Alma Mater Studiorum · Università di Bologna

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
Corso di Laurea Triennale in Informatica

STUDIO COMPARATIVO DEI SISTEMI DI GESTIONE DEI METADATI PER LE FEDERAZIONI DI IDENTITÀ

Tesi di Laurea in Reti di Calcolatori

Relatore:
Chiar.mo Prof.
FABIO PANZIERI

Presentata da:
MALAVOLTI MARCO

Correlatrice:
Dott.ssa
MARIA LAURA
MANTOVANI

Sessione II
Anno Accademico 2011/2012

Indice

1 Introduzione

Le Federazioni di Identità nel settore dell'istruzione e della ricerca [1] coinvolgono ormai decine di milioni di utenti e centinaia di organizzazioni [2].

Dal 2009 in Italia è attiva la Federazione di Identità Italiana IDEM [3] (*IDEntity Management* [4]*)* che ha lo scopo di realizzare un'infrastruttura di autenticazione e di autorizzazione (*Authentication Authorization Infrastructure - AAI*) [5] della rete GARR (rete telematica italiana dell'Università e della Ricerca con l'obiettivo di fornire connettività ad altissime prestazioni e servizi avanzati alla comunità di ricerca ed accademica italiana). La rete GARR è parte integrante dell'Internet mondiale e viene gestita dal Consortium GARR [6].

Attraverso una Federazione le Istituzioni e gli Organi di formazione, quali Università e Centri di Ricerca, possono condividere risorse e servizi con gli utenti in modo sicuro.

Ogni *Identity Provider* [4] e *Service Provider* [4] possiede un proprio file di metadati [7] che lo descrive e la Federazione deve amministrarli in modo da garantirne l'autenticità e l'integrità. In seguito verranno definiti meglio chi sono questi attori e come realizzano la condivisione delle risorse e dei servizi tra le Organizzazioni servendosi dello standard SAML v2.0 [12].

La loro gestione, però, sta' diventando un impegno troppo oneroso per l'operatore che ora svolge manualmente questo compito in IDEM, pertanto, questa Tesi ha lo scopo di comparare e analizzare diversi sistemi di gestione dei metadati, realizzati da federazioni estere, per valutarne i pro e i contro in una loro possibile adozione nella Federazione Italiana.

Per il corretto svolgimento di questa tesi è stato necessario avvalersi di una rete virtuale (`lab.unimo.it`) che ricreasse l'ambiente ideale di una Federazione di Identità.

Tale Federazione virtuale si compone di:

- 1 *Identity Provider* : `idp.lab.unimo.it`
- 4 *Service Provider*: `sp.lab.unimo.it`, `spemb.lab.unimo.it`, `spwin.lab.unimo.it` e `janus.lab.unimo.it`
- 1 *Discovery Service* [8]: `dsc.lab.unimo.it`
- 1 *Server LDAP*: `t-rex.unimore.it`

Tutte le macchine virtuali sopra elencate sono dotate dei *software open source*, *Shibboleth* [9] o *simpleSAMLphp* [10], impiegati anche dalle Federazioni di Identità reali per instaurare rapporti di fiducia tra i suoi membri.

I 6 terminali Linux (`idp.lab.unimo.it`, `sp.lab.unimo.it`, `spemb.lab.unimo.it`, `janus.lab.unimo.it`, `dsc.lab.unimo.it`, `t-rex.unimore.it`) verranno utilizzati mediante il protocollo SSH e un terminale testuale, mentre quello Windows (`spwin.lab.unimo.it`) sarà acceduto attraverso RDP e *desktop* remoto.

Il traffico in entrata e in uscita di ogni elaboratore è governato da due *firewall*:

- uno di rete, amministrato dall'Università di Modena e Reggio Emilia.
- uno di bordo, gestito all'interno dell'elaboratore stesso.

Nel corso di questa tesi è stato necessario configurarli adeguatamente per raggiungere la completa operatività delle applicazioni studiate.

Il *Server* LDAP (*Lightweight Directory Access Protocol*) è stato impiegato per gestire le utenze aventi accesso all'*Identity Provider* ed i loro attributi [11] .

1 Introduzione

Ci si aspetta di trovare delle applicazioni che consentono di gestire i metadati di una Federazione in modo semplificato, automatizzando quelle operazioni che ora, nella nostra federazione, sono eseguite manualmente, tra cui:

- Adeguata gestione delle nuove entità, *Identity Provider* e *Service Provider*.
- Firma dei metadati della Federazione e loro pubblicazione.
- Esposizione di un catalogo degli IdP e degli SP amministrati.
- Controllo degli attributi da rilasciare ad una o più risorse.

Nel 2° Capitolo verranno introdotti i concetti fondamentali alla comprensione delle Federazioni di Identità, mostrati quali sono i motivi che hanno avvantaggiano la loro formazione, illustrati cos'è un Identità Digitale [4,5] e descritti gli attori che permettono il corretto funzionamento delle Federazioni.

Nel 3° Capitolo verrà definito cos'è lo standard SAML 2.0, da cosa è composto e come realizza la comunicazione tra un *Identity Provider* e un *Service Provider*.

Nel 4° Capitolo verranno mostrati, non in dettaglio, cosa sono e come si compongono i metadati che una Federazione è chiamata ad amministrare e i cui membri utilizzano per scambiarsi reciprocamente informazioni.

Nel 5° Capitolo verranno analizzate le tre applicazioni web, utilizzate da altre federazioni estere, per la gestione dei metadati che saranno oggetto di studio di questa Tesi.

Le applicazioni esaminate saranno:

- *Resource Registry* [13] *(SWITCHaai Federation)*
- *Janus* [14] *(Danish Federation)*
- *PEER* [15] *(TERENA)*

Tali applicazioni sono state scelte perché usate dalle maggiori Federazioni di Identità estere e proposte dalla Federazione Italiana IDEM.

Al termine del capitolo, servendoci di una tabella riassuntiva, riporteremo l'esito del loro confronto.

Il 6° Capitolo verrà riservato alle conclusioni, con la possibilità di muovere proposte in merito all'adozione di uno o più di questi programmi per la Federazione Italiana IDEM.

2 Federazione di Identità

In questo capitolo verrà affrontato un breve percorso sulle motivazioni che hanno spinto diverse Istituzioni a riunirsi in un'unica Federazione di Identità e verranno illustrati i concetti di Identità Digitale, *Identity Provider*, *Service Provider* e Asserzione.

Al giorno d'oggi l'aspetto che certamente accomuna maggiormente le diverse Organizzazioni come Università, Centri di Ricerca o Biblioteche è sicuramente la gestione degli utenti.

Questi enti non si compongono solo di aule o lezioni, ma offrono anche numerosi servizi sul web, quali:

- Sistema di gestione dei prestiti di una biblioteca.
- Sistema di gestione delle riviste digitali.
- Sistema di eMail universitarie.
- Gestione del Database della documentazione per la ricerca.
- Sistema di *accounting* per l'accesso alla rete Wireless.

Una risorsa si dice protetta quando, per accedervi, è necessario fornire:

1. **Autenticazione**: un utente dimostra di essere davvero lui ad averne fatto richiesta. (es. *Username* e *Password*, impronte digitali, [...])
2. **Autorizzazione**: un utente utilizza la risorsa secondo i permessi che gli sono stati assegnati. (es. permesso di scrivere in un file, permesso di creare un file, [...])

In principio la soluzione adottata dalle Istituzioni per fornire agli utenti le loro risorse, era quella di proteggerle mediante un servizio di autenticazione e di autorizzazione come illustrato dalla **Figura 2.a**:

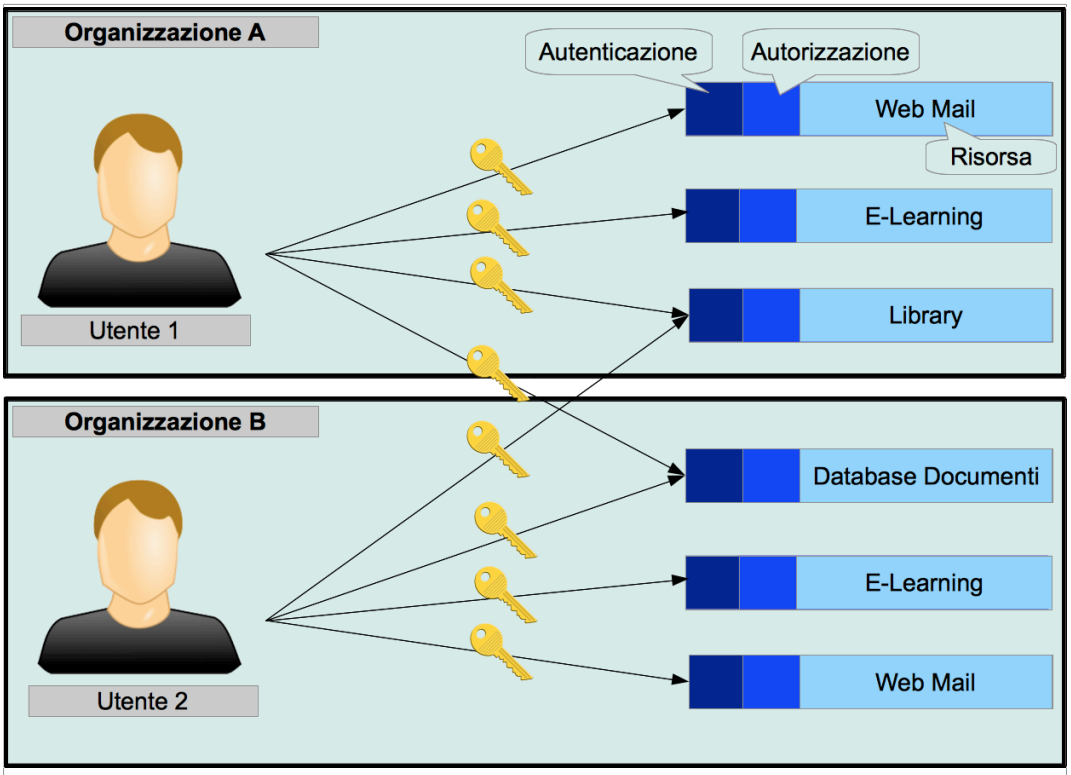


Figura 2.a - *Accesso alle Risorse senza AAI*

Come si può osservare, sia l'Utente 1 che l'Utente 2 sono costretti a collezionare una lunga serie di credenziali per riuscire ad avere accesso a quello che desiderano.

Si evince facilmente che questa soluzione non è conveniente in quanto:

- produce una notevole moltiplicazione dei dati personali
  (ogni risorsa richiede una registrazione per poter essere utilizzata).

- aumenta la probabilità di intrusione non autorizzata
  (gli utenti più pigri spesso riutilizzano i medesimi dati per registrarsi a più risorse aumentando così le possibilità che un malintenzionato se ne impossessi).
- rende oneroso il compito di gestire le credenziali di tutti gli utenti
  (il crescente aumento del numero di registrazioni porta a una sempre maggiore difficoltà della loro gestione e memorizzazione).

In risposta alle precedenti problematiche si è fatta avanti l'idea di poter impiegare un'infrastruttura dedicata alla sola Autenticazione e Autorizzazione (AAI) con la quale sarebbe stato possibile riunire tutte le Istituzioni aderenti in un'unica Federazione di Identità.

Lo sviluppo di una AAI Federata consente all'utente di disporre di tutte le risorse, condivise dalle Organizzazioni che ne fanno parte, attraverso una sola, ed unica, operazione di autenticazione e risolve i problemi dovuti alle credenziali multiple precedentemente incontrati.

La **Figura 2.b** della pagina seguente ne mostra un chiaro esempio.

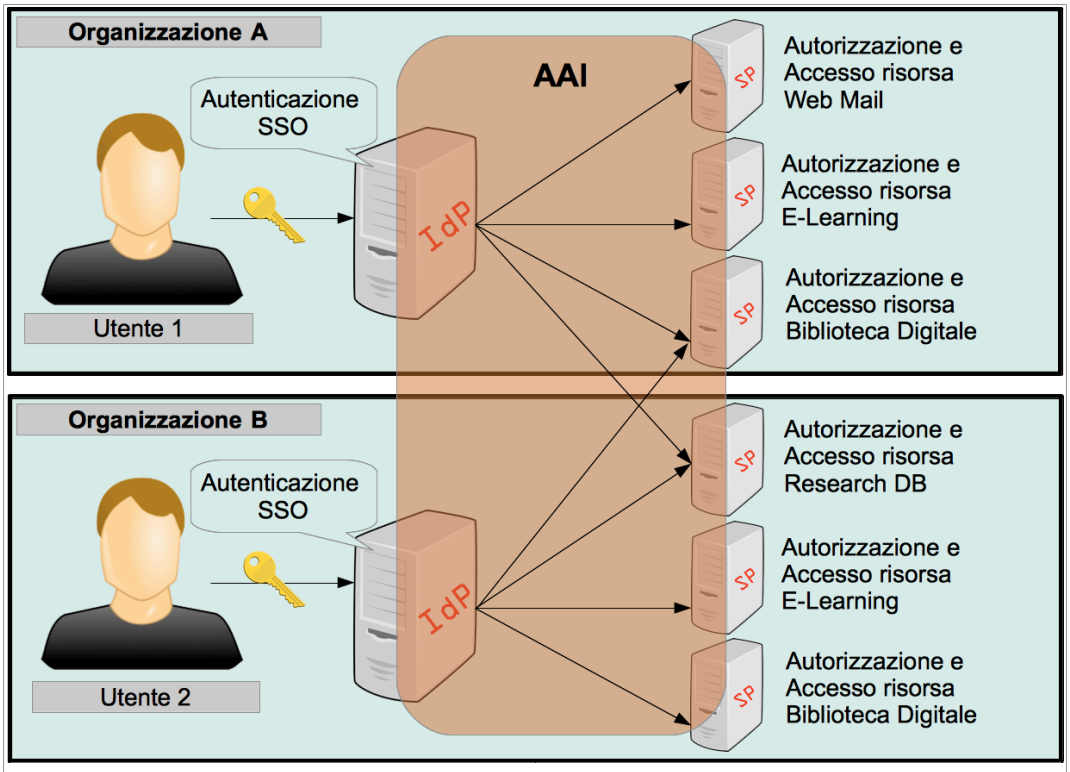


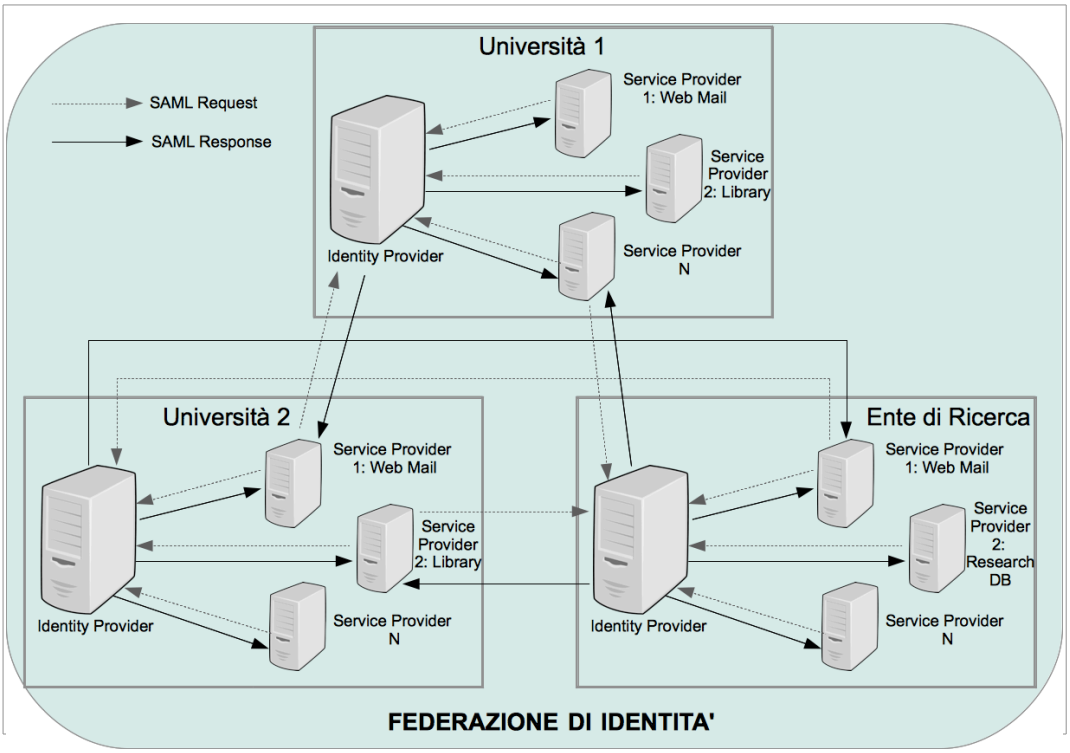
Figura 2.b - *Accesso alle Risorse con AAI Federata*

Dall'immagine si possono notare i seguenti cambiamenti:

- il numero delle registrazioni richieste all'utente è drasticamente diminuito.
- l'introduzione di un *Identity Provider (*IdP*)*.
  L'IdP si preoccupa di amministrare i dati personali degli utenti appartenenti alla medesima Organizzazione e fornire il servizio di autenticazione (*Single-Sign-On* - SSO) necessario per identificarli.
- l'introduzione di tanti *Service Provider* (SP).
  Il SP protegge una o più risorse di interesse e decide cosa l'utente, una volta autenticato presso l'IdP, può fare e cosa no con esse.

2 Federazione di Identità

<<*Una* Federazione *di Identità è un gruppo di Istituzioni e Organizzazioni che aderiscono ad un insieme di politiche condivise, riguardo gli utenti e le risorse, per permettere l'accesso a risorse e servizi. La federazione, congiunta ai sistemi di identity e access management delle istituzioni e delle organizzazioni, permette la gestione federata dell'accesso alle risorse.*>> [16]



Nel contesto italiano la Federazione di Identità prende il nome di IDEM, le istituzioni che vi aderiscono sono esclusivamente quelle rivolte all'istruzione (Università, Centri di Ricerca, Biblioteche) e gli utenti che possono utilizzare le risorse offerte possono essere studenti, ricercatori, docenti [...] che appartengono a tali organizzazioni.

Ad oggi, 21 Ottobre 2012, IDEM consta di 60 *Identity Provider(IdP)* e 72 *Service Provider(SP),* ma il loro numero è in continuo aumento.

Cosa spinge le diverse Organizzazioni a formare una Federazione di Identità? [17]

1. Fornire all'utente maggiori risorse (interne ed esterne).
2. Semplificare l'accesso alle stesse.
3. Proteggere i dati personali dell'utente, la sua Identità Digitale [5].

2.1 Identità Digitale

Una persona può essere rappresentata digitalmente attraverso un'insieme di informazioni che la contraddistinguono dalle altre e che la descrivono in modo inequivocabile e univoco.

La digitalizzazione di queste caratteristiche produce le Identità Digitali.

Spesso l'utilizzo di una risorsa o di un servizio non richiede l'intera Identità Digitale, ma si accontenta di una sua parte: l'Identità Parziale.

Tale Identità Parziale identifica in egual modo la persona, ma utilizza un numero minore di informazioni.

La successiva **Figura 2.d** mostra un esempio di questi due tipi di Identità.

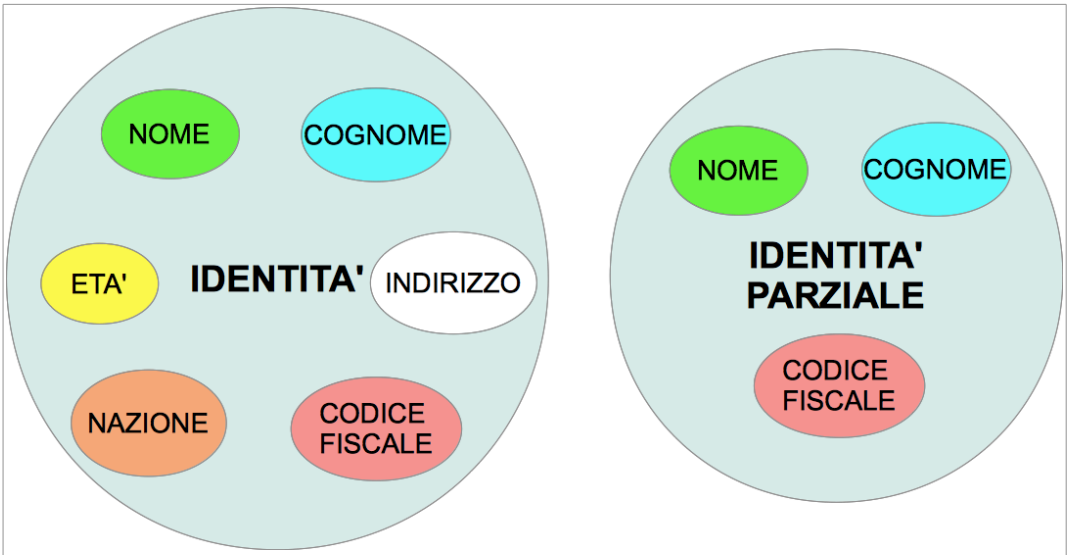


Figura 2.d - *Identità Totale e Parziale*

In una Federazione di Identità l'unità informativa che contiene la singola informazione, o dato, riguardante l'utente prende il nome di Attributo. L'identità digitale, parziale o totale, è formata da uno o più attributi.

2.2 Identity Provider, Service Provider e Asserzioni

Un'asserzione è il trasferimento di informazioni contenenti attributi e valori associati all'identità digitale di un utente.

Una Federazione di Identità si compone di due importanti attori:

1. *Identity Provider* (IdP) o *Asserting Party [4,5]*
2. *Service Provider* (SP) o *Relying Party [4,5]*

Un *Asserting Party* è l'entità che produce asserzioni contenenti le informazioni richieste da un *Relying Party*.

Un *Relying Party* è, invece, l'entità che consuma le asserzioni prodotte da un *Asserting Party* e inviate in risposta alla richiesta fatta dal *Relying Party* per concedere all'utente di utilizzare una sua risorsa protetta.

Un *Asserting Party* può produrre le seguenti asserzioni:

1. ***Authentication assertions*****:** asserzioni contenenti il risultato, positivo o negativo, del processo di autenticazione dell'utente.
2. ***Attribute assertions*****:** asserzioni contenenti gli attributi dell'utente.
3. ***Authorization Decision assertions*****:** asserzioni contenenti il risultato, positivo o negativo, del processo di autorizzazione che l'utente ha eseguito per ottenere la risorsa richiesta.

3 SAML 2.0 - Simple Assertion Markup Language

In questo capitolo definiremo cos'è lo standard SAML 2.0 e come viene utilizzato dalle entità di una federazione nello scambio di informazioni.

SAML 2.0, nato nel marzo del 2005, è uno standard che definisce un *framework XML-based* realizzato per la trasmissione e lo scambio di informazioni circa:

- gli attributi utente trasmessi e/o disponibili.
- le autorizzazioni che l'utente possiede sulla risorsa richiesta.
- l'autenticazione, avvenuta o meno, dell'utente al suo *Identity Provider*.

Questo standard sfrutta le tecnologie di protezione già esistenti offrendo, con i suoi profili, interoperabilità per una moltitudine di casi. Se non ve ne fosse uno adeguato, esso consente di crearne di nuovi in base alle necessità.

Si compone di:

- **Asserzioni:** porzioni di codice XML prodotte da una *SAML Authority*, come l'*Asserting Party*,contenenti informazioni riguardanti l'utente, la macchina o la persona e scambiate tra gli attori della Federazione durante le operazioni di autenticazione, autorizzazione e accesso alle risorse protette. Sono facilmente riconoscibili perché iniziano con `<saml:Assertion>` e sono contenute all'interno delle *SAML Response* inviate dall'*Asserting Party* al *Relying Party*.

- **Protocolli [18]:** definiscono come si richiedono e si ricevono le asserzioni da un *Asserting Party* attraverso lo scambio di questi messaggi:
  - *SAML Request*
    contenente la richiesta di informazioni circa un soggetto.
  - *SAML Response*
    contenente una o più asserzioni con le informazioni richieste dalla *SAML Request*.
- **Binding [18]:** definiscono come realizzare in SAML lo scambio di informazioni di sicurezza attraverso alcuni dei principali protocolli di trasporto (es.: HTTP o *Simple Object Access Protocol* - SOAP).
  Il *Binding* è un'operazione di mappatura dei messaggi "*SAML Request*" e "*SAML Response*" sui protocolli di comunicazione standard come HTTP o SOAP.
  Ne esistono diversi in SAML: *HTTP POST Binding*, *SAML SOAP Binding*, [...] [19]
- **Profili [18]:** definiscono alcuni scenari d'uso comune e come le asserzioni, i protocolli e i *binding* debbano essere combinati tra loro in procedure sufficientemente definite per soddisfare gli scopi per i quali tali scenari d'uso sono stati pensati.
  Tra i numerosi profili descritti nella documentazione ufficiale di OASIS [20], ne presentiamo qui, in modo generico, un esempio abbastanza completo: l' *Identity Provider Discovery Profile.*

Questo profilo ne utilizza un altro, il "*Web Browser SSO Profile*" [20], per descrivere l'interazione ai fini di autenticazione tra il *browser web* dell'utente, il *Relying Party* e *l'Asserting Party*.
Questo profilo coinvolge una terza entità, il *Discovery Service*, che permette all'utente di autenticarsi al proprio *Asserting Party* scegliendolo tra quelli aderenti alla Federazione e presentati in forma di elenco.
La sua implementazione è libera di scegliere una qualsiasi combinazione di *binding* come per esempio l' *HTTP POST Binding* realizzabile con i seguenti passi:

1. Richiesta di accesso alla risorsa: l'utente richiede di accedere alla risorsa che desidera utilizzare, il *Relying Party* produce una *SAML Request*.
2. *Redirect* verso il *Discovery Service*: l'utente sceglie il suo *Asserting Party* dall'elenco che gli viene mostrato e ne dà conferma.
3. *Redirect* verso il servizio di *Single Sign-On (SSO)*: l'utente viene rediretto al suo *Asserting Party* con la *SAML Request* del *Relying Party* che protegge la risorsa richiesta dall'utente.
4. Richiesta delle credenziali dell'utente mediante una *form* HTML: l'utente visualizza la *form* HTML di autenticazione del suo *Asserting Party*.
5. Risposta utente: l'utente inserisce le sue credenziali di autenticazione.
6. Preparazione *SAML Response* con *asserzione*/i e invio della stessa al *Assertion Consumer Service* del *Relying Party* che ha effettuato la *SAML Request*.

7. *Redirect* verso la risorsa richiesta: l'utente viene re-direzionato, ora abilitato al suo utilizzo, alla risorsa richiesta.
8. Accesso, dell'utente, alla risorsa protetta voluta.

3.1 Shibboleth e SimpleSamlPhp

Shibboleth e s*impleSAMLphp* sono framework *open source* basati sullo standard SAML. Attraverso di essi è possibile realizzare il *Single Sign-On* oltre i confini dell'istituzione autenticando gli utenti ai loro *Identity Provider* e autorizzandoli poi ad accedere alle risorse protette dai *Service Provider* in base alle informazioni scambiate tra le Organizzazioni.
Shibboleth e *simpleSAMLphp* vengono impiegati per realizzare le AAI federate consentendo ai diversi membri di una Federazione di offrire maggiori risorse ai propri utenti mediante l'impiego di un unico servizio di autenticazione.

Questi *framework* forniscono il supporto *software* in grado di creare gli *Identity Provider,* i *Service Provider* e il *Discovery Service* con i quali realizzare l'infrastruttura AAI.

4 Metadati

In questo capitolo verranno definiti cosa sono i metadati per le Federazioni di Identità, perché sono necessari, che struttura seguono e come si compongono. Nel paragrafo 4.3 verrà specificata una loro estensione destinata alle Interfacce che un utente può incontrare.

Le Organizzazioni che partecipano ad una Federazione hanno bisogno di:

1. conoscere quali risorse (SP) e quali servizi di gestione e verifica delle Identità (IdP) sono disponibili.
2. stabilire con certezza che il destinatario delle proprie asserzioni è realmente chi dice di essere.
3. conoscere quali attributi rilasciare per soddisfare la richiesta di una determinata risorsa.
4. conoscere cosa una risorsa è in grado di fare, qual è l'ente che lo gestisce e a chi rivolgersi per eventuali problemi.
5. stabilire una relazione di fiducia con gli altri membri della Federazione.

Per tutti questi punti, e non solo, una Federazione di Identità si occupa del raccoglimento e della gestione dei metadati provenienti dai suoi partecipanti.

I metadati sono file, in formato XML, in grado di descrivere le entità di un'Organizzazione in modo standardizzato. Con essi è possibile stabilire i rapporti di fiducia necessari tra i membri della Federazione e realizzare gli scenari d'uso comune delineati dai profili definiti dalle specifiche SAML [20].

In questo capitolo non si entrerà nei dettagli della descrizione completa di un file di metadati, ma si darà uno sguardo alle sole componenti principali che la **Figura 4.a** mostra.

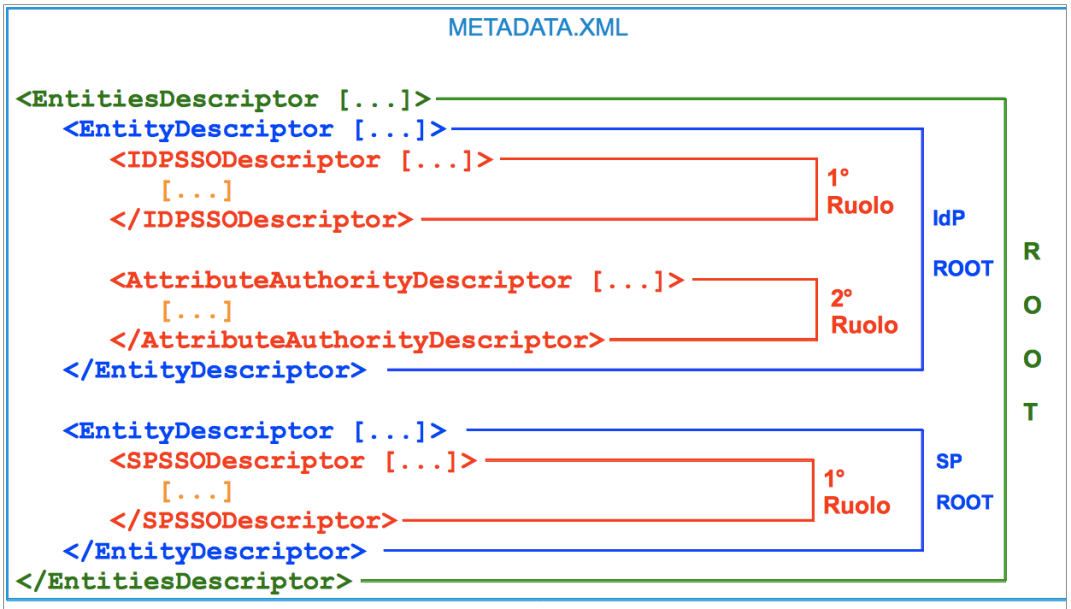


Figura 4.a - *Struttura di un file di Metadati*

Per la stesura di questo capitolo, utilizzeremo i seguenti documenti OASIS:

1. *Metadata for the SAML V2.0* [21]
2. *Metadata Extensions for Login and Discovery User Interface Version 1.0* [22]

4.1 Elementi ROOT

Un file di metadati SAML può contenere o una singola entità o molteplici entità. Nel primo caso l'elemento ROOT (radice) dovrà essere `<md:EntityDescriptor>`, mentre nel secondo caso dovrà essere `<md:EntitiesDescriptor>`.

Elemento <md:EntitiesDescriptor>

Questo elemento contiene i metadati di un gruppo di entità SAML e può, eventualmente, avere un nome per essere facilmente reperibile.
Gli attributi e gli elementi XML che lo compongono sono:

- `ID` [Facoltativo]
  identificatore univoco dell'elemento nel documento XML.
- `validUntil` [Facoltativo]
  periodo di tempo in cui i metadati, contenuti nell'elemento, restano validi.
- `caheDuration` [Facoltativo]
  massimo periodo di tempo concesso all'entità consumatrice (SP) per memorizzare i metadati qui contenuti.
- `Name` [Facoltativo]
  nome identificativo dell'insieme di entità SAML che l'elemento raccoglie.
- `<ds:Signature>` [Facoltativo]
  firma XML che autentica l'intero contenuto dell'elemento.
- `<md:Extensions>` [Facoltativo]
  estensioni che completano i metadati nella descrizione dell'entità e che vengono concordate tra l'entità che li pubblica e quella che li consuma.

- `<md:EntitiesDescriptor>` o `<md:EntityDescriptor>` [Uno o più]

  contengono i metadati di una o più entità SAML.

Quando `<md:EntitiesDescriptor>` viene usato come elemento ROOT deve contenere o l'attributo XML `validUntil` o `cacheDuration`.
La documentazione ufficiale OASIS raccomanda di usare tali attributi SOLO per l'elemento ROOT.

Elemento <md:EntityDescriptor>

Questo elemento definisce e contiene i metadati di una singola entità SAML.
Si compone dei seguenti elementi e attributi XML:

- `entityID` [Obbligatorio]

  identificativo univoco per l'entità SAML i cui metadati vengono descritti dall'elemento.

- `ID` [Facoltativo]

  identificatore univoco dell'elemento nel documento XML.

- `validUntil` [Facoltativo]

  periodo di tempo in cui i metadati, contenuti nell'elemento, restano validi.

- `cacheDuration` [Facoltativo]

  massimo periodo di tempo concesso all'entità consumatrice (SP) per memorizzare i metadati qui contenuti.

- `<ds:Signature>` [Facoltativo]

  firma XML che autentica l'intero contenuto dell'elemento.

- `<md:Extensions>` [Facoltativo]

  estensioni che completano i metadati nella descrizione dell'entità e che vengono concordate tra l'entità che li pubblica e quella che li consuma.

- Elementi Descrittori di Ruoli:

  `<md:RoleDescriptor>` , `<md:IDPSSODescriptor>` , `<md:SPSSODescriptor>`, `<md:AuthnAuthorityDescriptor>`, `<md:AttributeAuthorityDescriptor>`, `<md:PDPDescriptor>` [Uno o più]

  OPPURE

  `<md:AffiliationDescriptor>` [Obbligatorio se utilizzato al posto dei ruoli soprastanti]

  può contenere, in modo esclusivo:

  - uno o più elementi descrittori di ruoli (elencati precedentemente).
  - uno specifico descrittore che definisce l'affiliazione tra le diverse entità.

- `<md:Organization>` [Facoltativo, ma caldamente consigliato]

  contiene le informazioni sull'Organizzazione responsabile dell'entità SAML descritta.

  Si compone dei seguenti attributi ed elementi XML:

  - `<md:Extensions>` [Facoltativo]

    estensioni che completano i metadati nella descrizione dell'entità e che vengono concordate tra l'entità che li pubblica e quella che li consuma.

  - `<md:OrganizationName>` [Uno o più]

    sequenza di traduzioni del nome dell'Organizzazione che possono, o no, essere comprensibili all'uomo.

- `<md:OrganizationDisplayName>` [Uno o più]

  sequenza di traduzioni del nome dell'Organizzazione rese comprensibili all'uomo.

- `<md:OrganizationURL>` [Uno o più]

  uno o più URI che indicano all'utente dove ottenere maggiori informazioni sull'Organizzazione.

- `<md:ContactPerson>` [Facoltativo, ma caldamente consigliato]

  contiene le informazioni sui diversi contatti e che gestiscono l'entità descritta.

  Si compone dei seguenti attributi ed elementi XML:

  - `contactType` [Obbligatorio]

    tipo di contatto che si andrà a presentare.

    I valori possibili sono: `technical`, `support`, `administrative`, `billing` e `other`.

  - `<md:Extensions>` [Facoltativo]

    estensioni che completano i metadati nella descrizione dell'entità e che vengono concordate tra l'entità che li pubblica e quella che li consuma.

  - `<md:Company>` [Facoltativo]

    nome della compagnia a cui appartiene il contatto.

  - `<md:GivenName>` [Facoltativo]

    nome del contatto.

  - `<md:SurName>` [Facoltativo]

    cognome del contatto.

  - `<md:EmailAddress>` [Zero o più]

    indirizzo/i email del contatto.

- `<md:TelephoneNumber>` [Zero o più]

  numero/i di telefono del contatto.

- `<md:AdditionalMetadataLocation>` [Zero o più]

  sequenza di locazioni di *namespaces* qualificati in cui esistono dei metadati aggiuntivi per le entità SAML. Possono includere metadati in formati alternativi o descrivere l'adesione a specifiche non-SAML.

Quando `<md:EntityDescriptor>` viene usato come elemento di ROOT esso deve contenere o l'attributo XML `validUntil` o `cacheDuration`.
La documentazione ufficiale OASIS raccomanda di usare tali attributi SOLO per l'elemento ROOT.

4.2 Elementi che descrivono i ruoli delle entità SAML

Elemento <md:RoleDescriptor>

`<md:RoleDescriptor>` è l'elemento alla base della costruzione di tutti quelli che descrivono i ruoli che un'entità può ricoprire per realizzare i profili SAML.

Si compone dei seguenti elementi e attributi XML:

- `ID` [Facoltativo]

  identificatore univoco dell'elemento nel documento XML.

- `validUntil` [Facoltativo]

  periodo di tempo in cui i metadati, contenuti nell'elemento, restano validi.

- `cacheDuration` [Facoltativo]

  massimo periodo di tempo concesso all'entità consumatrice (SP) per memorizzare i metadati qui contenuti.

- `protocollSupportEnumeration` [Obbligatorio]

  insieme di URI, separate da spazi, che identificano il set di protocolli supportati dall'entità. Per le entità SAML v2.0 questo set deve includere "`urn:oasis:names:tc:SAML:2.0:protocol`"

- `errorURL` [Facoltativo]

  URI che specifica la locazione a cui re-direzionare un utente per ricevere supporto o maggiori informazioni sull'entità descritta.

- `<ds:Signature>` [Facoltativo]

  firma XML che autentica l'intero contenuto dell'elemento.

- `<md:Extensions>` [Facoltativo]

  estensioni che completano i metadati nella descrizione dell'entità e che vengono concordate tra l'entità che li pubblica e quella che li consuma.

- `<md:KeyDescriptor>` [Zero o più]

  sequenza opzionale di elementi che forniscono informazioni sulle chiavi di criptazione che l'entità andrà ad utilizzare.
  Si compone dei seguenti attributi ed elementi XML:

  - `use` [Facoltativo]

    attributo che descrive la funzione dalla chiave.
    Può assumere i valori "`signing`" o "`encryption`".

  - `<ds:KeyInfo>` [Obbligatorio]

    elemento che identifica direttamente o indirettamente la chiave.

  - `<md:EncryptionMethod>` [Zero o più]

    uno o più elementi opzionali che indicano gli algoritmi di criptazione supportati dall'entità.

- `<md:Organization>` [Facoltativo]

  contiene le informazioni sull'Organizzazione responsabile del ruolo descritto da questo elemento.

- `<md:ContactPerson>` [Zero o più]

  contiene le informazioni sui diversi contatti responsabili del ruolo descritto da questo elemento.

Elementi che descrivono i ruoli di IdP e SP

I descrittori dei ruoli ricoperti da un *Identity Provider* e da un *Service Provider* sono `<md:IDPSSODescriptor>` e `<md:SPSSODescriptor>`.

Entrambi gli elementi estendono l'elemento base `<md:RoleDescriptor>` con i propri elementi XML di cui condividono i seguenti:

- `<md:ArtifactResolutionService>` [Zero o più]

  zero o più elementi indicanti gli *endpoint* che supportano il protocollo *Artifact Resolution* definito in [20].
  Il `ResponseLocation` DEVE essere omesso.

- `<md:SingleLogoutService>` [Zero o più]

  zero o più elementi indicanti gli *endpoint* che supportano il *Single Logout profile* definito in [20].

- `<md:ManageNameIDService>` [Zero o più]

  zero o più elementi indicanti gli *endpoint* che supportano il *Name Identifier Management profile* definito in [20].

- `<md:NameIDFormat>` [Zero o più]

  zero o più elementi indicanti i *name identifier* [23] supportati dall'entità.

Elemento <md:IDPSSODescriptor>

Questo elemento aggiunge ai metadati i contenuti propri di un *Identity Provider* per supportare il SSO nei profili SAML che lo richiedono.

Si compone dei seguenti elementi e attributi XML:

- `WantAuthnRequestsSigned` [Facoltativo]

  attributo booleano opzionale indicante la volontà o meno di ricevere richieste di autenticazione firmate da parte del *Service Provider.*

- `<md:SingleSignOnService>` [Uno o più]

  uno o più elementi indicanti gli *endpoint* che supportano l' *Authentication Request profile* definito in [20]. Tutti gli *Identity Providers* supportano almeno uno di questi *endpoint* e il `ResponseLocation` DEVE essere omesso.

- `<md:NameIDMappingService>` [Zero o più]

  zero o più elementi indicanti gli *endpoint* che supportano l' *Identifier Mapping profile* definito in [20].
  Il `ResponseLocation` DEVE essere omesso.

- `<md:AssertionIDRequestService>` [Zero o più]

  zero o più elementi indicanti gli *endpoint* che supportano il *SAMLAssertion Query/Request profile* definito in [20] o una URI speciale di *binding* per le richieste di asserzioni definita in [19].

- `<md:AttributeProfile>` [Zero o più]

  zero o più elementi che elencano i *SAML Attribute Profiles* supportati dall'*Identity Provider* descritto.

- `<saml:Attribute>` [Zero o più]

  zero o più elementi che rappresentano gli attributi SAML supportati dall'*Identity Provider.*
  In questo contesto il supporto di un attributo indica la capacità, da parte dell'IdP, di poterlo includere in un'asserzione.

Elemento <md:SPSSODescriptor>

Questo elemento aggiunge ai metadati contenuti propri di un *Service Provider* per supportare il SSO nei profili SAML che lo richiedono.
Si compone dei seguenti elementi e attributi XML:

- `AuthnRequestSigned` [Facoltativo]

  attributo booleano opzionale che indica se le richieste di autenticazione inviate all'*Identity Provider* debbano essere firmate o no. Se omesso assume il valore di `false`.

- `WantAssertionsSigned` [Facoltativo]

  attributo booleano opzionale che indica la volontà, del *Service Provider,* di ricevere asserzioni firmate dall'IdP.
  Se omesso assume il valore di `false`.

- `<md:AssertionConsumerService>` [Uno o più]

  uno o più elementi che indicano gli *endpoint* che supportano il protocollo *Authentication Request* definito in [20].
  Ogni *Service Provider* supporta almeno uno di questi *endpoint*.

- `<md:AttributeConsumingService>` [Zero o più]

  zero o più elementi che descrivono un'applicazione o un servizio fornito dal *Service Provider* che richiede o desidera utilizzare attributi SAML. Uno solo di questi elementi può contenere l'attributo XML `isDefault` settato a `true`.
  Si compone dei seguenti attributi ed elementi XML:

  - `index` [Obbligatorio]

    attributo richiesto che identifica l'elemento con un numero intero unico.

  - `isDefault` [Facoltativo]

    attributo booleano indicante il servizio predefinito fornito dal *Service Provider*.
    Se omesso assume il valore di `false`.

  - `<md:ServiceName>` [Uno o più]

    elementi contenenti il nome dei servizi offerti tradotti in diverse lingue.

  - `<md:ServiceDescription>` [Zero o più]

    elementi contenenti la descrizione dei servizi offerti tradotti in diverse lingue.

  - `<md:RequestAttribute>` [Uno o più]

    elementi contenenti gli attributi SAML richiesti dal servizio per poter essere utilizzato.
    Si compone dei seguenti attributi ed elementi XML:

    - `isRequired` [Opzionale]

      indica l'obbligatorietà di fornire l'attributo SAML richiesto.

- `Name` [Obbligatorio]

  indica il nome dell'attributo SAML richiesto.

- `NameFormat` [Facoltativo]

  indica il formato del nome dell'attributo SAML richiesto.

- `FriendlyName` [Facoltativo]

  contiene il nome dell'attributo SAML in un formato umanamente comprensibile.

- `<md:AttributeValue>` [Zero o più]

  zero o più elementi contenenti i diversi valori che l'attributo SAML fornito può assumere.

Elemento <md:AttributeAuthorityDescriptor>

Questo elemento estende i metadati dell'elemento base `<md:RoleDescriptor>` con contenuti specifici delle *Attribute Authority*, l'autorità SAML che risponde ai messaggi `<samlp:AttributeQuery>`.
Esso si compone dei seguenti elementi XML:

- `<md:AttributeService>` [Uno o più]

  uno o più elementi che indicano gli *endpoint* che supportano l' *Attribute Query profile* definito in [20].
  Tutte le *Attribute Authority* supportano almeno uno di questo *endpoint.*

- `<md:AssertionIDRequestService>` [Zero o più]

  uno o più elementi che indicano gli *endpoint* che supportano l' *Assertion Query/Request profile* definito in [20] o una URI speciale di *binding* per le richieste di asserzione definite in [19].

- `<md:NameIDFormat>` [Zero o più]

  zero o più elementi indicanti i formati di *name identifier* [23] supportati dall'autorità.

- `<md:AttributeProfile>`

  zero o più elementi che elencano i profili sugli attributi SAML supportati da questa autorità.

- `<saml:Attribute>`

  zero o più elementi che indicano gli attributi SAML supportati dall'*Attribute Authority.* Si possono specificare, eventualmente, i valori permessi per tali attributi SAML.

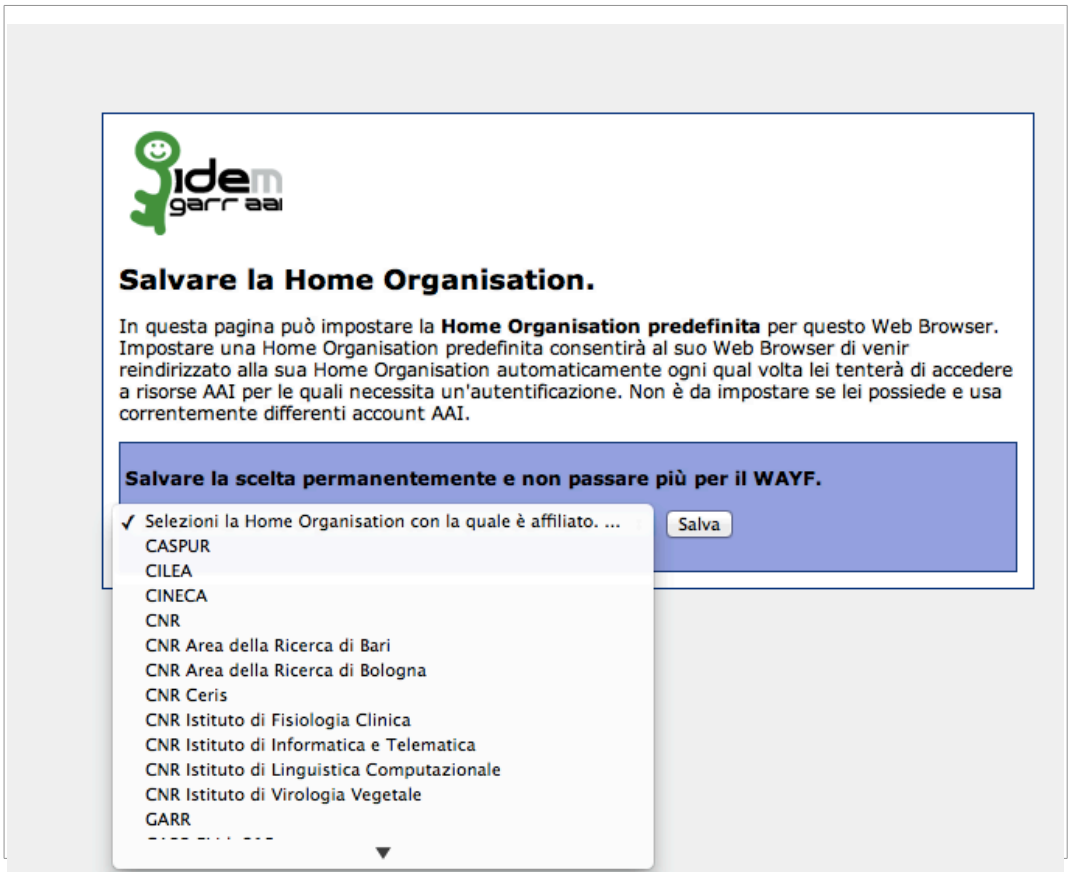
4.3 Estensione dei Metadati per le Interfacce Utente (MDUI)

La maggior parte dei profili SAML coinvolge uno *user-agent* per interagire con l'utente nella realizzazione del singolo profilo SAML.

Lo *user-agent* comunica con l'individuo attraverso delle interfacce, presentate dalle entità, che gli consentono di accedere e utilizzare le risorse da loro protette.

Esistono, in genere, tre interfacce utente:

1. **Interfaccia Utente del *Discovery Service (DS)*:**
   il *Discovery Service (DS)* è un *Service Provider* che permette all'utente di individuare, selezionare e raggiungere il proprio IdP per effettuare l'autenticazione richiesta dalla risorsa desiderata.
   Questa interfaccia si presenta come una pagina web in cui è possibile scegliere il proprio IdP. Spesso viene impiegato un elenco, come mostrato nella **Figura 4.3.a**, ma è il suo proprietario a decidere come mostrare i diversi *Identity Provider* selezionabili.



2. **Interfaccia Utente dell'*Identity Provider* (IdP):**
   Attraverso di essa l'utente può autenticarsi mediante le credenziali previste dalla sua Organizzazione.
   Non sono posti vincoli al suo miglioramento grafico, ma solo chi ha accesso all'*Identity Provider* può modificare tale interfaccia.

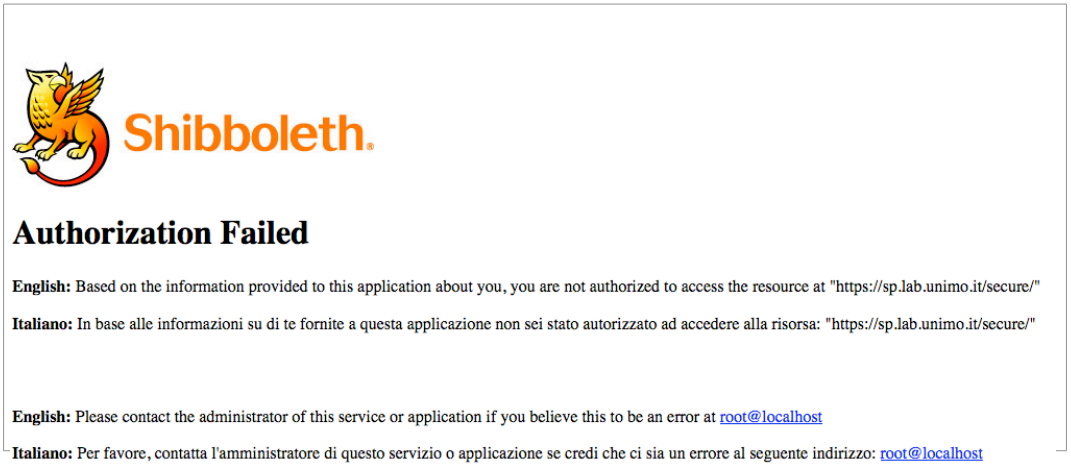
   Le informazioni che dovrebbero comparire in questa interfaccia sono:

   - I dati caratteristici dell'IdP su cui si effettua l'autenticazione. (utili all'utente per riconoscerlo)
   - Un riferimento alla risorsa protetta (SP). (utile per non dimenticare il motivo per cui è stata richiesta l'autenticazione)
   - Uno o più riferimenti di supporto. (utili per fornire aiuto in caso di necessità)

   La **Figura 4.3.b** ne mostra, nella pagina successiva, un semplice esempio.


Single SignOn UniMoRe
idem garr aai
UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA
Login
User ID
Password
Login
Accedi con Smart-Card. (Informazioni)
Versione italiana
English Version
Il servizio a cui si sta accedendo richiede l'inserimento delle proprie credenziali UniMORE:
per gli studenti: le credenziali (User ID e PIN) le sono state rilasciate al momento della registrazione sul servizio Esse3;
si può cambiare la password semplicemente inserendo l'ultima password impostata.
Ulteriori informazioni nella sezione delle domande ricorrenti.
per tutti gli altri utenti: le credenziali sono state scelte da lei al momento dell'identificazione o alla presa di servizio; nel caso non le ricordi può rivolgersi all'incaricato dell'identificazione presso la sua Struttura di afferenza. Consultare l'elenco degli incaricati
Contattare il supporto tecnico, specificando il servizio cui state accedendo (https://sp-elearning.u-gov.it /shibboleth), lo username usato e il tipo di errore ricevuto.

3. Interfaccia Utente del *Service Provider* (SP):
   Un *Service Provider* protegge le risorse che ospita e ne gestisce gli accessi mediante specifiche regole sugli attributi utente, forniti da un IdP in seguito alla sua autenticazione.
   Se l'identità autenticata possiede i giusti requisiti l'SP concede l'utilizzo della risorsa desiderata, altrimenti mostra una pagina web di errore per un'autorizzazione fallita e ne blocca l'accesso.
   Il miglioramento grafico della pagina non è limitato così come le informazioni che essa può contenere, ma solo chi ha accesso al *Service Provider* può modificarle.
   Le informazioni che dovrebbero comparire in questo tipo di interfaccia sono:
   - Contatto per la risoluzione dell'errore.
   - Riferimento alla risorsa non concessa.
   - Causa dell'errore riscontrato.
   - Nella **Figura 4.3.c** sottostante ne viene mostrato un esempio.



Di seguito verrà presentata l'estensione dei metadati usata dagli *user-agent* per la costruzione delle interfacce da presentare agli utenti.

Tutti gli elementi che si andranno a definire in questa sezione dovranno essere inclusi SOLAMENTE all'interno dell'elemento `<md:Extensions>` di un descrittore di ruolo (`<md:IDPSSODescriptor>`, `<md:SPSSODescriptor>`, `<md:AttributeAuthorityDescriptor>`, [...]) e tale `<md:Extensions>` potrà contenere al massimo una di queste estensioni.

Elemento <mdui:UIInfo>

Questo elemento contiene le informazioni necessarie alla creazione delle interfacce utente per operazioni quali: selezione di un *Identity Provider*, autenticazione utente, consenso al rilascio degli attributi [...]

Questo elemento può contenere un numero qualsiasi, e in qualsiasi ordine, dei seguenti elementi XML:

- `<mdui:DisplayName>`

  nome localizzato, tradotto nella lingua locale, dell'entità da mostrare all'utente. Questo elemento dovrebbe aiutare nello sviluppo di interfacce utilizzabili da persone diversamente abili. Possiede l'attributo XML:

  - `xml:lang` [Obbligatorio]

    lingua di con cui è scritto il contenuto dell'elemento.

  Non è possibile avere più elementi `<mdui:DisplayName>` con `xml:lang` identici all'interno di un singolo descrittore di ruolo.

- `<mdui:Description>`

  descrizione localizzata, tradotta in lingua locale, dell'entità.
  Nel caso di un `<md:SPSSODescriptor>` questo elemento DEVE contenere un testo che descrive il servizio offerto dal *Service Provider*.
  Nel caso di un `<md:IDPSSODescriptor>` questo elemento DEVE contenere un testo che descrive la comunità di utenti che mantiene. Possiede l'attributo XML:

  - `xml:lang` [Obbligatorio]

    lingua con cui è scritto il contenuto dell'elemento.

- `<mdui:Keywords>`

  Questo elemento contiene una elenco di parole chiave localizzate, inerenti al ruolo svolto dall'entità, da utilizzare nelle ricerche. Queste parole sono separate tra loro da spazi che vengono codificati come "`+`" impedendo a tale carattere di comparire tra loro. Possiede l'attributo XML:

  - `xml:lang` [Obbligatorio]

    lingua di con cui è scritto il contenuto dell'elemento.

  Non è possibile avere più elementi `<mdui:Keywords>` con `xml:lang` identici all'interno di un singolo descrittore di ruolo.

- `<mdui:Logo>`

  URL contenente l'immagine del logo localizzata per l'entità.
  Possiede i seguenti attributi XML:

  - `height` [Obbligatorio]

    altezza del logo in pixel.

  - `width` [Obbligatorio]

    larghezza del logo in pixel.

- `xml:lang` [Facoltativo]

  lingua di appartenenza del contenuto dell'elemento.

  Per facilitare l'inserimento dei loghi all'interno delle interfacce utente, i loghi dovrebbero:

  1. usare uno sfondo trasparente ove necessario.
  2. usare immagini PNG o GIF.
  3. usare URL HTTPS.

  L'ordine degli elementi `<mdui:Logo>` non è importante.
  Se l'elemento non presenta l'attributo `xml:lang`, tale logo potrà essere usato in qualsiasi caso e lingua.

- `<mdui:InformationURL>`

  URL contenente le informazioni localizzate sull'entità descritta. Il contenuto indicato dall'URL dovrebbe fornire informazioni più complete rispetto a quelle fornite dall'elemento `<mdui:Description>`. Possiede l'attributo XML:

  - `xml:lang` [Obbligatorio]

    lingua con cui sono scritte le informazioni date dall'URL contenuto dell'elemento.

  Non è possibile avere più elementi `<mdui:InformationURL>` con `xml:lang` identici all'interno di un singolo descrittore di ruolo.

- `<mdui:PrivacyStatementURL>`

  URL contenente le informazioni localizzate circa la *privacy* dell'entità. Questa dichiarazione dovrebbe fornire informazioni su come i dati dell'utente vengono usati dall'entità per svolgere il suo ruolo. Possiede il seguente attributo XML:

- `xml:lang` [Obbligatorio]

  lingua con cui sono scritte le informazioni sulla Privacy date dall'URL contenuto dell'elemento.

Non è possibile avere più elementi `<mdui:PrivacyStatementURL>` con `xml:lang` identici all'interno di un singolo descrittore di ruolo.

Element <mdui:DiscoHints>

Questo elemento contiene informazioni utili a un *Discovery Service* per determinare l'*Identity Provider* (o gli IdP), a cui l'utente è associato. Esso deve essere necessariamente contenuto in un `<md:Extensions>` di un elemento `<md:IDPSSODescriptor>`.

Un *Discovery Service* è un particolare SP in grado di fornire all'utente un'interfaccia grafica con cui scegliere l'IdP al quale autenticarsi.
Questo elemento può contenere un numero qualsiasi dei seguenti elementi XML e in un qualsiasi ordine:

- `<mdui:IPHint>`

  Questo elemento specifica i blocchi di indirizzi IPv4 o IPv6 associati o serviti dall'entità. Ogni `<mdui:IPHint>` può contenere un blocco di indirizzi IP sottoforma di stringa.

- `<mdui:DomainHint>`

  Questo elemento indica i nomi dei DNS associati o serviti dall'entità. Ogni elemento `<mdui:DomainHints>` può contenere un DNS associato all'entità sottoforma di stringa.

- `<mdui:GeolocationHint>`

  Questo elemento specifica un insieme di coordinate geografiche associate o servite dall'entità. Ogni elemento `<mdui:GeolocationHint>` può contenere una coppia di coordinate associate all'entità in forma di URI usando lo schema RFC5870 [24].

4.4 Metadati per le Federazioni

Non esiste alcun vincolo per cui un'Organizzazione sia obbligata a federarsi, basta che si accordi privatamente con l'ente che fornisce la risorsa desiderata e scambi con essa i propri metadati.

Con un accordo privato gli unici servizi e risorse, resi a disposizione degli utenti, sono quelli distribuiti dalle istituzioni che lo hanno preso. Inoltre, i metadati che le entità coinvolte producono, devono essere gestiti autonomamente dagli enti coinvolti, causando un notevole dispendio di tempo e un aumento generale delle probabilità di errore.

Il continuo progresso delle Organizzazioni e delle loro risorse, potrebbe portare alla necessità di gestire molti più metadati di quanti fossero stati preventivati al momento dell'accordo.

Questo, come già anticipato, produrrebbe un enorme dispendio di tempo nella sola gestione dei metadati e nella correzione di eventuali errori insorti.

Qual è, dunque, l'utilità di una Federazione nel gestire i metadati?

- essa raccoglie e firma i metadati dei suoi partecipanti, garantendo che quelli da lei trasmessi saranno autentici e integri.
- mantiene i metadati aggiornati e corretti.
- fornisce, eventualmente, il supporto tecnico e *software* necessari per l'installazione delle entità da utilizzate.
- crea e gestisce, eventualmente, un catalogo dei servizi di autenticazione (IdP) e un catalogo delle risorse (SP).
- organizza, eventualmente, giornate di formazione per presentarsi alle nuove organizzazioni o formare meglio quelle già inserite.

Coinvolgere un sempre maggior numero di utenze vuol dire aumentare la visibilità dei servizi e delle risorse condivise.

5 Applicazioni per la Gestione dei Metadati nelle Federazioni di Identità

In questo capitolo si andranno a comparare diverse applicazioni, attualmente utilizzati da altre Federazioni europee, con l'obiettivo di individuare una possibile soluzione alla gestione dei metadati manuale adottata ora nella Federazione Italiana IDEM.

Il *Resource Registry*, *Janus* e *PEER* dovrebbero semplificare le operazioni di inserimento da parte dei nuovi membri, permettere di visionare i metadati altrui e gestire i propri nel modo più chiaro e dettagliato possibile.

Verrà affrontato come ultimo elemento dell'analisi l'applicazione *PEER*, in quanto si diversifica dalle altre per il pubblico a cui è rivolto.

Mentre *Resource Registry* e *Janus* sono pensati per gestire servizi e risorse localmente alla propria Federazione, *PEER* è stato progettato per gestire i metadati delle risorse, rese disponibili dalle Organizzazioni, col mondo.

Ogni entità utilizzata in questo studio dispone di 2 certificati distinti:

1. Un certificato rilasciato da una CA:
   utilizzato da un *web server* per fornire il servizio HTTPS.
2. Un certificato autofirmato o *self-signed*:
   - nel caso sia un IdP questo certificato viene utilizzato per firmare le asserzioni SAML e per essere identificato da un SP che lo interroga per ricevere gli attributi necessari.
   - nel caso sia un SP esso viene utilizzato per firmare le richieste di autenticazione che vengono trasmesse ad un IdP e per decriptare le asserzioni SAML che vengono ricevute dall'IdP autenticato.

Ogni applicazione verrà esaminata trattando i seguenti punti:

A) Introduzione: Rapido sguardo alla caratteristiche preliminari definite per l'applicazione e eventuale configurazione stabilita per lo studio.

B) Funzionamento: Breve descrizione sull'utilizzo che uno o più utenti possono farne e come vengono gestite le informazioni da loro inserite.

C) Considerazioni Personali : Raccolta di osservazioni personali sulle caratteristiche dell'applicazione ritenute utili ai fini di questa tesi.

D) Problemi riscontrati : Collezione delle difficoltà incontrate nella sua installazione e nel suo uso.

5.1 Resource Registry

Il *Resource Registry* (RR) è l'applicazione per la gestione dei metadati prodotta dalla Federazione svizzera SWITCHaai.
Il *software* amministra tre federazioni differenti:

1. *SWITCHaai Federation*: ambiente reale in cui vengono gestite le diverse entità, IdP e SP, che fanno parte della federazione svizzera. Nel corso di questa tesi verrà sostituita con "*IDEMaai Federation*" e ricoprirà il ruolo di *Production Federation*.
2. *AAI Test Federation*: ambiente di test in cui le nuove Organizzazioni possono testare il *Resource Registry* prima di aderire alla *IDEMaai.* Questa costituirà la *Test Federation*.
3. *Interfederation*: ambiente predisposto per quelle Organizzazioni che vogliono condividere le proprie risorse e/o i propri servizi con il resto del mondo.

SWITCHaai utilizza *eduGAIN* [25], il più vasto servizio di interfederazione che interconnette diverse federazioni tra loro e semplifica l'accesso ai servizi e alle risorse federate a favore della Ricerca e degli organi di formazione.

Le Organizzazioni che vi aderiscono potranno utilizzare i servizi e le risorse trasmesse da *eduGAIN* e aggiungere ad essa le proprie entità.

Le entità inserite nel *Resource Registry* vengono memorizzate in *description* e si dividono in:

- *Home Organization Description*
- *Resource Description*

Le prime descrivono gli *Identity Provider* (fornitori di servizi di autenticazione), mentre le seconde i *Service Provider* (produttori di risorse che gli utenti autenticati utilizzeranno).

La composizione di queste *description* è fondamentale in quanto ne determinano l'inserimento, o il rifiuto, nella *Production Federation*.

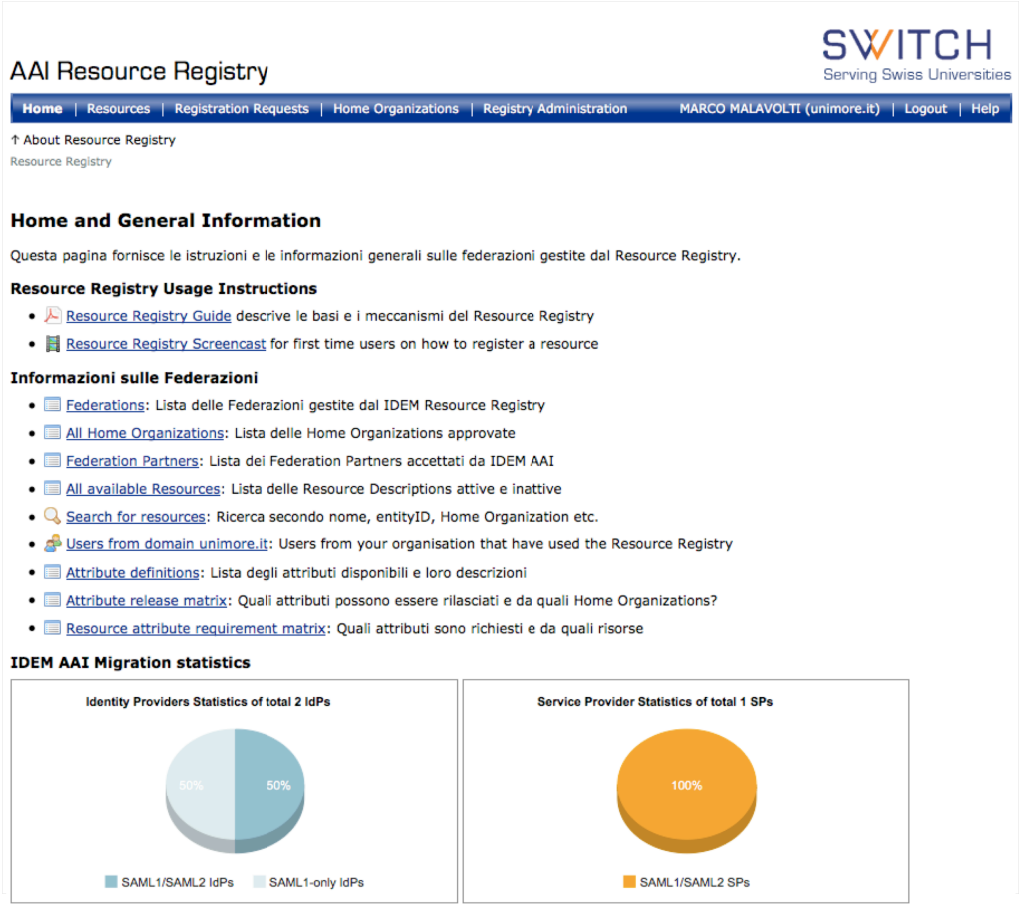
Ogni utente può ricoprire uno o più ruoli assumendo privilegi differenti:

- **Resource administrator (R Admin)**:
  può aggiungere nuove *Resource Description* o modificare quelle già esistenti che gli competono e appartenenti ad una certa *Home Organization*.
  A qualsiasi utente è dato il privilegio di aggiungere nuove risorse.
- **Home Organization administrator (HO Admin)**:
  può modificare la *Home Organization Description*, e decidere quali attributi rilasciare alle risorse.
  Tale ruolo deve essere ricoperto da almeno 1 persona per Organizzazione.
- **Resource Registry Authority administrator (RRA Admin)**:
  può approvare o rifiutare la registrazione di *Resource Description* nuove o modificate e deve istruire i *Resource Administrator* sulle regole da osservare nella Federazione.
  Tale ruolo deve essere ricoperto da almeno 1 persona per *Home Organization.*
- ***Attribute Policy Administrator (AP Admin):***
  può solamente editare la configurazione delle *Attribute Policy* di una *Home Organization* e definire quali attributi il suo *Identity Provider* è in grado di rilasciare.
  Un HO Admin possiede già questi privilegi.
- ***Resource Registry Operator (RR Operator):***
  godono di un accesso completo all'applicazione, al suo contenuto e al database che la supporta.
  E' la più alta carica all'interno del *Resource Registry*.

L'applicazione è suddivisa in 5 sezioni principali:

1. ***Home***: accessibile a qualsiasi utente
2. ***Resources****:* riservata agli utenti aventi i privilegi di *Resource Administrator*
3. ***Registration Requests****:* riservata agli utenti aventi i privilegi di *Resource Registry Authority Administrator*
4. ***Home Organizations****:* riservata agli utenti aventi i privilegi di *Home Organization Administrator*
5. ***Registry Administration****:* riservata agli utenti aventi i privilegi di *Resource Registry Operator*.

5.1.1 Funzionamento



L'utente, responsabile dell'Organizzazione che vuole aderire alla Federazione, completa un apposita *form* con la quale chiede la registrazione del suo *Identity Provider*.

Se è in grado di fornire tutte le informazioni richieste ed il suo IdP rispetta i requisiti imposti dalla Federazione e dal *Resource Registry*, l'utente otterrà i privilegi di HO Admin e RRA Admin e un accesso all'applicazione.

Inizialmente la sua Organizzazione prenderà parte della *AAI Test Federation* per impedire eventuali problemi nella *IDEMaai Federation*.

Le due federazioni hanno le medesime caratteristiche, ma vengono separate per offrire un ambiente sicuro in cui testare le entità prima di entrare in produzione.

La pagina principale è la **Home** dove tutti gli utenti possono eseguire le seguenti operazioni:

- Vedere le Federazioni, le Organizzazioni, i *Federation Partner* [26], le risorse e gli attributi gestiti dal programma.
- Ricercare una o più risorse all'interno di una Federazione attraverso una *form* dotata di auto-completamento per una ricerca più rapita.
- Controllare quali utenti appartengono alla propria *Home Organization* e qual'è stato il loro ultimo accesso.
- Scoprire quali attributi vengono rilasciati dalle diverse *Home Organization* e quali vengono richiesti dalle *Resources* nelle 3 Federazioni presenti.
- Verificare l'evoluzione della *Production Federation* attraverso grafici a torta.

Per iniziare ad inserire le proprie risorse un utente o, più specificatamente, un R Admin, dovrà accedere al pannello **Resources** dove gli è consentito di:

1. aggiungere nuove risorse.
2. visualizzare quelle già inserite o di cui è diventato amministratore.
3. generare il codice del *Discovery Service Embedded* per una risorsa specifica della *IDEMaai Federation* o *AAI Test Federation*.

L'inserimento di una nuova risorsa prevede il superamento di 8 passi, senza i quali non è possibile presentare la richiesta della sua approvazione.

Tale percorso, reso agile dall'automazione introdotta grazie agli assistenti forniti dal *Resource Registry*, permetterà di raccogliere una descrizione dettagliata dell'entità e di comporre i relativi metadati privi di errori.

L'applicazione, una volta raggiunta la completa preparazione della risorsa, permette all'utente di richiedere la sua approvazione agli RRA Admin nominati della sua *Home Organization*.

Questi amministratori, dopo aver verificato la correttezza delle informazioni inserite, potranno approvare l'entità nella Federazione di appartenenza della loro Organizzazione.

Per determinare l'autenticità della risorsa, nel caso in cui si utilizzi un certificato autofirmato, il RRA Admin sarà obbligato a prendere contatti con il proprietario dell'entità e ricevere le prime 8 cifre del *fingerprint* del suo certificato. Questo sistema prende il nome di *Four Eyes Approval Procedure*.

Per i certificati firmati, approvati da una CA che fa parte del *Microsoft Root Certificate Program* o accettata dalla *Mozilla Foundation,* tale controllo non è richiesto.

Per ogni risorsa così aggiunta un R Admin sarà in grado di:

- Visualizzare le informazioni in essa contenute
- Modificarla.
- Crearne una nuova partendo dalla descrizione di una esistente.
- Rimuoverla.
- Trasferirne i privilegi di gestione ad altri utenti dell'Organizzazione.
- Visualizzare una guida per la sua corretta installazione.

I RRA Admin, oltre a gestire le richieste di approvazione delle risorse, dal pannello **Registration Request** a loro riservato potranno:

- Aggiungere un nuovo attributo SAML utilizzabile localmente dalla propria Organizzazione.
  (funzione condivisa anche dagli utenti non RRA Admin)
- Visualizzare le *Resource Description* già attive ed appartenenti alla propria *Home Organization.*
- Trasferire, o revocare, i privilegi di RRA Admin ad altri membri dell'Organizzazione.
- Vedere tutti gli amministratori dell'istituzione.

La scheda **Home Organization**, destinata agli HO Admin, è progettata per la gestione delle *Home Organization Description* che descrivono le Istituzioni.

Per ogni *Home Organization* qui presentate, gli HO admin potranno:

- Visualizzare un resoconto di tutte le sue informazioni.
- Vedere le risorse che gli appartengono.
- Migliorare la propria *Home Organization Description* e specificarne le regole per il rilascio degli attributi (ARP) alle risorse.
- Prelevare i file *arp.site.xml (Shibboleth 1.x) o attribute-filter.xml (Shibboleth 2.x)* da utilizzare sul IdP per applicare le ARP.
- Estendere ad altri membri dell'Organizzazione i permessi di HO Admin.

Infine, ma non meno importante, il pannello **Registry Administration** riservato ai soli RR Operator e con il quale è possibile gestire il *software* e le entità in esso presenti.

Qui i RR Operator non solo saranno in grado di gestire tutte le *Home Organization* e le risorse presenti nel *Resource Registry*, ma potranno anche:

- Visualizzare le risorse temporanee o inattive presenti nelle 3 federazioni gestite.
- Esaminare i certificati usati da tutte le entità presenti nell'applicazione.
- Disabilitare un utente trasferendo i suoi permessi ad un altro.
- Aggiungere un certificato intermedio ai metadati della Federazione.
- Vedere alcune statistiche dettagliate della IDEMaai.
- Prelevare i metadati NON firmati prodotti dall'applicazione.
- Gestire i *Federation Partner*, ovvero gli enti privi di *Identity Provider* a cui autenticarsi.

5.1.2 Considerazioni Personali

Riteniamo che il *Resource Registry* non sia solo un'applicazione per la gestione dei metadati, ma anche un programma per l'amministrazione delle informazioni sulle Organizzazioni stesse.

Questo programma infatti non mostra mai all'utente i suoi metadati, ma lo guida con assistenti e procedure automatiche che provvedono per lui alla loro formazione priva di errori.

Attualmente il programma supporta egregiamente sia lo standard SAML 1.0, utilizzato ancora da alcune entità presenti nella Federazione Svizzera, che il nuovo SAML 2.0 a cui si aggiunge l'estensione MDUI [22] per le interfacce utente.

Il programma, interamente realizzato in PHP e Javascript, viene supportato da un *database* SQL facilmente gestibile con phpMyAdmin e, non essendo progettato per un uso esterno alla Federazione Svizzera, non è dotato di una una documentazione adeguata alla sua corretta installazione e configurazione.

Il Resource Registry è stato implementato per supportare completamente e specificatamente il *framework* Shibboleth. Questo ha permesso alla Federazione Svizzera di creare strumenti specifici per semplificare l'adesione, e la gestione, delle Organizzazioni che utilizzano prettamente tale tecnologia.

Ciò nonostante l'inserimento di entità s*impleSAMLphp* non diverge troppo da quelle *Shibboleth* ed è quindi possibile impiegare l'applicazione in una Federazione eventualmente mista.

La generazione automatica dei file "*attribute-filter.xml*" o "*arp.site.xml*", semplifica di molto il ruolo di un HO Admin nel mantenere aggiornate le *Attribute Release Policies* (Politiche per il rilascio degli attributi) dei propri Identity Provider.

Ad ogni utente è data la possibilità di inserire nuovi attributi nell'applicazione, ma solo al RR Operator è dato il potere di definirne di nuovi per la Federazione. In questo modo non le Organizzazioni possono impiegare anche attributi non stabiliti per l'intera Federazione.

Il sistema di SMS utilizzato per la *Strong Authentication* necessaria per compiere modifiche a una *Home Organization* è ben progettata e la *Four Eyes Approval Procedure* è un ottimo modo per dimostrare l'autenticità di una risorsa.

Molto ben gestito anche il *Rollover* dei certificati prossimi alla scadenza, che impedisce la sospensione delle risorse e dei servizi di autenticazione in un procedimento semplice e rapido.

La capacità del RR di rimuovere automaticamente dai metadati della Federazione le entità il cui certificato è scaduto lascia libero l'operatore da un loro costante controllo.

Abbiamo potuto osservare che il *Resource Registry* non solo tiene conto del periodo di validità dei certificati, ma permette agli amministratori di configurarne uno ulteriore, per le risorse, permettendone l'interruzione prima della scadenza del proprio certificato.

L'applicazione, inoltre, si avvale di un ottimo sistema di email con cui comunica agli utenti eventuali problemi riscontrati con le loro entità.

L'utilizzo del file "rr-cron-file" permette alla federazione di automatizzare diverse operazioni, tra cui:

- controllo dei certificati delle entità presenti.
- verifica della sincronizzazione delle *Attribute Release Policy* tra RR e i diversi *Identity Provider* registrati.
- aggiornamento dei metadati dell'interfederazione.

- esecuzione di *backup* del database mantenuto dall'applicazione.

Il *Resource Registry*, mediante cartelle specifiche, permette al suo amministratore di includere dei metadati nella giusta Federazione senza che le entità da loro rappresentati possano comparire nel programma stesso. Questo può rivelarsi utile nel caso in cui si vogliano nascondere certe entità alle Organizzazioni, ma permettere loro di prender parte alla Federazione.

L'applicazione non si limita alla sola gestione delle Organizzazioni aventi un *Identity Provider*, ma consente anche a chi non ne avesse uno, i *Federation Partner*, di poter usufruire ugualmente dei suoi servizi attraverso Organizzazioni virtuali, le *Virtual Home Organization*.

Riteniamo molto importante questo aspetto in quanto la percentuale di risorse fornite dai *Federation Partner* spesso è molto maggiore di quella data dalle Organizzazioni.

Abbiamo avuto modo di notare un elevato numero di *backup* del *database* effettuati per le risorse e le organizzazioni in seguito ad una qualsiasi modifica della loro *description*. Visto un così elevato numero di salvataggi, possiamo dedurre che, nel medio-lungo termine, si dovrà provvedere ad una loro rimozione manuale e mirata per non saturare la macchina che ospita l'applicazione.

Per concludere abbiamo avuto modo di notare che la SWITCHaai usa lo *Shibboleth Metadata Aggregator* [27] per fornire alle sue entità interfederate una rielaborazione dei metadati distribuiti da eduGAIN.

Tale servizio viene utilizzato dal *Resource Registry* per:

1) Dividere i metadati raccolti da eduGAIN in più parti:

- metadati degli IdP: forniti agli SP interfederati della Federazione
- metadati degli SP: forniti agli IdP interfederati della Federazione

2) Rimuovere eventuali entità ripetute e già presenti nella Federazione

Suddividendo i metadati nelle 2 componenti specifiche, SWITCHaai si assicura una buona scalabilità nel tempo e il poter distribuire al meglio il carico di lavoro da eseguire sui metadati, qualora fosse necessario.

5.1.3 Problemi riscontrati

Il *Resource Registry* è un'applicazione web che non è stata creata per essere condivisa con le altre Federazioni, ma per essere utilizzata unicamente dalla Federazione Svizzera SWITCHaai.

Per questo motivo sono stato costretto ad effettuare numerose correzioni al codice sorgente per adattarla alle esigenze della nostra Federazione.

Per riuscire ad analizzare completamente l'applicazione è stata necessario ricorrere alla creazione di ben 4 utenti fittizi in grado di ricoprire i 4 principali ruoli che un utente del RR può avere.

Si è reso necessaria una corretta, e precisa, configurazione del nostro IdP e dei nostri SP per soddisfare tutte le richieste del *software* e risultare così idonei al loro inserimento nell'applicazione.

La scarsa documentazione sul *Resource Registry* ha richiesto un notevole sforzo nella correzione degli errori che l'applicazione riscontrava con le nostre entità.

Tale programma non è dotato di un supporto multilingue e, quindi, nel caso una Federazione decida di adottarlo, sarebbe necessaria una completa traduzione del testo presente nei vari file che ne compongono l'interfaccia.

L'unica proposta che ci permettiamo di avanzare e che potrebbe rivelarsi utile è quella di implementare un meccanismo con cui gli utenti possano recuperare i metadati delle proprie entità in modo diretto senza doverli filtrare da quelli della Federazione forniti, oltre che a dotare l'applicazione di una documentazione adeguata se intenzionati a promuoverla ad altre Federazioni.

5.2 Janus

Janus è un modulo per s*impleSAMLphp* scritto in PHP e utilizzato dalla Federazione Danese per la gestione dei propri metadati.

Configurando il file "`module_janus.php`" è possibile adattare l'applicazione ai requisiti caratterizzanti la propria Federazione.

Presentiamo di seguito alcune delle potenzialità che riteniamo essere di maggior importanza:

1. pieno controllo della struttura dei metadati registrabili.
2. definizione dettagliata dei ruoli e dei privilegi degli utenti.
3. gestione semplice e accurata del *workflow* delle entità ed i loro stati.
4. esportazione dinamica dei metadati della Federazione.
5. pubblicazione dei metadati su più destinazioni: Feed RSS, *filesystem*, FTP, [...]
6. Utilizzo delle REST API per il recupero di importanti informazioni

Nell'analisi di questo programma si sono applicate le seguenti modifiche alla configurazione predefinita per rispondere meglio alle esigenze della nostra Federazione:

a) Si sono definiti solo 3 stati per il *workflow* di un entità:
   1. Test (di colore **rosso**)
   2. In attesa di approvazione (di colore **arancio**)
   3. Produzione (di colore **verde**)

b) Si sono stabiliti solo 3 tipologie di utenze:
   1. Amministratore della Federazione (**admin**): gode di un accesso completo a Janus e alle entità in esso registrate.
   2. Membro del Team di Supporto (**support**): gode di un accesso limitato alla sola gestione delle entità e all'esportazione dei metadati della Federazione.
   3. Amministratore di un entità (**technical**): gode di un accesso limitato al solo inserimento dei propri IdP o SP e può esportare i propri metadati soltanto in fase di **Test**.

c) Si è stabilito che la scheda ARP, destinata alla definizione delle politiche sul rilascio degli Attributi, sia ad uso esclusivo degli utenti di tipo '**admin**' e '**support**'.

d) Si è scelto di considerare le sole entità SAML 2.0, benchè Janus supporti anche le entità SAML meno innovative.

e) E' stato permesso agli utenti di visionare le sottoscrizioni alle entità da loro registrate e di poter scegliere come ricevere le rispettive notifiche: eMail o Inbox.

f) Si è concesso agli utenti di vedere il catalogo degli IdP e degli SP della Federazione, ma è stato privato loro la capacità di esportare i metadati della stessa. Tale privilegio spetta solo agli utenti '**admin**'.

g) Si è imposto agli utenti l'inserimento del *fingerprint* del certificato utilizzato dalle entità per verificarne l'autenticità.

5.2.1 Funzionamento

L'utente, amministratore di un'Organizzazione composta da IdP e SP, richiede di entrare a far parte della Federazione sottoscrivendo un contratto che lo istruisce sugli attributi da lei adottati.

Una volta adattato il proprio *Identity Provider* per far fronte alle richieste, l'utente accede a Janus attraverso un collegamento ricevuto per email.

Qui l'utente technical, a seconda della configurazione dell'applicazione, può:

1. Completare le informazioni sul suo *account.*
2. Registrare nuove entità e visualizzare quelle già inserite.
3. Visionare la *Inbox* contenente le notifiche a lui destinate.
4. Visualizzare, eventualmente, il catalogo degli IdP e degli SP presenti nella Federazione.

L'inserimento di una nuova entità è semplice e veloce grazie all'interfaccia grafica sobria e munita delle sole operazioni necessarie.

L'applicazione implementa tre metodi per importare i metadati:

- tramite una URL
- tramite codice XML
- tramite codice JSON

L'utente può autonomamente controllare la validità delle proprie entità dalla scheda 'Validate' e correggerne, o espanderne, i metadati dalla scheda 'Metadata'.

Una scheda, 'History', è destinata alla raccolta dei progressi fatti dall'entità sottoforma di revisioni.

Per ogni entità è possibile abilitare una *Blacklist* ed una *Whitelist* attraverso cui gestire gli accessi alle risorse(SP) o ai servizi di autenticazione(IdP).

Agli IdP registrati è concesso, grazie all'apposita scheda '*Consent disabling*', di scegliere quali SP sono meritevoli di fiducia e verso cui è permesso il rilascio degli attributi di un utente senza il suo esplicito consenso.

Infine, dalla scheda 'Export' e solo in fase di Test, è possibile esportare i metadati delle singole entità in varie forme:

- simpleSAMLphp
- XML
- JSON

o inviarli per email all'**admin** in formato simpleSAMLphp.

Una volta pronta, l'entità può essere presentata per l'approvazione alla Federazione che provvederà a verificarne la correttezza e promuoverla alla produzione dove sarà a disposizione delle altre Organizzazioni.

L'amministratore della Federazione **admin**, oltre a visualizzare tutto quello che un utente normale già vede, dispone di due schede aggiuntive, 'ARP' e 'Admin'.

La prima, 'ARP', permette di definire le *Attribute Release Policies* (Politiche sul Rilascio degli Attributi) dei *Service Provider* in base agli intenti delle loro risorse. Ogni SP dovrà assumere l'ARP più adatta alle sue esigenze, questo modificherà i suoi metadati aggiungendo gli elementi SAML necessari a soddisfarla.

Questo consente di:

a) ridurre al minimo il numero degli attributi che una risorsa può richiedere.
b) limitare il numero di utenti aventi accesso alle risorse specificando il valore dei loro attributi.

La scheda 'Admin', invece, consente di amministrare gli utenti e le entità presenti nell'applicazione Janus.

Qui un **admin** può:

1. Modificare i permessi di un utente.
2. Attivare, Disattivare o Eliminare un utente.
3. Assegnare o rimuovere un utente da un'entità.
4. Rimuovere o disabilitare temporaneamente un'entità.

Dalla scheda 'Federation', infine, un **admin** può esportare dinamicamente i metadati della Federazione attraverso una semplice *form* con cui potrà scegliere:

- quali tipologie di entità (SAML 2.0 SP, SAML 2.0 IdP)
- quali stati del *workflow* (Test, In attesa di Approvazione, Produzione)
- quale formato MIME utilizzare (XML , simpleSAMLphp)
- quali entità escludere
- quale *post processor* usare per pubblicare i metadati su altre destinazioni: Filesystem, FTP, [...]

5.2.2 Considerazioni Personali

Janus è un modulo per la gestione dei metadati orientato alle entità con lo scopo di offrire un servizio *self-service* utilizzabile da chiunque voglia aderire alla Federazione che lo impiega.

Il suo semplice utilizzo e la sua alta configurabilità ne fanno un'applicazione potente e in grado di migliorare con l'avanzare delle versioni.

Il programma è facilmente installabile sia sui sistemi Linux che sui sistemi Windows in quanto richiede la presenza del solo PHP e di un database MySQL oltre al *framework simpleSAMLphp* che estende.

Essendo solamente un modulo, Janus utilizza il *framework* per effettuare il *parsing* dei metadati. Questo legame fa sì che ogni progresso che simpleSAMLphp compie si ripercuota anche sul *software,* sia in positivo che in negativo.

Al momento *simpleSAMLphp* è in grado di utilizzare le seguenti estensioni per metadati:

1. *SAML V2.0 Attribute Extensions* [28]
2. *SAML V2.0 Metadata Extensions for Login and Discovery User Interface Version 1.0* [22]

L'applicazione usa un sistema di revisioni per tenere traccia nel tempo delle modifiche apportate alle entità. Tale funzionalità facilita la condivisione delle stesse tra più utenti e ne consente, in caso di errore, il ripristino a una versione precedente.

Janus possiede un buon supporto multilingua che consente, ai suoi utenti, di visualizzarne i contenuti secondo la lingua da loro preferita. La lingua predefinita è l'inglese, ma la traduzione dell'interfaccia non è per niente complessa.

La validazione dei metadati è decisa dalla Federazione attraverso la definizione di funzioni che regolano i valori dei singoli elementi SAML che li formano, ma non è possibile combinare un confronto tra più valori per stabilire la validità di un elemento.

Uno dei punti di forza di questo programma è certamente l'esportazione dinamica dei metadati, che si fa più forte se affiancata da uno o più *post processor* capaci di personalizzarne l'esportazione.

La via suggerita da Janus per la loro distribuzione è quella del Feed RSS che mette a disposizione, ma non mancano anche altre forme di distribuzione quali *File System* o FTP.

L'applicazione è in grado di firmare i metadati esportati con il certificato della Federazione e di definire gli attributi XML 'validUntil' e 'cacheDuration' per tutte le entità presenti, operazione che ora viene eseguita manualmente in IDEM.

Il programma implementa anche un sistema di *rollover* dei certificati per evitare l'interruzione dei servizi offerti, ma non è stato possibile verificare che cosa avvenga alla loro scadenza. L'unica segnalazione osservata deriva dalla scheda 'Validate' dove, a certificato scaduto, Janus segnala il problema visivamente.

Per concludere, apprezziamo l'impiego di REST API con cui è possibile recuperare le seguenti informazioni da Janus:

- l'ARP di un'entità
- i dati di un utente
- la capacità di un SP e di IdP di comunicare tra loro (true o false)
- una lista di tutti gli IdP che possono usufruire di uno specifico SP
- una lista di tutti gli SP accessibili da uno specifico IdP

- il catalogo degli SP
- il catalogo degli IdP
- uno o più entityID di specifiche entità

5.2.3 Problemi riscontrati

Al momento della stesura di questa tesi e dell'analisi di questa applicazione, il *framework simpleSAMLphp* non è in grado di effettuare un *parsing* accurato dei metadati prodotti dalle entità *Shibboleth*.

Questo non preclude l'uso di tale strumento anche in federazioni che usano tale *framework*, ma può portare alla perdita di alcuni elementi SAML dei metadati ritenuti utili dalla Federazione.

Le funzioni di validazione definite di *default* da Janus sono per la maggiorparte banali, quasi studiate apposta per una loro ridefinizione più accurata che, senz'altro, è necessaria.

Non siamo riusciti a utilizzare il *post processor* 'FileSystem' che, per motivazioni sconosciute, non riesce ad esportare i metadati nel percorso stabilito e di cui gode del pieno controllo.

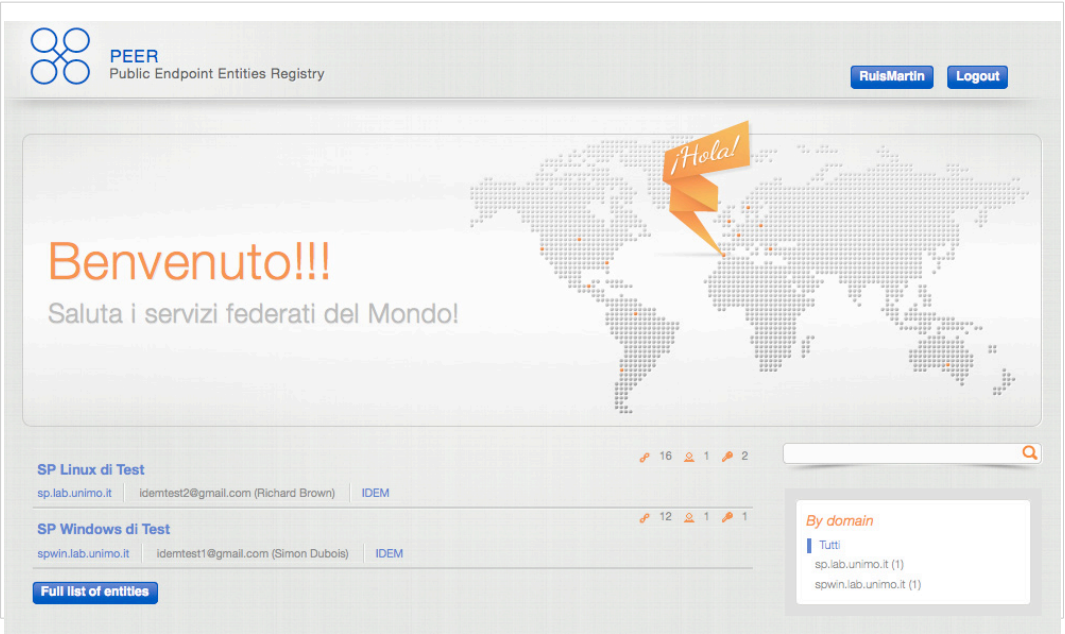
La mancata sottoscrizione dell'utente alle notifiche, riguardanti i cambiamenti avvenuti all'entità di cui è diventato gestore, in seguito a un trasferimento di privilegi effettuato da un admin, porta l'utente a non intervenire in caso di necessità.

Pensiamo sia necessaria una modifica al codice dell'applicazione affinché tale problema si risolto.

Riteniamo possa essere utile fornire agli utenti una sezione apposita per il trasferimento dei privilegi delle proprie entità in modo tale da distribuire il carico di lavoro gravante sull'**admin** e i suoi collaboratori agli utenti stessi.

Avanziamo infine la proposta di sviluppare un *plugin* in grado di automatizzare la creazione dei *metadatafields*, utilizzati per strutturare i metadati che possono essere registrati in Janus, per ridurre i tempi di configurazione del *software* stesso.

5.3 PEER



PEER è un'applicazione web sviluppata da **TERENA** (**T**rans-**E**uropean **R**esearch and **E**ducation **N**etworking **A**ssociation) che mira a creare un registro globale autonomo per la registrazione e la pubblicazione dei metadati rappresentanti *endpoint* SAML e non.

Questo servizio focalizza la sua attenzione sull'uso internazionale dell'identità federata e deve necessariamente basarsi sulle *PEER User Stories* [29] e sulle *PEER Service Descriptions* [30].

Una o più federazioni dovrebbero adottare questo programma come supporto esterno per le proprie risorse interfederate per essere in grado di migliorarle senza preoccuparsi della loro gestione.

5.3.1 Funzionamento

L'applicazione prevede 3 tipologie di utenze:

1. L'utente Anonimo.
2. L'utente Iscritto a PEER.
3. L'utente amministratore di PEER.

L'utente Anonimo può solo recuperare i metadati delle entità presenti in PEER.

L'utente iscritto a PEER, diventato tale solamente dopo aver completato il processo di registrazione ed aver ottenuto i dati utili all'accesso, può:

- Aggiungere o rimuovere uno o più domini di sua proprietà.
- Aggiungere o rimuovere una o più entità appartenenti ai propri domini.
- Creare uno più gruppi di entità preferite.
- Cambiare i dati personali forniti con la sua registrazione.
- Cambiare la password di accesso.
- Invitare un altro utente ad iscriversi al servizio.

Per ogni risorsa aggiunta, l'iscritto può:

- Visualizzarne i dettagli registrati.
- Modificarne i metadati.
- Eliminarla.
- Concedere ad altri utenti la sua gestione o nominare un nuovo proprietario.
- Configurare un periodo per l'aggiornamento dei suoi metadati.

- Abilitare un monitor che lo avverta via email nel caso in cui uno degli *endpoint* dell'entità smetta di rispondere.

La gestione dei metadati di un'entità è affidata al plugin JQuery SAMLmetaJS che trasforma una semplice area di testo in una collezione di schede e *form* con cui è possibile creare, modificare e rimuovere i diversi elementi SAML.

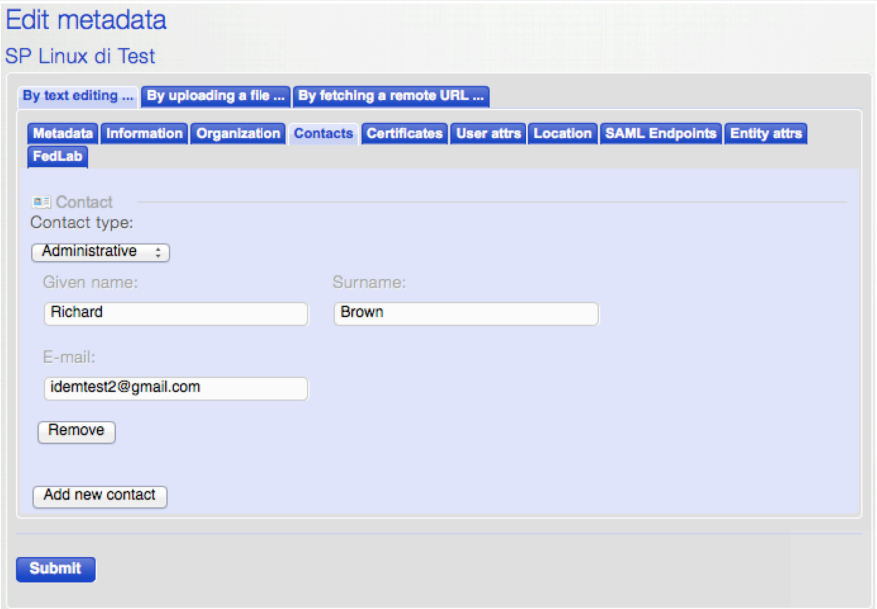


Figura 5.4.b - *Modifica dei metadati di un'entità*

Attualmente sono 9 le sezioni implementate da SAMLmetaJS:

1. **Information**: raccoglie le informazioni di base della risorsa.
2. **Organization**: contiene i dati dell'Organizzazione ospitante.
3. **Contacts**: gestisce i contatti collegati all'entità.
4. **Certificates**: amministra i certificati del servizio.
5. **User attrs**: definisce gli attributi richiesti.
6. **Location**: specifica la posizione geografica dell'entità.
7. **SAML Endpoint**: organizza gli *endpoint* della risorsa.
8. **Entity attrs**: permette di definire attributi specifici utilizzati dal servizio.

9. **FedLab**: aggiunge alcuni parametri aggiuntivi richiesti dalle *Federation Lab Tools* [31].

Una volta apportati i cambiamenti necessari è possibile inserire un testo che alimenta il sistema di revisioni che il software implementa per tracciare l'evoluzione delle entità inserite.

L'utente amministratore di PEER, oltre a detenere il completo controllo dell'applicazione e la possibilità di visualizzare e/o modificare tutte le risorse presenti, può nominare altri utenti a grado di amministratore per suddividere il carico di lavoro tra più individui.

5.3.2 Considerazioni Personali

PEER è un'applicazione davvero ben progettata per lo scopo alla quale è stata destinata.

Il non dover rispondere alle esigenze di una specifica Federazione trasforma questo programma in un'applicazione dal carattere universale che le consente di essere uno strumento utilizzabile da Federazioni eterogenee.

Inoltre, essendo completamente orientata allo standard SAML, può importare i metadati sia di entità basate sul *framework Shibboleth* che s*impleSAMLphp*.

Il dover dimostrare di essere il proprietario del dominio inserito consente a PEER registrare solo entità autentiche e realmente possedute da un individuo fisico in grado di gestirle. Si assicura, inoltre, che non vi siano entità ripetute in quanto il nome del dominio *(domain name)* deve essere univoco sulla rete.

Riteniamo molto utile la capacità di PEER di eseguire un controllo sintattico sui metadati. Questo evita che siano composti metadati sbagliati ed elimina la necessità di un *workflow* per le risorse inserite.

Attualmente l'applicazione supporta le seguenti estensioni per i metadati:

- *SAML V2.0 Metadata Extensions for Login and Discovery User Interface Version 1.0* [22]
- *SAML V2.0 Metadata Extension for Entity Attributes Version 1.0* [32]

Abbiamo apprezzato molto l'interazione con le mappe di *Google Maps* per geolocalizzare le entità e la possibilità di creare gruppi da cui prelevare i metadati attraverso un semplice *Feed* RSS. Troviamo che quest'ultima capacità sia particolarmente utile per limitare i metadati alle sole entità di interesse per l'utente.

Il *software* è stato implementato per supportare pienamente il solo standard SAML v2.0, ma l'importazione di entità precedenti ha come unico effetto l'inserimento di elementi SAML normalmente esclusi.

5.3.3 Problemi riscontrati

Non è stato possibile verificare come PEER gestisca la scadenza dei certificati delle sue entità.

Riteniamo necessario che il *software* venga dotato di un sistema in grado di riconoscere le risorse il cui certificato è scaduto prima della loro registrazione al servizio per evitare che vengano condivise risorse inattive.

Proponiamo, anche, la rimozione di quelle entità il cui certificato è scaduto. Dalle verifiche effettuate, esse continuano ad esistere e ad essere a disposizione delle altre Federazioni nonostante non siano più valide.

In PEER non è ancora possibile decidere quali attributi un SP può richiedere ad un IdP. Confidiamo che, in un futuro aggiornamento del *plugin* SAMLmetaJS, tale *feature* possa essere inclusa.

5.4 Raccolta dei risultati osservati

Feature/Software	RR	Janus	PEER
Viene controllata l'unicità del nome delle entità?	si	si	si
E' possibile aggiungere una descrizione dell'entità?	si, è obbligatorio	si	si
E' possibile aggiungere l'organizzazione che ospita l'entità?	si, è obbligatorio	si	si
Quali estensioni dei metadati sono supportate?	MDUI Extension	MDUI Extension, Attribute Extension	MDUI Extension
E' possibile riconoscere eventuali mancanze/errori presenti nei metadati?	no, non è necessario.	si	si
E' possibile visualizzare quali attributi un IdP può rilasciare?	si	no	no
E' possibile visualizzare quali attributi un SP richiede?	si	no	no
E' possibile esportare i metadati di una sola entità?	no	si	si
E' possibile esportare metadati che non rispettano i requisiti della Federazione?	no	no	si, perchè utilizzato da Federazioni differenti
E' possibile definire le Attribute Release Policy per i propri IdP?	si	no	no
E' possibile aggiungere entità SAML 1.0?	si	si	si, anche se non perfettamente riconosciute
E' possibile aggiungere entità SAML 2.0?	si	si	si
Quali tipologie di entità gestisce?	IdP e SP	IdP e SP	IdP e SP
Espone un Catalogo degli SP e degli IdP disponibili nella Federazione?	si	si	si
Linguaggi utilizzati per lo sviluppo dell'applicazione	PHP, javascript	PHP, JSON	Python, javascript
E' di facile installazione?	no	si	si
Possiede il supporto multi-lingue?	no	si	no

Feature/Software	RR	Janus	PEER
Gestisce correttamente i certificati delle entità?	si, ed elimina le entità il cui certificato è scaduto	non in modo predefinito	si, ma non è stato possibile verificarlo
Riesce a gestire anche le entità interfederate?	si	si	si
E' possibile gestire le ARP?	si	si	no
E' concepita per evolversi?	no	si	si
Permette il ripristino a versioni precedenti dei metadati?	no	si	si
Riesce a firmare digitalmente i metadati della Federazione?	no	si	no
E' in grado di pubblicare i metadati delle Federazione?	si	si	si, ma solo singolarmente

6 Conclusioni

Dai risultati ottenuti dallo studio di questa Tesi si può affermare che nessuna delle 3 applicazioni esaminate rappresenta una soluzione definitiva alle esigenze di gestione dei metadati richieste da una Federazione di Identità mista.

Attualmente, la necessità di un *software* in grado di amministrare i metadati di entità che utilizzano *framework* diversi per implementare lo standard SAML v2.0, è molto sentita dalle Federazioni nate su scala nazionale.

Ad essa si aggiunge anche quella di essere compatibile ad aggiornamenti futuri e in grado, quindi, di rispondere a nuove esigenze insorte con il progresso delle risorse e dei servizi.

Lo scopo di questa tesi, inteso come confronto valutativo di queste applicazioni, è pienamente soddisfatto ed evidenzia le carenze ancora esistenti nel soddisfacimento delle esigenze sopra riportate.

Il *Resource Registry* è uno dei programmi meglio progettati per il suo ruolo, ma, date le sue scarse capacità di espansione senza il supporto del suo *developer*, risulta debole in un confronto con lo *Janus,* le cui doti vengono aumentate ad ogni nuova *release* del *framework simpleSAMLphp*.

Al momento, però, *simpleSAMLphp* non è in grado di effettuare un *parsing* completo dei metadati *Shibboleth* e, quindi, non è ancora pronto per essere impiegato in una Federazione di Identità che deve amministrare entità miste.

Infine c'è *PEER* che, nonostante la si possa ritenere una soluzione valida alla condivisione di entità interfederate, non può essere impiegato come gestore dei metadati di una Federazione perchè, per i motivi per cui è stato concepito, non è prevista ne la pubblicazione e la firma dei suoi metadati, ne la gestione delle politiche sul rilascio degli attributi (ARP).

6 Conclusioni

Per concludere, ad oggi non esiste ancora un *software* in grado di soddisfare le richieste di ogni Federazione di Identità e, pertanto, le soluzioni finora proposte potrebbero essere adottate solo temporaneamente, ma, viste le doti dimostrate, pensiamo che uno sviluppo ulteriore di *Janus* e *simpleSAMLphp* permetterebbe di raggiungere tale traguardo.

7 Bibliografia

[1] Federazioni di Identità:

https://refeds.org

[2] Organizzazioni:

https://refeds.org/resources.html

[3] Federazione di Identità Italiana IDEM:

https://www.idem.garr.it/documenti/regolamento

https://www.idem.garr.it/documenti/normepartecipazione

[4] Identity Management:

Dr. Jean-Marc Seigneur , Dr Tewfiq El Maliki, Cap. 17 - Identity Management, Capitolo del volume Vacca, J.R. (2009). *Computer and Information Security Handbook*, ISBN: 978-0123743541, Morgan Kaufmann

[5] Authentication Authorization Infrastructure:

https://www.idem.garr.it/it/documenti/doc_download/170-introduzione-alle-infrastrutture-di-autenticazione-e-autorizzazione

[6] Consortium GARR:

http://www.garr.it/a/garr/documenti-chisiamo/doc_download/39-chi-e-garr

[7] Metadati:

https://www.idem.garr.it/documenti/specifichetecniche

[8] Shibboleth Discovery Service Centralized:

https://wiki.shibboleth.net/confluence/display/SHIB2/DiscoveryService

[9] Shibboleth:

http://shibboleth.net

[10] simpleSAMLphp:

http://simplesamlphp.org/

[11] Specifiche Tecniche per la compilazione e l'uso degli attributi in IDEM:

https://www.idem.garr.it/documenti/specificheattributi

[12] Standard SAML v2.0:

http://saml.xml.org/

[13] Resource Registry Guide:

http://www.switch.ch/aai/docs/AAI-RR-Guide.pdf

[14] Janus Wiki:

http://code.google.com/p/janus-ssp/

[15] PEER Documentation:

http://packages.python.org/peer/index.html

[16] *Maria Laura Mantovani,* 2008*,* "IDEntity Management federato per l'accesso ai servizi":

http://www.garr.it/eventiGARR/ws08/presentazioni/Tutorial%20-%20IDEM/Mantovani_WSGARR08.pdf

[17] *Maria Laura Mantovani*, 2010, "Federazione IDEM: come partecipare":

https://www.idem.garr.it/index.php/it/documenti/doc_download/118-presentazione-ml-mantovani

[18] *Massimiliano Panciamore*, "Attori nella Federazione e Profili SAML":

http://www.garr.it/eventiGARR/idem-day/docs/pianciamore_pres_idemday09.pdf

[19] *OASIS*, 15 Marzo 2005, "Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0" :

http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf

[20] *OASIS*, 15 Marzo 2005, "Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0":

http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf

[21] *OASIS*, 15 Marzo 2005, "Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0":

http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf

[22] *OASIS*, 10 Gennaio 2012, "SAML V2.0 Metadata Extensions for Login and Discovery User Interface Version 1.0":

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-metadata-ui/v1.0/sstc-saml-metadata-ui-v1.0.pdf

[23] *OASIS*, 15 Marzo 2005, "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0":

http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[24] RFC5870:

http://tools.ietf.org/html/rfc5870

[25] eduGAIN:

http://www.geant.net/service/edugain/Pages/home.aspx

[26] What is a Federation Partner?:

http://www.switch.ch/aai/join/partners.html

[27] Shibboleth Metadata Aggregator:

https://wiki.shibboleth.net/confluence/display/MA1/Home

[28] *OASIS,* 4 Agosto 2009, "SAML V2.0 Attribute Extensions":

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-attribute-ext.pdf

[29] PEER User Stories:

https://spaces.internet2.edu/display/PEER/PEER+Software+User+Stories

[30] PEER Service Descriptions:

https://spaces.internet2.edu/display/PEER/PEER+Service+Description

[31] FedLab:

https://fed-lab.org/

[32] *OASIS,* 4 Agosto 2009, "SAML V2.0 Metadata Extension for Entity Attributes Version 1.0":

http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-attr.pdf

[33] OASIS, 15 Marzo 2005, "Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0" :

http://docs.oasis-open.org/security/saml/v2.0/saml-glossary-2.0-os.pdf

8 Ringraziamenti

Ringrazio innanzitutto la mia famiglia che mi ha permesso di affrontare questi studi e sostenuto nei momenti di maggiore difficoltà.

Ringrazio il Professor Fabio Panzieri per avermi guidato con interesse e cordialità nella stesura di questa tesi.

Ringrazio la Dottoressa Maria Laura Mantovani per aver seguito con interesse lo sviluppo di questa tesi e per avermi fornito i materiali necessari allo studio svolto.

Ringrazio tutte le federazioni e il personale che ha saputo, con professionalità e disponibilità, assistermi nell'installazione, configurazione e correzione dei problemi da me riscontrati nello studio di questa tesi.
Rivolgo un particolare ringraziamento a Lukas Hämmerle (Resource Registry), Lorenzo Gil Sanchez (PEER) e Jacob Christiansen (Janus) per l'enorme aiuto che mi hanno fornito in questa tesi.

Ringrazio la Dottoressa Barbara Monticini del Servizio di supporto IDEM GARR AAI per la disponibilità rivoltami nella risoluzione di numerosi problemi incontrati nella configurazione ed uso della mia federazione di identità virtuale impiegata in questa Tesi.

8 Ringraziamenti

Ringrazio il personale dell'Università di Modena e Reggio Emilia che mi ha concesso di avere accesso alle loro strutture e fornito il supporto tecnico indispensabile allo studio delle applicazioni comparate in questa Tesi, in particolare ringrazio Francesco Malvezzi, Stefano dall'Olio, Andrea Ghidoni e Danilo Crecchia per la loro disponibilità e la loro assistenza.