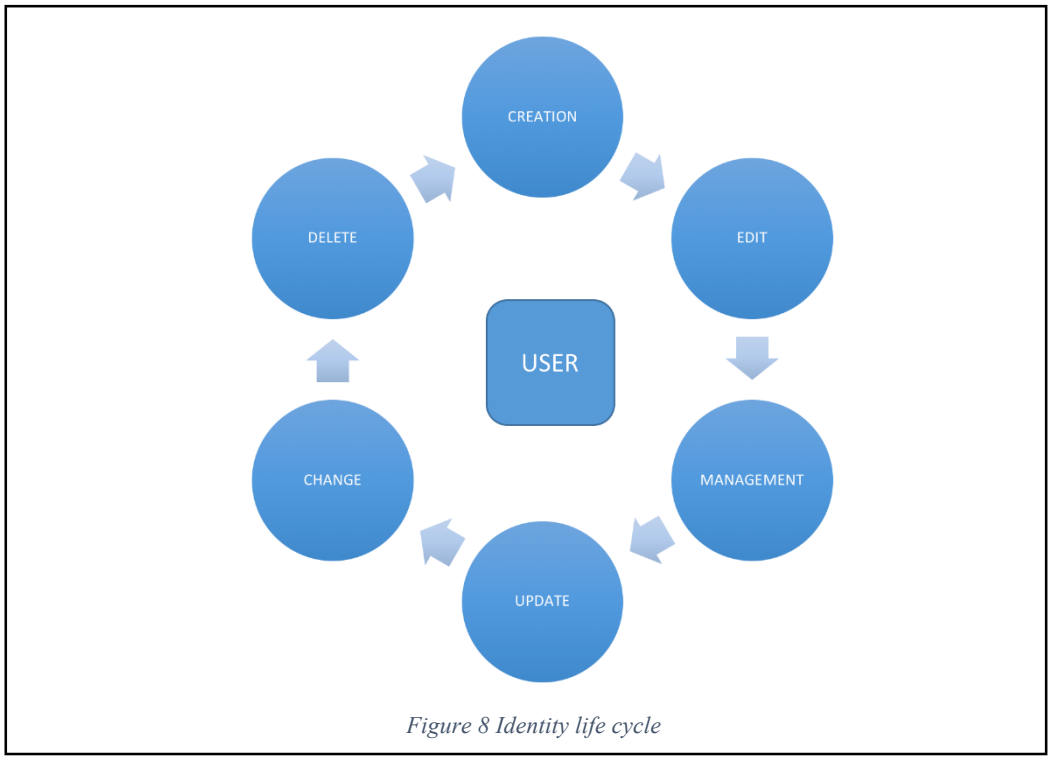

Figure 8 Identity life cycle

The management of the Identity lifecycle includes processes and technologies that enable the implementation, the implementation delete, the management, and the synchronization of digital identities. [7]

The success of the identity and the access management is mainly based on the efficiency of the management of the digital identities' life cycle.

All the administrative operations are defined as User Provisioning which define the user's account and the access rights. Through a good IAM, a centralized management system for the whole administrative process can be successfully reached.

3.3 Systems of identity management

In this context, it is possible to classify the systems of identity management, following the approach adopted during the design and the implementation of the system.

3.3.1 Service-oriented management

The service-oriented approach is available when the system of the identity management is exclusively connected to the resource or to the local service of the same system. The website is a clear example since it has two areas: a public one and one restricted to registered users. An identity management system will be implemented to allow the registration of new users, their access and the customization of the reserved area. This approach, where the Identity Provider and the Service Provider merge into a single system, is currently the most widespread, thanks also to the large availability of systems able to implement this architecture. The ease of this kind of systems' configuration is hindered by the increase in the number of credentials that users must have (one for each different service) and by the high probability of inconsistencies between the attributes of different profiles concerning the user.

3.3.2 Institution-oriented management

When the same system of identity management is used to access different resources or services, it is possible to talk about an institution-oriented approach. A typical example of this approach is that used by the university, which provides students and teachers with different services. The system will associate each user with a single profile, that will allow the service to authorize or not access to the e-mail system, to pay university fees or to register exams.
This approach, which involves a single Identity Provider and several Service Providers, is spreading rapidly among large institutions due to the management simplification of both profiles and attributes.
The introduction of a new service within the institution implies the system's implementation in order to connect the service with the centralized system.

3.3.3 Federated Identity Management

It is possible to talk about federated approach to identity management when different institutions agree to mutually recognize their own principles, so that single systems of identity's management can interact with each other, while keeping at the same time the management of

local profiles. Libraries adopt this approach by offering the possibility to consult online periodicals to other universities. Once the institution has been identified, the system get in contact with the institution in order to find the profile.

The whole group of identity and service providers, along with the agreements between them and those mechanisms that allow them to interoperate are defined identity federation, or authentication and authorization infrastructure (AAI).

The federated identity management and the institution-oriented management have the same advantages. However, the first one is applied in a context in which several institutions operate with the common purpose, such as teaching or research, sharing resources and services.

3.3.4 Management oriented to the individual identity

An approach is defined individual-oriented, when the identity management system, owned by the user, is local to a specific device, hardware or software.

Smartcards adopt the over mentioned approach: once the presence of the smart card is detected, the reading device identifies the principal and it is responsible for the profile's communication (which is registered within the same reading device) to the services the user wants to access. This approach, characterized by an identity provider for each user or access point, and different service providers, is emerging in relation to services such as certified e-mail or Internet banking. The main advantage and the main architectural difference compared to the federated model is the decentralization of the identity provider, which avoids the costs related to the management of the profiles within the institute. However, it is particularly difficult to create the infrastructure and to distribute devices to users.

Chapter 4

Structure of a federated IAM

An IAM server is an infrastructure that manages and maintains information on the digital identity of users in a single centralized point. Resources and services use this infrastructure in order to manage the authentication and authorization.

An IAM server have basic functions:

- must memorize and store large amounts of information, giving the possibility to organize them through schemes and attributes;
- must be safe in terms of safety and security. For this reason, it must have specific mechanisms to ensure that data is not lost in case of malfunctions or technological failures (security) and that data is not read by malicious people (security) ;
- must be reliable and must guarantee continuity of the service, in compliance with the performance specification over time;
- must be performed during read operations and in queries, since they constitute the majority of operations performed on it;
- must be easily interfaced with the access technologies that read the stored information and with automatic procedures for data entering from the sources.

4.1 Standards used for exchange credentials

The credential, a certified document verifying the identity, is the main relevant information when considering the issues connected with Identity Management. Any Identity Management solution has to deal with digital identities, and each digital identity cannot be considered as such without those credentials that supports it.

4.1.1 SAML

Security Assertion Markup Language (SAML[11]) is a XML-based standard, which creates and communicates security tokens. It has been created since 2001 from OASIS[12] Security Services Technical Committee. [8]

[11] Security Assertion Markup Language

[12] Organization for the Advancement of Structured Information Standards , retrieved from https://www.oasis-open.org

The most widespread SAML's Open Source implementation is OpenSAML[13]. The main objective of SAML is the creation of a support system for problems concerning the SSO and, more generally, the IDEM. However, over the years, other standards have emerged, such as proxies' limitation in CAS[14] or the security of SOAP[15] messages.

4.1.2 X.509

X.509 is one of the ITU-T16[16] standards that defines a PKI[17]. Over the years, thanks to the adaptations made by IETF18[18], it has become the main standard with regard to strong authentication in Internet. [9].

PKC[19] is the standard's basic informative element, which is the document that connects the name of an individual with his/her public key. The certificates, which have a fixed temporal validity, are issued and signed by a CA20 upon request from an individual.

A PKI X.509 is typically composed of more CAs organized in a tree structure: the CA is located at the root. Each CA is denoted by a certificate issued and signed by a CA[20] with a higher level, except for the root, which uses a self-signed certificate. The leaves are the individuals for which a certificate has been issued. This structure guarantees the authenticity of an individual through the authenticity of the CA root.Those, who need this guarantee, have to go back to the tree up to the root, verifying that each intermediate node has a certificate signed by the parent node.

In the case of compromise of an individual associated with a certificate (it can happen for instance that his/her private key is discovered), the certificate can be revoked by publishing it in a CRL[21].

In its most recent developments, the X.509 standard specifies the AC[22] certificate. Instead of the public key of a subject, an AC contains privileges and credentials useful for an

---

13 Open Source Security Assertion Language implementation. Retrieved from: http://www.opensaml.org

14 Central Authentication Service

15 Simple Object Access Protocol

16 International Telecommunication Union, Telecommunication standardization sector, retrieved from http://www.itu.int/ITU-T/

17 Public key infrastructure

18 Internet Engineering Task Force, retrieved from http://www.ietf.org/

19 Public Key Certificate

20 Certificate Authority

21 Certificate Revocation List

22 Attribute Certificate

authorization by an appropriate authority. ACs are allowed and signed by an AA[23] and can be stored in a database that uses the LDAP[24] service.

[23] Attribute Authority
[24] Ligthweight Directory Access Protocol

Chapter 5

Single Sign On

Single Sign On is a particular form of authentication and authorization, which allows access to more protected resources starting from a single initial authentication made by the user. There are different types of SSOs depending on the application contexts and the involved individuals.[10]

This section is focused on the Single Sign On. The SSO is only applicable to authenticated access to web resources, static or dynamic, typically performed through a browser. Firstly, the users who access such resource, for the first time, during the same time session, are redirected to an authentication service, more precisely the Identity Provider. Secondly, if the Identity Provider was successful, users are redirected back to the first resource, in particular the Service Provider, to which the updates of the user authentication status will be communicated.

SSO has clear advantages on the web, compared to the traditional solutions characterized by a distinct credential for each site are evident:

- **Usability**: the user must carry out less interactions in order to access the same services;
- **Security**: More attention has to be paid on the authentication phase since it has become a rare occurrence by implementing strong cryptographic techniques. For the SSO of a federative type, this is true because the Identity Providers could be the only ones to have the information necessary to carry out this operation on their subjects
- **Scalability:** When the user authenticates to a specific Identity Provider, he/she can access resources protected by a Service Provider who does not need to keep information internally. The Provider can rather obtain it from the Identity Provider. This phenomenon implies a greater scalability of the SSO solutions with the consequent increase of the subjects that can be used for the same investment;
- **Incentives to the federation**: a Service Provider joins a federation in order to access a wider user group, i.e. that of the Identity Providers of the federation. However, an Identity Provider, that enters a federation, guarantees to its users a better offer of services in terms of quality, quantity and diversification.

5.1 LDAP

The LDAP (Lightweight Directory Access Protocol) is basically a directory and access management protocol. The LDAP operates onTCP/I[25] or on other connections oriented to transfer services.

A directory service is used to associate names to objects, in which each object is characterized by a number of attributes consisting of name and a set of values. Directory services are optimized in order to carry out objects' researches, carried out through the name of the object and that of the value of a given attribute. The objects in a directory service are typically an element of the environment in which the service is used, such as a user, a computer, a printer, or a network, and each object will contain a set of attributes which describes what it represents. A directory is therefore a set of objects and a directory service is a service designed to manage the objects of a directory and to perform researches on them.

Before the introduction of the LDAP, in order to access the data stored in an X.500 directory, a client had to support the DAP[26], which imposed a considerable penalization of the resources involved as it required the use of the OSI[27] specification. This specification is now largely replaced by the TCP/ IP protocols TCP / IP and others. LDAP was introduced with the aim of replacing the DAP as the latter one was quite onerous when considering the used resources.

LDAP is a client-sever:

- The LDAP client sends a request to an LDAP server;
- LDAP server processes the received request;
- LDAP server possible accesses a database directory and finally returns the results to the client.

In LDAP, information are organized hierarchically, in a tree structure, where each node represents an object, called entry.

[25] Transmission Control Protocol (TCP) and Internet Protocol (IP)

[26] Directory Access Protocol

[27] Open System Interconnection

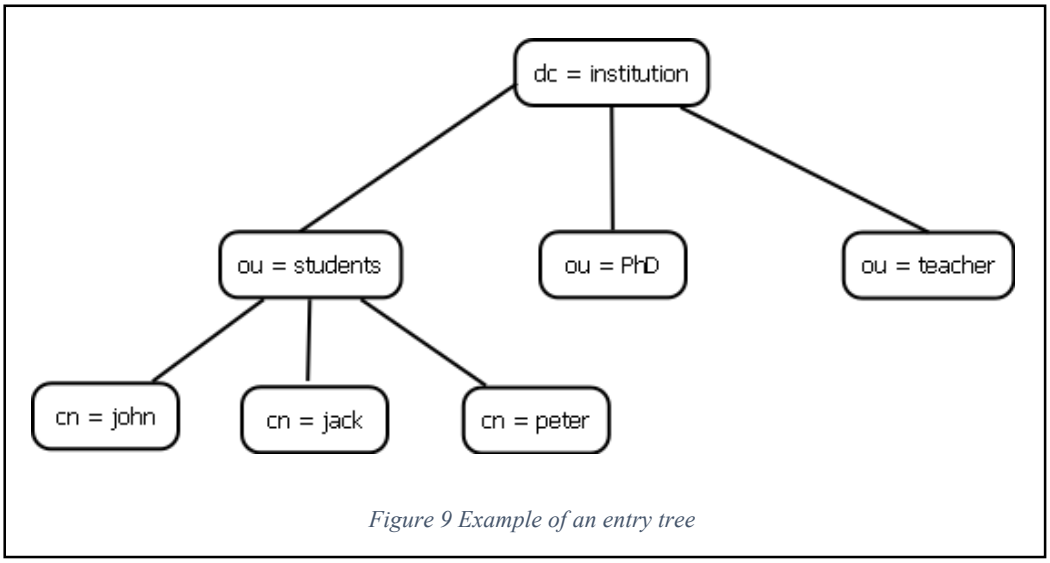

Figure 9 Example of an entry tree

The Entry tree is also called DIT[28]. Each entry of the DIT is uniquely identified by its **DN**[29], consisting of the concatenation of the names of its ancestors to the root.

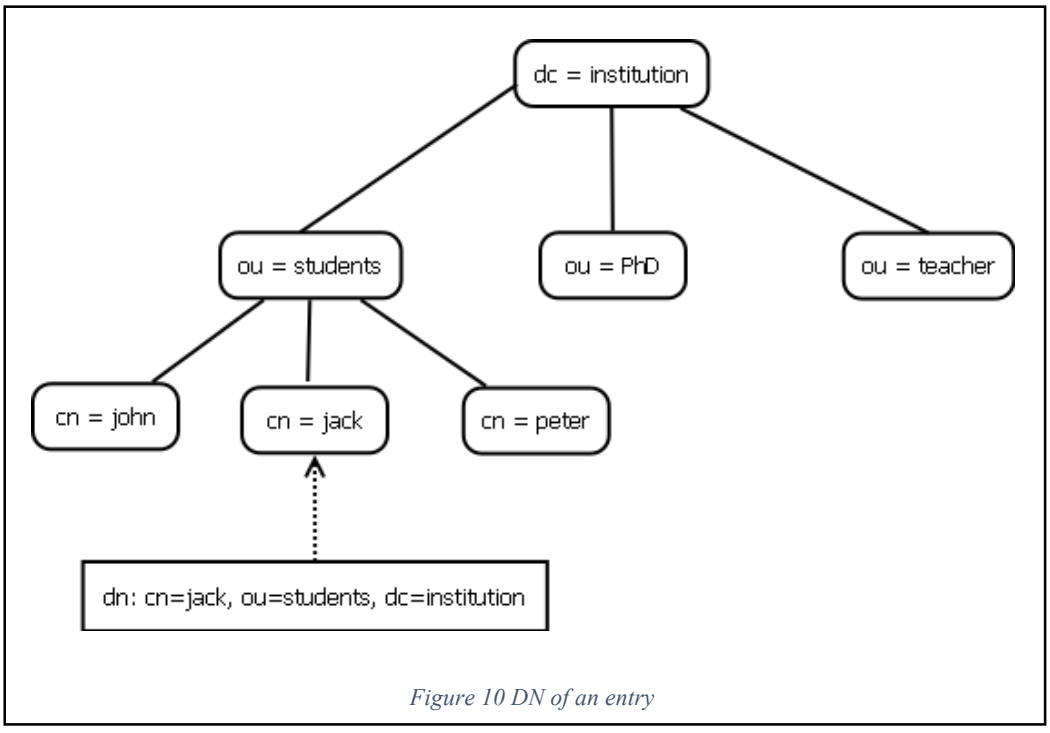

Figure 10 DN of an entry

An entry is a collection of attributes having a single global name, the **DN**.

Each attribute of the entry has a type and one or more values.

Types are usually mnemonic strings, such as ***cn*** for common names, or ***mail*** for e-mail addresses.

[28] Directory Information Tree

[29] Distinguished Name

In LDAP, the directory entries are structured in a hierarchical tree structure: at the root of the tree, the entry that represents the institution can be found (**DC** = Domain Component) then the entry represents the categories of the staff of the institution (**Ou** = Organizational Unit).
Finally, there are other types of entries that can represent individuals (UID = User ID).
The ***ObjectClass*** attribute defines the structure of the entry, this means the attributes that are present and those mandatory (they must necessarily contain a value). A class of objects is a set of attributes that have in general a common meaning. For example, the person of the object class has a set of attributes that are used to describe a person, such as the first name, surname or telephone number. During the definition of a class of objects, it is possible to indicate which attributes are mandatory and which are optional. The value of the special ***ObjectClass*** attribute in an entry must be the name of an object class and the structure of the entry will consist of attributes defined in that object class. The structure of each entry is independent from other objects and position in which it is located in the tree.

5.1.1 Attributes

LDAP uses the X.500 syntax for the definition of attributes. For each attribute is specified:

- ***OID***[30]***:*** Object Identifier issued free of charge from IANA[31]
- ***Name***: Name of the attribute with any alias below.
- ***Meaning***: a short description of the attribute use.
- ***Syntax***: Standard ID of the syntax used for the value of the attribute.
- **Comparison rules**: rule used to compare the value of the attribute during a research.
- **Multiplicity**: indicates whether it is possible to memorize more values in the task.

5.1.2 Object class

The X.500 syntax is also used to define the object classes. For each object class the followings

- ***OID***[32]***:*** Object Identifier issued free of charge from IANA[33]
- ***Description***: a brief description of the meaning of the object class
- ***Category***: This could be abstract, auxiliary or structural. Each entry needs to have at least one object class structural and cannot just have an object class auxiliary. An object class abstract can instead be used just to derive other class of objects.

[30] Object IDentifier
[31] **Internet Assigned Numbers Authority, sito di riferimento: https://www.iana.org**
[32] Object IDentifier
[33] Internet Assigned Numbers Authority, retrieved from: https://www.iana.org

- **Mandatory attributes**: list of attributes that must exist in the entry.
- **Optional attributes**: list of attributes that may exist in the entry.

5.1.3 Schema

A schema is a set of attribute definitions and a class of objects. There are some standard schemes, such as RFC4519, but in general it is possible to define a personalised scheme.
Some organizations realize and promote their schemes in order to standardize the use of some common attributes and encourage the creation of federated structures. For instance, Internet2[32] created the eduPerson[33] scheme, which can be used to describe people in a University environment.

5.1.4 Active Directory

Active Directory is a set of network services, better known as directory services, adopted by Microsoft Operatve Systems, starting from Windows 2000 Server, and managed by a domain controller; it allows to centrally catalogue and manage different resources such as: users, work groups, printers, shared folders, etc. [11]
The database structure is hierarchical with containers that contain objects and other containers. Active directory is a structure that must be present in every IT environment in which one or more servers must have control over all clients on the network. Active Directory is a set of network services better known as directory service adopted by Microsoft operating systems starting from Windows 2000 Server. It is based on the concepts of domain and Directory, which in English means "telephone directory".
Active Directory can be imagined as a "telephone directory" and the Domain as a world in which all the resources of the network are concentrated starting from:

- user account;
- computer account;
- shared folders;
- printers, etc.

The set of Active Directory network services, in particular the authentication service, create another important feature of the Single Sign-On (SSO). Through this mechanism, a user, once entered into the domain and then logged in from any of the domain machines, can have access

[32] Sito di riferimento: https://www.internet2.edu
[33] https://www.internet2.edu/products-services/trust-identity/eduperson-eduorg/

to resources available on the network (shares, mailboxes, intranets, etc.) without having to re-authenticate, facilitating the management of users, despite what happens in peer to peer networks. Active Directory is the name that Microsoft uses to refer to its implementation of security on a distributed computer network. It uses various protocols (mainly LDAP, DNS, DHCP). In Active Directory, LDAP is used as a database that centrally stores all the information of a network domain, in terms of authentication and access to services, with the advantage of keeping all this information synchronized between the various authentication servers for access to the network. Active Directory networks can vary from a single installation with a few hundred of objects to large installations with millions of objects.

Unlike the old account management and server systems like User manager for domain and Server manager for domain, AD includes in a single monitoring system of all the objects in three broad categories of the domain: resources (e.g. printers), services (e.g. e-mail) and users (user and group accounts). AD provides information about the objects and organizes them. Moreover, it controls accesses and sets security. AD is a logical grouping of users and computers in a domain, centrally managed by servers called "domain controllers". An 'Active Directory' structure is a hierarchical framework of objects. AD provides information about the objects and organizes them. Plus, it controls access and sets their security. Each object represents a single entity - perhaps a user, a computer, a printer, or a group - with its own attributes. Some objects can also be containers of other objects. An object is uniquely identified by its name and it has a set of attributes - the characteristics and information that the object can contain - defined by a schema, which also determines the type of objects that can be stored in Active Directory. Each attribute object can be used in different object classes of a schema. The schema object exists to allow the schema to be extended or modified when it is necessary. However, since each schema object is part of the definition of Active Directory objects, the disabling or the modifying of the schema object can have serious consequences because it will permanently modify the Active Directory structure. A schema object, if modified, will automatically propagate through Active Directory and, once created, it can only be deactivated or not deleted. The Websites contain objects called subnets and can be used to assign "group policy" objects, simplify the identification of resources, manage the replication of the AD and manage the traffic related to the connection to the network. Websites can be linked to other websites. Costs can be assigned to the objects of a linked website. Those costs represent the speed, reliability, availability or other real properties of a physical resource. A planning can also be assigned to the links to the websites. The AD structure can be logically divided into three different entities: domains, trees and forests. A domain represents a set of machines

connected to each other, which share a common database directory in which the objects are inserted. Domains are identified based on the structure of their DNS name, the namespace. A tree is the set of one or more domains that share a contiguous namespace. These domains are connected to each other in a hierarchical way and the different domain controllers can exchange information with each other. This is defined as a transitive trust relationship. At the highest level of the structure, the forest is defined, that is, the set of trees in the directory. These trees share a global catalogue, a directory schema, a logical structure and a configuration. The forest therefore represents the area where users, computers, groups and other objects are accessible. In a multi-domain forest, the server that runs the domain on which the primary AD (the highest-level one) is installed, is called "Primary Domain Controller". Active Directory objects within a domain can be grouped together in Organizational Units (OU). Using these structures, it is possible to form a hierarchy within the domain and thus facilitate its administration, for instance by subdividing the structure into geographical terms. Organizational units are the recommended level to implement group management policies, which are Active Directory objects officially called Group Policy Objects (GPOs) for the delegation of administrative powers. In any case, organizational units represent a simple abstraction that can be realized for administrative purposes and, therefore, do not provide a physical container of objects. In this context, it is not possible to define, for instance, a user account with the same name in two organizational units belonging to the same domain, since the effective visibility is always limited to the chosen domain and not to the organizational units contained within it. Active Directory was released as a beta in 1996, for the first time with Windows 2000. Then, in Windows Server 2003, a version with new features was released. Further improvements have been made in Windows Server 2003 R2. It has been further refined with Windows Server 2008 and Windows Server 2008 R2 and has been renamed Active Directory Domain Services.

5.2 Shibboleth

Among the various technologies applied to Federated Identity Management, Shibboleth represents the most widespread open solution. It is no coincidence that it is widely used in the largest federations, in particular in IDEM federation.

Shibboleth, is an interuniversity project of the MACE[34] group, belonging to the Internet2 consortium. Its purposes are the design, specification and implementation of Open Source

[34] Website reference: http://middleware.internet2.edu/MACE/

systems for the interinstitutional sharing of web resources, which are subject to the control of the accesses.

This project was initially created to simplify the issue of access to educational contents, which are reserved or paid by multiple University campuses, each with a different authentication infrastructure.

However, Shibboleth can be applied to other contexts such as e-Business and Public Administration and it is, probably, the most complete Open Source implementation of a web-based SSO system. Moreover, unlike other similar systems, Shibboleth has adopted SAML to implement most of its protocols, ensuring the interoperability with other systems belonging to the Identity Management area.

The architecture of Shibboleth includes the following elements:

• **Identity Provider** (IdP): the entity able to authenticate the user and provide additional information or attributes on his account;

• **Service Provider** (SP): the system that manages the web resource to which the user may apply; in addition, it has the task of protecting the web resource above-mentioned through some form of access policy;

• **User Agent** (UA): it is the application that, on behalf of the user, activates the SSO protocols that require access to a Web resource protected by a Shibboleth-interoperable SP;

• **Where Are You From** (WAYF): concerns a service managed by a third party or by the same SP whose job is to discover the user's IDP.

There is no hierarchy between suppliers. In fact, each organization presents the following duties: IdP is responsible for its users; SP is responsible for its users and / or its own resources. The resulting network of trust is horizontal and in Shibboleth it is called federation. The SP that chooses to join a federation, implicitly accepts a bond of trust with all the IdPs that are part of it.

Symmetrically, an IdP agrees to release assertions, upon request, of any SP federation. However, Shibboleth does not prohibit the possibility of specifying relations of trust at bilateral level between single SPs and IdPs. This latter approach has reduced scalability, but it may be useful for small configurations or to specify differentiated relationships between providers within a pre-existing federation.

The Identity Provider administers the information of the users who is responsible for and can be used for:

1. register the users and keep the information;

2. manage authentication sessions. It must be able to authenticate its users, such as the request for credentials, or verify that the user already has a valid session (single sign on);
3. to release attributes of the authenticated users for which it is responsible and to release them to those who make requests, protecting their personal data. This information is requested by the ServiceProviders to authorize the user to use a given service. Not all data are sent to the SP but only those which are necessary.

Each resource, whose access must be protected, requires a Service Provider (SP), which takes care of directing the user to the organization in which it belongs (or Identity Provider) for his recognition and the release of attributes. The SP collects information that have been sent by the IdP about the user, then it uses them to protect the service and grant the authorisation to the user who has requested it. For a SingleSignOn protocol, the interactions between the different entities are the following:

1. the UA, UserAgent (for the user), requires access to a web resource at the SP. It is assumed that the UA does not have an active session with the SP yet;
2. the SP redirects the AU to the WAYF or directly to the IDP of the registration. The content of the URL of destination constitutes an authentication request and contains information about the requested resource, an identifier of the SP, and the endpoint to which the SP intends to receive the authentication assertion;
3. if asked, the WAYF processes the authentication request of the SP (transported by the UA) and interacts with the user to know the IdP with which it intends to authenticate. This information is stored in a long-term session between WAYF and UA (for instance a cookie). The WAYF, then, redirects the UA to the IdP selected by the user;
4. the IdP identifies the user by activating an authentication mechanism or by taking advantage of a session that is still active. Shibboleth does not consider the particular authentication mechanism used;
5. the IdP sends an authentication assertion to the SP using the SAML / POST browser;
6. the SP sends a request for attributes from the user to the IdP;
7. if asked, the IdP answers to the request for attributes of the SP;
8. based on user information, the SP takes the decision for the access of the user to the requested resource. Depending on the outcome of this decision, the SP sends an appropriate HTTP response to the AU.

To support the SSO protocols, the providers involved expose a series of distinct functionalities. In Shibboleth terminology, these functions are called roles. A typical IdP will have a set of roles

disjoint from a typical SP. However, it should not be excluded the case in which the same provider supplies some of the functionality of IdP and SP at the same time.

5.2.1 Metadata

Metadata can be seen as the "Identity Cards" (in XML[35] format) of the "trusted" participants, that is, those who belongs to the federation and has been used as an instrument with which trust relationships are built up among the members. Shibboleth uses metadata to communicate information to trusted IdPs and Service Providers; moreover, it distributes information about CAs (certification authorities). They are collected in a file based on SAML 2.0 standards, containing:

• Certificates
• Scope of the IdPs
• Textual description of the participants

The use of self-signed certificates for Sp-IdP (back-channel) communication is allowed. Each participant uses the relevant certificate contained in the metadata in order to verify the identity of the counterparty and communicate. The SP interacts only with a known IdP (whose data are in the MD file).

[35] eXtensible Markup Language

Chapter 6

Case study: SSO integration on Moodle platform

In this part of the dissertation, the method that has been used to create three different SSO authentication configurations, for the Unicam E-learning platform, will be exposed.
For each of the three configurations, a dedicated Virtual Machine, having Linux (Ubuntu 17.10) as Operative System, has been implemented to test and verify whether it works properly before configuring the Unicam E-learning platform.
The first authentication takes place using a centralized identity and access management system internal to the University of Camerino, configured using the internal LDAP authentication service connected to Microsoft Active Directory. This configuration consists of:

- installation of the Ubuntu 17.10 LTS virtual machine;
- installation of Apache Web Server, Mysql and Php;
- installation of Moodle;
- installation of the Php-Ldap module;
- Ldap plugin configuration and data mapping;
- connection of the Moodle Platform to the Ldap Server;
- configuration Test.

In the second and in the third authentication, the Shibboleth package has been implemented, involving the installation of a Service Provider connected to the Unicam Identity Provider. On the one hand, in the second authentication the service involves only the individual Unicam SP and Idp; therefore, the authentication remains within the configuration of the institution.
The steps of this procedure can be summarized as follows:

- installation of the Ubuntu 17.10 LTS virtual machine;
- installation of Apache Web Server, Mysql and Php;
- installation of Moodle;
- configuration of the SSL Certificate;
- installation of the Service Provider;
- configuration of files by using Unicam IdP specifications;
- connection to the Unicam Identity Provider;
- configuration of the Moodle Shibboleth Plugin;
- configuration test.

On the other hand, in the third authentication a Federated IAM system is used, where the Service Provider and the Identity Provider act within the Federation:

- installation of the Ubuntu 17.10 LTS virtual machine;
- installation of Apache Web Server, Mysql and Php;
- installation of Moodle;
- configuration of the SSL Certificate;
- installation of the Service Provider;
- federation of the SP;
- configuration of files using IDEM Federation Specifications;
- connection to the Unicam Identity Provider via the Federation;
- configuration of the Moodle Shibboleth Plugin;
- configuration test.

6.1 Moodle

Moodle[36] is a free and constantly evolving Learning Management System (LMS) and Course Management System (CMS), which allows the creation of virtual classes aimed at e-learning or on-line learning.

The fact that this software is free and easy to use has contributed to its rapid diffusion: currently, the total installations are 95,538, and the number of its users is more than 129,000,000. [12]

It is mostly used by training centres and, in Italy, also by Public Administrations for training projects (especially schools and universities).

It is the best possible choice for those who are looking for many functionalities and those who are planning to invest considerable resources in the training program at an organizational level.

This project was created in the 90s, and it is still developing with the aim of creating a support for teaching, which is based on social constructivism. This is realised by integrating various tools for collaborative work (for instance, forum, messaging snapshot, integrated e-mail system, workgroup management, shared agenda, blog, wiki).

Moodle can be installed on any platform that supports Php, Apache and MySQL and that is translated into more than 80 languages. Another aspect that further qualify Moodle is the administrative ease of tailoring the user interface with the possibility to move, delete or add

[36] Modular Object-Oriented Dynamic Learning Environment

modules. The system uses style sheets or CSS and PHP web pages to which it is relatively easy to make changes.

6.1.1 User management in Moodle

The Moodle platform also allows advanced user management. In fact, thanks to the installation and integration of Mysql it is possible to administer the users directly from the platform, dividing them by groups with the possibility of assigning, for each user, attributes and custom fields.

The platform also allows the possibility to add, edit or delete users, but also to assign roles within the platform, such as: the possibility to be enrolled or not in a given course, the possibility of creating courses (through the teacher role), and the possibility to administer the entire platform (through the role of administrator).

It is also possible to query within the platform, by filter the research by attributes, groups, date or even by number of accesses. With the selected users there is the possibility to modify groups, delete users, send messages or e-mails, etc.

6.1.2 User authentication methods in Moodle

Moodle is a platform that allows a centralized management of accesses. The platform independently manages the authentication and authorization of users and gives them the chance to have access to the diverse services present within it.

Basically, the platform allows users to make a registration via e-mail, then, they should enter their data in the opportune fields and choose both username and password to access the platform.

Another way to import users is via ".csv files", which allows the integration of a list of users in the Moodle database; moreover, it automatically send temporary credentials to all users via e-mail.

However, once the users have been logged in, the administrator will indicate them the permissions to access the different available courses, by enabling them (through access codes) to view those courses or not. It also enables users to a manual registration, by payment or through the assignment of roles, which allows them to have privileges in certain courses (for instance, the role of teacher editor enables a user to realise an instructional design within the course, while with the role of non-editor teacher the user can only view students' marks and their statistics).

Over the years, due to the proliferation of passwords for users, the Moodle platform had to adopt novel solutions to allow the integration between the system and different authentication systems with centralized management implemented by a given organization.
For this reason, several plugins have been developed in order to allow the integration of different systems such as:

- Ldap;
- Shibboleth;
- MNet[36] ;
- Server Cas[37];
- DB Esterno;
- Server Imap[38];
- PAM[39];
- Pop3[40];
- Radius[41].

Obviously, the implementation of these plug-ins foresees, from the developer, an appropriate configuration of the services inside the machine that hosts Moodle; hence, to provide the installation of services, php extensions, appropriate file configurations, permissions to directories. However, the most important task is make Moodle perfectly configured to avoid anomalies.

6.2 State of the art

Currently, the E-learning platform of the University of Camerino (elearning.unicam.it), used for the realisation of courses in e-learning mode, is installed on a Virtual Machine, which have: Centos 7[42] as Operative System, the Apache Server[43], database Mysql[44] and Php 5.6[45]. It is managed internally by the E-learning Office of the University, which deals with the technical management of the Virtual Machine and makes changes and customizations of the technical management of the application. Those changes are realised with graphic customizations

[36] Moodle Network
[37] Central Authentication Service
[38] Internet Message Access Protocol
39 Pluggable Authentication Modules
40 Post Office Protocol
41 Remote Authentication Dial-In User Service
42 Website reference: https://www.centos.org/
43 Website reference: https://httpd.apache.org/
44 Website reference: https://www.mysql.com/it/
45 Hypertext Preprocessor

(importing user technical management and courses) of the design of online courses with innovative techniques in the field of teaching (this is subject to various publications), ensuring support for educational activities for both students, tutors and teachers of Unicam.

The platform, which hosts the Moodle LMS, has 3260 registered users (almost half of the total enrolled in the Camerino University) and nowadays includes 97 active courses for all the students of the University of Camerino, representing all five faculties (Architecture and Design, Biosciences and Veterinary Medicine, Law, Pharmaceutical Sciences and Health Products, Sciences and Technologies) delivered in different ways: e-learning, blended, or as support material.

E-learning has had, in recent years, a great developent in the world of Universities. In particular, the potential of e-learning 2.0 has led many teachers to create learning objects for their subjects and has brought considerable traffic of users on the platform. This could be possible thanks to the tools applied to the evaluation and monitoring of activities carried out by students.

To make the use of e-learning contents more rapid, it was decided to allow students access to the platform through manual registration, and then to request, through a form (which is automatically sent to the administrator's email address), the access to the various courses.

The University of Camerino has implemented an internal system for managing digital identities. This is realised with an institution-oriented management by using, mainly, the internal Ldap authentication service connected to the database Microsoft Active Directory, which consists of a hierarchical directory system, in which the profiles of all the professors, researchers, PhDs, students and administrative staff of Unicam are managed.

This service is widely used, as it allows all users to access all the services on the network, both internal and external.

In fact, through the Ldap service, teachers and students can perform the following activities: consult the service for the reservation of exams (called 'Esse3'), consult the e-mail, use the library database, and access different portals. The administrators also can log in to consult the e-mail and access the portal (for instance to view the salaries).

This directory service is the most used inside the University, so that users of other universities or institutions cannot access with their own credentials to services offered by the University of Camerino.

The only case in which an authentication service has been used for a web service that has been opened to other universities is the library portal of the University of Camerino.

For this reason, there were the need to implement a federated access system. Therefore, a SSO system based on Shibboleth was set up, and the relative Service Provider and Identity Provider

were installed to allow the interaction and validation of the certificates and metadata for the purposes of authentication and authorization of digital identities wishing to access to the resource.

Hence, it has been decided to try to set up the LDAP service, trying firstly the authentication with centralized non-federated management, and then the federated one, by using the Shibboleth service.

6.3 Integration of the LDAP service

6.3.1 Authentication

The first implementation that it was decided to develop was concerning the integration of the Moodle platform with the LDAP authentication service.

The LDAP service is a client-server service that, once integrated, acts as an intermediary between the user and the Moodle platform.

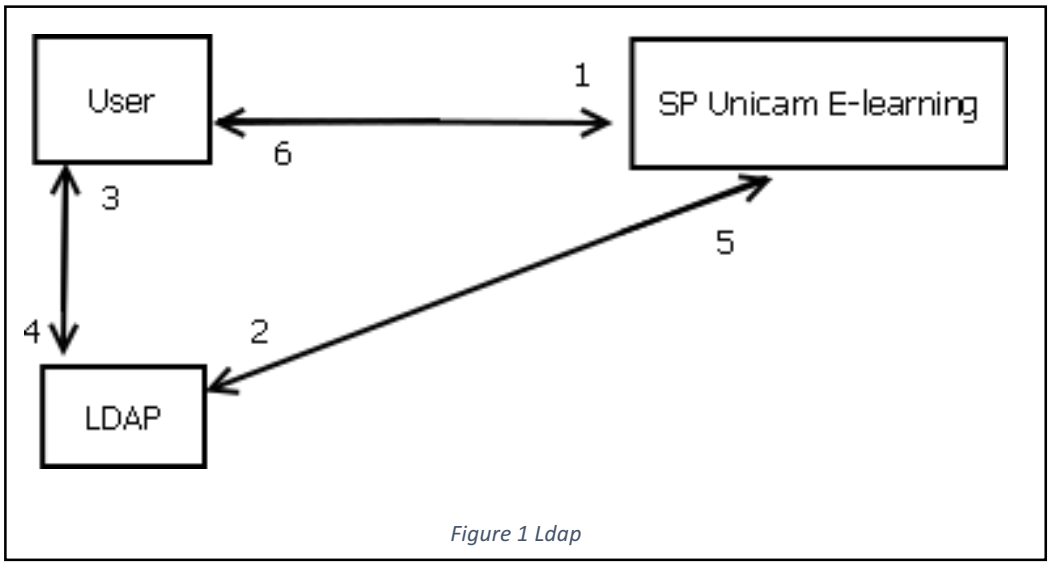

Figure 1 Ldap

As it can be seen from Figure 11:

1. the user requests access to the e-learning platform;
2. Moodle, which acts as a client, submits the request to the LDAP server;
3. the LDAP server processes the request received by accessing AD to verify the identity of the user;
4. the LDAP server has verified the identity and returns the results and attributes of the user to the Moodle Client;
5. the Moodle platform recognizes the user and authorizes him to access.

To perform this integration, it was decided to create a new virtual machine to avoid changes directly in the machine hosting the Unicam E-learning platform, and thus create possible system anomalies.
For this reason, a new Virtual Machine was installed with the Operative System Ubuntu 17.10, using the VMware[46] program.
After having installed Ubuntu, the Moodle CMS had to be installed, and then Apache, Mysql and Php.
After having installed Moodle, it was necessary to install the Php-Ldap module, which is fundamental to allow the Ldap Plugin, which is by default in the platform, to function correctly by executing the following command:

- `apt get php-ldap`

After installing Php-Ldap, it was necessary to enable some extensions to allow both MySQL and Ldap databases to be included in Php.
Then, it was necessary to access to "php.ini", present at "/etc/php/7.0/apache2/php.ini" and add the following lines of code:

- `extension=mysql.so`
- `extension=gd.so`
- `extension=ldap.so`

For an optimal functioning, it was also necessary to properly set the cron, which is fundamental for synchronizing Ldap users with the Moodle system.
Thus, from the terminal it was necessary to open "crontab" :

- `crontab -u www-data –e`

and insert the following caption:

- `*/1 * * * * /usr/bin/php /path/to/moodle/admin/cli/cron.php >/dev/null`

This allows the reload of the cron every minute, and therefore a continuous updating of the pages. This would allow, above all, an optimal interaction between the Moodle modules and the e-mail.

[46] Website reference: https://www.vmware.com/it.html

After having set the cron, in order to permit the users syncronization, it was essential to access the platform as an administrator and click on “Site Administration”-> “Server” -> “Scheduled tasks”-> “Ldap users sync job”, enabling the service (disabled by default) and entering the related data to customize the synchronization.
In our case it was decided to enable it every 5 minutes, so the character * was placed in all the fields, inserting in the box of minutes */5.
After having perfectly set up all the extensions and plugins related to Ldap and Cron, it was the time for the real configuration.
To do this, I needed to contact the Data Processing Centre of the University of Camerino (Cinfo) to obtain fundamental information about the functioning of the LDAP service. It was therefore necessary to find:

- the host address of the Ldap server configured with students' data;
- the distinguished name of an account designed to scroll through all the entries of the database for the validation of users and the retrieval of their attributes.

In order to view and scroll through all the attributes of each Unicam student, an AD explorer was used, downloading the free Apache Directory Studio[47] tool.
Thanks to the DN of the account provided, it was possible to scroll through the hierarchical data structure of Unicam Active Directory, starting from the DC contexts, to the OU sub-contexts, up to the individual CN.
In this way it was possible to view all the attributes configured according to the student, in order to be able to evaluate which ones should be recalled in the mapping phase of the data on the platform.
Among the various attributes related to the students, below it is presented a list with the attributes selected:

- sAMAccountName (account);
- givenName (Name);
- sn (surname);
- mail (e-mail);
- department (degree course);
- employeeNumber (identification number).

[47] Website reference http://directory.apache.org/studio/

Further information such as EmployeeID (fiscal code), physicalDeliveryOfficeName (place and date of birth) have not been used, as they are not relevant for the purposes of access to the e-learning platform.

Finally, to allow the operation of LDAP authentication, it was necessary to configure the Moodle Plugin dedicated to the management of authentication with the LDAP server, by accessing directly on the Moodle platform and clicking on “Site administration”->”Plugins”->”Authentication”, enabling the "Ldap" function (making the Plugin visible).

The following values have been entered in the configuration form:

- Host URL;
- Bind Settings:
  - insertion of the DN of the account and of the relative password, in order to allow the research of the users within the AD data structure;
- User search settings:
  - Type of user (in this case Active Directory);
  - Contexts: list of contexts where users are present;
  - User attribute: attribute used to research users.
- Data mapping:
  - Possibility to find the values users' attributes and insert them as descriptive fields of users' accounts in the Moodle platform, inserting the same syntax of LDAP attributes.

Once configured, students could access the platform by simply entering their credentials on the login page (https://elearning.unicam.it/login).

Then, identity verification occurs (through the cn attribute). Thus, it would create an account in the e-learning platform, importing the values of the attributes retrieved in the data mapping phase.

One of the advantages of this solution is the synchronization of LDAP users with the platform thanks to the cron: in this way, we have an optimal centralized management of the users. In fact, we are not worried about deleting or modifying the profiles within the e-learning platform anymore, as any changes made in AD are automatically updated also on the platform when the user authenticates.

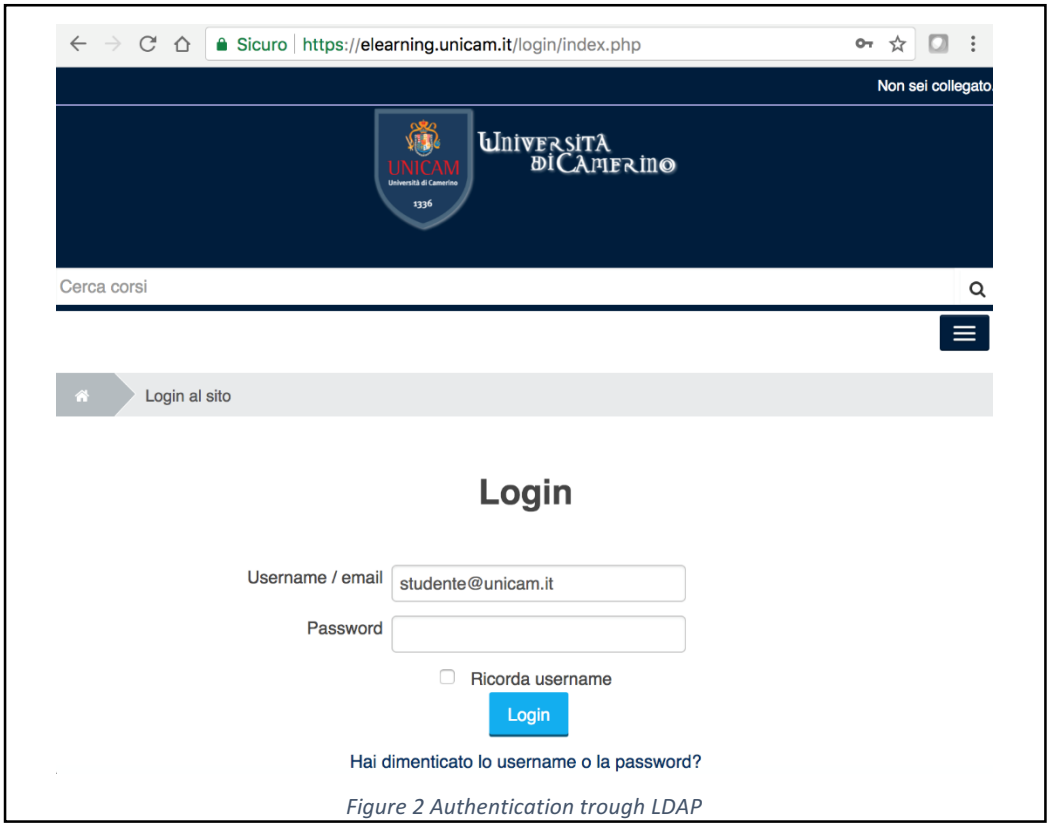

Figure 2 Authentication trough LDAP

6.3.2 Moodle courses enrollment

The Moodle platform includes a Plugin that allows, through the Ldap service, to have automatically access to certain online courses.

To do this, it is necessary to have, in Active Directory, a structure provided with Groups (objectClass = group), where users are collocated.

Once configured, the plugin allows the members of a specific group to be accessed in a given course, which must have the same identification code of the course. It is possible to decide for each group that must be included in the platform, the role that have to be assigned, allowing more or less privileges depending on whether the user is a student or a teacher.

The plugin must be configured by entering the following values:

- host;
- bind settings;
- role mapping: in this section, the plugins must be specified next to the roles that have to be assigned in the platform:
  - LDAP contexts, which the related groups belong;
  - the univocal LDAP identifiers (usually "member").

- Identification Code attribute: fundamental to determine the membership of a user to the group (it must be the same that has been configured in the mapping of the authentication LDAP plugin data).

The part regarding the course enrolment settings must, instead, contain the following data:

- Object Class: used by LDAP to research the courses (group);
- Identification Code: attribute that contains the identification code of the course.

Finally, for the automatic creation of courses settings, it was necessary:

- enable the automatic course creation;
- choose the category in which the new course should be created.

In this case, if the ID attribute of the LDAP group does not match any ID of any course, Moodle will create a new course (which will take the name of the group as its name).
In the case of the University of Camerino, it was decided to try this function by using LDAP profiles that were already part of LDAP Groups.
The Unicam Active Directory structure has been initially implemented for the management of administrative staff profiles; hence, it is organized by Organization Unit (OU). This has also been spread to students, which enrolments to a specific degree course are divided into OUs and not into groups.
Furthermore, there is no attribute in LDAP that refers to the degree course to which it belongs, so that the recognition of users on the platform is difficult.
To test the service, it was therefore necessary to make reference to the LDAP group of second-year students of the Computer Science Degree Course, as it is one of the few groups implemented in the structure.
After filling in the fields related to the host and the bind settings, only the student role was mapped, inserting the context related to the faculty of science to the degree course in Computer science, which includes the Ldap groups that Moodle consults. The distinguishedName (which had already been mapped as an attribute in the LDAP authentication phase) was then entered as the identifying attribute of the user, which is fundamental as it is the attribute that identifies the user and allows it to belong to the group. On the one hand, in the ObjectClass field, the type of object used to create courses (group) was defined. On the other hand, the cn of the defined objectClass was used as the idNumber, title and abbreviated title of the course.
In this way the course that will be created (if the id does not correspond to an existing course) will have the same name of the Ldap group that have been set.

Once the plugin was configured and enabled, a new login was made with the LDAP credentials of a user belonging to the group who had never accessed the course.

Now, Moodle will look for Ldap Objects where:

- the value of the Ldap "member" attribute contains the value "ID number" of the user;
- the type of objectClass is equal to the value of the ObjectClass specified in the plugin.

If these two conditions are verified, in addition to the recognition and generation of the account in local, Moodle checks the course id to enrol the user in the respective course. If no course is associated with the group's cn, a new course will be created, where the user is enrolled.

Since the groups are divided by year of course, the name of the course refers to the group of students of the 2 year of Computer Sciences (StdLM-CS.A2).

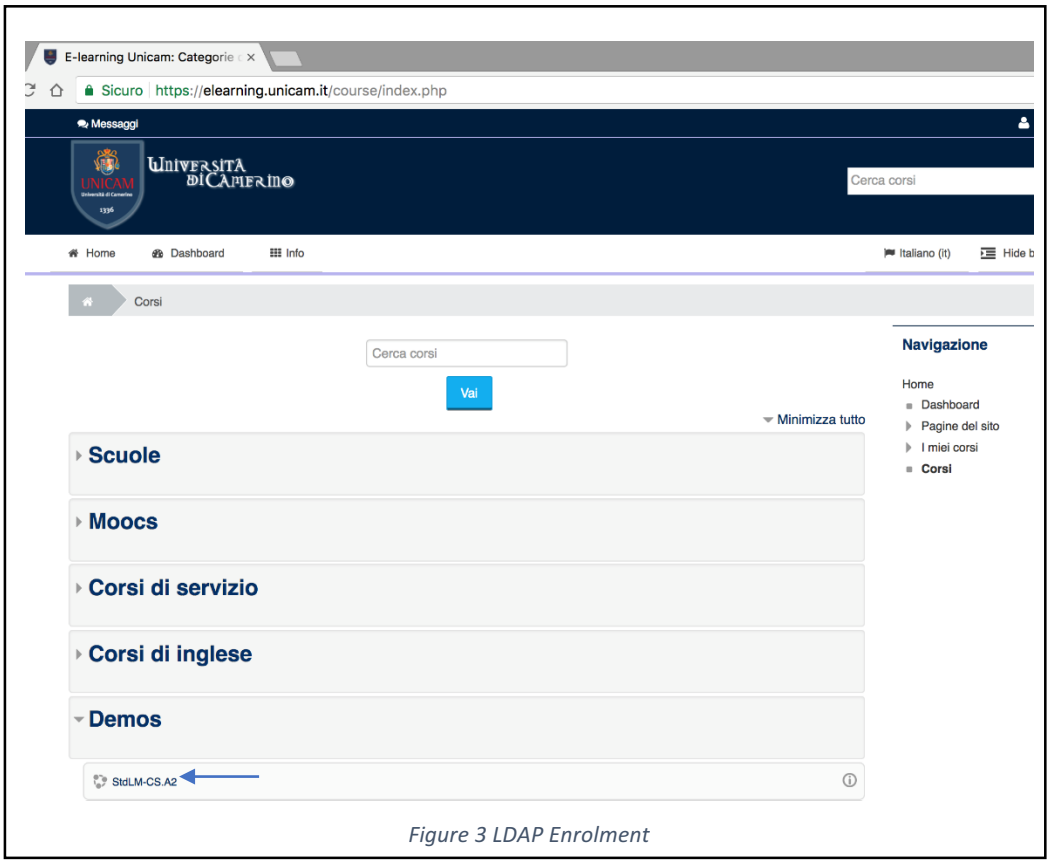

Figure 3 LDAP Enrolment

To make the user enrolment system via LDAP efficient, it would be necessary to reorganize the AD structure, in order to set up the groups with the names of the Moodle courses, in which the students should be enrolled through the Ldap.

For each course the administrator should provide an **Ou** with the list of the enabled roles.

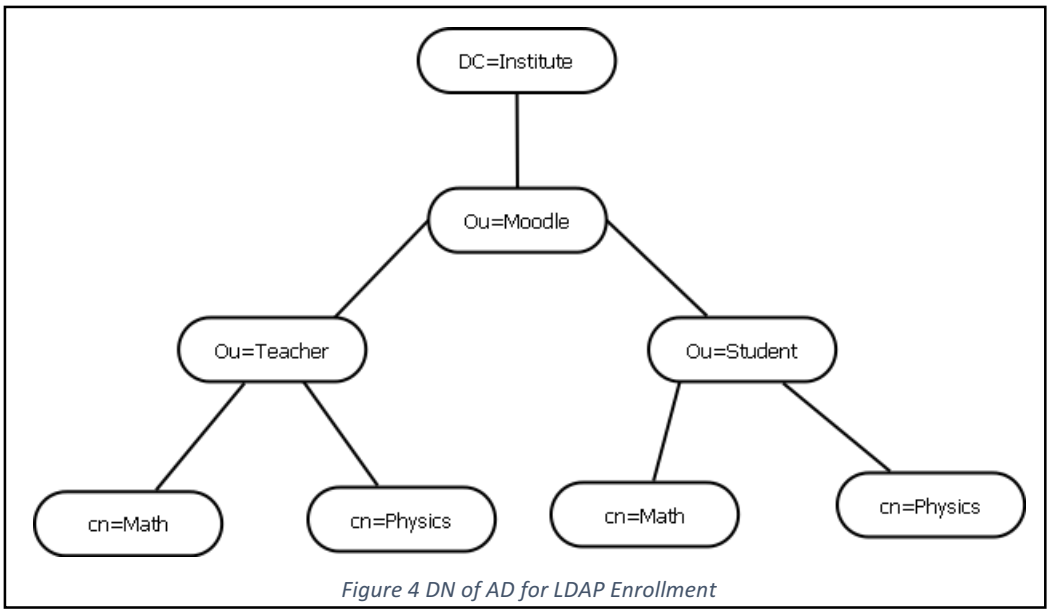

Figure 4 DN of AD for LDAP Enrollment

In this way two courses will be created:

- Math;
- Physics.

For each course, users will be enabled: teachers for the Teacher context and students for the Student context.

Their role will then be differentiated at the time of mapping the data of the Moodle plugin with the differences in platform privileges.

An example of a structure for the registration via LDAP in Moodle is below provided:

```
MSAD LDAP structure export
*****************************
** Structure of ou=moodle **
*****************************
dn: OU=moodle,DC=myInstitute,DC=it
changetype: add
objectClass: top
objectClass: organizationalUnit
ou: moodle
distinguishedName: OU=moodle,DC=myInstitute,DC=it
..some more ou details..
*********************************
** End of structure of ou=moodle **
*********************************

*****************************
** Structure of ou=Student **
*****************************
dn: OU=Student,OU=moodle,DC=myInstitute,DC=it
```

```
changetype: add
objectClass: top
objectClass: organizationalUnit
      ou: Student
distinguishedName: OU=Student,OU=moodle,DC=myInstitute,DC=it
..some more ou details..
************************************
** End of structure of ou=Student **
************************************

**************************************************************************
** Structure of cn=course01 (inside ou=Student) with one belonging user **
**************************************************************************
dn: CN=course01,OU=Student,OU=moodle,DC=myInstitute,DC=it
changetype: add
objectClass: top
objectClass: group
      cn: course01
member: CN=user01FistName.user01LastName,CN=Users,DC=myInstitute,DC=it
member: CN=user02FistName.user02LastName,CN=Users,DC=myInstitute,DC=it
distinguishedName: CN=course01,OU=Student,OU=moodle,DC=myInstitute,DC=it
..some more cn details..
******************************************************************************
** End of structure of cn=course01 (inside ou=Student) with one belonging user **
******************************************************************************

**************************************************************************
** Structure of cn=course02 (inside ou=Student) with one belonging user **
**************************************************************************
dn: CN=course02,OU=Student,OU=moodle,DC=myInstitute,DC=it
changetype: add
objectClass: top
objectClass: group 70 cn: demo
member: CN=user03FistName.user03LastName,CN=Users,DC=myInstitute,DC=it
distinguishedName: CN=course02,OU=Student,OU=moodle,DC=myInstitute,DC=it
..some more cn details..
******************************************************************************
** End of structure of cn=course02 (inside ou=Student) with one belonging user **
******************************************************************************

****************************
** Structure of ou=Teacher **
****************************
dn: OU=Teacher,OU=moodle,DC=myInstitute,DC=it
changetype: add
objectClass: top
```

```
objectClass: organizationalUnit
     ou: Teacher
distinguishedName: OU=Teacher,OU=moodle,DC=myInstitute,DC=it
..some more ou details..
**********************************
** End of structure of ou=Teacher **
**********************************

*************************************************************************
** Structure of cn=course01 (inside ou=Teacher) with one belonging user **
*************************************************************************
dn: CN=course01,OU=Teacher,OU=moodle,DC=myInstitute,DC=it
changetype: add
objectClass: top
objectClass: group
     cn: course01
member: CN=user03FistName.user03LastName,CN=Users,DC=myInstitute,DC=it
distinguishedName: CN=course01,OU=Teacher,OU=moodle,DC=myInstitute,DC=it
..some more cn details..
*************************************************************************
** End of structure of cn=course01 (inside ou=Teacher) with one belonging user **
*************************************************************************
```

Considering this model, it would be desirable to modify the structure of Active Directory, at least in the part concerning students, which could be grouped for Degree Courses. Moreover, it should also be reported as Ldap attributes to make it possible that they can be mapped locally in Moodle.

For an optimal configuration, it would also be necessary to create further sub-connections with other groups, identified with the CdL course names. In this way, it could be possible to have a system which is directly integrated with the data structure, eliminating the authorization steps. This would lead to a gain in efficiency, both from the point of view of the user (who does not have to do any registration procedure), and from the teacher / administrator of Moodle point of view (who has not to be worried about having to manage the registration procedure anymore). In addition, the grouping of students permits also to monitoring them more easily.

6.4 Integration of the Shibboleth Sp / IdP service

6.4.1 Service Provider Configuration

After having successfully verified the integration of the Moodle platform with the LDAP authentication service, the second implementation that it was decided to develop is the integration with the Shibboleth Service.

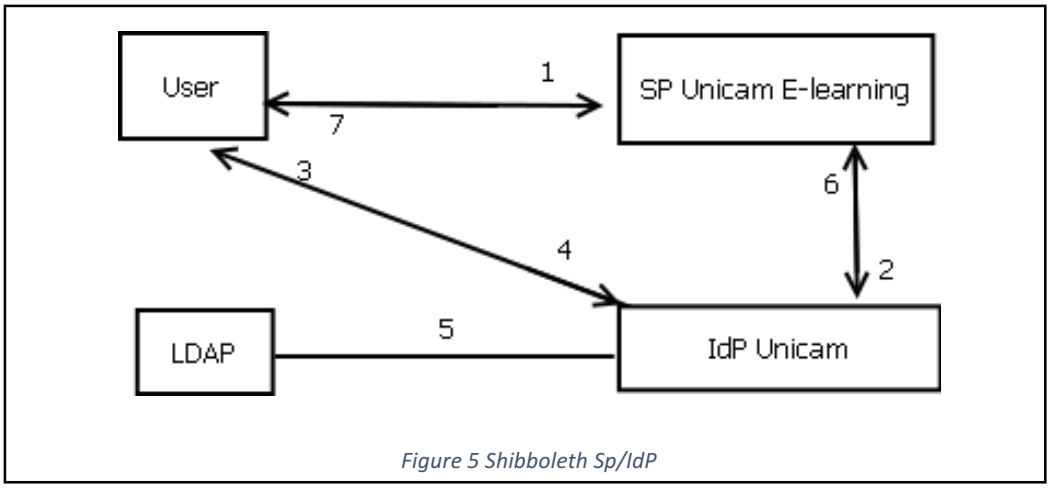

Figure 5 Shibboleth Sp/IdP

As it can be seen in Figure 12, Shibboleth's architecture includes:

- Identity Provider (IdP): the entity that can authenticate the user and provide additional information or attributes on his account;
- Service Provider (SP): the system in which the web resource to which the user requests is managed and which has the task of protecting the user through the access policy.

For this reason, it was necessary to implement an IdP and an SP, in order to test the SSO authentication with the Moodle platform.

In recent years, the University of Camerino has adopted some authentication solutions using Shibboleth, such as those used for the library portal.

In that case it was decided to adopt the SSO policy, as the portal had to be available to different universities; for this reason, the University has implemented its own Federated Identity Provider to allow authentication through the IDEM Federation.

Since a configured IdP was already present, it was decided to proceed with the configuration of a Service Provider, which should then be able to interface with the Unicam IdP.

In order not to make changes directly on the virtual machine hosting the Unicam IdP, it was decided to clone the machine and to import it internally into a small local internal test network, in order to carry out the tests without interfering with other systems.

So, in this first phase it was decided to try the SSO authentication system with Shibboleth, configuring the service internally without using the Federated system.
Regarding for the Service Provider, a dedicated Virtual Machine was created with Ubuntu[48] as Operative System, with the Moodle CMS installed.
After having installed Moodle, the first change made was to assign a temporary DNS to the machine, so that it can be recalled in the configuration steps of the SP.
To do so, it was necessary to modify the hosts file, present in the "/etc/apache2/hosts" directory, by inserting:

- `192.168.1.1.  eleprova.unicam.it`

After having changed and restarted both the network card and the apache server, it was necessary to install the SSL security certificate, so that the website could result secure and accessible via https.
In fact, Shibboleth needs the SP to have this certificate as a guarantee of safety.
For this reason, it was necessary to install the SSL module through the command:

- `sudo a2enmod ssl`

and then the openssl certificate was created through the command:

- `sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/apache.key -out /etc/apache2/ssl/apache.crt`

After having created the certificate and the openssl key, in order to configure Apache to the use of SSL, it was necessary to modify the ssl configuration file, which is located at the following path:

- `sudo nano /etc/apache2/sites-available/default-ssl.conf`

[48] Website reference: https://www.ubuntu.com

Entering the following values:

```
<IfModule mod_ssl.c>
    <VirtualHost _default_:443>
        ServerAdmin admin@example.com
        ServerName  eleprova.unicam.it
        DocumentRoot /var/www/html
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
        SSLEngine on
        SSLCertificateFile /etc/apache2/ssl/apache.crt
        SSLCertificateKeyFile /etc/apache2/ssl/apache.key
        <FilesMatch "\.(cgi|shtml|phtml|php)$">
                        SSLOptions +StdEnvVars
        </FilesMatch>
        <Directory /usr/lib/cgi-bin>
                        SSLOptions +StdEnvVars
        </Directory>
        BrowserMatch "MSIE [2-6]" \
                        nokeepalive ssl-unclean-shutdown \
                        downgrade-1.0 force-response-1.0
        BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown
    </VirtualHost>
</IfModule>
```

Once Apache is restarted, the dns will be available also with the https protocol.

After configuring the SSL certificate, it was necessary to continue with the installation of the Shibboleth Service Provider.

For this reason, we installed the Shibboleth module with the command:

- `sudo apt-get install libapache2-mod-shib2`

Since this test will be carried out internally without federating the SP and the IdP, we could not use the key and the certificate related to the Federation's Shibboleth service. However, also in this case, it was necessary to create them autonomously with the SSL service and insert them in the Shibboleth folder:

- `sudo openssl req -x509 -sha256 -nodes -days 3650 -newkey rsa:2048 -subj "/CN=$HOSTNAME" -keyout /etc/shibboleth/sp-key.pem -out /etc/shibboleth/sp-cert.pem`

These certificates must then be specified in the configuration file "shibboleth2.xml", which is fundamental for the functioning of the SSO.

A further fundamental element is certainly metadata, which Shibboleth uses to communicate information to trusted IdP, to Service Providers and also to distribute information about CAs (certification authorities). Metadata basically act as a key that allows to associate the Idp with the SP, validate the access and thus allow users who authenticate themselves in the IdP to access the resource.

So, it is necessary to download the IdP metadata and insert them in the folder located at "/var/run/shibboleth".

In the same way, the IdP will also have to download the SP metadata and insert them into its own folder.

At this point, it is necessary to modify the file "shibboleth2.xml", inserting (where specified) the address of the SP, in the part concerning the entityID, and the link related to the generation of metadata of the Id, within the section of the MetadataProvider.

```
<SPConfig xmlns="urn:mace:shibboleth:2.0:native:sp:config"
    xmlns:conf="urn:mace:shibboleth:2.0:native:sp:config"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    logger="syslog.logger" clockSkew="180">
    <ApplicationDefaults id="default" policyId="default"
        entityID="https://eleprova.unicam.it/shibboleth"
        REMOTE_USER="eppn persistent-id targeted-id"
        signing="false" encryption="false">
         <Sessions lifetime="28800" timeout="3600" checkAddress="true"
            handlerURL="/Shibboleth.sso" handlerSSL="true" cookieProps="https">
      <MetadataProvider type="XML" uri="https://idp.test.it/idp/shibboleth"
              backingFilePath="idp-test-metadata.xml" reloadInterval="7200">
         </MetadataProvider>
        <AttributeExtractor type="XML" validate="true" path=
              "attribute-map.xml"/>
        <AttributeResolver type="Query" subjectMatch="true"/>
        <AttributeFilter type="XML" validate="true" path=
              "attribute-policy.xml"/>
        <CredentialResolver type="File" key="sp-key.pem" certificate=
              "sp-cert.pem"/>
    </ApplicationDefaults>
    <SecurityPolicies>
        <Policy id="default" validate="false">
            <PolicyRule type="MessageFlow" checkReplay="true" expires="60"/>
            <PolicyRule type="Conditions">
                <PolicyRule type="Audience"/>
            </PolicyRule>
```

```
            <PolicyRule type="ClientCertAuth" errorFatal="true"/>
            <PolicyRule type="XMLSigning" errorFatal="true"/>
            <PolicyRule type="SimpleSigning" errorFatal="true"/>
        </Policy>
    </SecurityPolicies>
</SPConfig>
```

Now Shibboleth is configured.

At this point, the virtual machine was inserted into the same local network together with the test IdP; then, ports 80 and 443 were opened to allow communication with the IdP.

Before trying to log in, it was necessary to open the file "/etc/shibboleth/attribute-map.xml" and remove the block comment under "<! - Examples of LDAP-based attributes, uncomment to use these ... -> " to allow the Idp to map the Ldap attributes.

Then, accessing the link: "https://eleprova.unicam.it/Shibboleth.sso/Login", the login screen of the IdP will be opened, allowing users of the University of Camerino to access with AD's credentials.

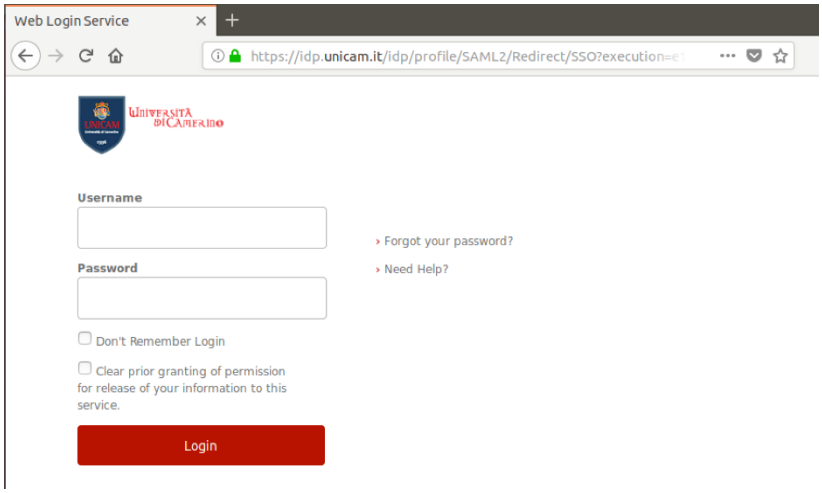

Figure 6 Screenshot Login IdP-test

To allow the access to Moodle, however, some settings must be configured correctly.

Initially it was necessary to modify the configuration of apache2 to support the authentication of Shibboleth, inserting the following code:

```
<IfModule mod_alias.c>
  Alias /moodle /var/www/html/moodle/
  <Directory /var/www/html/moodle/>
    Options Indexes MultiViews FollowSymLinks
    Require all granted
  </Directory>
  <Directory /var/www/html/moodle/auth/shibboleth/index.php>
    AuthType shibboleth
    ShibRequireSession On
    require valid-user
```

```
  </Directory>
</IfModule>
```

Later, it was necessary to configure the Shibboleth plugin in the Moodle platform and enable it.

Then, it is needed to access "Site administration" -> "Plugins" -> "Authentication" -> "Manage authentication".

In this plugin it is important to insert:

• Username: the name to be used for validating accounts (usually eppn);

• Identity Provider: Identity Provider list to be enabled for access to Moodle:

  - Data Mapping: Possibility of retrieving the values of users' attributes and inserting them as descriptive fields of users' accounts in the Moodle platform, by inserting the same syntax of LDAP attributes.

Once the plugin is configured and enabled, students can access the platform simply by entering their University credentials on the login page "https://eleprova.unicam.it/auth/shibboleth/index.php", select access with Shibboleth and enter their credentials, which will allow them to access the Moodle platform.

Once the system has been tested in the test machine, it was decided to follow the procedure also for the official platform "https://elearning.unicam.it/", as follows:

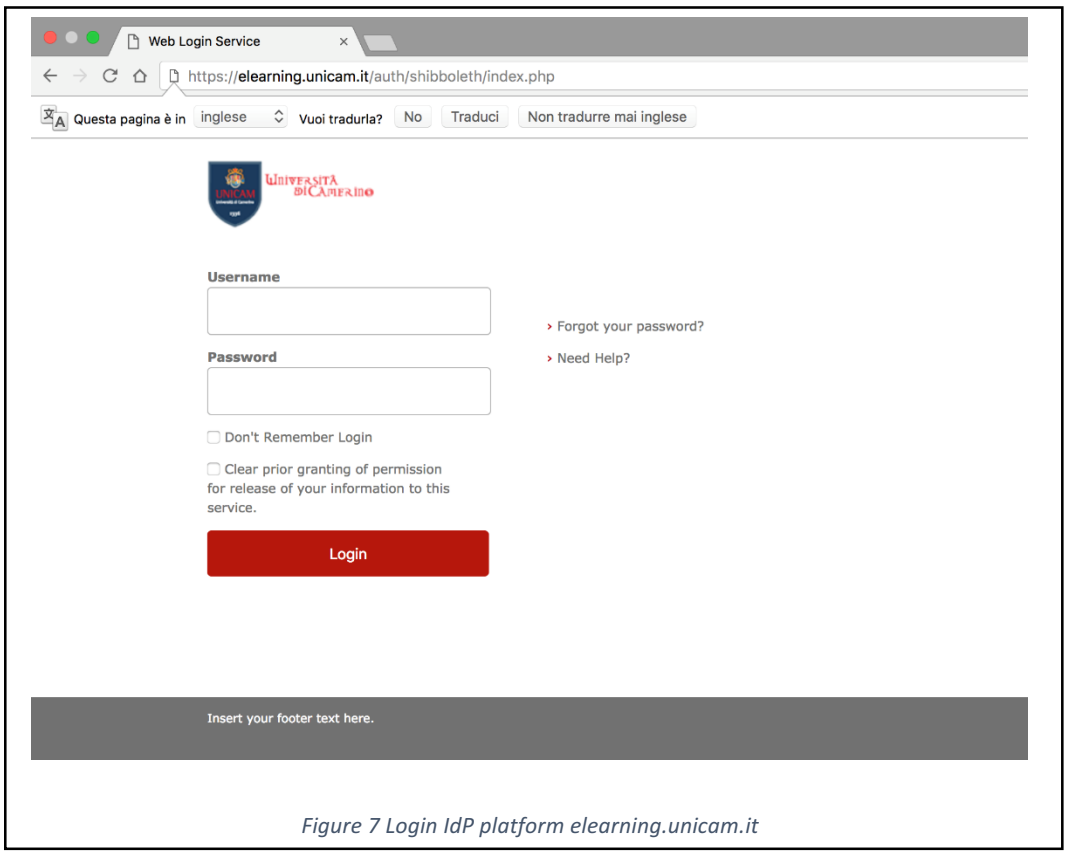

Figure 7 Login IdP platform elearning.unicam.it

6.4.2 Identity Provider configuration

Regarding the part related to the configuration of the Identity Provider, the procedure described below has been carried out.

Firstly, the metadata of the SP (which we called "unicam-moodle.xml") was taken and then inserted into the "/opt/shibboleth/metadata" folder, by executing the following command:

- `wget https://eleprova.unicam.it/Shibboleth.sso/Metadata -O /opt/shibboleth/metadata/unicam-moodle.xml`

Then, it was necessary to modify the "attribute-filter.xml" file, specifying the attributes of the users that are intended to be imported into the SP. Finally, it is necessary to modify the "metadata-provider.xml" configuration file, specifying the path where the metadata are taken. In this way, the IdP recognizes the validity of the SP, inserting the following code:

```
<MetadataProvider id="MoodleLocalMetadata"
  xsi:type="FilesystemMetadataProvider"
  metadataFile="%{idp.home}/metadata/unicam-moodle.xml"/>
</MetadataProvider>
```

Once these files have been modified successfully, and after restarting the server, it will be possible for the user to authenticate "locally" through the IdP.

6.5 Integration between the Shibboleth service and the Federated IdP

6.5.1 Service Provider Configuration

After having verified the functioning of the authentication system with Shibboleth in the local Unicam network, it has been considered the integration of the Shibboleth service with the Federated IdP.

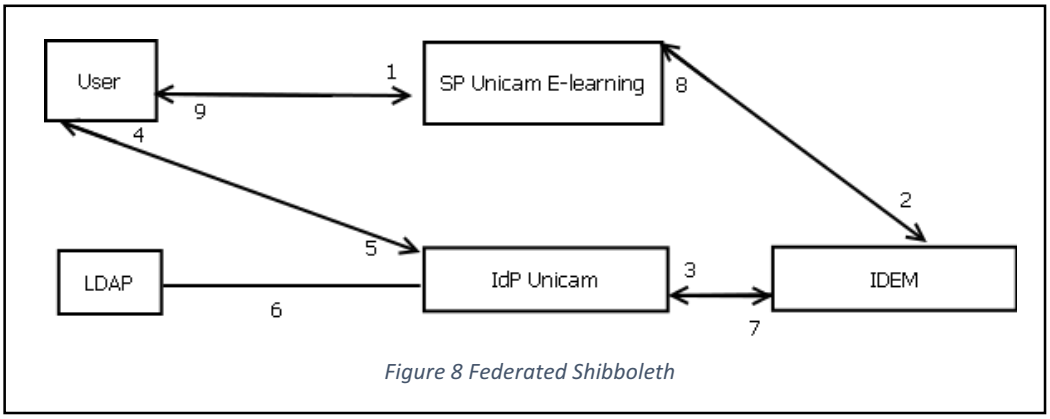

Figure 8 Federated Shibboleth

As it can be seen in Figure 18, the access is regulated by the Idem Federation, which acts as an intermediary between SP and IdP for Unicam users, but also regulates accesses to other authorized IdPs.

Therefore, this configuration also allows users of the organization to have the possibility have access to the resources of the University of Camerino; for this reason, it was not possible to "federate" a test machine, but it was decided to use directly the one that hosts the University platform, which is "elearning.unicam.it", installed on the Operative System called Centos 7.

This machine has already the ssl certificate, hence the Shibboleth module can be installed immediately:

- `yum install shibboleth.x86_64`

Afterwards, it was necessary to download the public key of the IDEM federation and save it in the directory "/etc/shibboleth". This key will be used by the federation to verify its authenticity:

- `wget https://www.idem.garr.it/documenti/doc_download/321-idem-metadata-signer-2019 -O /etc/shibboleth/idem_signer_2019.pem`

These lines, in addition to other information, will need to be specified in the "shibboleth2.xml" configuration file to allow it to function properly:

```
<SPConfig xmlns="urn:mace:shibboleth:2.0:native:sp:config"
    xmlns:conf="urn:mace:shibboleth:2.0:native:sp:config"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    logger="syslog.logger" clockSkew="180">
    <ApplicationDefaults id="default" policyId="default"
        entityID="https://elearning.unicam.it/shibboleth"
        REMOTE_USER="eppn persistent-id targeted-id"
        signing="false" encryption="false">
        <Sessions lifetime="28800" timeout="3600" checkAddress="true"
            handlerURL="/Shibboleth.sso" handlerSSL="true" cookieProps="https">
       <SSO discoveryProtocol="SAMLDS"
        discoveryURL="https://wayf.idem-test.garr.it/WAYF"> SAML2
        </SSO>
         <MetadataProvider type="XML"
          uri=" http://www.garr.it/idem-metadata/idem-test-metadata-sha256.xml"
              backingFilePath="idem-test-metadata-sha256.xml"
            reloadIn-terval="7200">
```

```
          <MetadataFilter type="Signature" certificate="idem_signer_2019.pem"/>
           <MetadataFilter type="EntityRoleWhiteList">
           <RetainedRole>md:IDPSSODescriptor</RetainedRole>

            <RetainedRole>md:AttributeAuthorityDescriptor</RetainedRole>

           </MetadataFilter>
          </MetadataProvider>

         </MetadataProvider>
        <AttributeExtractor type="XML" validate="true" path=
              "attribute-map.xml"/>

        <AttributeResolver type="Query" subjectMatch="true"/>
        <AttributeFilter type="XML" validate="true" path=
              "attribute-policy.xml"/>

        <CredentialResolver type="File" key="sp-key.pem" certificate=
              "sp-cert.pem"/>
    </ApplicationDefaults>
    <SecurityPolicies>

        <Policy id="default" validate="false">
            <PolicyRule type="MessageFlow" checkReplay="true" expires="60"/>
            <PolicyRule type="Conditions">
                <PolicyRule type="Audience"/>
            </PolicyRule>

            <PolicyRule type="ClientCertAuth" errorFatal="true"/>
            <PolicyRule type="XMLSigning" errorFatal="true"/>
            <PolicyRule type="SimpleSigning" errorFatal="true"/>
        </Policy>

    </SecurityPolicies>
</SPConfig>
```

After having properly modified the fields in the "shibboleth2.xml" file, it was necessary to download the metadata through the "https://elearning.unicam.it/Shibboleth.sso/Metadata" command and register it in the IDEM Entity Registry, which will allow the Federation to obtain the Metadata and other information related to the insertion of the SP into the Federation.

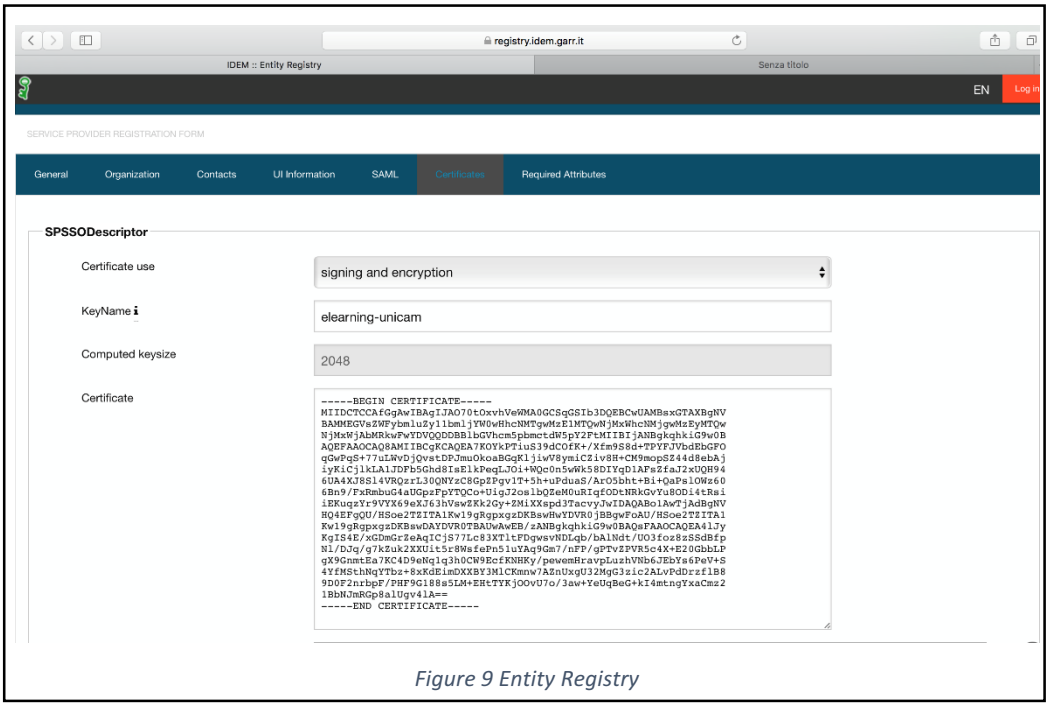

Figure 9 Entity Registry

Once this procedure has been carried out, the Federation will complete the procedure by enabling the SP to access the IdP in the Federation through the wayf service. This will allow the user to identify himself with the Unicam IdP (which is already federated) and have access to the resources present in the federation.

As it has already been done with the local Shibboleth service, it will be necessary to set up properly the Shibboleth plugin present in Moodle in order to allow the users to access.

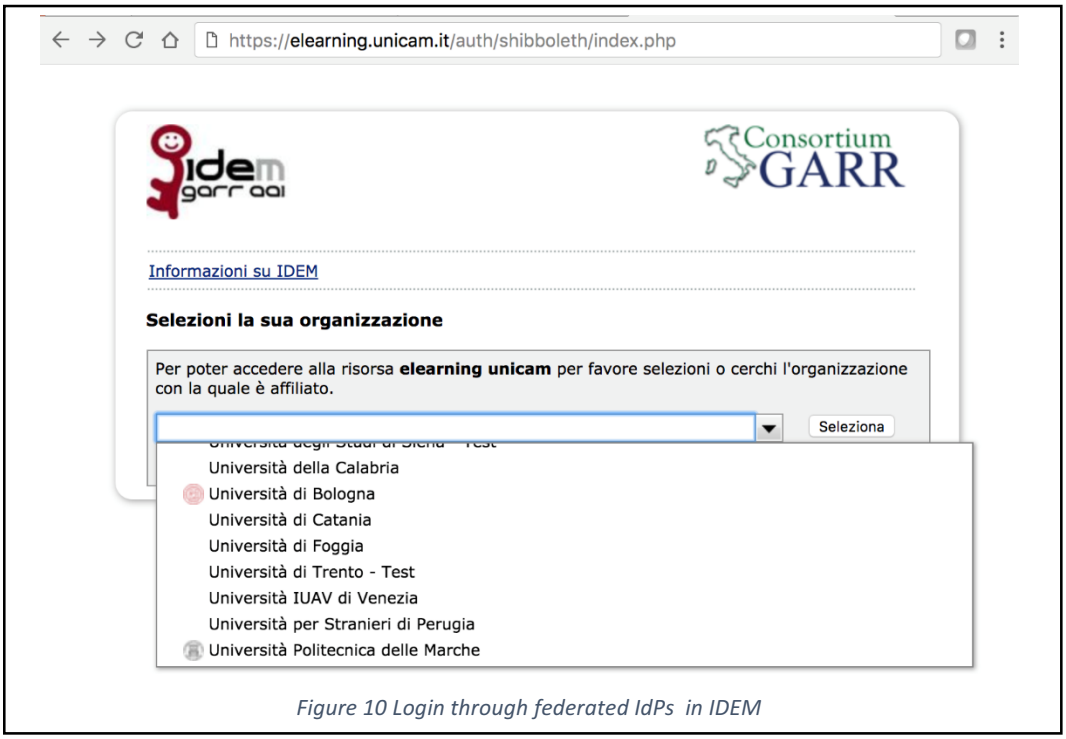

Figure 10 Login through federated IdPs in IDEM

6.5.2 Identity Provider configuration

Regarding the part related to the configuration of the Identity Provider, no change has been made since the service was already in IDEM federation. The procedure of introducing a federation IDP consists in submitting the metadata in the IDEM entity registry, compiling all the required textboxes. Then, it should be downloading the key "idem_signer_2019.pem" and the metadata to be loaded into the Shibboleth folder. Finally, "attribute-filter.xml" and "metadata-provider.xml" files must be appropriately modified according to the indications provided by the Idem website.

Chapter 7

Results

The future related to the Identity and Access Management field is undoubtedly that of being able to bring new methods of authentication within its own infrastructure. This will bring considerable flexibility in accessing the resource, delivering benefit for the entire organization. In the case study, following the analysis of several SSO services, it was possible to implement three different types of authentication:

- LDAP;
- Shibboleth;
- Shibboleth (Federated).

In all three cases, it was possible to reach the prefixed objective, that is, to allow students, teachers and all the Unicam staff to be able to authenticate themselves to the Moodle resource with the University credentials.
The Moodle plugins related to the services used have made it possible to map the attributes that allowed the collection of data related to the digital identity of the users, as well as the import and the update within the platform.

LDAP

Advantages:

- ease of access to the service by the user, by accessing with the university credentials;
- easy installation of the Php-Ldap module and configuration of the information, as well as the mapping of the attributes for the access to the Moodle platform;
- large online documentation with specific guides to better configure the service;
- integrate in an autonomous and rapid way, by getting data directly from AD and without the need for certificates, metadata or requests to external services.

Disadvantages:

- difficulties in configuring the LDAP Server, as in Unicam more than one server has been configured for students and teachers, which are difficult to access by the Bind user;
- data security limit since validation certificates are not required;
- this service allows authentication only in University services and cannot be used for access to resources in federation.

Shibboleth

Advantages:

- Increased security thanks to the SSL service configuration and Metadata exchange;
- Allows all Unicam students, teachers and staff to access through the same IdP, with the domains "studenti.unicam.it" and "unicam.it";
- It is easy for user to access the service, by University credentials.

Disadvantages:

- Data security limit, since no metadata signing public keys are required, which allows effective validation;
- This service allows authentication only for the University services, and cannot be used to access to resources in federation;
- Lacking documentation regarding the installation and configuration of the service.

Federated Shibboleth

Advantages:

- Greater security thanks to the configuration of the SSL service certified by a Certificate Authority, to the exchange of Metadata with the federation and to the public key (idem_signer_2019.pem), which allows the federation to recognize and validate the metadata of the SP resource;
- Allows all Unicam students, teachers and staff to access through the same IdP, with the domains "studenti.unicam.it" and "unicam.it", not only to the resources of Unicam but also to all the resources present in the IDEM federation;
- Possibility that other users of federated universities can access their e-learning resources with their University's credentials.

Disadvantages:

- Lacking documentation regarding the installation and configuration of the service;
- Obligation to keep the resources opened also to potential access by other users of federated organizations.

7.1 Comparison between different authentication systems

	DISADVANTAGES	ADVANTAGES
LDAP	• Difficulty of access for all users • Security • Access for internal resources only	• Easy to use by the user • Easy to install • Large documentation • No external services
Shibboleth (local)	• Lack of public validation key • Access for internal resources only • Lacking documentation	• Increased security • Possibility of access for all Unicam users
Shibboleth (federated)	• Obligation to keep resources open to potential access by other users of federated organizations • Lacking documentation	• Increased security • Possibility of access for all Unicam users, not only by internal Unicam resources but also by Federated resources • Possibility of access for users of other Federated organizations • Public validation key • Support from the Federation

Table 3

The results obtained by using these three services that permit to have access to the Unicam E-learning platform can be identified in the following peculiarities, as it allows:

- solve digital identities security issues and personal data confidentiality (thanks to the validation of information through the use of certificates);
- avoid the proliferation of passwords for users;
- avoid multiplication of credentials;
- avoid muddled enrolment in a service, which can discourage a user from using a certain service;
- possibility to have always updated the fields of digital identities;

- give users the opportunity to take advantage not only of internal resources, but also resources of other organizations that are placed in the federation;
- reduction of stress due to the management of numerous identities by the Service Provider;
- possibility to expand users on the platform thanks to the access of users of federated organization;
- reduce the administrative burden for the management of identities and credentials within the e-learning platform.

Conclusions

After several attempts with the three authentication services, during the implementation, configuring and testing correctly differently according to the procedures related to each of the services, the choice fell in using the service Shibboleth in IDEM Federation.

Although the simple Ldap service would have been sufficient for the current needs of the e-learning platform, it was decided to develop a Federated service as it involves the elimination of the number of credentials, the reduction of the stress (due to the management of numerous identities by the Service Provider) and allows a more flexible organization in case of policies to be managed. Furthermore, the possibility for users of the University of Camerino to have access to other federated resources can represent an additional service to those already offered by the University. At the same time, the possibility of allowing other users to access the e-learning platform can also be seen as an opportunity, for instance, by facilitating the exchange of information between different universities in a safe way, while minimizing the heavy costs of authentication.

Another reason why this service has been chosen, is the fact that the Federations (in this case IDEM) offer technical support regarding the implementation and configuration of SP and IdP as resources to be Federated.

This aspect should not be underestimated as Shibboleth is an inflexible and not intuitive system, which has led, during the configuration phase of the service, to a series of unexpected events and errors, which are difficult to solve even due to the lacking documentation related to the new version of Shibboleth.

This choice has been taken due to the will to develop, in recent years, a unique digital identity in Italy, with which it could be possible to have access to different federations. Currently, this reality is widely spread in the academic field, but more and more projects have been developed with an applicative interoperability and a unique identity. Among these, we can certainly find the SPID project (a Public System for the management of the Digital Identity of citizens and businesses), which represents an Italian Authentication and Authorization system with which the public administrations and private individuals can allow access to their services on the Internet. This, in the future, could be widespread also in the academic field. In this field, obviously, the IDEM federation would remain irreplaceable for the management of specific roles and information related to training and research. Furthermore, the institutionalization of the SPID system could lead to the transformation of the IDEM Federation into a source of qualified attributes, especially in terms of "Attribute Authority", for the access to both national

(IDEM) and European (EduGain) academic resources. For this reason, it is important to develop this service to allow citizens to have access to resources efficiently from their home or their office. In this way, they can have the information they need with lower costs in terms of bureaucracy and inconvenience.

The purpose of this work was, therefore, to clarify the development of an SSO access through different types of service, highlighting all the limits and potential in order to give to those who want to develop an SSO authentication service for their organization, a choice suitable for their situation, minimizing configuration problems.

Bibliography and Sitography

[1] Mayuri D., Sridevi K.
Identity and access management: concept, challenges, solutions, International Journal of Latest Trends in Engineering and Technology, 2017

[2] Metz C.
AAA protocols: Authentication, authorization, and accounting for the Internet, IEEE Computer Society, IEEE Computer Society, Institute of Electrical and Electronics Engineers, 1999

[3] Calabritto V.
Identity and access management e federazione, Scires IT, 2012
http://caspur-ciberpublishing.it/index.php/scires-it/article/viewFile/9576/8937

[4] IDEM Federation
L'infrastruttura di Autenticazione e Autorizzazione della rete GARR
http://progetto-idem.idem.garr.it/

[5] Abelson H., Lessig L.
Digital Identity in Cyberspace, white paper submitted for 6.805/Law of Cyberspace: Social Protocols, 1998

[6] Windlew P. J.
Digital Identity: Unmasking Identity Management Architecture, IMA, O'Reilly, 2005

[7] Tanlongo F, Tumini S.
Il modello dell'Accesso Federato per favorire processi di integrazione e diffusione della conoscenza in E-learning, Proceedings della Multiconferenza EM&M ITALIA, Genova University Press, 2016

[8] Oasis

Security Assertion Markup Language (SAML) V2.0 Technical Overview,

https://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf

[9] Information technology, Open systems interconnection

The Directory: Public-key and attribute certificate frameworks

https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-X.509-200811-S!!PDF-E&type=items

[10] Radha V. , Reddy Hitha D.

A Survey on Single Sign-On Techniques, Elsevier, 2011

[11] Policelli J.

Active Directory Domain Services 2008 How-To, Network World, 2009

[12] Moodle

Moodle Statistics
http://moodle.net/stats/

Appendix

1. Moodle Service Provider

This appendix describes the procedure for configuring the Unicam Moodle platform as SP for SSO authentication.

1.1 Requirements

In this case, it has been opted for an installation on the Ubuntu Linux 17.10 Operative System. In order to configure the Moodle cms, it was necessary to install the following packages:

- Apache2;
- Mysql;
- Php.

This has been possible by using the following commands:

- `sudo apt-get install apache2`
- `sudo apt-get install mysql-server mysql-client`
- `sudo apt-get install php`

1.2 Moodle installation

After installing the server lamp, it is necessary to install Moodle 3.4.
It is therefore necessary to download the "moodle-latest-34.tgz" package and then unpack it, enabling the permissions.

- `sudo su -`
- `cd /usr/local/src`
- `wget https://download.moodle.org/download.php/direct/stable31/moodle-latest-34.tgz`
- `tar xzf moodle-latest-34.tgz`
- `mv moodle/ /var/www/html/`
- `chown -R root:root /var/www/html/moodle ; chown www-data /var/www/html/moodle`

Then, connecting to the link "http://localhost/moodle", the cms installation is refined, entering the database data and then adding the general information that will then appear on the platform.

1.3 Create a SSL Certificate

After having installed Moodle, it is then necessary to install an SSL certificate, which allows the Moodle application to transmit information securely by using the https protocol.
For greater security, it is desirable to use an SSL certificate of a Certificate Authority, or, alternatively, it can also be generated independently.
To do this, it is necessary to activate the SSL module:

- `sudo a2enmod ssl`
- `sudo service apache2 restart`

Then, it is necessary to create the self-signed certificate:

- `sudo mkdir /etc/apache2/ssl`
- `sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/apache.key -out /etc/apache2/ssl/apache.crt`

Afterwards, it should be entered the information about the organization.
The next step concerns the configuration of apache, modifying the default-ss.conf file:

- `sudo nano /etc/apache2/sites-available/default-ssl.conf`

inserting the following commands:

```
<IfModule mod_ssl.c>
    <VirtualHost _default_:443>
        ServerAdmin admin@example.com
        ServerName your_domain.com
        ServerAlias www.your_domain.com
        DocumentRoot /var/www/html
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
        SSLEngine on
        SSLCertificateFile /etc/apache2/ssl/apache.crt
        SSLCertificateKeyFile /etc/apache2/ssl/apache.key
        <FilesMatch "\.(cgi|shtml|phtml|php)$">
                        SSLOptions +StdEnvVars
        </FilesMatch>
        <Directory /usr/lib/cgi-bin>
                        SSLOptions +StdEnvVars
        </Directory>
        BrowserMatch "MSIE [2-6]" \
                        nokeepalive ssl-unclean-shutdown \
                        downgrade-1.0 force-response-1.0
```

```
        BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown
    </VirtualHost>
</IfModule>
```

Finally, after saving, restart the service and apache:

- `sudo a2ensite default-ssl.conf`
- `sudo service apache2 restart`

Now, the SSL protocol is configured, and can it be accessed on the platform by specifying the protocol:

- `https://server_domain_name_or_IP`

1.4 LDAP configuration

Regarding the LDAP configuration, it is necessary to enable the extensions in the php.ini file, by uncomment the following lines:

- `extension=mysql.so`
- `extension=gd.so`
- `extension=ldap.so`

Once this change has been made, it is sufficient to enable the Moodle plugin, going to "Site administration"-> "Plugins"-> "Authentication" -> "Manage authentication" and fill in the following fields:

LDAP server setting	
Host URL	Ldap://<school server's>:389
Version	3
Bind settings	
Distinguished Name	<DN of the authentication account>
Password	<Password of the authentication account>
User lookup setting	
User type	MS ActiveDirectory
Contexts	cn=users,ou=<school>,dc=<domain>,dc=<domin>,
Search subcontexts	yes
User Attribute	samaccountname

Data mapping	
First name	givenName
Surname	Sn
Email address	mailPrimaryAddress
ID number	distinguishedName

Then the users can log in with them LDAP account.

1.5 Shibboleth installation

In order to use Shibboleth, it is necessary to install the shibboleth module through the command and set the activation of the daemon on start-up:

- `sudo apt-get install libapache2-mod-shib2`
- `sudo chmod +x /etc/init.d/shibd`
- `sudo update-rc.d shibd defaults`

The Shibboleth daemon was installed and configured during the "run at startup".
At this point you need to enter the following commands to install the key "/cert" files and restart the Shibboleth service:

- `sudo openssl req -x509 -sha256 -nodes -days 3650 -newkey rsa:2048 -subj "/CN=$HOSTNAME" -keyout /etc/shibboleth/sp-key.pem -out /etc/shibboleth/sp-cert.pem`
- `sudo service shibd restart`

Now, it is necessary to download the metadata that contains information useful for the IdP on:

- `/usr/bin/shib-metagen -c sp-cert.pem -h localhost -e https://localhost/shibboleth > sp-test-metadata.xml`

and then save the IDP metadata:

- `wget https://idp.example.it/idp/shibboleth -O /var/run/shibboleth/idp-test-metadata.xml`

At this point, before proceeding to the configuration of the file "shibboleth2.xml", it is necessary to "protect" the directory "moodle/auth/shibboleth/index.php" with Shibboleth.
To do this, it is necessary to log in to "/etc/apache2/sites-available/moodle.conf" and enter the following code:

```
<IfModule mod_alias.c>
  Alias /moodle /var/www/html/moodle/
  <Directory /var/www/html/moodle/>
    Options Indexes MultiViews FollowSymLinks
    Order deny,allow
    Allow from all
  </Directory>
  <Directory /var/www/html/moodle/auth/shibboleth/index.php>
    AuthType shibboleth
    ShibRequireSession On
    require valid-user
  </Directory>
</IfModule>
```

1.6 Shibboleth Non-Federated SSO configuration

Regarding the system that uses Shibboleth as a SSO service, through direct access between SP and IDP without the federation intermediation, it is sufficient to modify the configuration file "shibboleth2.xml" in this way:

```
<SPConfig xmlns="urn:mace:shibboleth:2.0:native:sp:config"
    xmlns:conf="urn:mace:shibboleth:2.0:native:sp:config"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    clockSkew="180">
<ApplicationDefaults entityID="https://sp.example.it/shibboleth"
                         REMOTE_USER="eppn persistent-id targeted-id"
cipherSuites="ECDHE+AESGCM:ECDHE:!aNULL:!eNULL:!LOW:!EXPORT:!RC4:!SHA:!SSLv2">
    <Sessions lifetime="28800" timeout="3600" relayState="ss:mem"
                  checkAddress="true" handlerURL="/Shibboleth.sso"
                  handlerSSL="true" cookieProps="https">
    <SSO entityID="https://idp.example.it/idp/shibboleth"
        discoveryProtocol="SAMLDS" discoveryURL="https://ds.example.org/DS/WAYF">
              SAML2 SAML1
            </SSO>
            <Logout>SAML2 Local</Logout>
            <Handler type="MetadataGenerator" Location="/Metadata"
             signing="false"/>
            <Handler type="Status" Location="/Status" acl="127.0.0.1 ::1"/>
            <Handler type="DiscoveryFeed" Location="/DiscoFeed"/>
        </Sessions>
        <Errors supportContact=example@root.com helpLocation="/about.html"
            styleSheet="/shibboleth-sp/main.css"/>
        <MetadataProvider type="XML" validate="true"
              uri="https://idp.example.it/idp/shibboleth"
             backingFilePath="idp-test-metadata.xml" reloadInterval="7200">
        </MetadataProvider>
        <AttributeExtractor type="XML" validate="true" reloadChanges="false"
         path="attribute-map.xml"/>
        <AttributeResolver type="Query" subjectMatch="true"/>
        <AttributeFilter type="XML" validate="true" path="attribute-policy.xml"/>
      <CredentialResolver type="File" key="sp-key.pem" certificate="sp-cert.pem"/>
    </ApplicationDefaults>
    <SecurityPolicyProvider type="XML" validate="true" path="security-policy.xml"/>
        <ProtocolProvider type="XML" validate="true" reloadChanges="false"
         path="protocols.xml"/>
</SPConfig>
```

At this point, after having properly configured the IdP, having imported the metadata of the SP in the metadata folder, set the attributes in the file "attribute-filter.xml" and insert in the file "metadata-provider.xml" the following code:

```
<MetadataProvider id="LocalMetadata"
    xsi:type="FilesystemMetadataProvider"
    metadataFile="path/sp-metadata.xml"/>
```

1.7 Shibboleth Federated SSO configuration

Regarding the federation of an SP system in IDEM, after submitting the Service Provider metadata in the Entity Registry (https://registry.idem.garr.it/rr3/) of IDEM, it is necessary to withdraw the "idem_signer_2019.pem" and change its permissions:

- `wget https://www.idem.garr.it/documenti/doc_download/321-idem-metadata-signer-2019 -O /etc/shibboleth/idem_signer_2019.pem`
- `chmod 444 /etc/shibboleth/idem_signer_2019.pem`

At this point, it is sufficient to properly modify the configuration file "shibboleth2.xml", as follows:

```
<SPConfig xmlns="urn:mace:shibboleth:2.0:native:sp:config"
    xmlns:conf="urn:mace:shibboleth:2.0:native:sp:config"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    clockSkew="180">
<ApplicationDefaults entityID="https://sp.example.it/shibboleth"
                     REMOTE_USER="eppn persistent-id targeted-id"
cipherSuites="ECDHE+AESGCM:ECDHE:!aNULL:!eNULL:!LOW:!EXPORT:!RC4:!SHA:!SSLv2">
    <Sessions lifetime="28800" timeout="3600" relayState="ss:mem"
                checkAddress="true" handlerURL="/Shibboleth.sso"
                handlerSSL="true" cookieProps="https">
    <SSO discoveryProtocol="SAMLDS" discoveryURL="https://wayf.idem.garr.it/WAYF">
              SAML2 SAML1
            </SSO>
            <Logout>SAML2 Local</Logout>
            <Handler type="MetadataGenerator" Location="/Metadata"
             signing="false"/>
            <Handler type="Status" Location="/Status" acl="127.0.0.1 ::1"/>
            <Handler type="DiscoveryFeed" Location="/DiscoFeed"/>
        </Sessions>
        <Errors supportContact=example@root.com helpLocation="/about.html"
            styleSheet="/shibboleth-sp/main.css"/>
        <MetadataProvider type="XML" validate="true"
              uri="https://www.garr.it/idem-metadata/idem-metadata-sha256.xml"
              backingFilePath="idp-metadata-sha256.xml" reloadInterval="7200">
        <MetadataFilter type=”Signature” certificate=”idem_signer_2019.pem”/>
        </MetadataProvider>
        <AttributeExtractor type="XML" validate="true" reloadChanges="false"
         path="attribute-map.xml"/>
```

```
        <AttributeResolver type="Query" subjectMatch="true"/>
        <AttributeFilter type="XML" validate="true" path="attribute-policy.xml"/>
      <CredentialResolver type="File" key="sp-key.pem" certificate="sp-cert.pem"/>
    </ApplicationDefaults>
    <SecurityPolicyProvider type="XML" validate="true" path="security-policy.xml"/>
        <ProtocolProvider type="XML" validate="true" reloadChanges="false"
         path="protocols.xml"/>
</SPConfig>
```

Finally, the IdP must have been properly configured, by importing the SP metadata into the metadata folder, setting the attributes in the file "attribute-filter.xml" and properly modifying the file "metadata-provider.xml".