European Commission

Communications Networks, Content and Technology

European Commission Directorate General

DG CONNECT

European Union’s FP7 Programme
DG Connect

Directorate C: Excellence in Science
Unit C1: e-Infrastructure

elcira

Europe Latin America
Collaborative e-Infrastructure
for Research Activities

Deliverable D2.6

Follow up Recommendations and Final Progress Report

SEVENTH FRAMEWORK PROGRAMME

A project of the Seventh Framework Programme (FP7)

European Commission

This project is funded by the European Commission

RedCLARA
+ Red + Ciencia

A project implemented by RedCLARA

Periodical Progress Report

ELCIRA Deliverable: D2.6 - Follow up Recommendations and

Final Progress Report

Document Full Name	**Follow up Recommendations and Final Progress Report**
Date	**October, 2014**
Activity	**WP2 (Coordinated Actions for AAI between EU and LA)**
Lead Partner	**RNP**
Document status	**Final**
Classification Attribute	**Public**
Document link	

Abstract: This deliverable will describe the achievements of the project aligned to the roadmap developed. The managerial aspects of the activities performed and the foreseen follow-up actions will be discussed with specific regard to the Consortium status and perspectives.

A project of the Seventh Framework Programme (FP7)

This project is funded by the European Commission

A project implemented by RedCLARA

COPYRIGHT NOTICE

ELCIRA (Europe Latin America Collaborative e-Infrastructure for Research Activities – Call (part) identifier: FP7-INFRASTRUCTURES-2012-1 – Project number: 313180) is a project co-funded by the European Commission within the Seventh Framework Programme (FP7), Infrastructures (DG Connect, Directorate C: Excellence in Science, Unit C1: e-Infrastructure). ELCIRA began on 1st June 2012 and will run for 24 months.

For more information on ELCIRA, its partners and contributors please see http://elcira.redclara.net

The information contained in this document represents the views of the copyright holders as of the date such views were published.

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED BY THE COPYRIGHT HOLDERS “AS IT IS” AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE MEMBERS OF THE ELCIRA COLLABORATION, INCLUDING THE COPYRIGHT HOLDERS, OR THE EUROPEAN COMMISSION BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THE INFORMATION CONTAINED IN THIS DOCUMENT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A project of the Seventh Framework Programme (FP7)

This project is funded by the European Commission

A project implemented by RedCLARA

DELIVERABLE ROUTE

	Name	Member/Activity	Date	Responsible
From	Leandro Guimarães	WP2	18/10/2013	RNP
From	Lalla Maria Laura Mantovani	WP2	20/10/2013	GARR
Revised by	Antônio Carlos Fernandes Nunes	WP2	22/10/2013	RNP
Approved by	Florencio Utreras	RedCLARA/CEO	30/10/2014	RedCLARA

TABLE OF CONTENTS

1.- INTRODUCTION

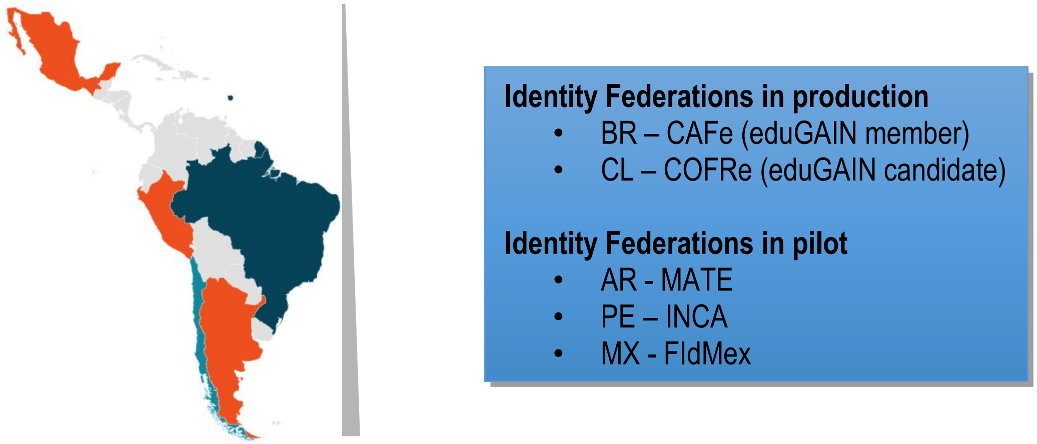
Since the beginning of ELCIRA Project, we were sure that this work package was one that would be one of the most complicated to implement, due the complexity of the theme Authentication and. Authorization Infrastructure (AAI - Identity Management) and the lack of knowledge and skills in the technical team in the target NRENs.

During the planning phase, we have discovered some heterogeneous environments. There was an NREN that was very interested in the topic, but they lacked the necessary manpower resources to work on it, at the other extreme, there was a NREN that planned to launch an Identity Federation, take it into production and study how to join the eduGAIN interfederation service.

Archived the project goals we need to foresee the future of AAI in Latin America. We are using as basis some work developed by RNP in this subject, adding some work developed by TERENA, GARR and other NRENs.

2.- INITIAL SCENARIO

When we have started this work package, in June 2012, we have defined the milestone to start the job; the MoU signed by the NREN's responsible. We thought that doing this we would raise the attention of the project to the member's directors while give the necessary power and support to technical staff to implement a federation. The Figure 1 shows the initial scenario in the beginning of 2013:



Identity Federations in production

- BR – CAFe (eduGAIN member)
- CL – COFRe (eduGAIN candidate)

Identity Federations in pilot

- AR - MATE
- PE – INCA
- MX - FIdMex

Figure 1: Identity Federation in January 2013.

As described in “Deliverable D2.1 - Roadmap for the delivery and deployment of National AAIs in Latin America”, during the kickoff meeting in Lima, Peru in July, 5th, 2012, three countries were identified as federation startups: Chile, Peru and Colombia. After this date, Mexico and Argentina also expressed interest.

While the project was developing, it was noticed that the gap in AAI knowledge was bigger than the project team have assumed, so it was needed a close approach to all countries identified as potentials federations in Latin America.

Doing that, the project team has found some alternative ways to help emerging NRENs to raise their federations in a short period, causing a quick response to the question of why a federation is important to them. Another point of interest was “selling” some services in order to maximize the reliability of federation.

3.- SUCCESSFUL ACTIONS DURING THE PROJECT

At the beginning of the project, it was ministered an Identity Federation and eduroam for NRENs Training in Cartagena de Indias, Colombia. Dedicated to engineers and technicians members of the National Academic Network and their associated institutions, the course was held on July 11 and 12 in the UTECNAR dependencies in Cartagena de Indias. In the course, taught by Camila Santos (RNP) and Jose Luis Quiroz Arroyo (INICTEL), participated 22 engineers from Argentina, Colombia, Ecuador and Uruguay who had the opportunity to learn about the theoretical fundaments and the necessary actions to create a federation and the required procedures for having eduroam in the assistants countries (if it is not yet implemented).

Figure 2: Participants of Identity Federation and eduroam for NRENs Training in Cartagena de Indias, Colombia.

It was very important to all NRENs responsible to be introduced in the AAI subject. It was presented, in a high-level view, the requirements, effort and steps to raise a federation, using Brazilian Federation case as basis. The results of this action were more understanding of AAI aspects and effort and more commitment of everyone, as they were introduced in the subject, so they couldn't ignore the benefits of building a federation anymore.

A second workshop on Identity Management in ELCIRA Project was successfully held in Cancun, Mexico, on May 28-29, 2014. This workshop – jointly organized by the CHAIN-REDS and ELCIRA EU - funded projects under the aegis of the European Commission (DG CONNECT) – was organized in co-location with the TICAL2014 Conference.

The workshop gave participants a comprehensive overview of Identity Management, with a focus on the existing solutions based on the Federations of Identity Providers promoted by the eduGAIN initiative.

The workshop presented the opportunities offered by such approach in terms of easiness of access to distributed resources and services. The presentations addressed the technical aspects of the implementation of Identity Providers and Service providers. Specific talks of the workshop also showed examples and current state of the art activities with an introduction to the Persistent Identifier (PID) system.

The workshop gathered around 40 participants, mainly high-level stakeholders from Universities, Computing Centers and National Research and Education Networks from all Latin America. It paved the way to the expansion of Identity Providers systems and Identity Federations in the continent. The strong interest shown by the participants points to a sustainable development of the CHAIN-REDS solutions in Latin America.

Figure 3: Participants workshop on Identity Management, in Cancun, Mexico.

Another successful action was and online course, although the number of participants was less than we had expected, at least two federations (MINGA and MATE) was born there.

In the last month of ELCIRA Project, the Caribbean Knowledge and Learning Network, CKLN is collaborating with the Trinidad and Tobago NREN, TTRENT, to facilitate a Federation Services Workshop at the Ministry of Tertiary Education and Skills Training in Port of Spain, Trinidad and Tobago from 27th – 30th October 2014. The workshop will not only train the participants, but will also use the opportunity to actually implement the Federated Identity system on a server in Trinidad, which will function as a root server for the Caribbean R&E Federation in the future.

4.- FUTURE OF AAI IN LATIN AMERICA

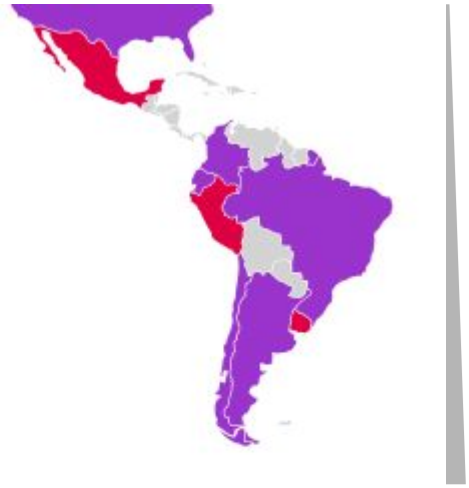
Latin America has reached the status of AAI region connected, as we have CAFe, COFRe, MINGA, COLFIRE and MATE in production state and some more federations in pilot basis. ELCIRA project has approached Latin America countries to share experience and knowledge in AAI subject, causing an acceleration in absorbing information, although all information was already available in some forums (TERENA, GÈANT, Chain-Reds, etc.) it is quite different when a neighbor helps.

Another project target was that all federations must be eduGAIN oriented, in order to speed up the adhesion process.

As it was described on *AAA-Study-Report-0907*: “eduGAIN has been designed to address inter-federation, to enable users from one federation to access services provided by another federation. This approach requires an infrastructure that supports the exchange of information between different entities (often located in different countries), a legal framework (such as a contractual agreement) and Data Protection Directives that ensure that the users’ personal data are securely handled.

At the end of the ELCIRA Project there are three federations in eduGAIN (CAFe-BR,COFRe-CL, MING–EC), with more two in eduGAIN pilot (COLFIRE-CO and MATE-AR), furthermore there are INCA-PE, RAUid-UR and FENIX-MX under the way. It shows that under ELCIRA project effort, it was created a very strong knowledge in AAI, helping to raise more Federations and IdPs (Identity Providers), remaining a gap in SPs (Service Providers), what was expected in the project. With the closure of the project, it is very important to continue this discussion under task forces in Latin America and worldwide in order to make eduGAIN stronger and evolve new technologies and governance in AAI.

The Figure 2 shows REFEDS (Research and Education Federations) map in the October of 2014, available at: https://refeds.org/resources/index.html.



Identity Federations in production

- BR - CAFe
- CL – COFRe
- EC – MINGA
- COL – COLFIRE
- AR – MATE

Identity Federations in pilot

- UR – RAUid
- PE – INCA
- MX - FENIX

Figure 4: Identity Federation in October 2014.

To help to create more federations, even when the NREN has no infrastructure, GARR has developed an *"IdPAsAService"*. The system offered is not simply limited to IdPs, but also implements an LDAP service and its management interface and is configured as an Identity Management System.

To obtain eduGAIN compliance and enable end users to access eduGAIN services, it was created metadata entities and identities attributes that follow the eduGAIN metadata profile and the eduGAIN attribute profile. Pointing on attributes, all eduGAIN recommended attributes are implemented in the LDAP directory and the web form for the IdP administrators helps in filling their values. The controlled vocabulary on Affiliation and OrganizationType is also implemented. The IdP implemented complies with the eduGAIN specifications and is technically ready to be registered in the inter-federation.

The IdP is also compliant with REFEDS discovery guide so the IdP's metadata are enriched with names and logos to be ready for smart discovery services. Moreover the IDP login page is designed for co-branding with the SP, taking a lot of user interface information from the SP metadata and displaying them on the IdP login page.

Another issue that RNP is working in last 2 year is how efficient is CAFe. In other words, Brazilian NREN is developing probes and tools to measure how many users are using a particular SP, doing that, CAFe administrators can strength relationship with those top SPs to provide feedbacks from local users and help them to improve even more the services. This effort will be shared with the community to help Federations admins to show all benefits of building and maintain an Identity Federation.

Regarding Governance and policy, it is very important to create a REFEDS chapter in Latin America. By mature level and similar concerns of Latin America Federations, IdPs and SPs, it might help to increase the relevance in this subject in the region. It was noticed during the last two TICAL (Network of Information and Communication Technologies Directors from Latin American Universities) conferences, when there were Identity Management Workshops that the demand was incredibly high, and the results doing that were spectacular. RNP until the end of this year will drive this subject to REFEDS.

APPENDIX1: FEDERATION AS A SERVICE – ACTIVITIES

Work Package/Activity: WP2/Federation as a Service

Authors A. Biancini (GARR), M. Malavolti (GARR),

Contributors M.L. Mantovani (GARR), D. Cresti (GARR)

1. INTRODUCTION

This document will present all the main activities shared between GARR and the Elcira project in the field of automating the provisioning of a virtual appliance to support the start phase of an Identity Federation. The goal of these activities is to lower the technological effort to install all the required software to operate effectively an Identity Federation.
During its experience of operating the Italian IDEM identity federation, GARR has had the opportunity to identify which were the processes that needed to be effectively implemented to operate a Federation in general. So now GARR is in a good position to propose a technical solution to ease and standardize the start-up phase of an Identity Federation. Leveraging the experiences on the automation project “IdP in the Cloud”, GARR started to develop a solution on the same technologies to automate the creation of a Federation.

Within this project, some automation code and scripts have been developed by GARR. In the Elcira collaboration, GARR shared the main ideas and concept upon which such components have been realized. In this way, the Latin American community could provide a good use case to test the project realized and verify the real benefits.

The rest of this document will present the conceptual framework for these activities. Then the Federation as a Service (FaaS) project will be presented briefly and the last part of the document will be a how-to guide that will summarize the activities to be performed to use the automation code developed in the FaaS activity.

2. FEDERATION AS A SERVICE, THE APPROACH

Managing effectively an Identity Federation is a task that may be overwhelming for a new National Research and Education Network (NREN) to start. GARR has been operating the Italian Identity Federation for the Academic and Research community for the last 5 years. Due to its experience, then, GARR has aimed at developing a solution that could ease the creation and management of a new Identity Federation by eliminating the entry barriers. This chapter will describe the general idea and the approach that has been followed to realize this goal.

2.1 WHAT AN IDENTITY FEDERATION IS

To understand which are the main activities and processes that need to be implemented to operate effectively an Identity Federation, we need to agree about what an Identity Federation is about. From a very general perspective, we can say that an Identity Federation is a collection of organizations that agree to interoperate under certain rules to manage user identities.

This definition contains some simple yet powerful concepts. On one side, it underlines that, within a Federation, different organizations cooperate in managing identities by taking care of their users and services. On the other side, the definition given stresses the concept of an agreement that must be realized among the different parts.

When we implement a Federation, each organization externalizes some tasks to other subjects. In the case of Identity Federations, the organization providing a service will delegate to other subjects the management of user authentication and will trust the users authenticated by any subject of the federation. For this delegation to be executed correctly, it is important that each organization has a reasonable trust in all the others. This trust is built by the Identity Federation: the Federation, in fact, is responsible for building global trust within the different organizations.

2.2 WHAT NEEDS TO BE DONE TO OPERATE A FEDERATION

As we have stated in the previous section, the mechanism of trust building is the key task of a Federation. To reach this goal, a Federation administrator has to pay strong attention to two different aspects:

1. the technology used to guarantee that each subject is accountable for the operation he can perform and that nobody can cheat or disrupt the Federation in a malicious way;

2. the processes implemented to manage the circle of trust and guarantee security to all participants in the Federation.

Regarding the technologies involved in implementing an Identity Federation, in the Research and Academic communities, the SAML protocol is used to express security (authentication/authorization) assertions. This protocol guarantees the needed security level and is implemented in different software packages, some of which are open source and freely available to be used and eventually extended.

Once agreed on the technology, it is important to share common processes that all entities have to follow in order to participate in the Federation. For this reason, a Federation administrator has to design and share processes describing how the main operations can be performed by the different organizations.

3. THE KEY PROCESSES AND COMPONENTS

The processes involved in managing an Identity Federation will be described in this section. The processes are presented from the perspective of the main subject responsible for the activity.

The participants in the Identity Federation have to define and manage two different procedures:

1. The procedure to create and manage an Identity Provider (IdP): if the participant wants to provide identities for its users to access federated services, it has to implement an IdP and register it in the Federation. The creation, installation and management of the software components to implement the IdP have to be performed by the participant.

2. The procedure to create and manage a Service Provider (SP): if the participant wants to provide a service to all (or part) of the federated users, it has to implement an SP on its application and register this SP in the Federation. The creation, installation and management of the SP software have to be performed by the participant.

The Federation administrators, on the other hand, have to implement all the processes to guarantee the realization of the circle of trust at the root of the Identity Federation. The processes involved are the following:

1. Registering an entity (IdP or SP) in the Federation: this process takes the entity metadata from an IdP or an SP and registers them to the Federation level. This registration process is composed of three steps:

    a. Validating entity metadata information toward Federation policies: this activity is related to a formal validation of the entity metadata information. Entity metadata for IdPs and SPs must contain some descriptive and informative information, as requested by the Federation policy. This step concerns the verification of the formal compliance to this policy.

    b. Performing all security controls: this step concerns the technical verification of the entity metadata. These controls take into consideration all the security aspects about the certificates used for securing communication and verifies the compliance to general technical practices.

    c. Guiding the participants in the implementation of an Identity Management policy: this step concerns the diffusion of a “federated culture” among the participants. This activity is intended to disseminate and promote good practices and a growing knowledge about identity federations and about the best way to satisfy user needs in these environments.

2. Signing and distributing the federation metadata: after the verification of each metadata entity, these must be signed with the federation certificate to guarantee the authenticity of the issuer of these federation metadata. After signing, the federation metadata are placed in a publicly available place (for example a URL over the Internet) where all participants can retrieve them.

3. Providing accessory services (like information pages or Discovery Service): an Identity manager must also manage some accessory services that may be of help to the Federation users. These services may include informative pages to guide the users in understanding the power and advantages of using an Identity Federation. The Discovery Service represents one other accessory service; it is the service that gives the user the possibility to easily identify the organization which is able to authenticate him while accessing a federated service.

3.3 REGISTERING AN ENTITY IN THE FEDERATION

The main process that involves a Federation administrator is the process of verifying, signing and publicly exposing the entity metadata from the different participants. Let's examine this process: this paragraph will show the process GARR implemented within the IDEM Italian Identity Federation.

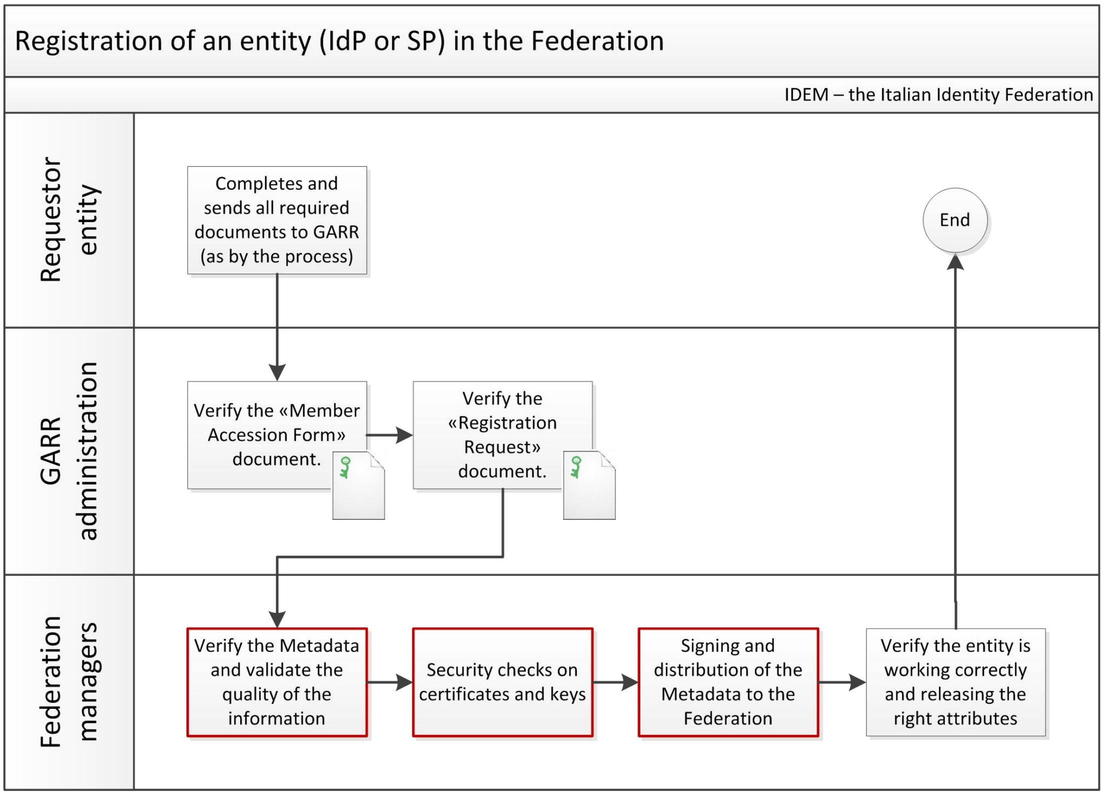

Registration of an entity (IdP or SP) in the Federation
IDEM – the Italian Identity Federation
Requestor entity
GARR administration
Federation managers
Completes and sends all required documents to GARR (as by the process)
End
Verify the «Member Accession Form» document.
Verify the «Registration Request» document.
Verify the Metadata and validate the quality of the information
Security checks on certificates and keys
Signing and distribution of the Metadata to the Federation
Verify the entity is working correctly and releasing the right attributes

Fig. 1: Process to register an entity (IdP or SP) in the Federation.

The steps involved in this process, as shown in the figure above, are the following:

1. Document compilation: the requestor has to compile some documents and send them to the Federation administrator to request participation in the Federation.
2. Verification of the "Member Accession Form" document: in the case of the Italian IDEM Federation, one of the documents sent to register an entity (IdP or SP) is a "Member Accession Form", which is verified and accepted into this step.
3. Verification of the "Registration Request" document: in the case of the Italian IDEM Federation, one of the documents sent to register an entity (IdP or SP) is a "Registration Request", which is verified and accepted into this step.
4. Verification and validation of the entity metadata: this step is responsible for executing the verification and validation of the entity metadata, in particular concerning the informative and human readable parts.
5. Security check of the entity metadata: this step is responsible for verifying the entity's metadata from a technical point of view. This step is also where the security checks are

performed on the certificates used by the entity to encrypt messages with other entities in the Federation.

6. Signing of the federation metadata and distribution at a Federation level: this step is responsible for signing the federation metadata and exposing them on a public link over the Internet.
7. Verification of the entity implementation: this step involves some human tests that are performed to verify that the newly registered entity works as defined in the entity metadata and interacts correctly with the federation.

3.4 THE TOOLS INVOLVED

The process for registering an entity in the Federation could be partially automated. In particular there are different software tools that can be used to guide and sustain the activities in this process.

In the picture showing the workflow, the red boxes contain activities that could be implemented with different software tools. The activities we are talking about are the following:

- Verification and validation of the entity metadata
- Security check of the entity metadata.
- Signing of the federation metadata and distribution at a Federation level.

These three activities could span two integrated software tools:

1. Resource Registry: to validate metadata information (and disseminate awareness). The tool we could use for this purpose is Jagger (http://jagger.heanet.ie), developed by HEAnet to manage the Edugate multiparty SAML federation. The main features of this tool are:

   a. Synchronise SAML metadata from another federation.
   b. Create and manage a federation.
   c. Create a single circle of trust containing metadata of all entities that your organization participates in via multiple federations.
   d. GUI to manage the attribute policy of identity providers based on the Shibboleth SAML implementation.
   e. Filter the RequestedAttribute's of a SAML service provider to allow an IdP release attributes to such providers based on a policy set in the Jagger GUI.
   f. Create and edit metadata of individual entities.
   g. Notification subsystem with subscription options.

2. Metadata Aggregator: to verify all the security aspects bound to certificates and to sign the federation metadata for distribution to the community and others. The Metadata Aggregator provides a command line tool and REST-based web service to support publishers and consumers of metadata. In both cases, the product supports reading in metadata from multiple sources and then verifying, filtering, and transforming the data. The command line takes this data and produces one or more output files which may be consumed by other products supporting SAML Metadata, whilst the web service provides a mechanism for dynamically querying the processed data. The key features of this tool are:

    a. Support for consuming metadata from the local file system or HTTP URL.
    b. Ability to verify digitally signed metadata using multiple trust models.
    c. Support filtering of information such as SAML roles, contact persons, or more generically specified elements.
    d. Support for SAML Registration and Publication Information specification
    e. Provides web service (REST-based) for querying a pool of consumed and processed metadata.

Together with these tools, the FaaS appliance is configured to also support another component which is very important in operating a federation. This component is a Discovery Service; it defines a generic browser-based protocol by which a centralized discovery service implemented independently of a given service provider can provide a requesting service provider with the unique identifier of an identity provider that can authenticate a principal.

As a service provider, connected to a large number of Identity Providers, you would need to ask the user in advance of the authentication process to select its Identity Provider. DiscoJuice is super simple to deploy at a Service Provider. It is as easy as copy and pasting a small javascript reference into the HTML source of your application.

The main features of this software are:

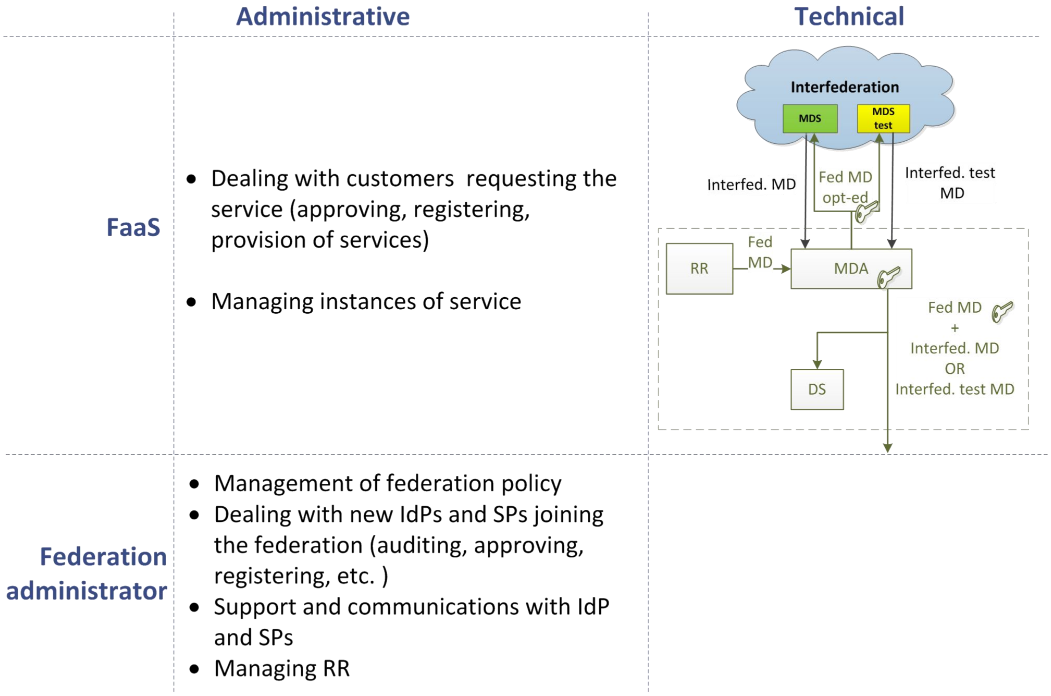
- Local Memory (cookie)
- Remote Memory (DiscoReadWrite protocol + IdP Discovery)
- Javascript only, super simple to deploy
- DiscoJuiceJSON compact UI-focused Metadata format (MDUI friendly)
- Presents logos, searchable keywords, name, descr, country...
- Automatic discovery of country
- HTML5 Geo-location API
- Graceful non-javascript fallback

- Inline incremental search
- Flexible integration API using JS callbacks.
- Protocol agnostics, demoed with alternative protocols.
- Multi-lingual provider list (from metadata) and UI is translated into 15 languages

4. THE FAAS SCENARIO

To simplify the adoption of the tools to implement entity registration in a Federation, GARR decided to extend the technologies and techniques used within the project “IdP in the Cloud” to obtain similar results.

In particular GARR decided to automate the installation and configuration of a FaaS appliance that could contain all the relevant software installed and configured to be used for implementing the processes at a Federation level. In this way, the technical entry barrier is completely lowered and a federation administrator can start operating a newborn federation very quickly and effectively.



	Administrative	Technical
FaaS	• Dealing with customers requesting the service (approving, registering, provision of services) • Managing instances of service	
Federation administrator	• Management of federation policy • Dealing with new IdPs and SPs joining the federation (auditing, approving, registering, etc.) • Support and communications with IdP and SPs • Managing RR	

Fig. 2: The FaaS perimeter (from an administrative and a technical point of view[1]) (imagine da GÉANT).

[1] Image from GÉANT “Milestone MS83 (DS5.4.1): Federation as a Service - Market Analysis and Pilot Service Definition”

The schema in the picture above, shows the general idea behind FaaS, we can divide the activities between the Faas project and the Federation administrator that leverages this project:

- FaaS: will take care of all technical aspects and of the installation and configuration of a FaaS appliance (in the dashed rectangle on the picture). FaaS will also be responsible for dealing with customers, requesting the service, and with the maintenance and administration of the service instances.

- Federation administrator: will not have to deal with any technical activity. The administrator will only have to manage federation policies, manage the process of entity registration and provide support and communication with participants' organizations.

4.1 THE PUPPET AUTOMATION

The problem of deploying and managing the FaaS appliance has been solved by leveraging the experience of the "IdP in the cloud" project. The FaaS appliance, from a technical point of view, is an Appliance as a Service, i.e. a virtual machine in the GARR cloud that includes a set of preconfigured services.

The virtual machine is based on Linux Ubuntu Server 12.04 LTS distribution and is provided, together with the use of Puppet recipes, with:

- an Apache2 Server with PHP support;
- the Jagger RR3 installed and configured to support a newborn Federation;
- the Metadata Aggregator installed and configured to integrate with eduGain metadata;
- a Discovery Service, based on DiscoJuice, to be used as a central service for the Federation;
- a MySQL Server: used by the IdP;
- an IPTABLES firewall: already configured to respond properly to the needs of the IdP;
- a rsyslog daemon: for IDP logs centralized management;
- a Nagios Server: for monitoring and alerting;
- a Collectd daemon: to collect system statistics;
- a certificate (needed by the Apache 2 server (HTTPS)) verification system and expiry notification;
- A data backup system.

5. HOW TO ADMINISTER A FEDERATION WITH THE FAAS APPLIANCE

Once the Puppet recipes have correctly installed the appliance, you should have generated two or more federations that can be managed by the Jagger Entity Registry.
From our point of view, the best suggestion that we can give you is to think of having at least three different federations:

1) **A Test Federation**: this will be useful to test the new entities provided by the different organizations that will want join into your Production Federation.
2) **A Production Federation**: This will be the real federation that contains all your Identity and Service Providers.
3) **A Federation for eduGAIN**: This will contain all the entities, IdP and SP, that will also want to join into eduGAIN interfederation.

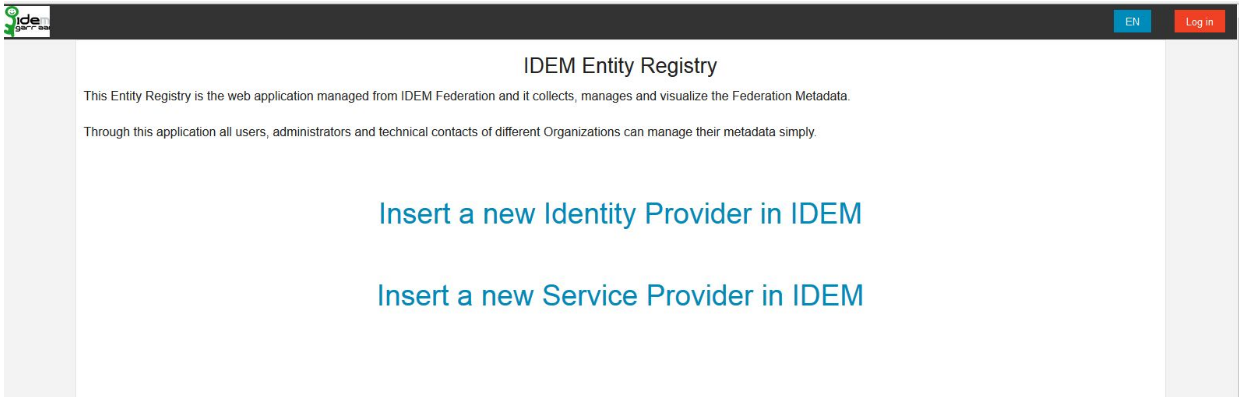
This document will present you **a couple of** tutorials **obtained from the** IDEM GARR AAI experience**. These tutorials** will help you with the management of an Identity Federation **leveraging the FaaS** appliance.

5.5 HOW TO ADD A NEW ENTITY TO THE PRODUCTION FEDERATION

In the following sentences, we will briefly describe the steps that an Organization must take to register its entity (IdP or SP) into the Production Federation (e.g. IDEM). We take, for example, an organization that wants to add an Identity Provider.
The same steps, with few changes, can be followed for a Service Provider registration.

5.5.1 Add the new entity to the Test Federation – User side

1) Open the Metadata Registry login page and follow the link for the registration of an Identity Provider. E.g.: "**Insert a new Identity Provider in IDEM**"



IDEM Entity Registry

This Entity Registry is the web application managed from IDEM Federation and it collects, manages and visualize the Federation Metadata.

Through this application all users, administrators and technical contacts of different Organizations can manage their metadata simply.

Insert a new Identity Provider in IDEM

Insert a new Service Provider in IDEM

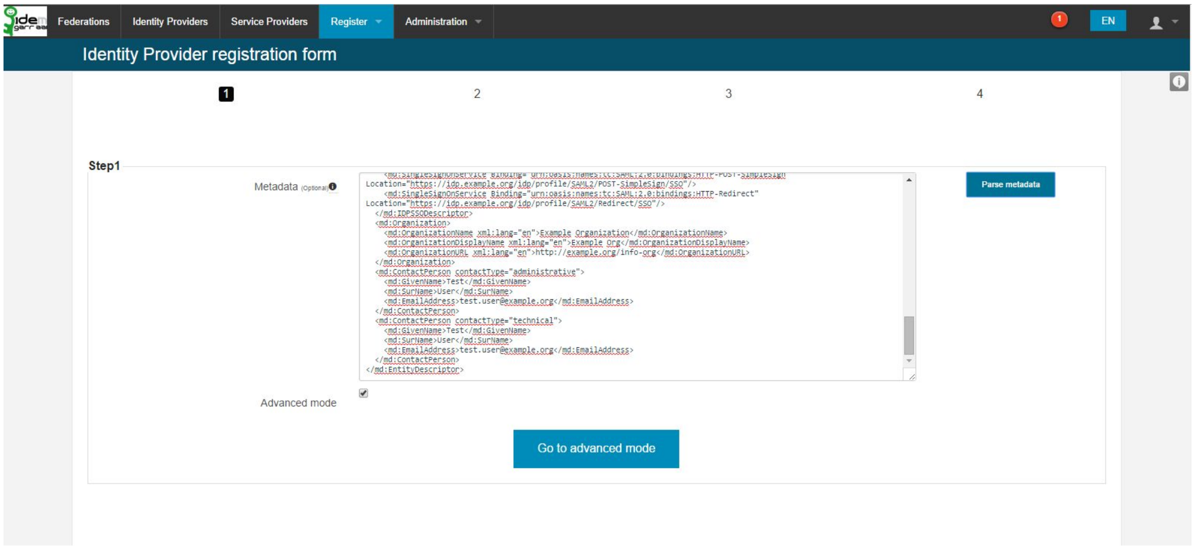
Fig. 3: Jagger Login page.

2) Paste the IdP metadata into the textbox that appears and choose the process to register your entity:

   a) **The simplest way**: this saves, on the Jagger instance, only the basic information needed for the exchange of the assertions between entities of the federation. This way can be useful for those organizations that provide their entity metadata without change to those created by their SAML "framework" for default.
   For this method you must paste your IdP metadata into the textbox, click on the "*Parse metadata*" small blue button and then, if you receive the "**Success**" message from the screen, click on the big blue button "*Next*" and follow the 4 steps to submit your entity to the Test Federation.

   b) **The best way**: this saves, on the Jagger instance, all kind of information about the entity contained into its metadata. *(This is the way followed by this tutorial)*
   This method should be followed by those users that provide complete entity metadata to Jagger and that want to provide additional information about their entity with its metadata.

   For this method you must paste your IdP metadata into the textbox, control that the checkbox near the sentence "**Advanced mode**" is selected and then press on the "**Go to advanced mode**" button to start the registration of the entity to the Test Federation:

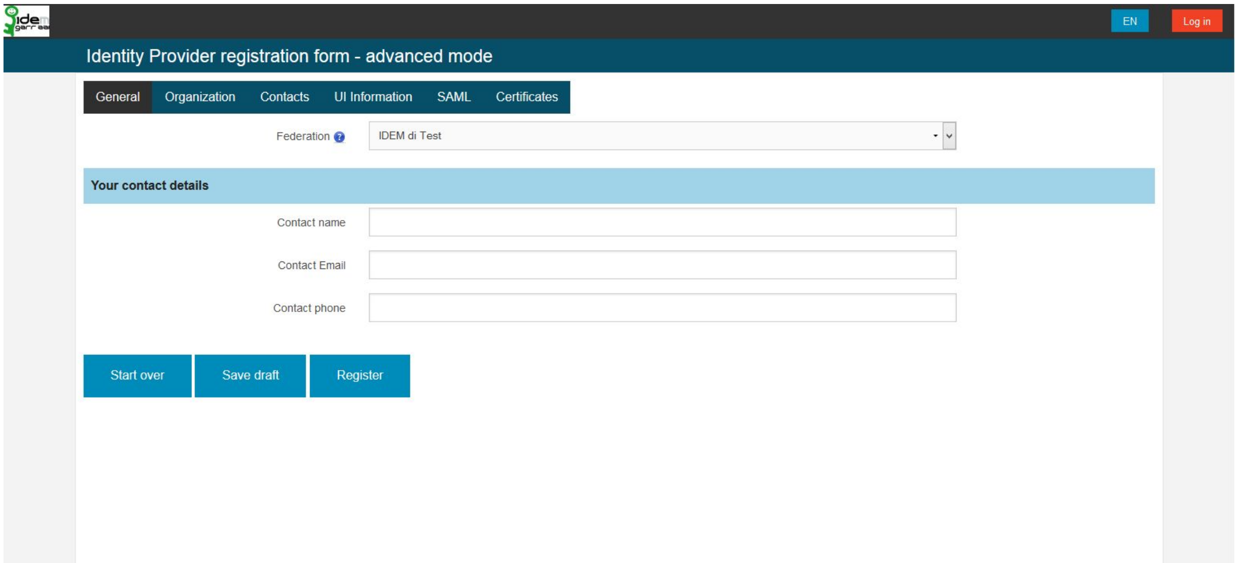


```
    <md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST-SimpleSign"
Location="https://idp.example.org/idp/profile/SAML2/POST-SimpleSign/SSO"/>
    <md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://idp.example.org/idp/profile/SAML2/Redirect/SSO"/>
  </md:IDPSSODescriptor>
  <md:Organization>
    <md:OrganizationName xml:lang="en">Example Organization</md:OrganizationName>
    <md:OrganizationDisplayName xml:lang="en">Example Org</md:OrganizationDisplayName>
    <md:OrganizationURL xml:lang="en">http://example.org/info-org</md:OrganizationURL>
  </md:Organization>
  <md:ContactPerson contactType="administrative">
    <md:GivenName>Test</md:GivenName>
    <md:SurName>User</md:SurName>
    <md:EmailAddress>test.user@example.org</md:EmailAddress>
  </md:ContactPerson>
  <md:ContactPerson contactType="technical">
    <md:GivenName>Test</md:GivenName>
    <md:SurName>User</md:SurName>
    <md:EmailAddress>test.user@example.org</md:EmailAddress>
  </md:ContactPerson>
</md:EntityDescriptor>
```

Advanced mode ☑

Fig. 4: IdP registration form page.

3) Complete all the sections, described below, that appear on the next page and click on “**Register**” button once you have finished:

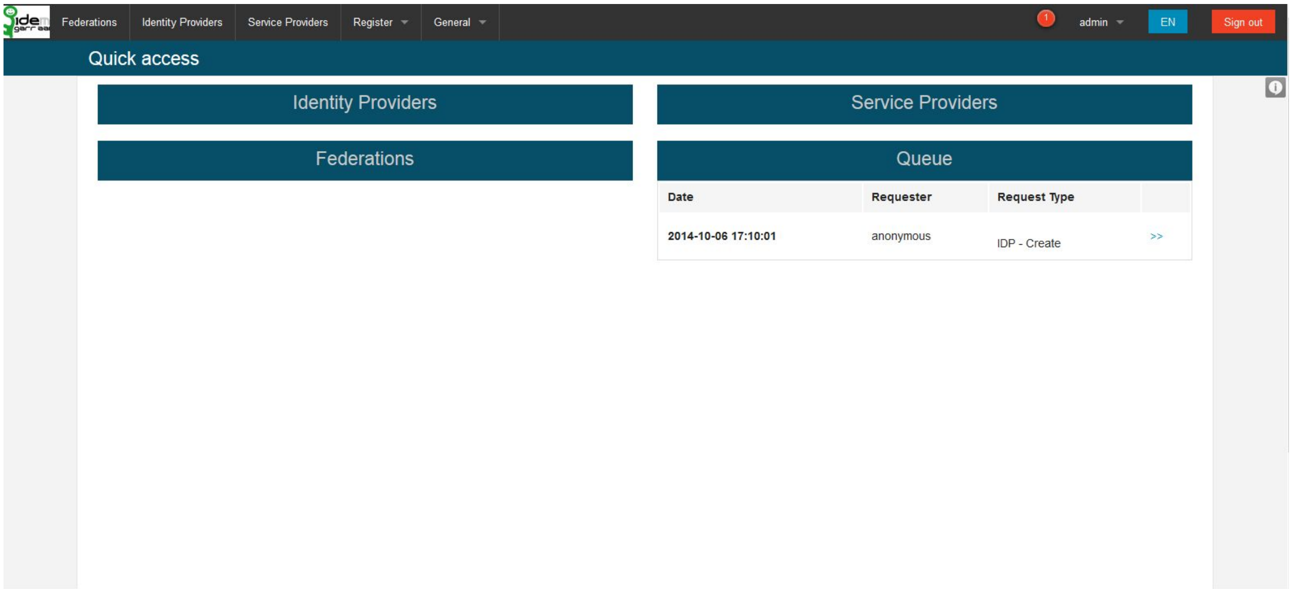


Federation: IDEM di Test

Fig. 5: IdP registration form page - advanced mode.

a) **General Tab**: contains some general information about your entity.
b) **Organization Tab**: contains the information about the organization/owner of the entity.
c) **Contacts Tab**: contains the information about the contacts that are responsible of the entity.
d) **UI Information tab**: contains the information useful for Login and Discovery User Interface (MDUI).
e) **SAML tab**: contains the information about SAML endpoints of the entity and its entityID.
f) **Certificates tab**: contains the information about the certificates used to sign and encrypt the assertions exchanged between the entities of the federation.

4) After that the “**Register**” button has been clicked, the registration request for the entity will be added to the queue and a message will be sent to the Federation Operator.

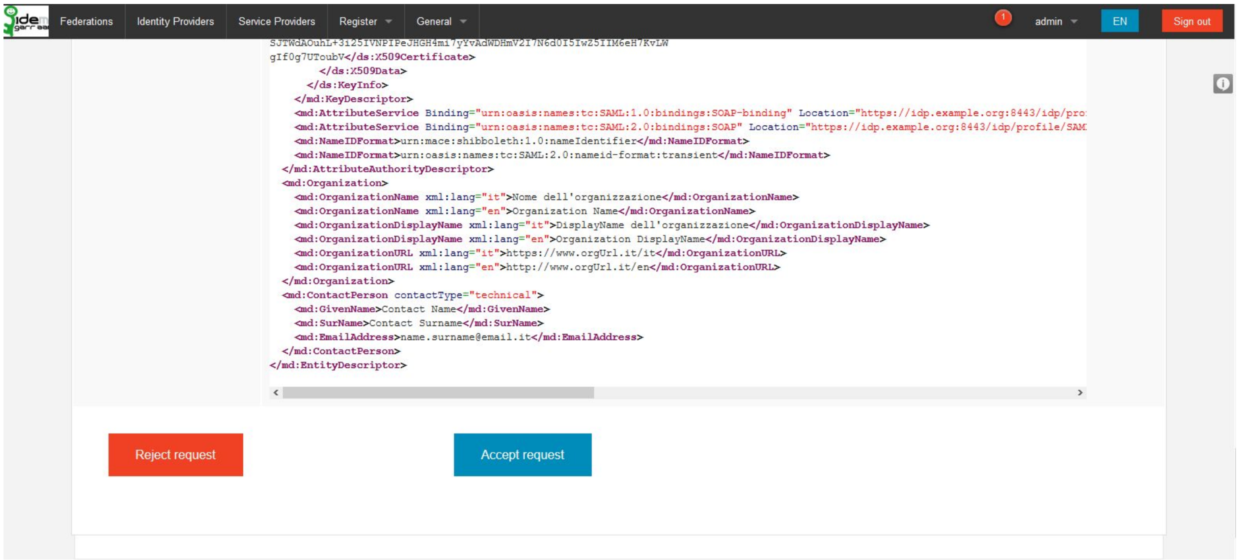


Date	Requester	Request Type
2014-10-06 17:10:01	anonymous	IDP - Create

Fig. 6: Jagger Main Page.

5.5.2 Add the new entity to the Test Federation – Operator side

1) The Federation Operator receives the request from the new entity and, if it complies with the rules designed for the Test Federation, accepts it.



```
SJTWdAOuhL+3i25IVNPIPeJHGH4mi7yYvAdWDHmV2I7N6d0I5IwZ5IIM6eH7KvLW
gIf0g7UToubV</ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </md:KeyDescriptor>
    <md:AttributeService Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding" Location="https://idp.example.org:8443/idp/pro
    <md:AttributeService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP" Location="https://idp.example.org:8443/idp/profile/SAM
    <md:NameIDFormat>urn:mace:shibboleth:1.0:nameIdentifier</md:NameIDFormat>
    <md:NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</md:NameIDFormat>
  </md:AttributeAuthorityDescriptor>
  <md:Organization>
    <md:OrganizationName xml:lang="it">Nome dell'organizzazione</md:OrganizationName>
    <md:OrganizationName xml:lang="en">Organization Name</md:OrganizationName>
    <md:OrganizationDisplayName xml:lang="it">DisplayName dell'organizzazione</md:OrganizationDisplayName>
    <md:OrganizationDisplayName xml:lang="en">Organization DisplayName</md:OrganizationDisplayName>
    <md:OrganizationURL xml:lang="it">https://www.orgUrl.it/it</md:OrganizationURL>
    <md:OrganizationURL xml:lang="en">http://www.orgUrl.it/en</md:OrganizationURL>
  </md:Organization>
  <md:ContactPerson contactType="technical">
    <md:GivenName>Contact Name</md:GivenName>
    <md:SurName>Contact Surname</md:SurName>
    <md:EmailAddress>name.surname@email.it</md:EmailAddress>
  </md:ContactPerson>
</md:EntityDescriptor>
```

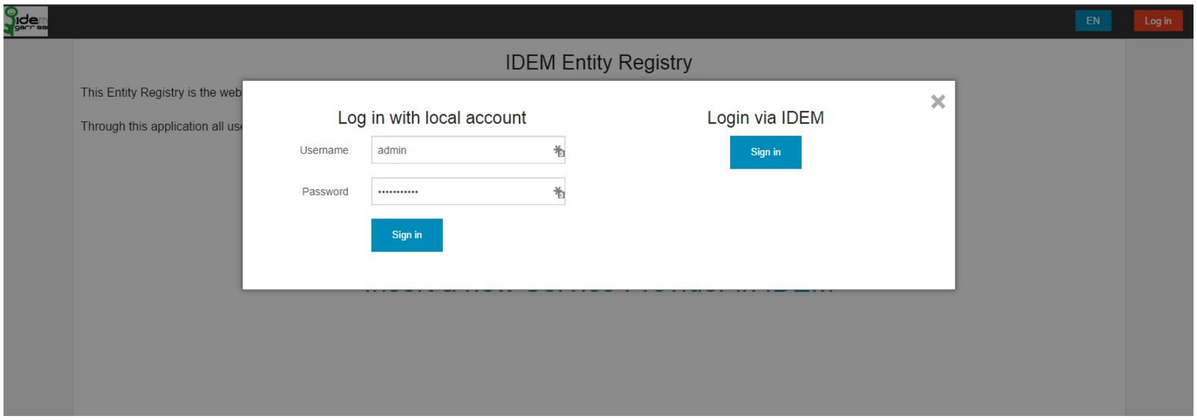
Fig. 7: Jagger Management Request Page.

2) After that operation, the metadata aggregate stored into Jagger instance for the Test Federation will be updated with the metadata of the new entity.
   Now the Federation Operator can retrieve the new metadata aggregate, sign it and publish it with the Shibboleth Metadata Aggregator(MDA) tool following these steps:

   a) Open a terminal on the virtual machine that hosts the MDA tool
   b) Move to the directory "**/opt/ukf-meta**" and execute one of these commands:
      i) '**ant #FED_ID#-test-check**': to perform several checks on Test Federation metadata aggregate
      ii) '**ant #FED_ID#-test-all**': to perform several checks and produce the following Test Federation metadata aggregates:
         (1) **#FED_ID#-test-metadata.xml**: not signed
         (2) **#FED_ID#-test-metadata-sha1.xml**: signed with algorithm SHA1
         (3) **#FED_ID#-test-metadata-sha256.xml**: signed with algorithm SHA256

All these metadata aggregates will be published to the web location:
https://#YOUR.APPLIANCE.FQDN#/mda/#FED_ID#-test

3) Once the Federation Operator has updated the Test Federation metadata aggregates, he informs the entity owner that will configure its machine to retrieve and use the proper metadata aggregate provided on the established public location.
   Finally the owner of the entity and the Federation operator can verify the correctness of the information provided by the entity metadata and the correct assertion exchange between the new entity and another one into the Test Federation: a SP if we have added an IdP or an IdP if we have added an SP.
4) The last operation that the Operator must do for the new entity is to assign the right privileges to its owner on the Registry instance by following the instructions:
   1) Ask to the entity owner to log into Jagger Metadata Registry through the Federated Access (e.g.: Login via IDEM) and his Identity Provider.
      This operation will create a new user into the Jagger Metadata Registry.


IDEM Entity Registry

This Entity Registry is the web

Through this application all us

Log in with local account

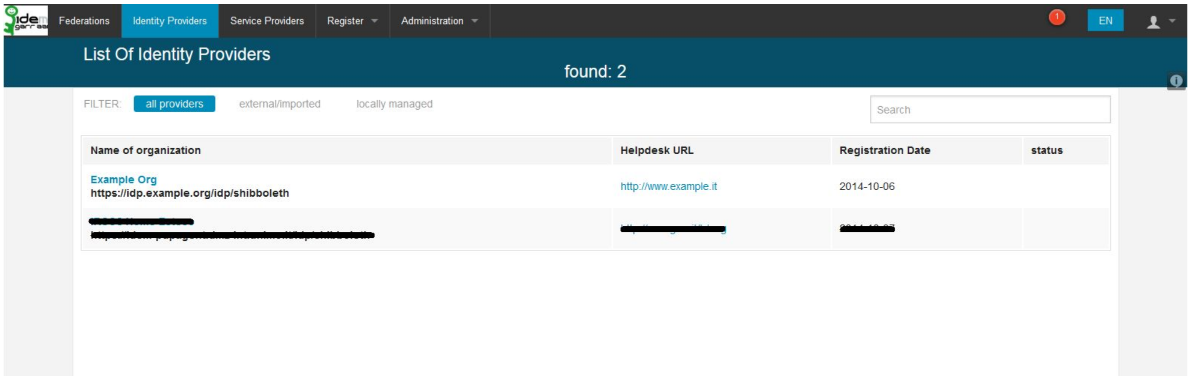
Username: admin

Password: ••••••••••••

Login via IDEM

Fig. 8: Jagger Authentication Page.

   2) Find the new entity inserted on Jagger and open it with a click on its display name:



List Of Identity Providers

found: 2

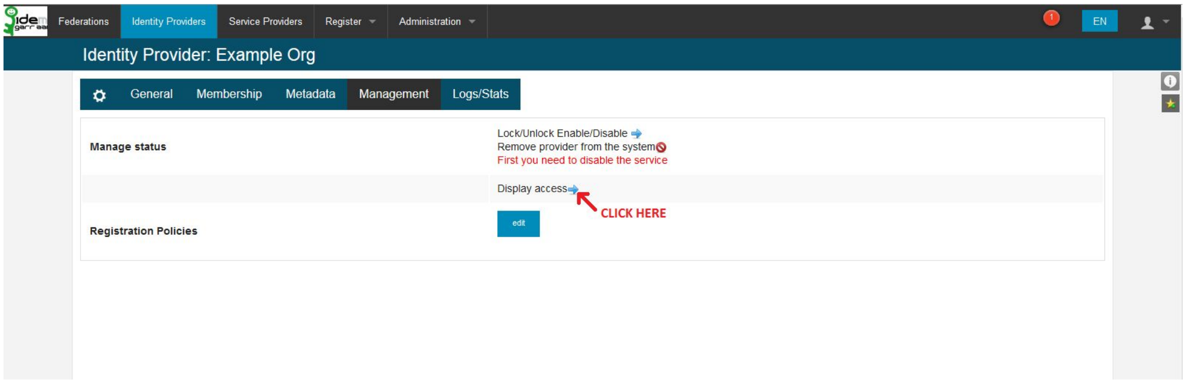
FILTER: all providers | external/imported | locally managed

Search

Name of organization	Helpdesk URL	Registration Date	status
Example Org https://idp.example.org/idp/shibboleth	http://www.example.it	2014-10-06	
[redacted]	[redacted]	[redacted]	

Fig. 9: Jagger List of IdP page.

3) Move to the “**Management**” section and open the Access Management (as shown by the image):

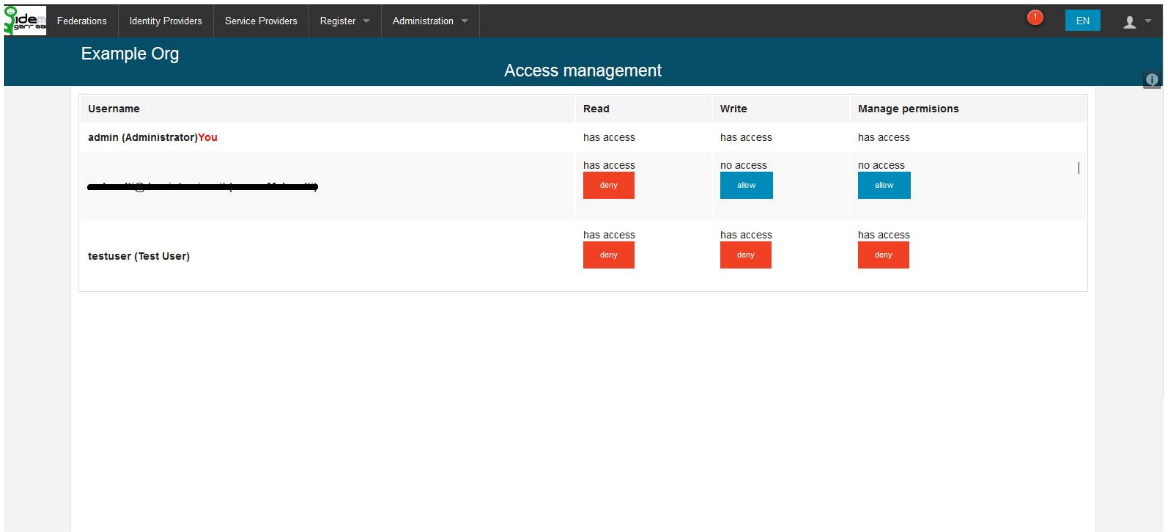


Identity Provider: Example Org

Manage status	Lock/Unlock Enable/Disable Remove provider from the system First you need to disable the service
	Display access
Registration Policies	edit

Fig.10: Jagger IdP page - Management section.

4) Assign the right permissions for the entity to its owner:



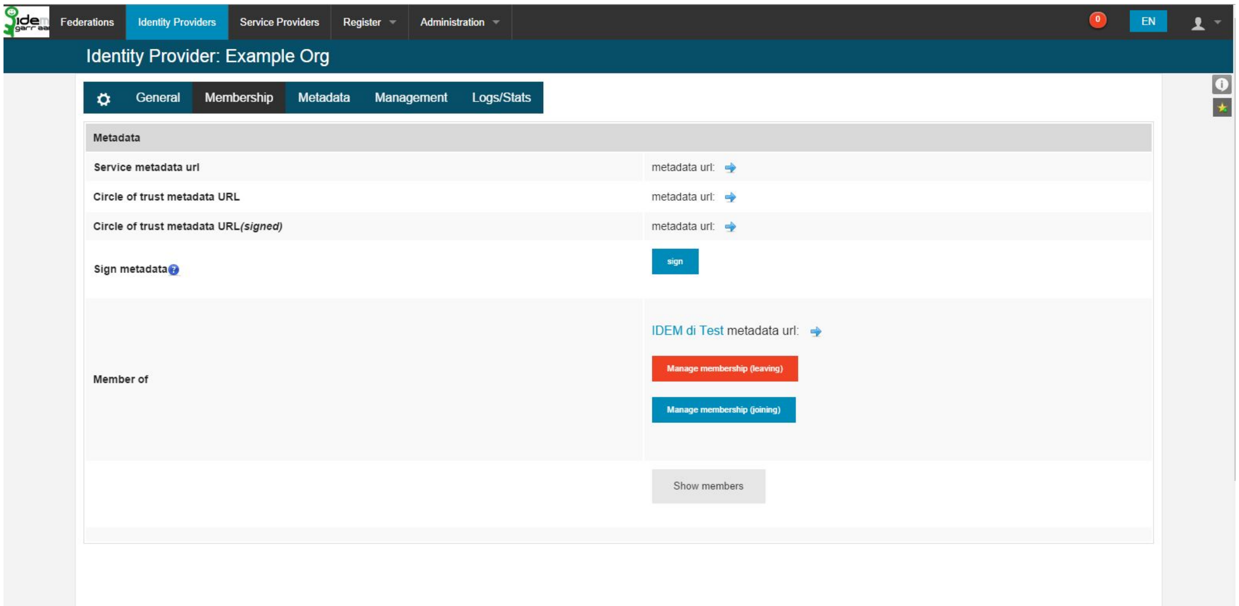
Username	Read	Write	Manage permisions
admin (Administrator)You	has access	has access	has access
[redacted]	has access deny	no access allow	no access allow
testuser (Test User)	has access deny	has access deny	has access deny

Fig. 11: Jagger IdP Access Management page.

5.5.3 Migrate the new entity to the Production Federation – User side

Once all tests are finished correctly, the owner of the entity can decide to migrate it to the Production Federation or maintain it into the Test Federation.
If he decides to migrate the entity to the Production Federation, he must follow the steps below:

1) After logging into Jagger, find your entity and move to "Membership" tab:

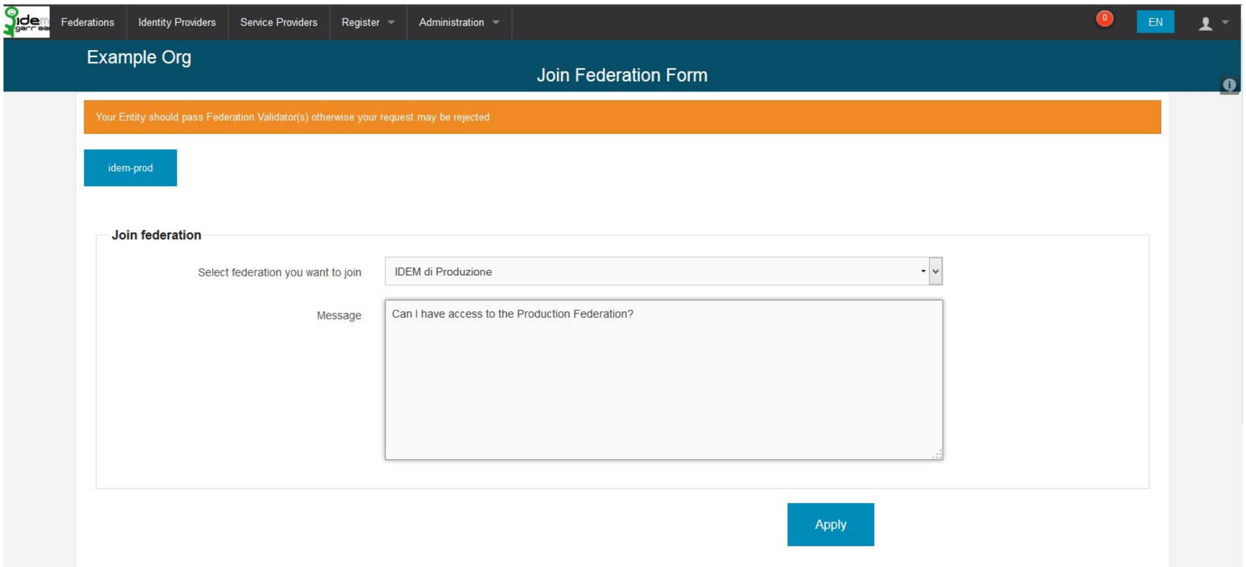


Identity Provider: Example Org

Metadata	
Service metadata url	metadata url: ➡
Circle of trust metadata URL	metadata url: ➡
Circle of trust metadata URL*(signed)*	metadata url: ➡
Sign metadata	sign
Member of	IDEM di Test metadata url: ➡ Manage membership (leaving) Manage membership (joining)
	Show members

Fig. 10:Jagger IdP page - Membership section.

2) Click on the blue button called “**Manage membership (joining)**”, select the federation desired, check that your entity has the requirements to join that federation with its validator (e.g.: *idem-prod*) and leave a message for the Federation Operator before clicking on the “**Apply**” button



Select federation you want to join: IDEM di Produzione

Message: Can I have access to the Production Federation?

Fig. 11:Jagger IdP page - Join Federation section.

3) The Federation Operator will receive the entity request to join the Production Federation, if needed will execute other checks, and finally will accept the request.

5.5.4 Migrate the new entity to the Production Federation – Operator side

1) After the request has been accepted, the metadata aggregate stored into the Jagger instance for the Production Federation will be updated with the metadata of the new entity.
2)

   Now the Federation Operator can retrieve the new metadata aggregate, sign it and publish it with the Shibboleth Metadata Aggregator(MDA) tool following these steps:

   a) Open a terminal on the virtual machine that hosts the MDA tool
   b) Move to the directory “**/opt/ukf-meta**” and execute one of these commands:
      i) ‘**ant #FED_ID#-prod-check**’: to perform several checks on Production Federation metadata aggregates
      ii) ‘**ant #FED_ID#-prod-all**’: to perform several checks and produce the following Production Federation metadata aggregates:
         (1) **#FED_ID#-prod-metadata.xml**: not signed

(2) **#FED_ID#-prod-metadata-sha1.xml**: signed with algorithm SHA1
(3) **#FED_ID#-prod-metadata-sha256.xml**: signed with algorithm SHA256

All these metadata aggregates will be publish to this web location:
https://#YOUR.APPLIANCE.FQDN#/mda/#FED_ID#-prod

3) Once the Federation Operator has updated the Production Federation metadata aggregate, he informs the entity owner that will configure its machine to retrieve and use the proper metadata aggregate provided at the established public location.
   Finally, the owner of the entity and the Federation Operator can test the correct assertion exchange between the new entity and another one in the Production Federation: an SP if we have added an IdP or an IdP if we have added an SP.

5.6 HOW-TO ADD AN ENTITY TO EDUGAIN INTERFEDERATION

The operations to join the eduGAIN federation through the appliance are very similar to the operations undertaken for joining the other federations.

We can summarize the needed operations into the following steps:

1) Register your entity into the Test Federation and migrate it into the Production Federation.
2) When the entity is ready, the owner can request to join the “Federation for eduGAIN” in a similar way to that undertaken by the owner to migrate its entity from Test Federation to Production Federation.
4) Once the metadata aggregate stored in the Jagger instance for the “Federation for eduGAIN” is updated with the metadata of the new entity, the Federation Operator can retrieve the new metadata aggregate, sign it and publish it with the Shibboleth Metadata Aggregator(MDA) tool following these steps:
   a) Open a terminal on the virtual machine that hosts the MDA tool.
   b) Move to the directory “**/opt/ukf-meta**” and execute one of these commands:
      i) ‘**ant #FED_ID#-to-edugain-check**’: to perform several checks on the entities that will be imported into the eduGAIN Federation metadata aggregate.
      ii) ‘**ant #FED_ID#-edugain-all**’: to perform several checks and produce the following metadata aggregates:
         (1) To be consumed by the own Federation:
            (a) **#FED_ID#-from-edugain-metadata.xml**: not signed
            (b) **#FED_ID#-from-edugain-metadata-sha1.xml**: signed with algorithm SHA1
            (c) **#FED_ID#-from-edugain-metadata-sha256.xml**: signed with algorithm SHA256

(2) To be consumed by the eduGAIN interfederation
    (a) **#FED_ID#-to-edugain-metadata-sha1.xml**: signed with algorithm SHA1
    (b) **#FED_ID#-to-edugain-metadata-sha256.xml**: signed with algorithm SHA256

All these metadata aggregates will be published to this web location:
https://#YOUR.APPLIANCE.FQDN#/mda/#FED_ID#-edugain

6. CONCLUSIONS AND OPEN POINTS

With this «appliance» we plan to standardize and support Federation operations.
By consuming this FaaS service, it will be possible to rapidly start the operations of a new Federation, by almost eliminating the technological steps in it. It will also be possible to leverage experiences and best practices to operate effectively a Federation even starting with little or no prior experience.

The code developed for Puppet, to permit the FaaS appliance installation and configuration, has been developed. Some more testing may be needed and some additional activity could be performed to generalize some process and adapt it to different Federations.

7. REFERENCES

- Openstack: http://www.openstack.org/
- Shibboleth: http://shibboleth.net/
- Puppet: http://puppetlabs.com/
- Jagger Resource Registry: https://github.com/Edugate/ResourceRegistry
- MDA: https://shibboleth.net/products/metadata-aggregator.html
- DiscoJuice: http://discojuice.org/

APPENDIX2: IDP IN THE CLOUD - ACTIVITIES

Work Package/Activity: WP2/Federation as a Service

Authors A. Biancini (GARR), M. Malavolti (GARR), M.L. Mantovani (GARR)

1. INTRODUCTION

This document will present all the main activities shared between GARR and the Elcira project in the field of automating an IdP installation. As we will describe, installing and configuring a new Identity Provider (IdP) to operate within an existing Identity Federation, is not an easy task.

During its experience of operating the Italian IDEM community, GARR has had the opportunity to identify which were the main problems and issues that could prevent an organization to participate to the Federation. From a participant's point of view, the more complex task is that of creating and managing an IdP. For this reason, GARR developed a project, called IdP in the Cloud, to tackle this problem in the Italian community.
Within this project some automation code and scripts have been developed by GARR. In the Elcira collaboration GARR shared the results and main components realized by this project to the Latin America community so that it could be possible to replicate the project in those areas and obtain similar benefits.

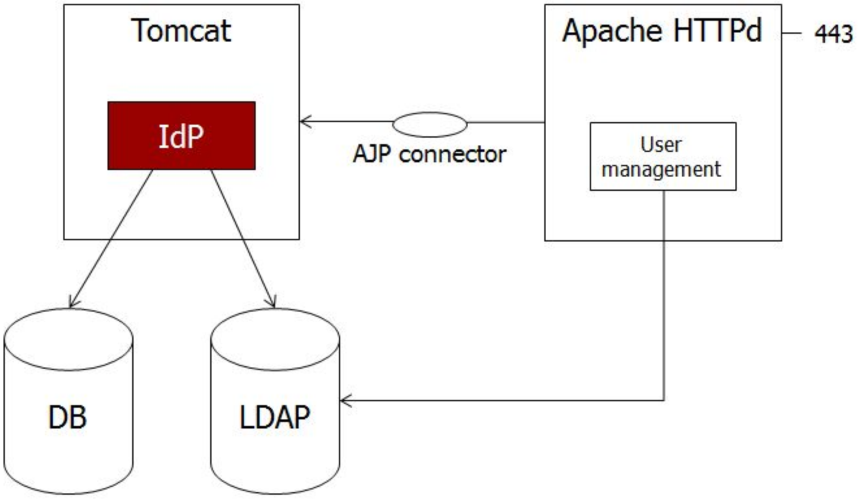
The rest of this document will present the main activities and tasks to install and configure an IdP. Then the IdP in the cloud project will be presented briefly and the last part of the document will be an how-to guide that will summarize the activities shared during the Elcira WP2.

2. INSTALLING AND IDP

The installation of an IdP, for an organization willing to participate to an Identity Federation, is not a simple task. This chapter will describe the main software components involved in a full IdP installation and will describe the main difficulties that may be encountered during the installation and configuration process.

3. THE MAIN SOFTWARE COMPONENTS

To have a fully functional Shibboleth IdP installed, different software components needs to be installed and configured properly. The picture below shows them all:



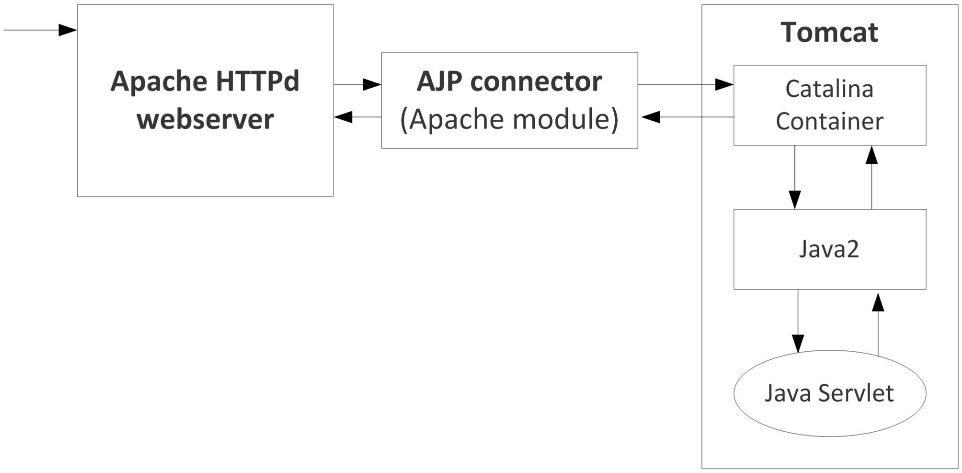
In the following, all the main components are described in respect to an IdP installation.

- **The Shibboleth IdP:**

  The Identity Provider is a web application, called Shibboleth IdP, developed in Java that can be downloaded here: http://shibboleth.net/downloads/identity-provider/2.4.0/shibboleth-identityprovider-2.4.0-bin.zip
  The installation starts by executing the install.sh script and is quite easy. The IdP needs a Java application server to be run, usually Tomcat is used for this purpose.
  The IdP can then be configured to access the proper LDAP and MySQL databases by modifying the files:
  - /opt/shibboleth-idp/conf/login.config
  - /opt/shibboleth-idp/conf/attribute-resolver.xml

- **Tomcat and Apache httpd:**

Very often, Tomcat is configured behind an Apache HTTPd instance, as shown in this picture:



There are different reasons for this choice:

- for security reasons (httpd is more robust than Tomcat)
- to make Tomcat run with a non privileged user (that cannot bind to port 443)
- to server more quickly static files from Apache
- to use other Apache features (mod_rewrite, mod_auth, ...)

The following configurations need to be realized as follows:

- Configuration for Tomcat in /etc/tomcat7/server.xml:

```
<Connector port="8009" protocol="AJP/1.3"
        connectionTimeout="20000"
        URIEncoding="UTF-8"
        redirectPort="443" enableLookups="false”
        address="127.0.0.1" />
```

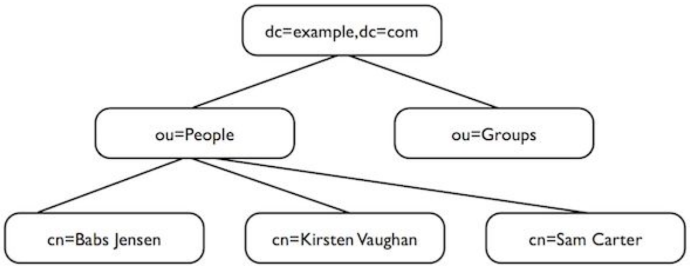
- Configuration for Apache in /etc/apache2/httpd.conf:

```
ProxyRequests Off
<Proxy *>
  Order deny,allow
  Allow from all
</Proxy>
ProxyPass /idp ajp://localhost:8009/idp
```

ProxyPassReverse /idp ajp://localhost:8009/idp

- **LDAP server:**
  To manage user information, most commonly, an LDAP database server is used. LDAP, in fact, is born to implement a directory service (like a user's database for an organization).
  The IdP application can access an external LDAP, if available, or a locally installed server on localhost. On Linux, usually, the OpenLDAP server is used for these goals.

  LDAP has a tree structure:



  Each LDAP entity may implement different schemas (i.e. has different attributes):
  - eduPerson (http://www.internet2.edu/products-services/trust-identity-middleware/eduperson-eduorg);
    and
  - SCHAC (http://www.terena.org/activities/tf-emc2/schac.html).

- **Database: MySQL:**
  Some of the user attributes needs to be managed by the IdP itself. Usually, however, the IdP should NOT write to the LDAP (this is a security best practice). For this reason another database is needed to store some additional user information. Usually a MySQL database may be used for this purpose.

  A field that works this way, for instance, is the eduPersonTargetedID which is a "persistent, non-reassigned, privacy-preserving identifier for a user shared between an IdP and a SP".

  Besides, some optional component to the Shibboleth IdP may need to store information in a relational database. An example is uApprove: User Consent Module for Shibboleth Identity Providers. This module stores the release allowance choices from the users into a relational database.

- **User Management Interface:**
  To manage users inside the LDAP server, an user management interface may be of help. There are different tools that can be used for this purpose. In particular, phpLDAPadmin has been chosen and customized by GARR to provide this functionality in a simplified and easy to use way to the IdP administrators.

4. DIFFICULTIES AND PROBLEMS

As it could be clear from the list of needed software just shows, the installation of an IdP is not a trivial task. By interacting with different subjects within the Academic and Research communities in Italy, GARR has identified the following main issues regarding the management of an IdP:

- The participant has to manage a lot of different technologies (Shibboleth, Tomcat, LDAP, security on the server, MySQL, …);
- An organization installing and IdP needs, after installation, to monitor and update constantly the technical infrastructure (for security and quality of service);
- After an IdP installation, the privacy and identity management policies have to be examined and implemented ;
- At last, an organization installing an IdP have, surely, to manage users and passwords.

Many entities, especially the smaller ones or the ones with smaller IT departments, do not have enough skills or resources to manage all these tasks and activities.

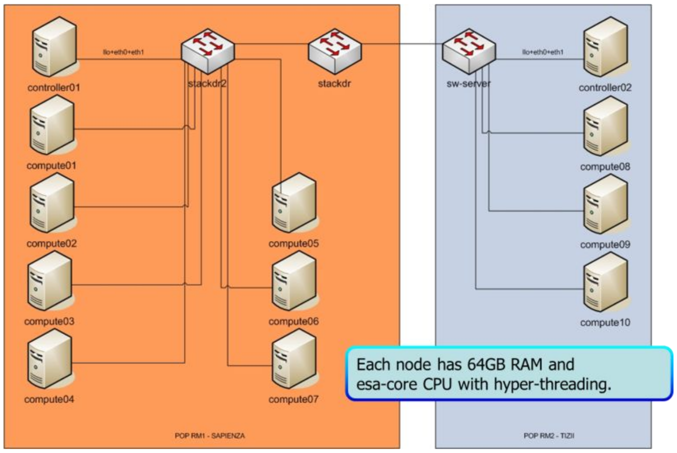
5. THE EXISTING CONTEXT: IDP IN THE CLOUD

To overtake these problems and limitations GARR developed a project, called IdP in the Cloud. This project has the goal to simplify IdP installation and management. GARR, within this project, is offering a cloud service to provide IdPs as a service upon user request. With this service all the technical and operational tasks bound to IdP installation and management will be performed by GARR, the requestor will "only" have to manage its own users and their password.

To implement this project a specific private cloud infrastructure has been installed in GARR. Moreover, GARR worked on some software automation tool to automatize the provisioning of an IdP starting from a unconfigured VM. In the following of this chapter, these two aspects will be described in detail.

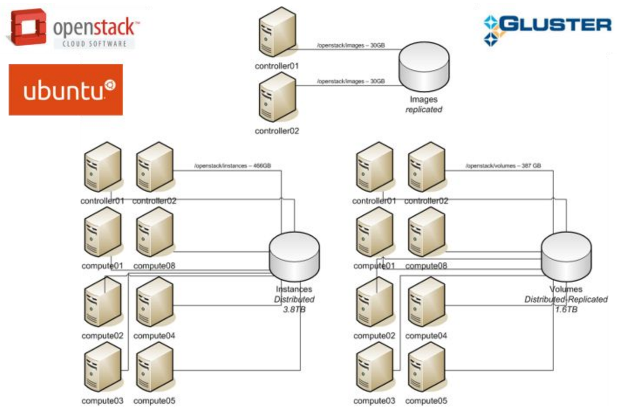
6. THE INFRASTRUCTURE

GARR decided to build its own Cloud infrastructure to have a fault tolerant and resilient system where we could offer advanced servers and services in "as a Service" fashion.
This infrastructure is made of 12 physical nodes. Each node has 64GB RAM and esa-core CPU with hyper-threading. The nodes are geographically distributed on two distant sites to maximize resilience in case of fault of systems or communication.

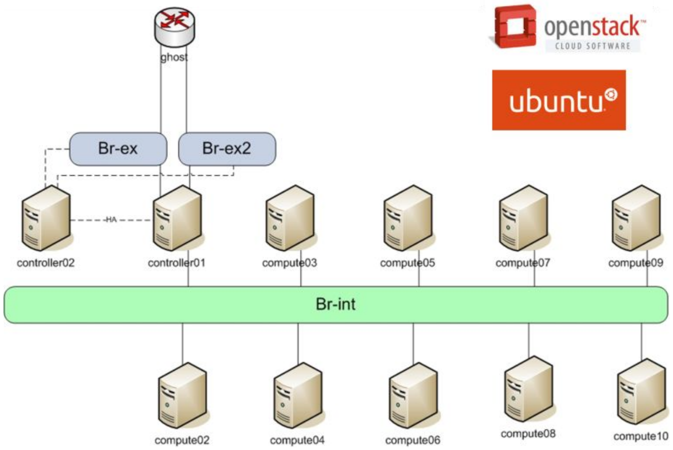


Each node has 64GB RAM and esa-core CPU with hyper-threading.

The GARR Cloud is built using OpenStack platform on Ubuntu Server distributions.
The storage present on the nodes is managed with GlusterFS in the distributed and replicated mode for the volumes. This ensures the data availability and the resilience.

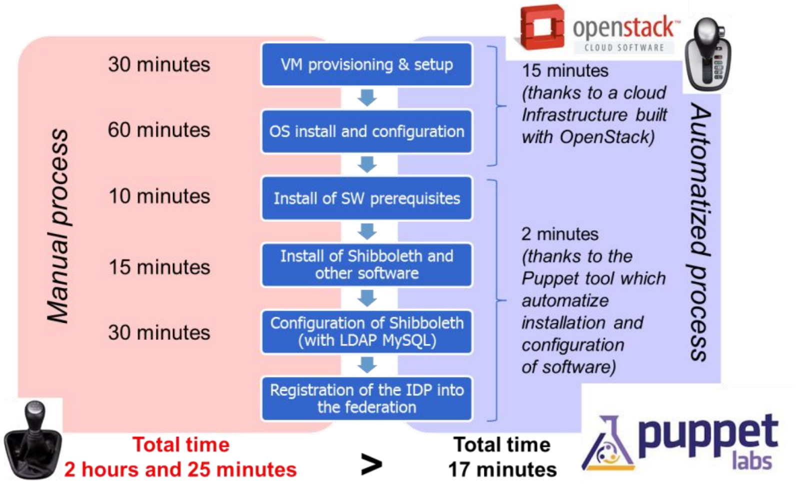


OpenStack is configured for using 2 controllers, located in the 2 different sites of the cloud, that control the set of nodes. The following image shows the redundancy and resilience also in communications.



7. AUTOMATING THE IDP PROVISIONING

The problem of deployment and management of hundreds of potential systems in the cloud was tackled automatizing and optimizing the provisioning process. In this image we see a comparison of times of single steps necessary for the provisioning of the IDP on the left during a manual process and on the right during an automatized process using OpenStack features and puppet recipes.


Manual process
30 minutes
60 minutes
10 minutes
15 minutes
30 minutes
VM provisioning & setup
OS install and configuration
Install of SW prerequisites
Install of Shibboleth and other software
Configuration of Shibboleth (with LDAP MySQL)
Registration of the IDP into the federation
openstack CLOUD SOFTWARE
15 minutes (thanks to a cloud Infrastructure built with OpenStack)
2 minutes (thanks to the Puppet tool which automatize installation and configuration of software)
Automatized process
Total time 2 hours and 25 minutes
>
Total time 17 minutes
puppet labs

Thanks to OpenStack features the time for the first 2 steps for VM provisioning and OS installation and configuration is reduced from one hour and an half to 15 minutes. Thanks to Puppet recipes the time for the next 3 steps: Install of SW prerequisites, Install of Shibboleth and other software, Configuration of Shibboleth (with LDAP and MySQL, and others) is reduced from 55 minutes to 2 minutes. So the provisioning total time of the IDP in the Cloud machine is reduced from 2 hours and 25 minutes to 17 minutes.

The “IDP in the cloud” technical solution is an Appliance as a Service, i.e. a virtual machine in the GARR cloud that includes a set of preconfigured services.

The virtual machine is based on Linux Ubuntu Server 12.04 LTS distribution and is provided, with the use of Puppet recipes, of:

- a LDAP Server (OpenLDAP): dedicated to organizations not provided with it or that require it explicitly;
- an Apache2 Server: as a front-end of Tomcat7 Servlet Container used by the IdP;
- a Shibboleth IdP (available versions: 2.3.3, 2.3.8 and 2.4.0) ;
- the uApprove module: version 2.4.1 or 2.5.0 for user attribute release with informed consent;
- the IDP Login Page: that can be customized according to the applicant Organization. This interface is designed to show two languages, typically for English and the national language;
- a MySQL Server: used by the IdP;
- an IPTABLES firewall: already configured to respond properly to the needs of the IdP;
- a rsyslog daemon: for IDP logs centralized management;
- a phpLDAPadmin interface: for simplified users' management on the IdP for organizations that required the installation of the LDAP server;
- a Nagios Server: for monitoring and alerting;
- a Collectd daemon: to collect system statistics;
- a certificate (needed to the Apache 2 server (HTTPS)) verification system and expiry notification;
- a data backup system.

The system offered is not simply limited to IDPs, but also implements an LDAP service and its management interface and is configured as an Identity Management System.

To obtain eduGAIN compliance and enable end-users to access eduGAIN services, we create metadata entities and identities' attributes that follow the eduGAIN metadata profile and the edugain attribute profile. Pointing on attributes, all eduGAIN recommended attributes are implemented in the LDAP directory and the web form for the IDP administrators helps in filling their values. The controlled vocabulary on Affiliation and homeOrganizationType is also implemented.

The IDP implemented complies with the eduGAIN specifications and is technically ready to be registered in the inter-federation.

The IDP is also compliant with REFEDS discovery guide so the IDP's metadata are enriched with names and logos to be ready for smart discovery services. Moreover, the IDP login page is designed for co-branding with the SP, taking a lot of user interface information from the SP metadata and displaying them on the IDP login page.

8. HOW TO REPLICATE THIS SOLUTION

The solution developed by GARR in the “IdP in the Cloud” project has been shared with Elcira to evaluate the opportunity to replicate it within the Latin American communities.

The automation code developed for this project has been released by GARR on GitHub with Apache 2.0 license. It is downloadable from here: https://github.com/ConsortiumGARR/Puppet-GARRShibbolethIdP. This code contains a set of recipes for Puppet to automatize the installation of all the software components needed to operate an IdP.

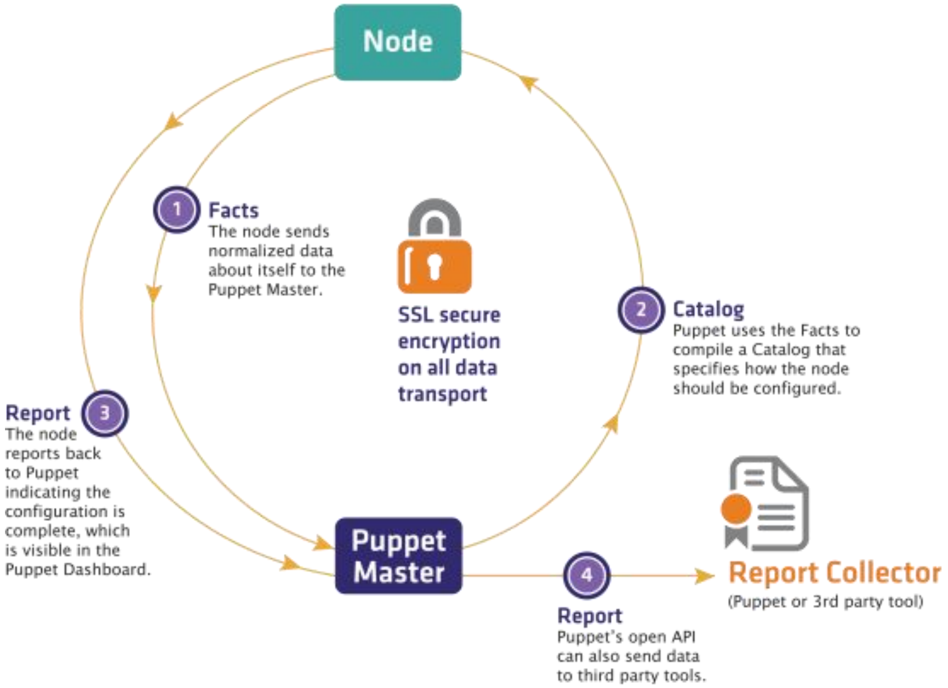
In the rest of this chapter, the sharing activities performed within Elcira project will be described. At first a description of Puppet and its internal mechanism will be presented. Than the other sections will focus on the steps needed to install an IdP with the code developed by GARR.

9. PUPPET, WHAT IT IS AND HOW IT CAN BE USED

Puppet is a framework able to automate repetitive system administration tasks. It is an open source tool (with Apache 2.0 license) and it is available on many different platforms: Linux/Windows/UNIX/…

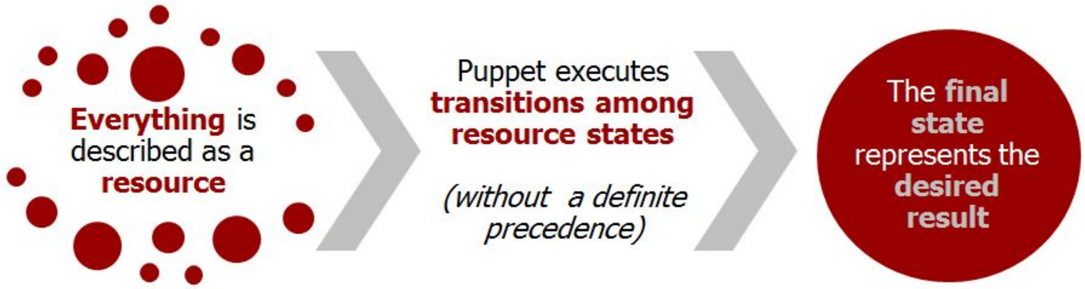
Puppet manages your servers: you describe machine configurations in an easy-to-read declarative language, and Puppet will bring your systems into the desired state and keep them there.

Puppet can be used to automatize the provisioning and configuration of IT servers and software components in a distributed environment. The installation and configuration of a software with Puppet follows the following path:


Node
1 Facts
The node sends normalized data about itself to the Puppet Master.
SSL secure encryption on all data transport
2 Catalog
Puppet uses the Facts to compile a Catalog that specifies how the node should be configured.
Report 3
The node reports back to Puppet indicating the configuration is complete, which is visible in the Puppet Dashboard.
Puppet Master
4
Report
Puppet's open API can also send data to third party tools.
Report Collector
(Puppet or 3rd party tool)

1. The node that needs to be installed contacts a central server called Puppet Master. This call will happen with the Node providing some information about itself. The information sent are called facts and are normalized data describing the main characteristics of the node.
2. The Puppet Master takes the facts sent by the Node, and with these information compiles a Catalog to be sent to the Node for executing. The Catalog represents a full description about which software needs to be installed on the Node and which should be the final configuration.
3. At this point the Node executes the Catalog and installs and configures all the software specified in it. At the end of this installation procedure, a Report is produced and sent back to the Puppet Master. This report contains the result (and eventually a description of the errors or warning) happened during the installation.
4. The Puppet Master can store this Report in external Collectors. These Collectors can query the Puppet Master with open APIs to retrieve installation status and results.

The key element to be used in this cycle, to perform a Puppet automation, is the Catalog. Within a Catalog, all the configuration instructions needs to be specified. The main principles upon which the catalog is build and composed are represented in the following:


Everything is described as a resource
Puppet executes transitions among resource states
(without a definite precedence)
The final state represents the desired result

There are three main concepts that must be taken into consideration:

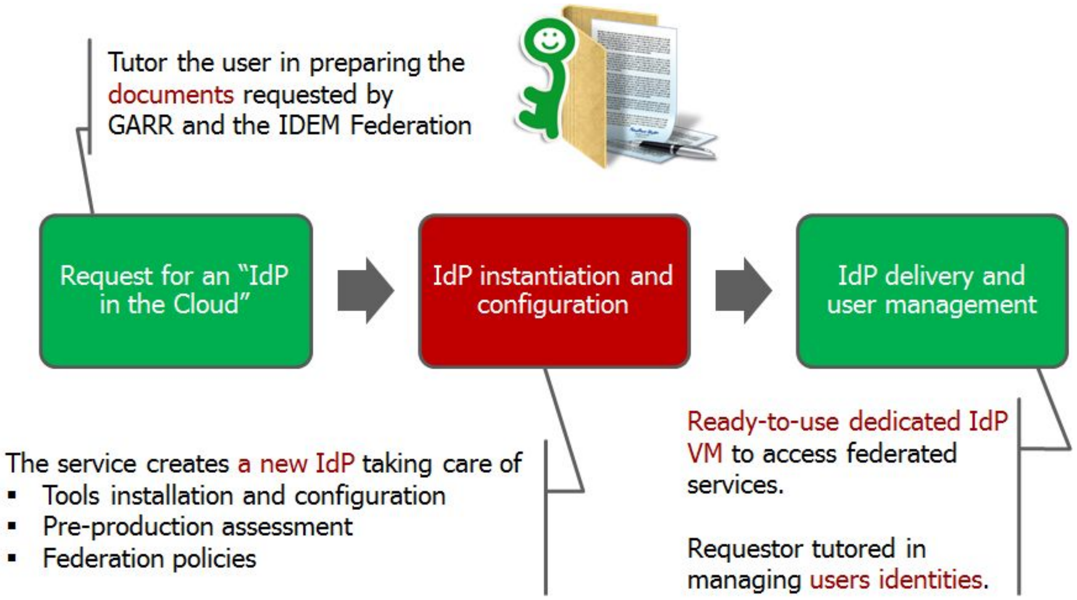
1. Everything, in a Puppet Catalog, is described as a resource. A resource can be, for instance, a file, a package, a process (or a daemon) and any other configuration item that can be manipulated by Puppet.
2. The Puppet Catalog describes a set of transitions among the resource states. Every resource has a desired final state and a current state. If the two states differ, Puppet takes the responsibility to execute a transition to bring that specific resource to the desired final state described in the Catalog.

   It is important to notice that Puppet will execute the transitions among different resource states, without a definite precedence. This means that, if not explicitly described in the Catalog, every resource transition may happen at any time (without a dependency schema among resources).
3. The final state of all the resources of the Catalog represents the desired configuration result for the configuration of the Node.

10. THE IDP INSTANTIATION PROCESS

To permit the realization of the “IdP in the cloud” project, GARR has developed a set of recipes that can be used to install and IdP on a Linux server.

Within the “IdP in the cloud” project, GARR has also defined a process to be followed to request an IdP and to have it installed and configured on the OpenStack virtual infrastructure. In the picture below, the three main steps of this simplified process are shown:


Tutor the user in preparing the documents requested by GARR and the IDEM Federation
Request for an “IdP in the Cloud”
IdP instantiation and configuration
IdP delivery and user management
The service creates a new IdP taking care of
▪ Tools installation and configuration
▪ Pre-production assessment
▪ Federation policies
Ready-to-use dedicated IdP VM to access federated services.
Requestor tutored in managing users identities.

1. The requesting organization requires for an IdP to be installed on the GARR virtual infrastructure. This request is accompanied by a small document that contains all the relevant information that can be used to specify the configuration parameters of the IdP. In particular this document contains:
   - Organization name
   - Organization internet domain
   - IdP name (or EntityID)
   - Description of the service
   - Organization public web site URL
   - Organization privacy policy page URL
   - IdP Informative web page URL (shown to users)
   - Organization logo images

  - Technical contact mailing list

2. The real IdP installation and configuration is executed by GARR. In this step the Puppet recipes are used and the provisioning of this IdP is almost completely automatized.
3. After the installation and configuration the IdP is ready to be delivered to the requesting organization. From this moment the requesting organization has the responsibility to manage its user identities registered on the IdP.

11. HOW-TO INSTALL AN IDP WITH PUPPET

Operations to be executed the first time to install the GARR code on a puppet master:

1. Download the Puppet code:

```
cd /opt
git clone https://github.com/ConsortiumGARR/Puppet-GARRShibbolethIdP
cd Puppet-GARRShibbolethIdPgit submodule init
git submodule update
```

2. Execute the install.sh script in the scripts folder:

```
cd /opt/Puppet-Shibboleth/scripts
./setup.sh
```

   This operation creates the module folders under the path /etc/puppet/modules.

Operations to be executed every time we install a new IdP:

1. Execute the prepare_puppetmaster.sh script:

```
cd /opt/Puppet-Shibboleth/scripts
./prepare_puppetmaster.sh
```

   This script creates all the necessary files to customize the IdP installation (logo, metadata information, styles, ecc). During the execution of the script different questions will be asked and an example of the answer will be provided.
2. The script prepare_puppetmaster.sh can generate certificate files for HTTPs and, in this case, it will also create a CSR request file in the /tmp folder to be sent to the CA for signing.
   The certificates can be replaced, even after the first installation, by changing the files in the folder: /etc/puppet/modules/shib2common/files/certs

3. After this script it is possible to overwrite styles and logos with new versions of the files. These style files will be created in the folder: /etc/puppet/modules/shib2idp/files/styles and will have names containing the hostname for the IdP which is going to be configured.

4. Execute the generate_sitepp.py script:

   ```
   cd /opt/Puppet-Shibboleth/scripts
   ./generate_sitepp.py
   ```

   This script will create a site.pp file for the IdP to be installed. During the execution of the script different questions will be asked and an example of the answer will be provided.
   The generated file for the IdP will be:
   /etc/puppet/manifests/nodes/$$IDP_HOSTNAME$$.pp
5. Restart puppetmaster service:

   ```
   service puppetmaster restart
   ```

Installation of the IdP on the final server:

To install the IdP it is now possible to login to the server that will operate the IdP and execute puppet:
puppet agent –test

12. CONCLUSIONS AND OPEN POINTS

The code developed for Puppet, to permit an IdP installation and configuration, has been released. Some work, however, still needs to be done:

- regarding internationalization: many pages and messages are still Italian or at least require that the Italian language is present (together with additional ones, like English or others);
- generalization of some logic: for the moment the policies implemented are very adherent to the Italian IDEM federation (where the IdP in the Cloud project is operating); for a wider use of this piece of software, this logic parts should be more generalized and maybe configurable by an administrator.

With what was shared within Elcira project, however, it should be possible to replicate the GARR "IdP in the Cloud" project on different infrastructures and so it could be possible to offer similar services in different Identity Federations.

13. REFERENCES

- Openstack: http://www.openstack.org/
- Shibboleth: http://shibboleth.net/
- Puppet: http://puppetlabs.com/