Tecniche pratiche per federare applicazioni

Stefano Gargiulo (GARR)

Indice

- Introduzione
- Perche` federare un'applicazione?
- Come federare un'applicazione
- Best practices ed interoperabilità

Identità Digitale

Concetti

Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

L'identità digitale è un concetto delicato

"On the Internet, nobody knows you're a dog."

Peter Steiner,
The New Yorker,
5 July 1993

Autenticazione

concetto

Potremmo definirla come «il processo mediante il quale un'applicazione discerne l'identità di un utente».

Si tratta di un processo dipendente da elementi esterni all'applicazione: **l'utente** (almeno).

Il tutto è infatti **basato su un operazione che può compiere solo l'utente**: (Ricordare una password, fornire un documento, presentare un certificato, far riconoscere un'impronta digitale ecc.)

Autenticazione

L'**identità** può essere provata da

- Qualcosa che si conosce (una password)
- Qualcosa che si ha (smart cards, USB tokens, or pk certificates)
- Ciò che si è: impronte digitali, riconoscimento biometrico, o altre tecniche simili

Autorizzazione

- Verificare se l'utente ha diritto ad accedere ad una risorsa
- Tipicamente i sistemi digitali eseguono i controlli di autorizzazione basandosi su informazioni associate all'identità dell'utente
  - «Chi è»
  - Qualcuno dei suoi «attributi» :
    - Ruolo (professore, studente, staff)
    - Fee pagata (yes|no)

Nota: «chi è», è definito sempre dagli attributi: da un insieme di **attributi identificativi**

Autenticazione vs. Autorizzazione

- Si tratta di due processi distinti
- Spesso eseguiti nella stessa riga del codice di un'applicazione ma comunque distinti
- Abbreviazioni:
  - AuthN (Authentication)
  - AuthZ (Authorization)

Identità digitale

- I sistemi digitali rappresentato gli utenti (ma anche dispositivi, risorse ecc..) nel loro dominio come **entità digitali**.
  Ogni entità digitale possiede un numero finito di **attributi** che la descrivono.
- Esempio di attributi che compongono **un'identità digitale**
  - Username
  - Password
  - Certificato X.509
  - Nome
  - Cognome
  - Organizzazione
  - Ruolo nell'organizzazione

Identità digitale

- **Attributi identificativi**: Un sottoinsieme di attributi che identifica l'entità in maniera univoca. Esempi:
  - N# Documento
  - Username e Scope (gargiulo@garr.it)
    - Email
  - (name, surname, place of birth, date of birth)
- Non facile da definire:

  Alla definizione di Platone, dell'uomo come l'unico animale bipide e inplume, Diogene prese un pollo, lo spenno` e disse: «Ecco l'uomo di Platone!»

Autenticazione

Requisiti Applicativi

Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Autenticazione

requisiti applicativi

Nella fase di autenticazione **l'applicazione** ha principalmente un solo ruolo: **fornire l'infrastruttura necessaria al compimento dell'operazione di identificazione da parte dell'utente:**

- interfaccia per la sottomissione di credenziali
- storage degli hash delle password
- relativo codice di riconoscimento

Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Autenticazione

requisiti applicativi

Bastano questi tre requisiti?

- Si, per il funzionamento di base del processo di autenticazione
- Non per manutenerlo:
  - Interfaccia di registrazione?
  - Interfaccia gestionale per la modifica/aggiunta utenti?
  - Procedure di recupero password?
  - Messa in sicurezza (HTTPS, SQL Injection ecc.)?
    - Evoluzione dell'assurance level: (e.g 2-3 factor auth ecc.)
  - Gestione dei dati sensibili (profilo utente)?

Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Autenticazione

requisiti applicativi

E dopo aver implementato un infrastruttura completa?

- Molto spesso non la riutilizziamo altrove (come sarebbe buona norma per ogni componente software implementato).
- Dobbiamo manutenerla e aggiornarla
- Lo sconforto e la confusione, nostri e dell'utente, crescono all'aumentare del numero di applicazioni che offriamo

Autenticazione

Conclusioni

Ma serve davvero a qualcosa avere una logica di autenticazione nelle applicazioni?

Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Autenticazione

Conclusioni

No,

Non di per se!

Alle applicazioni serve **identificare** un utente al solo fine di poter decidere se **autorizzarlo** o meno a compiere determinate operazioni, oppure per avere a disposizione alcuni suoi **attributi** a fini applicativi (es. email).

Applicazioni

Che rapporto hanno con le identità?

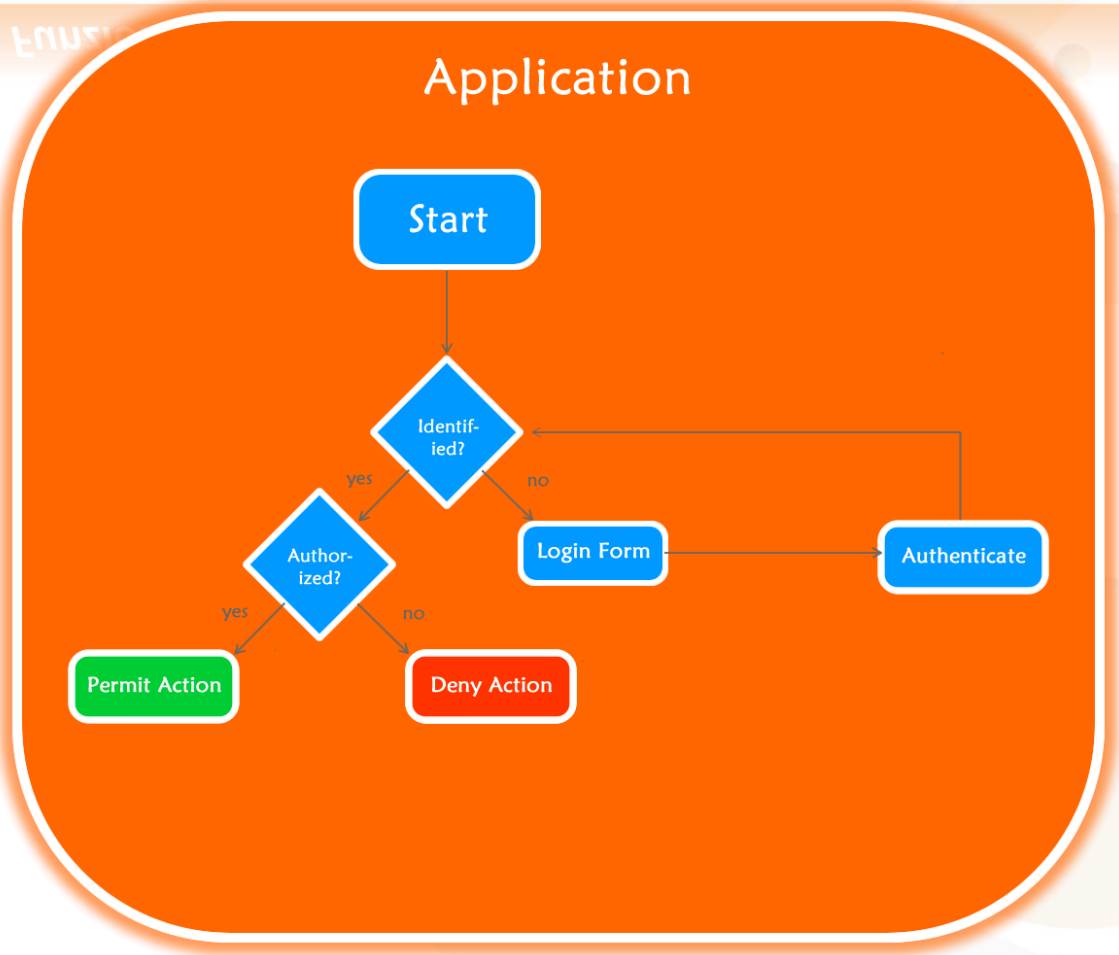
Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Applicazioni

Funzionamento generico

un'applicazione rapportata ad un'identità, può essere concepita come un insieme di azioni permesse o meno a quest'ultima.

L'immagine a destra è una generalizzazione della logica AA delle applicazioni

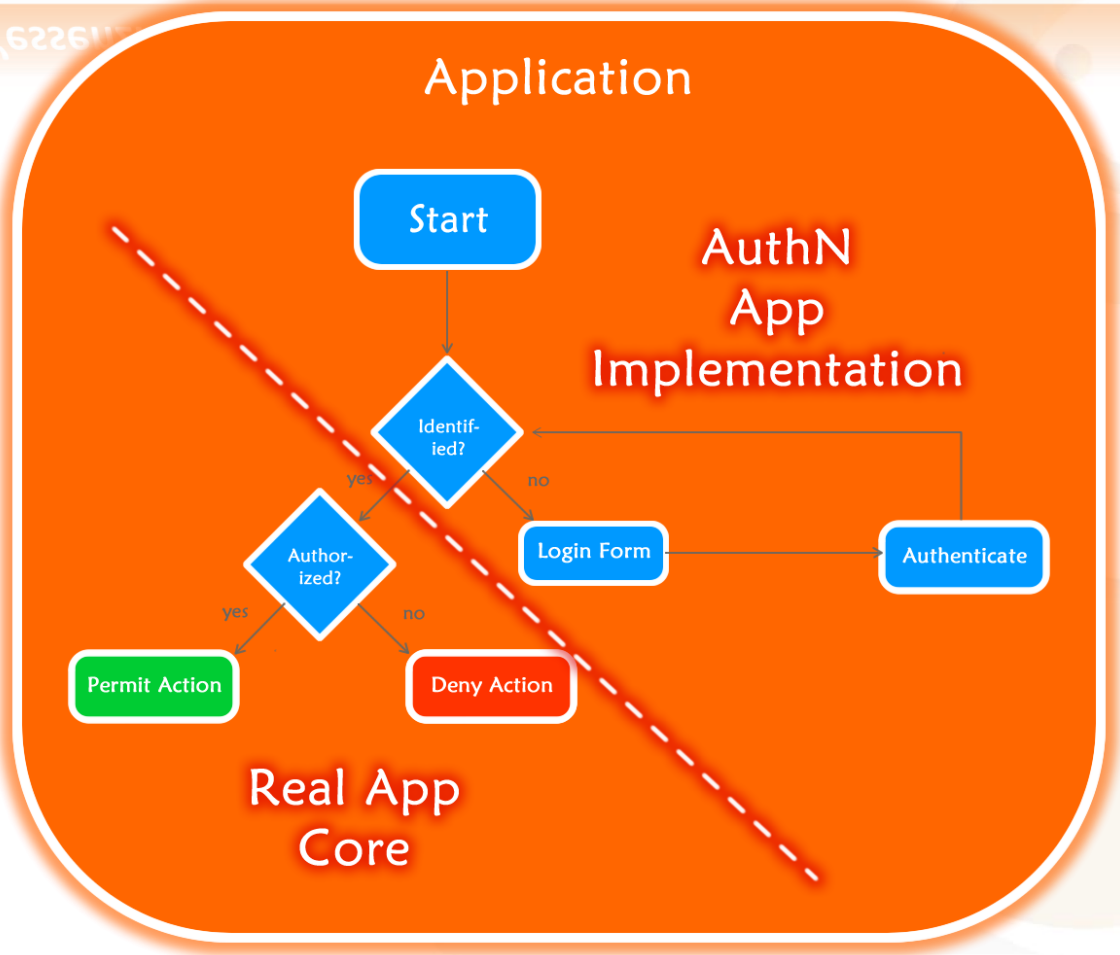


Applicazioni

L'essenziale ed il superfluo…

Il **nucleo di ogni applicazione** è essenzialmente rappresentato dall'insieme **azioni che essa mette a disposizione dell'utilizzatore**: l'identificazione è un operazione presente nelle applicazioni ma necessaria solo per determinare l'insieme di azioni concesse ad un utente (autorizzazione).

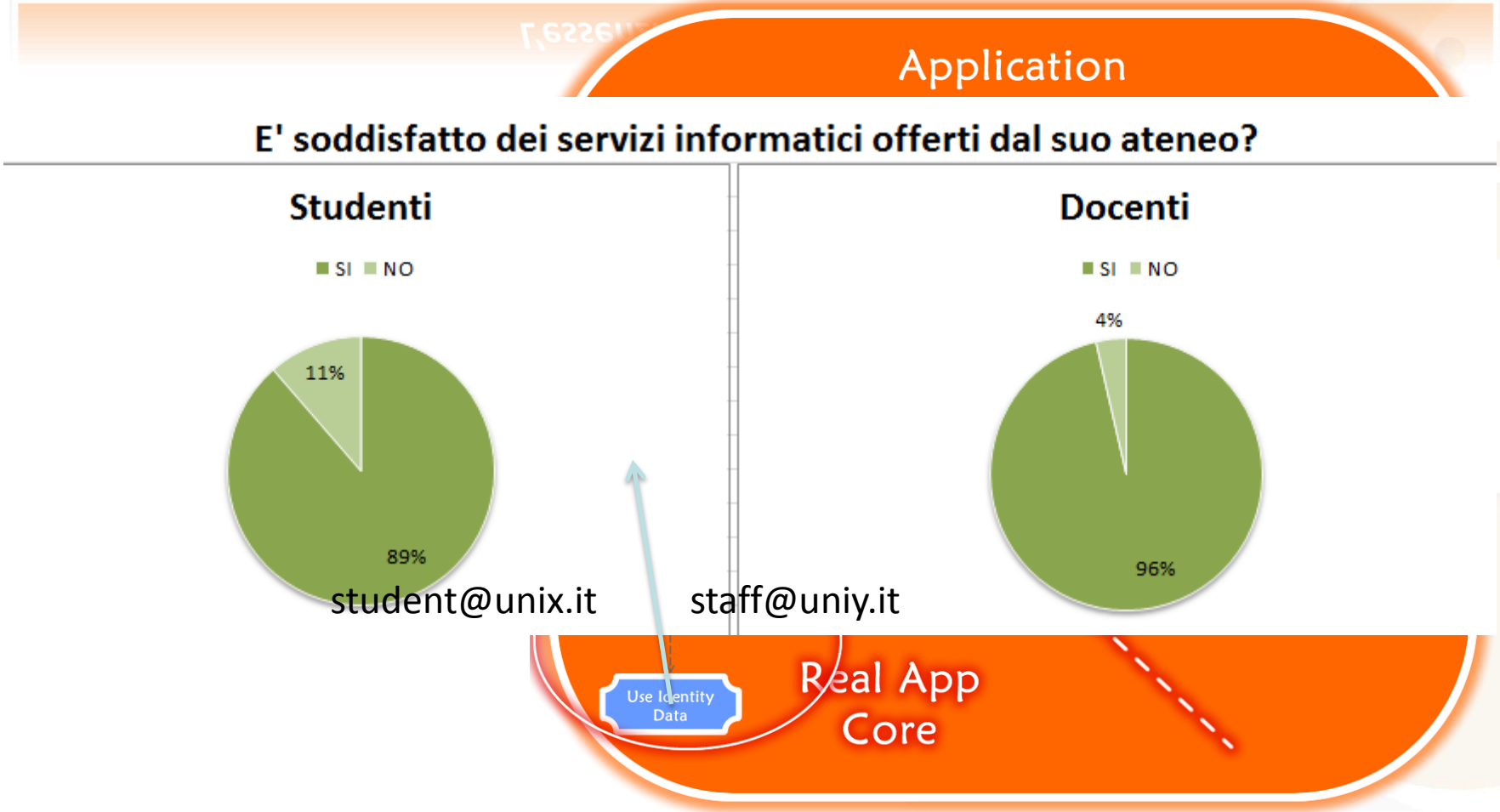
Per definire tale insieme ci si avvale di relazioni dirette o indirette tra gli attributi dell'utente ed azioni



Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Applicazioni

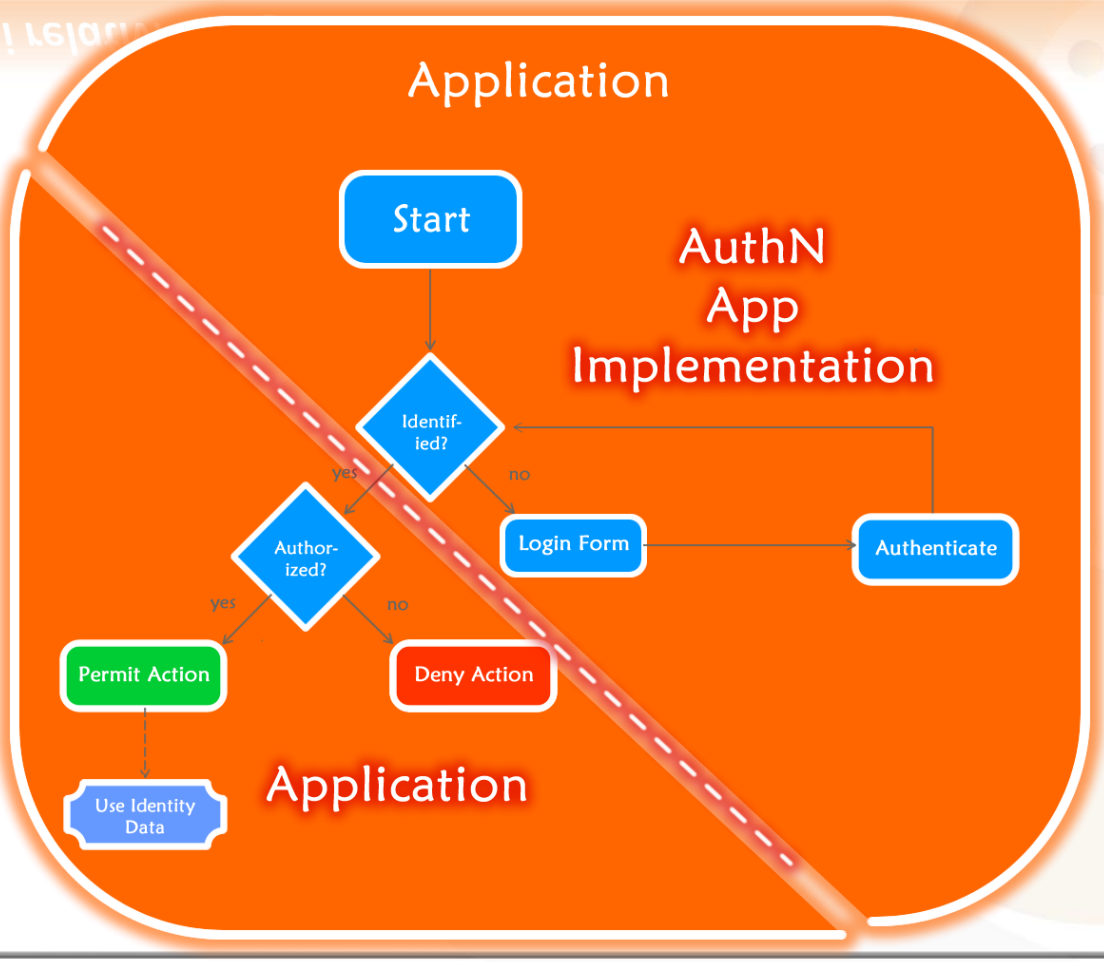
L'essenziale ed il superfluo…



Applicazioni

Problemi relativi alla gestione delle identità

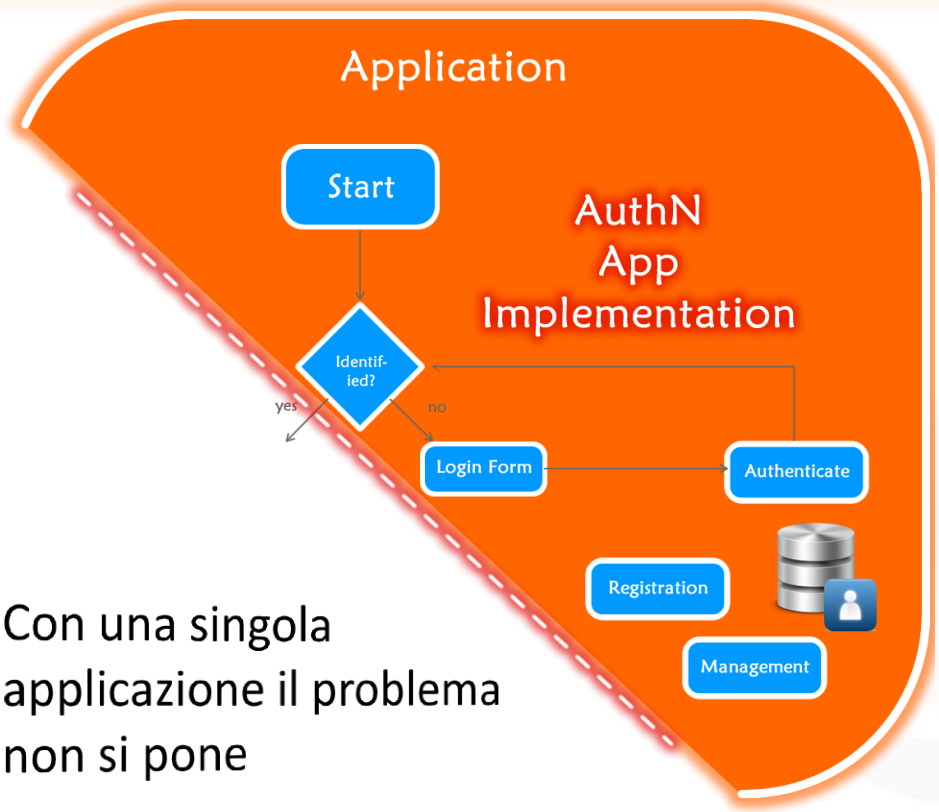
Ma perché dovremmo sentire l'esigenza di federare le applicazioni?



Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Applicazioni

Problemi relativi alla gestione delle identità



Con una singola applicazione il problema non si pone

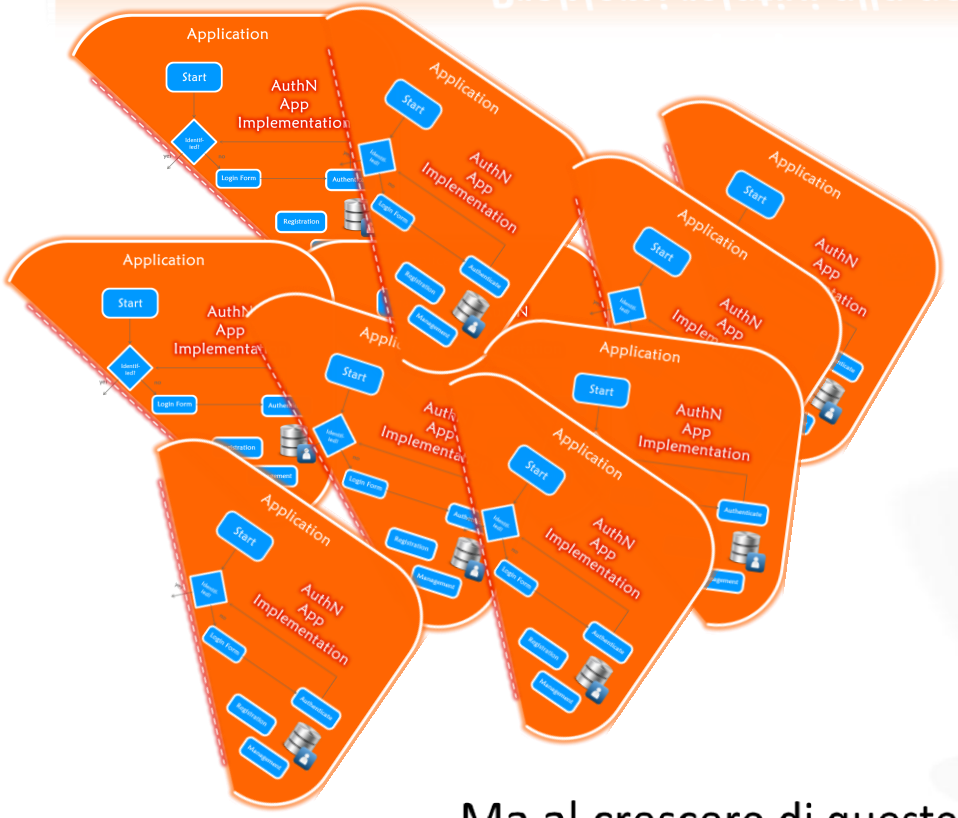

User


Developer


Sysadmin

Applicazioni

Problemi relativi alla gestione delle identità



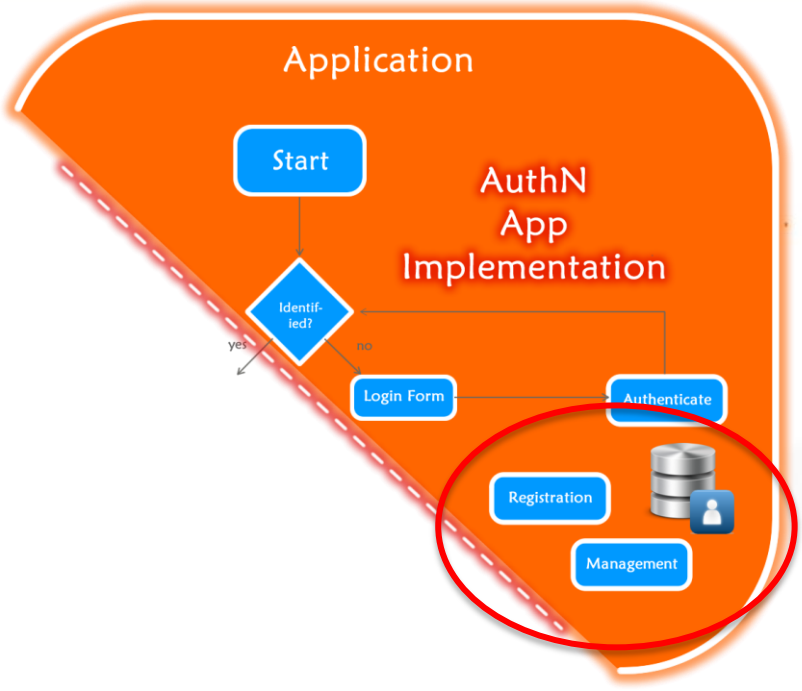
Ma al crescere di queste…

Applicazioni

Problemi relativi alla gestione delle identità

Applicazioni

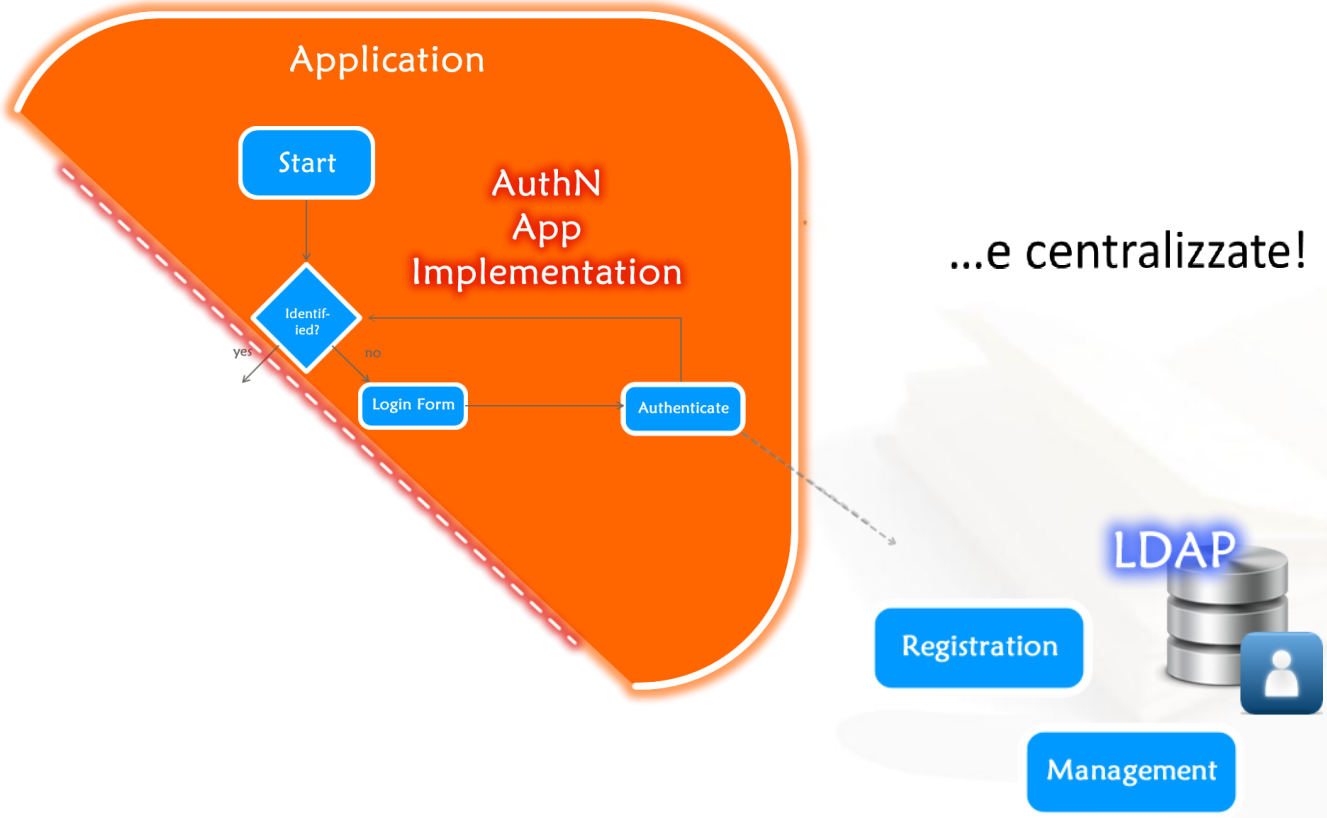
Razionalizzare: eliminare il superfluo



Le varie basi dati degli utenti e le relative interfacce di gestione e registrazione, sono tutte abbastanza simili, potrebbero quindi essere esternalizzate dalle singole applicazioni

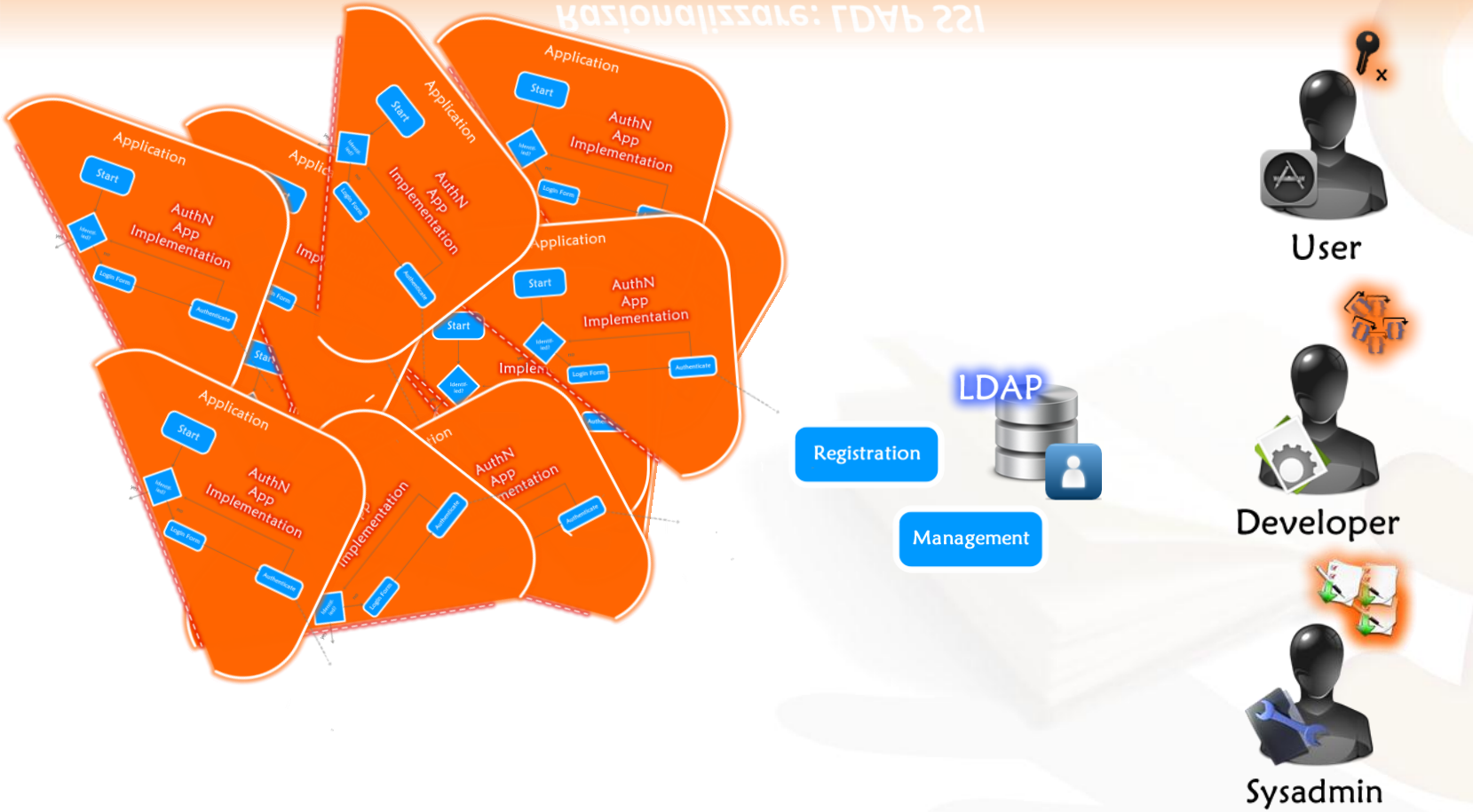
Applicazioni

Razionalizzare: eliminare il superfluo



Applicazioni

Razionalizzare: LDAP SSI



Applicazioni

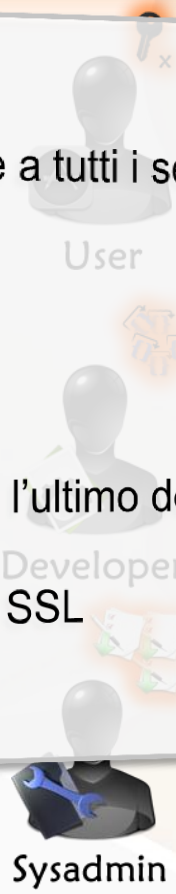
Razionalizzare: LDAP SSI

Limiti:

L'SSI non è SSO: l'utente userà una stessa password per accedere a tutti i servizi ma dovrà comunque singoli login su ognuno di questi

Il che` implica:

Scomodità per l'utente
Rischi di phishing da parte delle applicazioni (ovvero come rendere l'ultimo degli sviluppatori in outsource più potente del IT Manager)
Rischi di sniffing: obbligo ad avere tutte le applicazioni/backends in SSL
Carico per gli sviluppatori e sistemisti ridotto, ma non al minimo

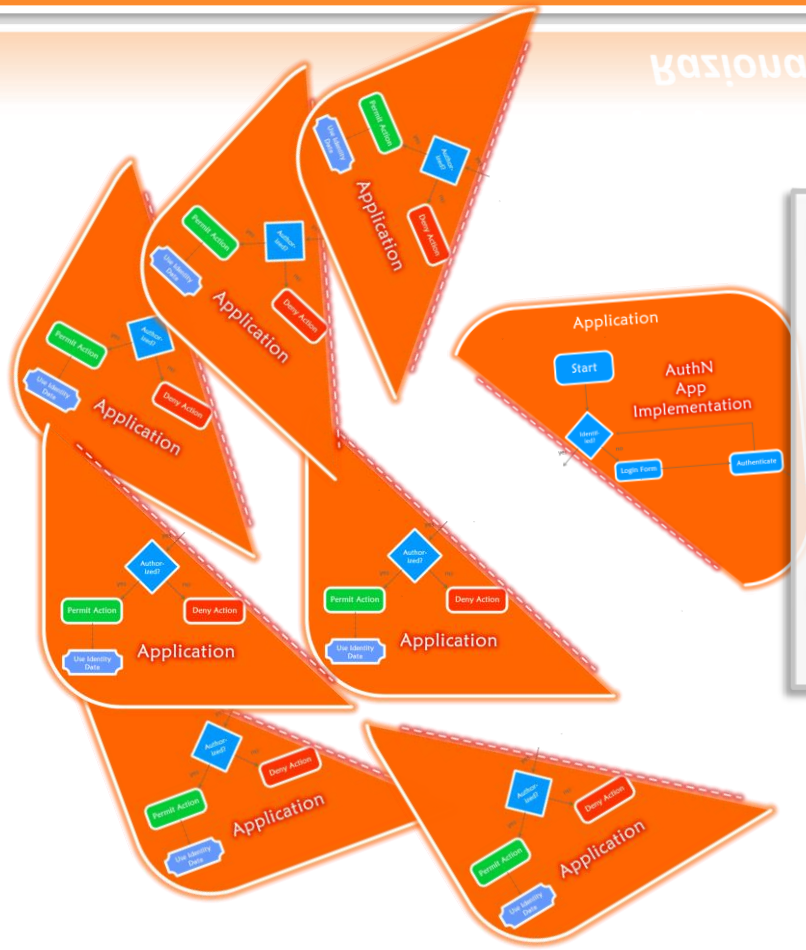


Applicazioni

Razionalizzare: SSO

Applicazioni

Razionalizzare: SSO



Limiti:

Il problema si pone solo quando le applicazioni necessitano di essere accedute anche da utenti esterni all'organizzazione

Applicazioni

Razionalizzare: Federare!

Applicazioni

Razionalizzare: Federare!

Vantaggi:

L'utente accede con le stesse credenziali (immesse una sola volta per sessione) a tutte le applicazioni federate **oltre che** a quelle interne alla sua organizzazione

Lo sviluppatore ed il sistemista trovano il supporto della federazione per implementare i singoli servizi e l'infrastruttura SSO

L'organizzazione non deve più gestire account per utenti esterni

User

Developer

Sysadmin

Federare un'applicazione

Tutorial tecnico

Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Federare un'applicazione

Introduzione

Sessione

- L'Interazione tra un utente ed un'applicazione, rappresentata mediante un insieme di variabili che ne descrivono lo **stato**: proprietà statiche e dinamiche, solitamente memorizzate in maniera tale da poter essere accedute dall'applicazione **solo in presenza dell'utente**

Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Federare un'applicazione

Introduzione: Sessione utente

JSESSIONID A5A582E39219

Valore

A5A582E39219F59942E8A79CEBD4786D

```
session_name("JSESSIONID");
session_start();
$user=$_SESSION['loggedUser'];
```

```
$sessionID=$_COOKIE["JSESSIONID"];
$user= unserialize(file_get_contents("/var/lib/php5/sess_".$sessionID));
```

['loggedUser']


```
session.save_path = /var/lib/php5
```

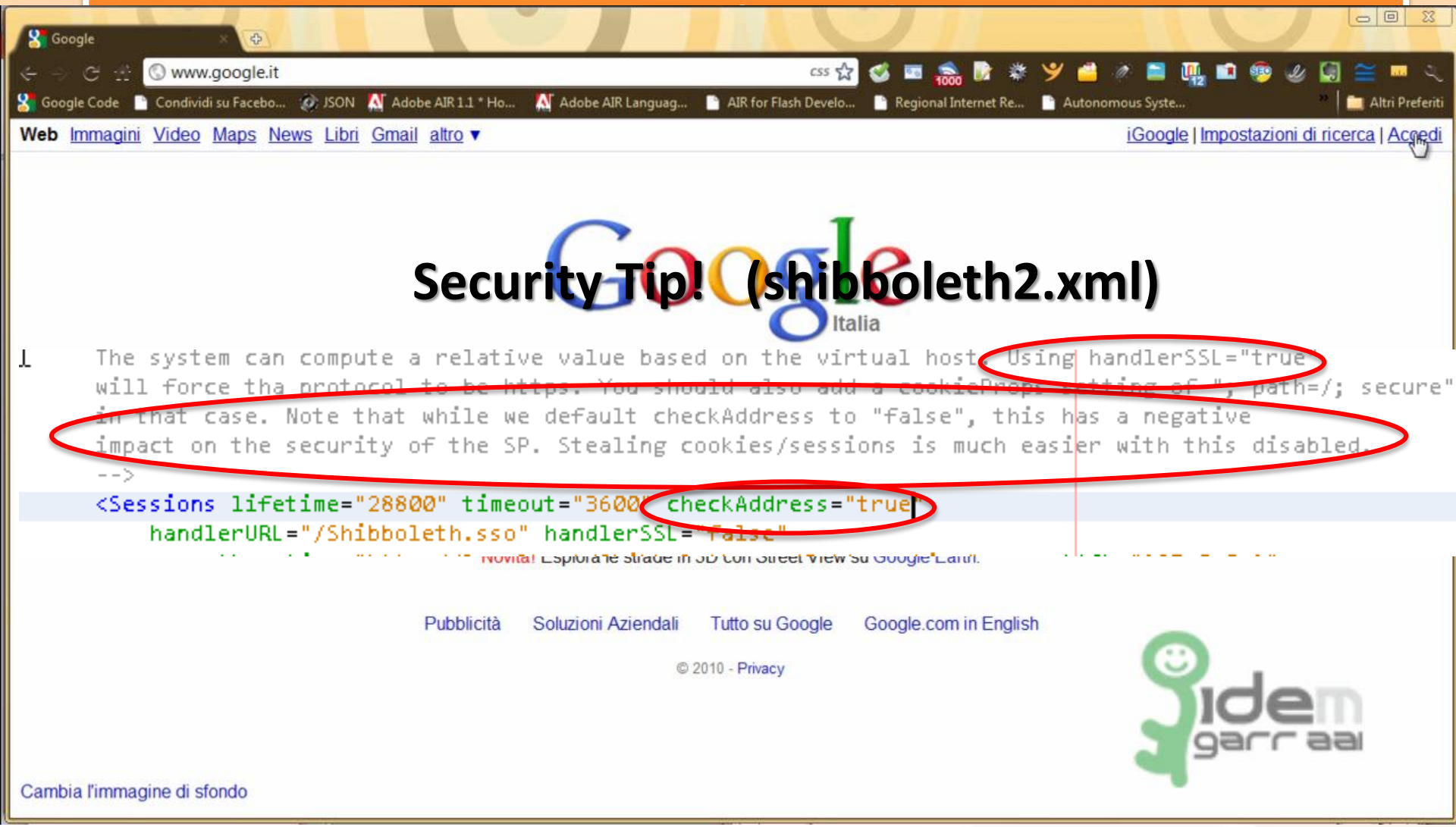
```
 Whether to use cookies.
session.use_cookies = 1

session.cookie_secure =
```

```
    root     124K 2010-11-25 17:17 .
    root     4.0K 2010-11-01 06:52 ..
data www-data    0 2010-11-25 17:17 sess_0003a0c84b530d19d2388cbf0f46324f
data www-data    0 2010-11-25 17:09 sess_001c973518e0aee3619a28012984004c
data www-data    0 2010-11-25 17:10 sess_004110228d517eb1db0510771c8c5bb3
```


Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Security Tip! (shibboleth2.xml)



Federare un'applicazione

Proteggere un'applicazione con Shibboleth

Oggi vedremo come federare risorse concrete, casi più complessi di una semplice pagina web da proteggere.

Prima di ciò, per rinfrescare la memoria, vedremo come federare un'applicazione usando Shibboleth SP, nella maniera più semplice e veloce che lo strumento mette a disposizione.

Federare un'applicazione

Proteggere un'applicazione con Shibboleth - Installazione

1. Installare un SP Shibboleth

- ~~Tutorial e guide sull'argomento sono ampiamente disponibili in rete~~
- Ad oggi i repository software ufficiali delle distribuzioni linux più diffuse offrono la possibilità di installare tutto l'occorrente in pochi minuti:

```
dev:~# apt-cache search shibboleth

libapache2-mod-shib - Federated web single sign-on system (Apache module)
libshib-dev - Federated web single sign-on system (development)
libshib-target5 - Federated web single sign-on system (target runtime)
libshib6 - Federated web single sign-on system (runtime)
libapache2-mod-shib2 - Federated web single sign-on system (Apache module)
```

https://www.idem.garr.it/index.php/it/informazioni-tecniche/147-installazione-di-un-service-provider-shibboleth-v2-con-debian

Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Federare un'applicazione

Proteggere un'applicazione con Shibboleth – Configurazione (Fed. side)

2. Configurare un SP Shibboleth nei confronti della federazione

- EntityID, Metadati e Discovery Service
  - Vedere: https://www.idem.garr.it/index.php/it/informazioni-tecniche/147-installazione-di-un-service-provider-shibboleth-v2-con-debian
- Mapping degli attributi
  - Serve ad associare attributi ricevuti dalla federazione a variabili di sessione (Info sull'identità e ruoli dell'utente)
  - Attributi disponibili: https://www.idem.garr.it/index.php/en/technical-information-idem/attributes-idem
  - How to: **decidere quali attributi servono alla nostra applicazione** e poi seguire le guide Shibboleth su come configurare attribute-map.xml (il supporto dello staff tecnico IDEM è sempre disponibile)

Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Federare un'applicazione

Proteggere un'applicazione con Shibboleth – Configurazione (App. side)

3. Configurare un SP Shibboleth nei confronti dell'applicazione

- Apache
  - Abilitare il modulo Shibboleth

```
a2enmod shib2
/etc/init.d/apache2 force-reload
```

- Apache/Applicazione
  - In un .htaccess o nel file di configurazione del virtual host:

```
1  AuthType shibboleth
2  ShibRequireSession On
3  require valid-user
```


Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Federare un'applicazione

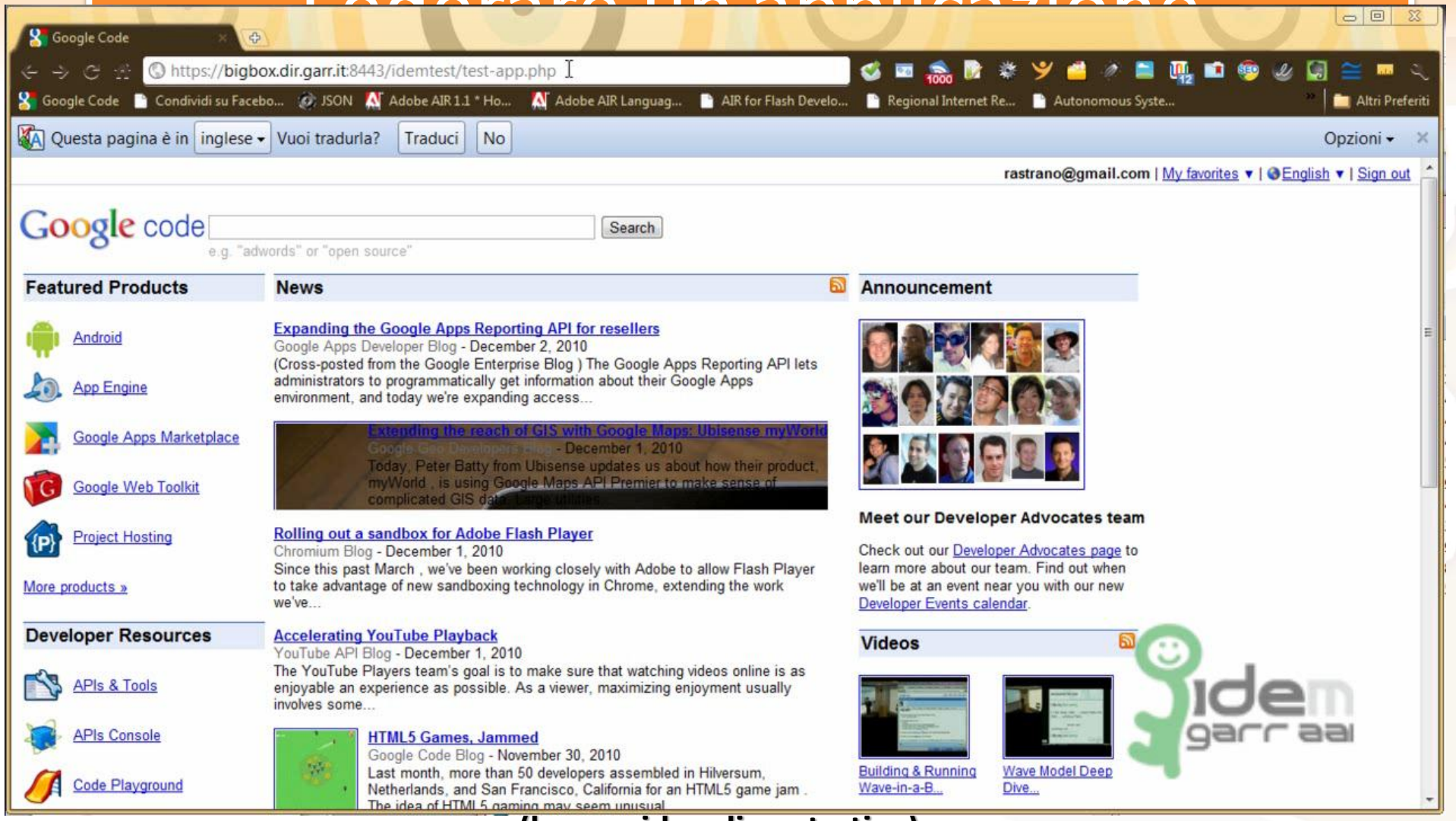
Proteggere un'applicazione con Shibboleth – La nostra applicazione

4. Un'applicazione d'esempio:

```
16
17  <?php
18  /*
19   * Sono un applicazione php molto complessa!
20   */
21  if ($_GET['action']=="a"){
22      echo "Hai eseguito l'azione A!";
23  }
24  if ($_GET['action']=="b"){
25      echo "Hai eseguito l'azione B!";
26  }
27  ?>
28  <h1>
29  Applicazione d'esempio
30  </h1>
31  <p/>
32  <a href="test-app.php?action=a">Esegui l'azione A</a>
33  <p/>
34  <a href="test-app.php?action=b">Esegui l'azione B</a>
35  </body>
36  </html>
```


Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Federare un'applicazione



(breve video dimostrativo)

Federare un'applicazione

Proteggere un'applicazione con Shibboleth - Risultato

Notare come, sebbene l'applicazione sia protetta, in essa non ci sia alcuna traccia di codice AuthN (è stato delegato ad IDEM), ma notare anche come, in questo esempio, l'Autorizzazione sia stata completamente trascurata!

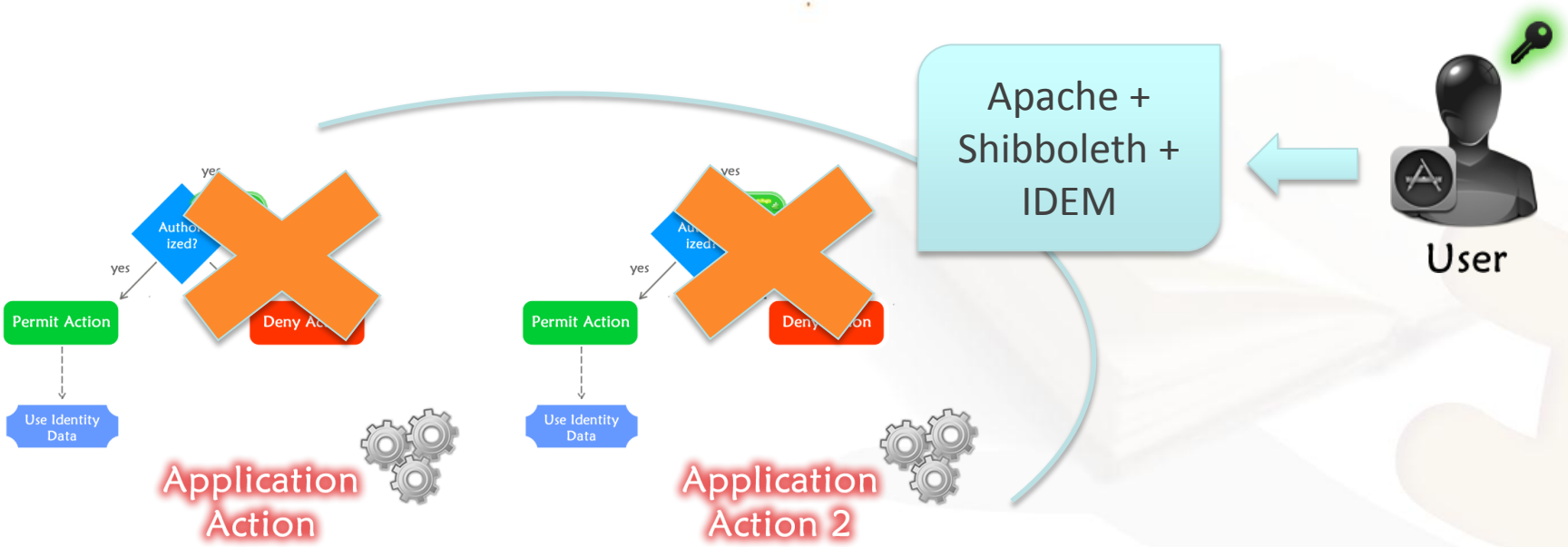
Questa soluzione è infatti semplicissima da implementare, ma è applicabile solo ad applicazioni altrettanto semplici, dato che non effettua alcun tipo di autorizzazione o meglio effettua un autorizzazione che coincide con l'autenticazione.

Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Federare un'applicazione

Proteggere un'applicazione con Shibboleth - Risultato

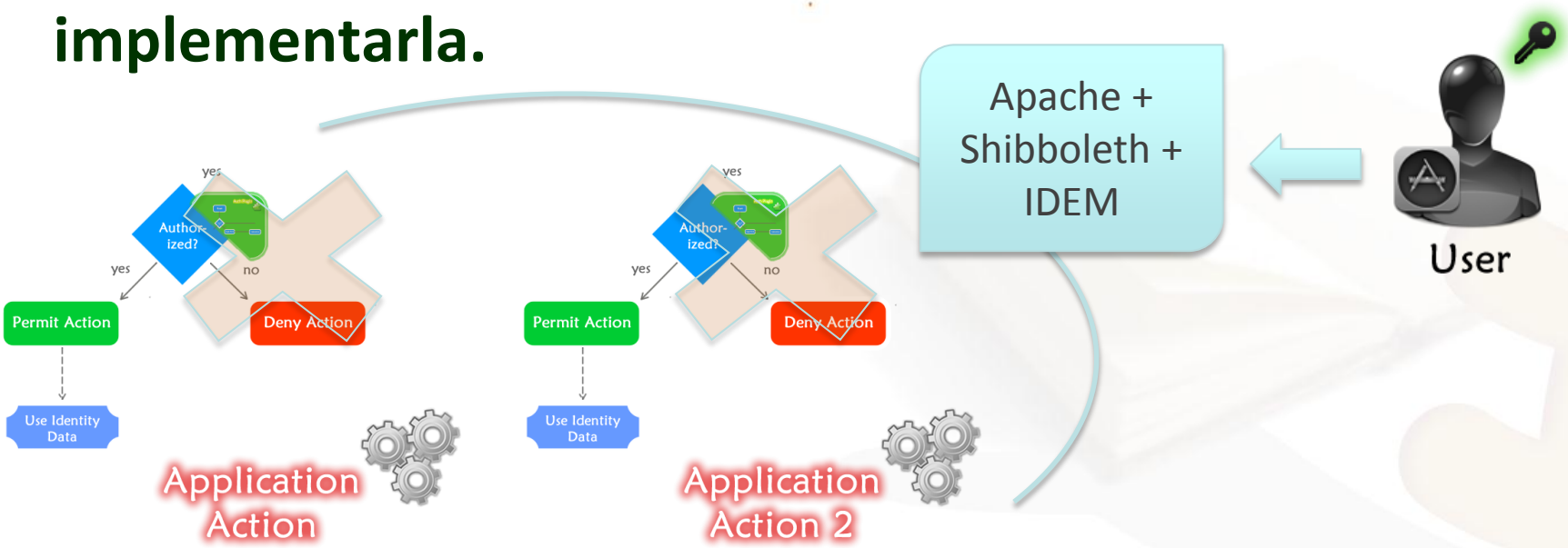
In pratica abbiamo rimosso tutta la logica AA dall'applicazione, permettendo «tutto» oppure «nulla»



Federare un'applicazione

Proteggere un'applicazione con Shibboleth - Risultato

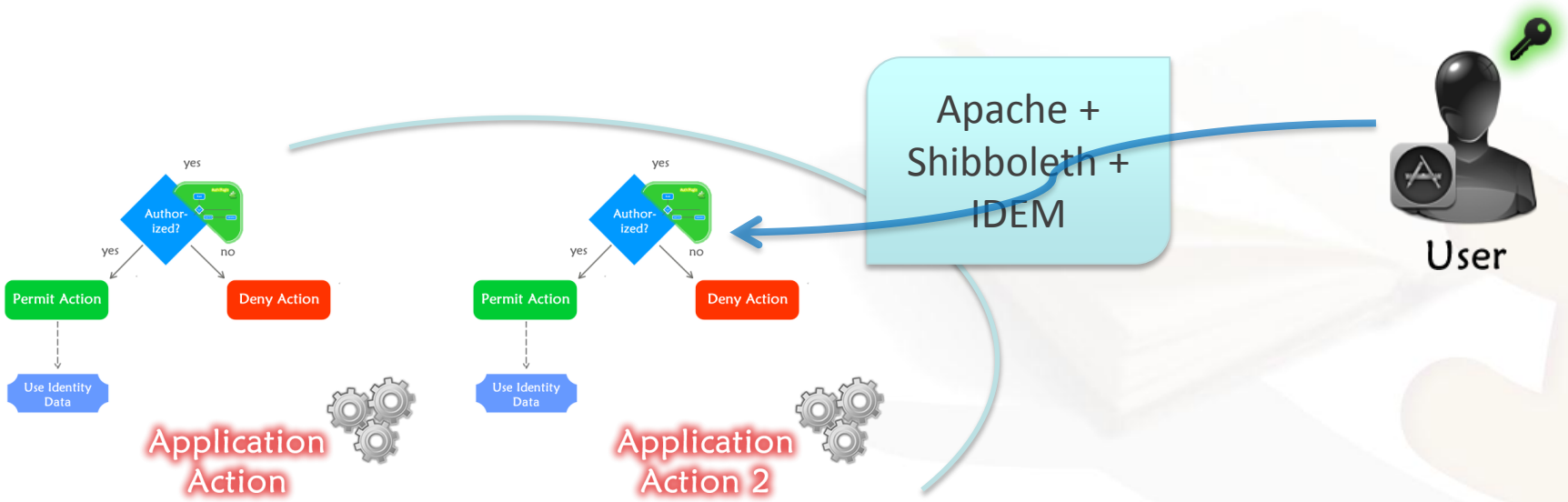
Sappiamo però che un'autorizzazione più o meno granulare sarà tipica della maggior parte delle applicazioni concrete, quindi vediamo come implementarla.



Federare un'applicazione

Proteggere un'applicazione con Shibboleth – Uso degli attributi

L'autorizzazione è possibile solo in presenza di attributi associati all'utente (elementi per discernere l'insieme di azioni concesse ad un identità), quindi vediamo come accedere ad essi dalla nostra applicazione protetta con Shibboleth.



Federare un'applicazione

Proteggere un'applicazione con Shibboleth – Uso degli attributi

In attribute-map.xml abbiamo assegnato un id ad ogni attributo SAML, questo sarà l'identificativo con il quale esso verrà esportato nelle variabili di sessione del server Apache…

```
<Attribute name="urn:mace:dir:attribute-def:eduPersonOrgUnitDN" id="eduPersonOrgUnitDN">
    <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
<Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.4" id="eduPersonOrgUnitDN">
    <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>

<Attribute name="urn:mace:dir:attribute-def:eduPersonScopedAffiliation" id="eduPersonScopedAffiliation">
    <AttributeDecoder xsi:type="ScopedAttributeDecoder" caseSensitive="false"/>
</Attribute>
<Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.9" id="eduPersonScopedAffiliation">
    <AttributeDecoder xsi:type="ScopedAttributeDecoder" caseSensitive="false"/>
</Attribute>
```


Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Federare un'applicazione

Proteggere un'applicazione con Shibboleth – Uso degli attributi

...e quindi reso disponibile alla nostra applicazione

PHP:

```
if ($_SERVER['eduPersonScopedAffiliation'] == 'staff@garr.it')
    { grantAccess() }
```

Perl:

```
if ($ENV{'eduPersonScopedAffiliation'} == 'staff@garr.it')
    { &grantAccess() }
```

Java (con Tomcat dietro Apache+Shib):

```
if(request.getHeader("eduPersonScopedAffiliation").equals("staff@garr.i
   t"))
    { grantAccess() }
```


Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Federare un'applicazione

Proteggere un'applicazione con Shibboleth – Uso degli attributi

Ecco come aggiungere uno strato AuthZ al nostro esempio precedente:

```php
/*
 * Sono un applicazione php molto complessa!
 */
$action = $_GET['action'];
$userrole = $_SERVER['eduPersonScopedAffiliation']; //sostituitelo con un oggetto complesso che faccia da wrapper a tu
$authUtils = new AuthUtils();
if ($authUtils->checkAuthZ($userrole, $action)) {
    if ($action == "a") {
        echo "Hai eseguito l'azione A!";
    }
    if ($action == "b") {
        echo "Hai eseguito l'azione B!";
    }
    if ($action == "c") {
        echo "Hai eseguito l'azione C!";
    }
    if ($action == "d") {
        echo "Hai eseguito l'azione D!";
    }
} else {
    echo "<span class='errorText'>Non sei autorizzato ad eseguire questa operazione ($action) </span>";
}
?>
```


Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Federare un'applicazione

Proteggere un'applicazione con Shibboleth – Uso degli attributi

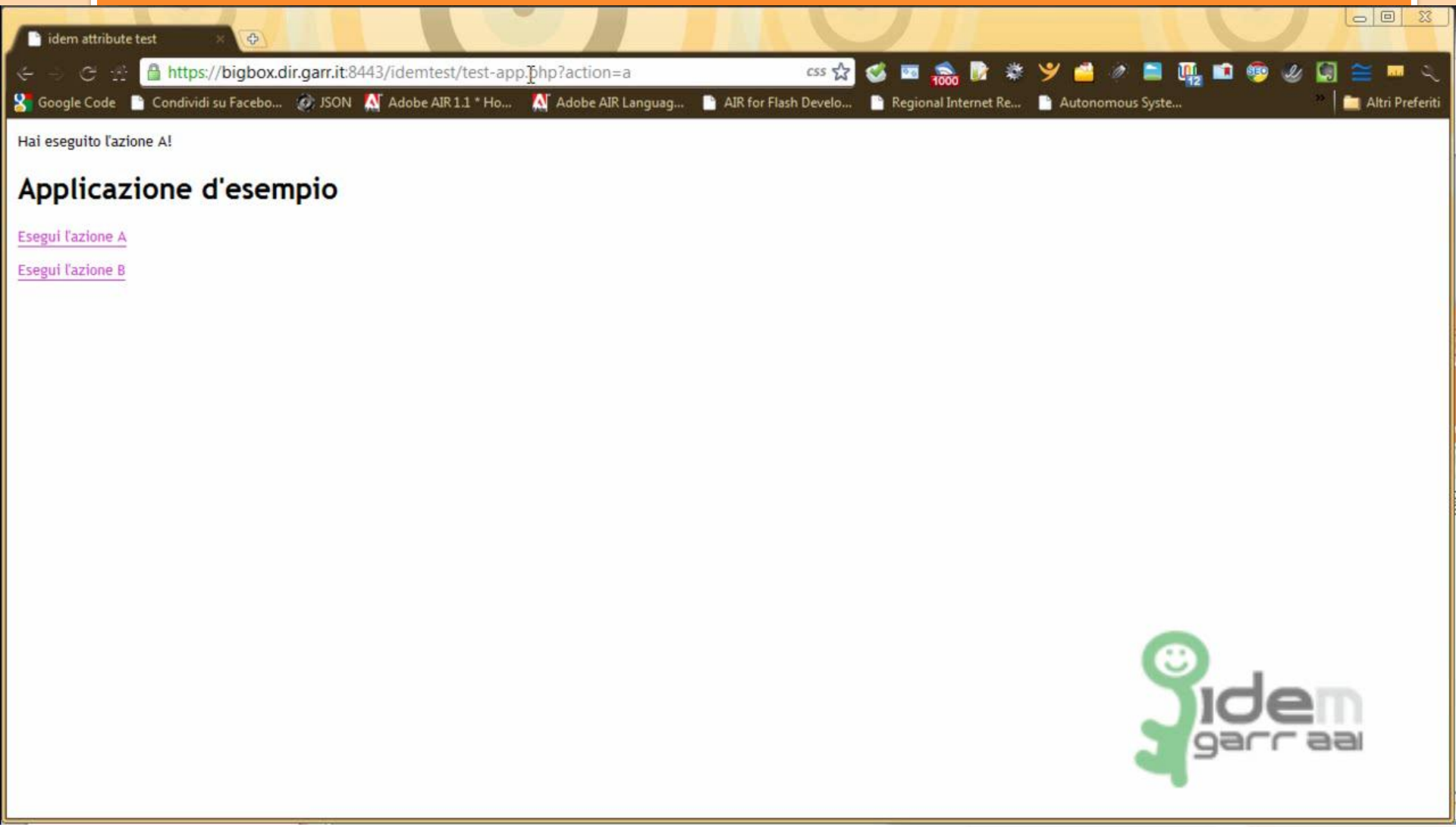
Un caso possibile in IDEM è sfruttare l'attributo eduPersonScopedAffiliation a fini autorizzativi, (per autorizzazioni piu` granulari si potrebbe richiedere un valore specifico di eduPersonEntitlement o avere un mapping dei ruoli in locale)

```
class AuthUtils {

    function checkAuthZ($ur, $action) {
        switch ($action) {
            case "a": if (stristr($ur, "member@cnr.it")) { //l'azione a è permessa a tutti i membri cnr
                    return true;
                }break;
            case "b": if (stristr($ur, "member@infn.it")) { //l'azione d è permessa a tutti i membri infn
                    return true;
                }break;
            case "c": if (preg_match("/student\@.+.it/",$ur)) { //l'azione c è permessa a tutti gli studenti di
            //tutte le università italiane
                    return true;
                }break;
            case "d": return true; //l'azione d è pubblica
        }
        return false;
    }

}
```

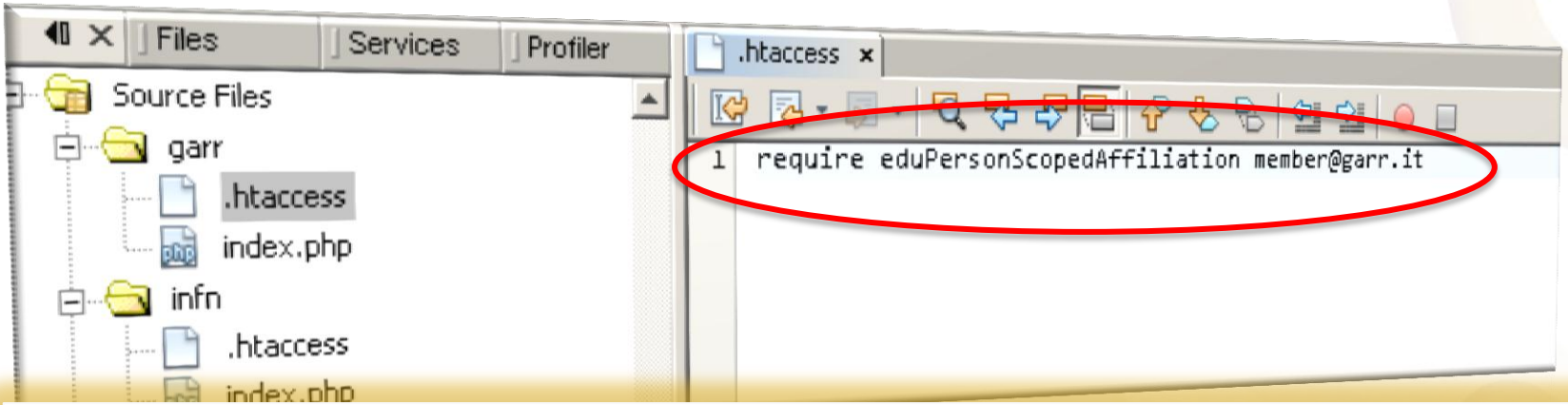

Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)


idem attribute test
https://bigbox.dir.garr.it:8443/idemtest/test-app.php?action=a
Google Code
Condividi su Facebo...
JSON
Adobe AIR 1.1 * Ho...
Adobe AIR Languag...
AIR for Flash Develo...
Regional Internet Re...
Autonomous Syste...
Altri Preferiti
Hai eseguito l'azione A!
Applicazione d'esempio
Esegui l'azione A
Esegui l'azione B
idem garr aai

Federare un'applicazione

Proteggere un'applicazione con Shibboleth – AuthZ Attributi Metodo2

Ci sono molti modi:



QUESTA TECNICA PUO` PERO` ESSERE MOLTO UTLITE PER PROTEGGERE DIRECTORY DI DOCUMENTI, OPPURE APPLICAZIONI SEMPLICI CHE DEVONO ESSERE ACCEDUTE SOLO DA DETERMINATE CATEGORIE DI UTENTI MA CHE NON PREVEDONO POI MOLTI RUOLI O PERMESSI DISTINTI AL LORO INTERNO

Federare un'applicazione

Proteggere un'applicazione con Shibboleth – LAZY SESSION

In ogni caso non ci accontentiamo: abbiamo visto esempi che «forzano» un redirect verso il flusso di login IDEM non appena l'utente tenta di accedere ad un'applicazione. Questo non è un modo molto elegante per dire all'utente che è possibile fare login con IDEM: lo disorienta, inoltre è molto limitante per la nostra applicazione dato che, ad esempio, potremmo voler supportare più sistemi di login contemporaneamente (eg. Più federazioni, login legacy, login facebook, twitter, google ecc.) oppure offrire delle azioni pubbliche fruibili anche da utenti non identificati.

Per questo dobbiamo introdurre il concetto di

lazy session

Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

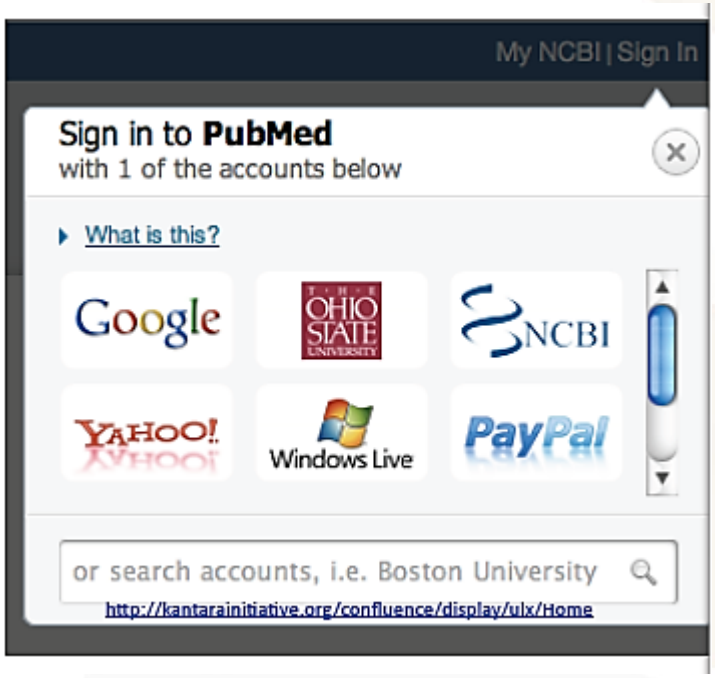
Federare un'applicazione

Proteggere un'applicazione con Shibboleth – LAZY SESSION

Il concetto è elementare: la richiesta di autenticazione SAML (AuthNRequest) deve essere inoltrata al DS o all'IdP solo quando ce ne è effettivamente bisogno: solo quando l'utente decide di effettuare il login con IDEM.

Oltre ad essere una soluzione elegante e flessibile è anche quella che offre più privacy e comodità all' utente finale:

Se un'applicazione ha dei contenuti pubblici, egli potrebbe fruirli senza doversi identificare.



Federare un'applicazione

Proteggere un'applicazione con Shibboleth – LAZY SESSION

Con Shibboleth il trucco è semplice: innanzitutto bisogna configurare l'.htaccess principale dell'applicazione in maniera tale da aspettarsi una sessione Shibboleth, ma non in modo da richiederla forzatamente.

```
1 AuthType shibboleth
2 ShibRequireSession On
3 require valid-user
```

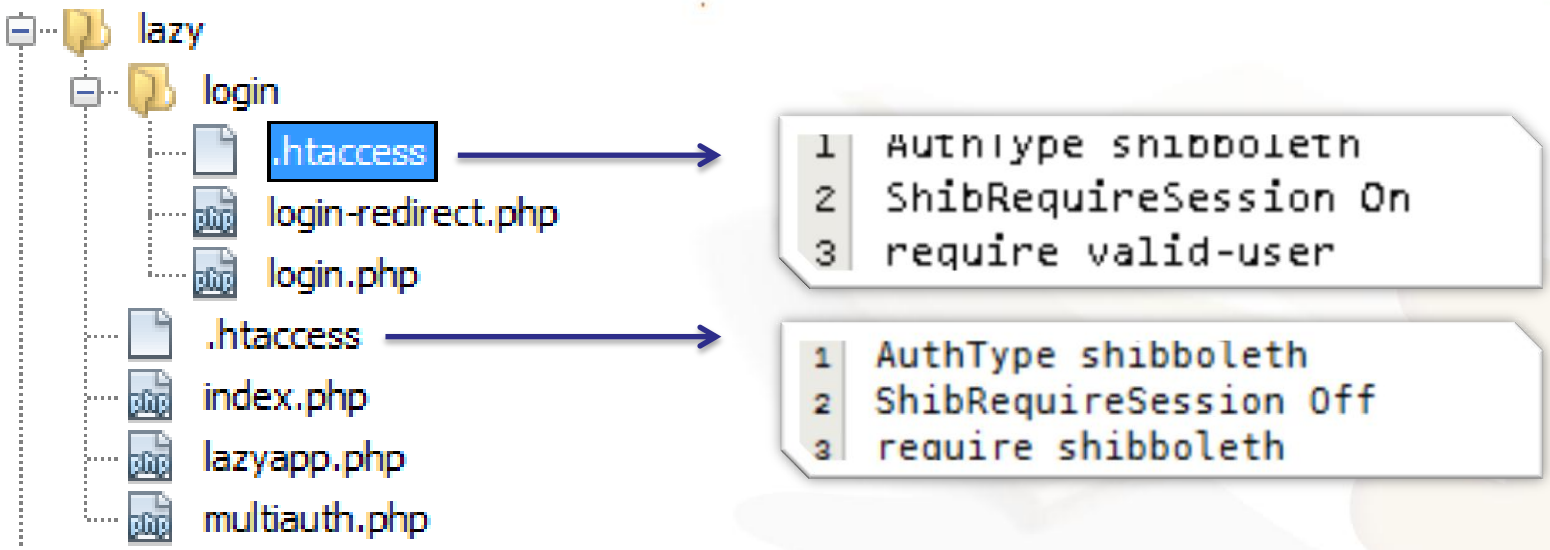
```
1 AuthType shibboleth
2 ShibRequireSession Off
3 require shibboleth
```


Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Federare un'applicazione

Proteggere un'applicazione con Shibboleth – LAZY SESSION

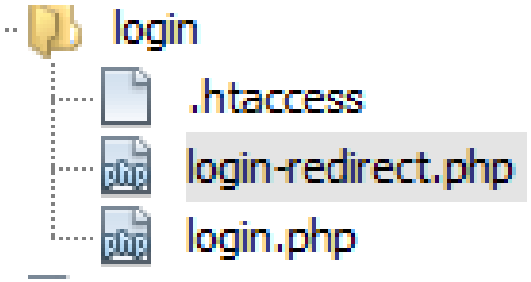
Poi bisogna creare una sottosezione della nostra applicazione che invece richieda forzatamente il login Shibboleth (causi l'invio di un AuthN Request).



Federare un'applicazione

Proteggere un'applicazione con Shibboleth – LAZY SESSION

Questa sezione servirà solo da inizializzatore del flusso di login SAML, i file al suo interno sono privi di ogni funzionalità applicativa, al più possono contenere un messaggio di benvenuto o meglio, un redirect che riporti alla pagina dalla quale si è effettuato il login.

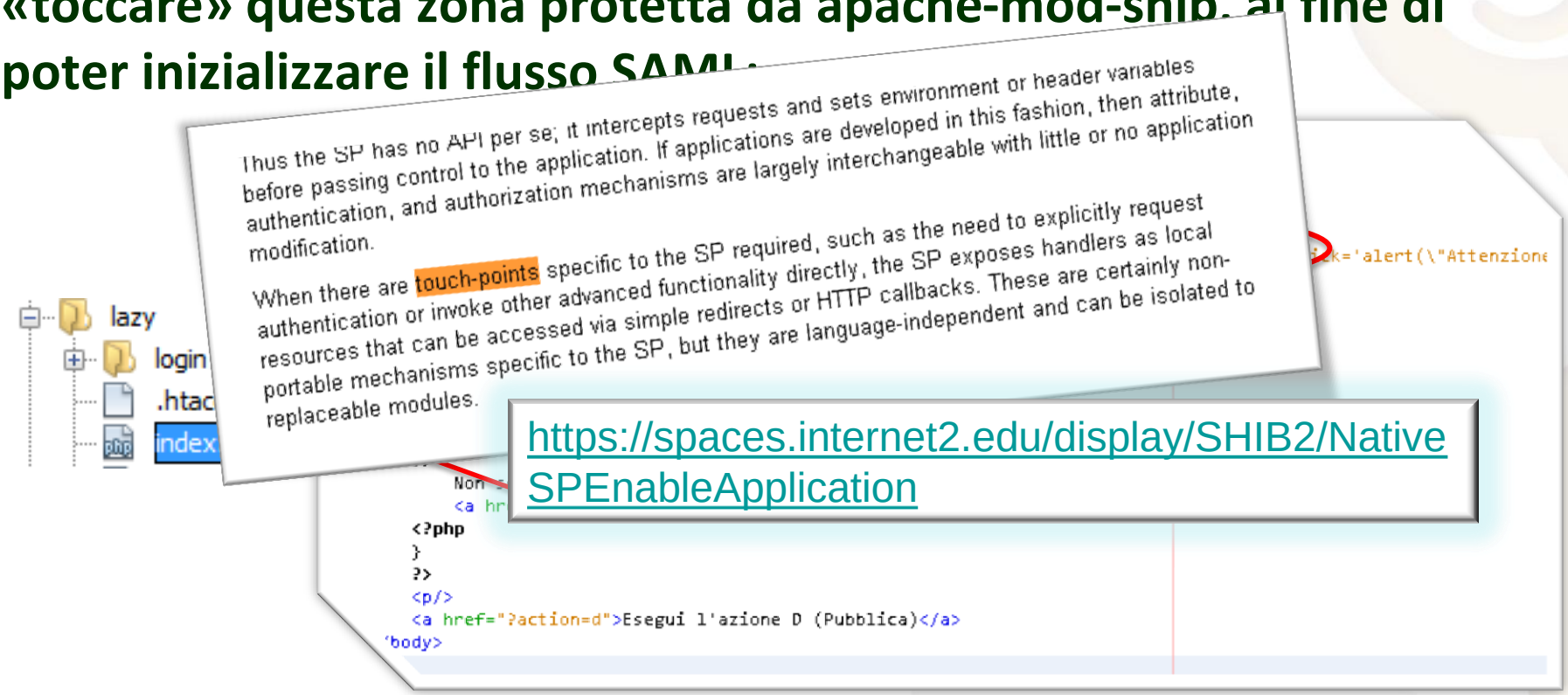



```php
<?php
header('Location: ../index.php');
?>
```

Federare un'applicazione

Proteggere un'applicazione con Shibboleth – LAZY SESSION

La nostra applicazione deve solo dare un modo all'utente di «toccare» questa zona protetta da apache-mod-shib, al fine di poter inizializzare il flusso SAML



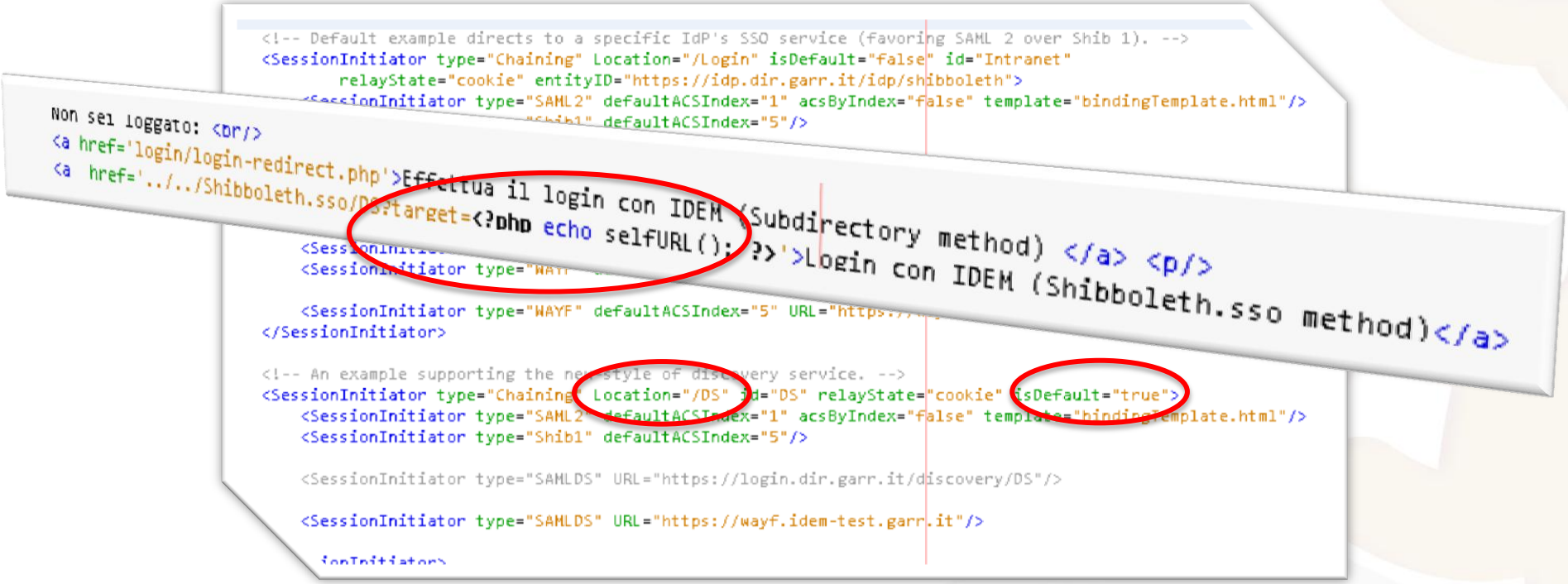
https://spaces.internet2.edu/display/SHIB2/NativeSPEnableApplication

Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

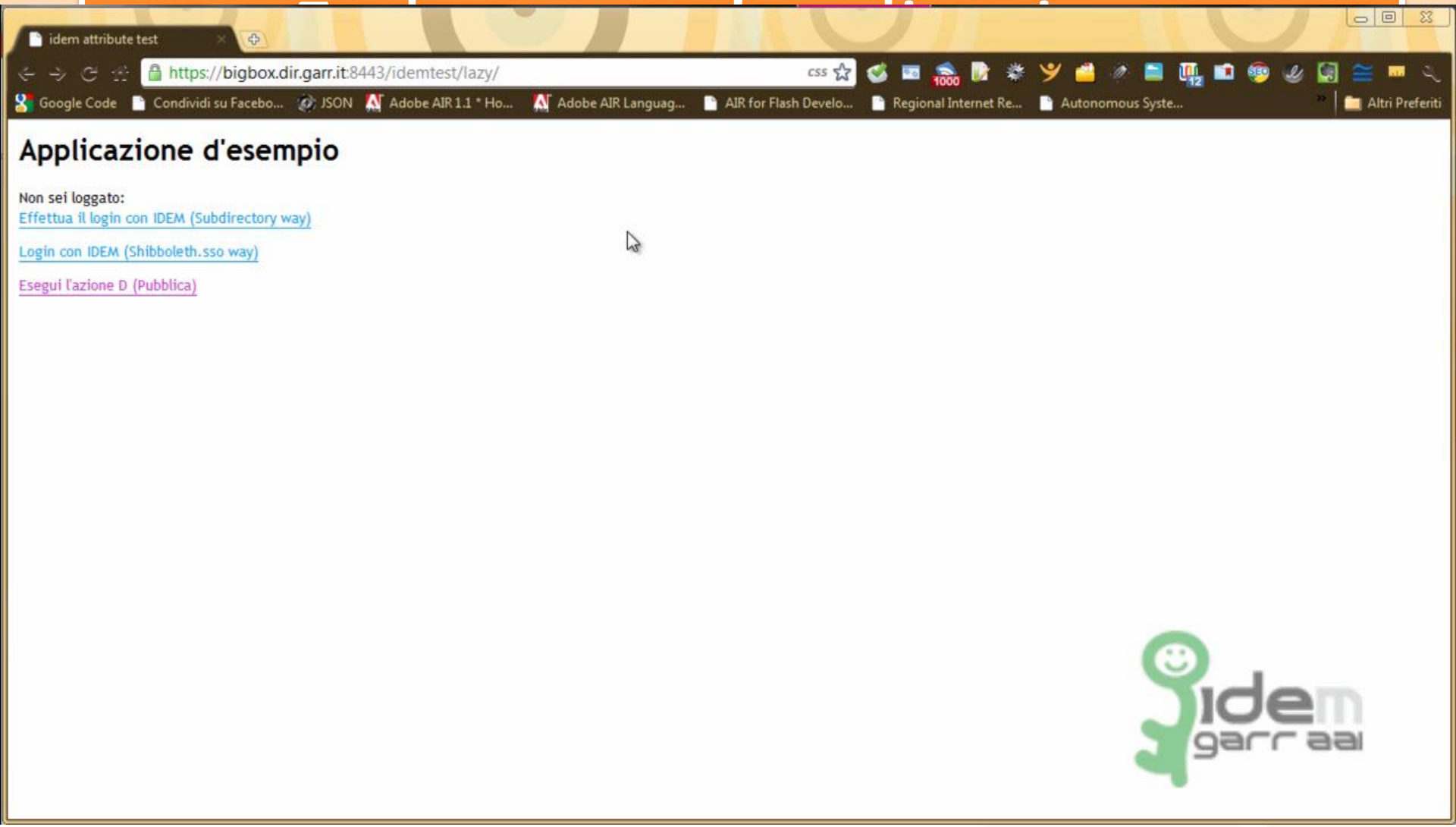
Federare un'applicazione

Proteggere un'applicazione con Shibboleth – LAZY SESSION

In pratica abbiamo creato dei punti che fungono da «Inizializzatori di Sessione», potremmo però anche usare direttamente quelli messi a disposizione da Shibboleth (consigliato)



Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)


idem attribute test
https://bigbox.dir.garr.it:8443/idemtest/lazy/
Google Code
Condividi su Facebo...
JSON
Adobe AIR 1.1 * Ho...
Adobe AIR Languag...
AIR for Flash Develo...
Regional Internet Re...
Autonomous Syste...
Altri Preferiti
Applicazione d'esempio
Non sei loggato:
Effettua il login con IDEM (Subdirectory way)
Login con IDEM (Shibboleth.sso way)
Esegui l'azione D (Pubblica)

Federare un'applicazione

Proteggere un'applicazione con Shibboleth – LAZY SESSION

Shibboleth SP è stato sviluppato come un modulo di Apache, non come API, se da una parte questo offre il vantaggio di supportare più linguaggi di programmazione (tutti quelli supportati da Apache), dall'altra per usarlo bisogna interagire con la configurazione del webserver e trovare un modo di sfruttarne le funzionalità nell'applicazione (come abbiamo visto con le lazy session). Ma IDEM non è solo Shibboleth. IDEM è una federazione basata sullo standard interoperabile SAML 2.0. Esistono anche altri frameworks utilizzabili per federare la nostra applicazione.

Vediamone un altro.

Federare un'applicazione

Proteggere un'applicazione con SimpleSAMLphp

SimpleSAMLphp è un implementazione pure PHP di SAML 2.0 (sia IdP che SP)

A differenza di Shibboleth SP, è stato progettato per essere utilizzato in maniera programmatica, quindi implementa le lazy session di default (L'AuthN viene richiesta via codice: l'AuthN Request viene invocata via API, non da Apache)

Federare un'applicazione

Proteggere un'applicazione con SimpleSAMLphp: Installazione

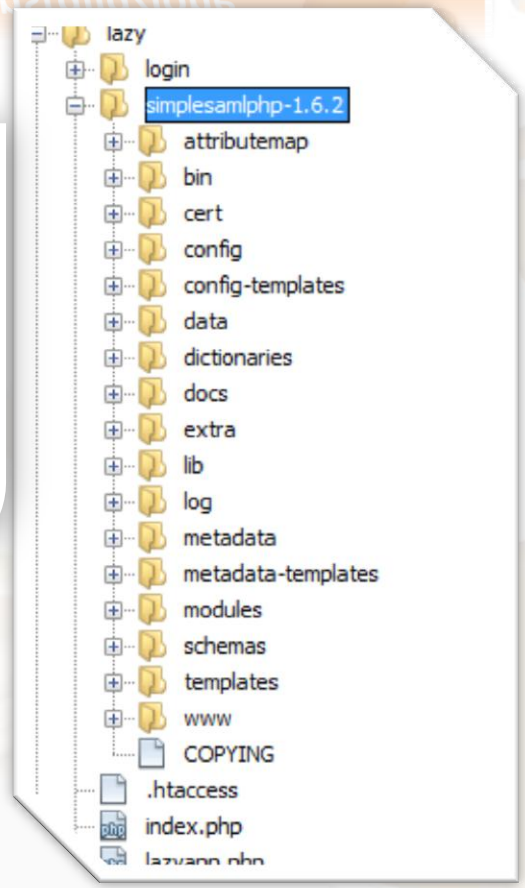
Installare simpleSAMLphp è un operazione molto semplice:

- Verificare i requisiti di sistema

Prerequisites

- Some webserver capable of executing PHP scripts.
- PHP version >= 5.2.0 if you are using simpleSAMLphp as an SP connected to a Shibboleth IdP; PHP version >= 5.1.2 if not.
- Suppoort for the following PHP extensions:
  - Always required: `date`, `dom`, `hash`, `libxml`, `openssl`, `pcre`, `SPL`, `zlib`
  - When encrypting assertions: `mcrypt`
  - When authenticating against LDAP server: `ldap`
  - When authenticating against RADIUS server: `radius`
  - When saving session information to memcache-server: `memcache`
  - When using database:
    - Always: `PDO`
    - Database driver: (`mysql`, `pgsql`, ...)

What actual packages are required for the various extensions varies between different platforms and distributions.



- Scaricare i sorgenti
  http://code.google.com/p/simplesamlphp/downloads/list
- Estrarli in una sottodirectory della nostra applicazione (ciò ne renderà portabile la configurazione ma può essere fatto altrove se si preferisce)

Federare un'applicazione

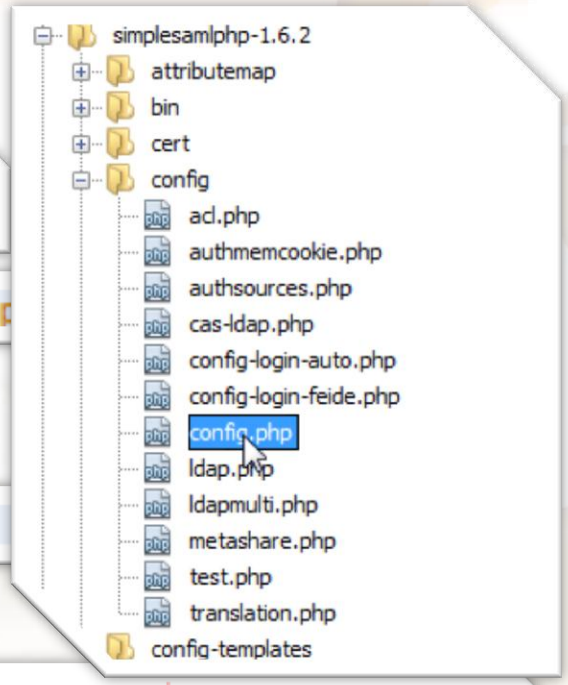
Proteggere un'applicazione con SimpleSAMLphp: Configurazione

- **Editare il file config/config.php**
  - Password dell'interfaccia di amministrazione:

```
'auth.adminpassword'            => 'newpassword',
```

```
 * A possible way to generate a random salt is by running the following command from a unix shell:
 * tr -c -d '0123456789abcdefghijklmnopqrstuvwxyz' </dev/urandom | dd bs=32 count=1 2>/dev/null;echo
 */
'secretsalt' => 'defaultsecretsalt',
```

```
'baseurlpath'                   => 'idemtest/lazy/simp
```



  - Generare nuovi certificati per l'SP e posizionarli in una location protetta

```
'certdir'                       => '/usr/local/certs/',
```

  - Configurare la sincronizzazione dei metadati di federazione

```
'metadata.sources' => array(
    array('type' => 'flatfile'),
    array('type' => 'xml', 'url' => 'http://www.idem.garr.it/docs/conf/idem-metadata.xml'),
//* no-cache: solo per test!! usare mod-metarefresh in produzione!!
),
```


Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Federare un'applicazione

Proteggere un'applicazione con SimpleSAMLphp: Configurazione SP

- **Editare il file config/autsources.php**

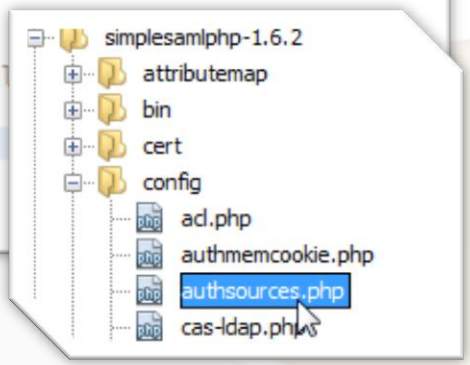
```
// An authentication source which can authenticate against both SAML 2.0
// and Shibboleth 1.3 IdPs.
'default-sp' => array(
        'saml:SP',

        // The entity ID of this SP.
        // Can be NULL/unset, in which case an entity ID is generated based on the metadata URL.
        'entityID' => NULL,

        // The entity ID of the IdP this should SP should contact.
        // Can be NULL/unset, in which case the user will be shown a list of available IdPs.
        'idp' => NULL,

        // The URL to the discovery service.
        // Can be NULL/unset, in which case a builtin discovery service wil
        // 'discoURL' => NULL,
        'discoURL' => 'https://wayf.idem-test.garr.it/',
```

```
'privatekey' => 'server.key',
'certificate' => 'server.crt',
'redirect.sign' => TRUE,
'redirect.validate' => TRUE,
```

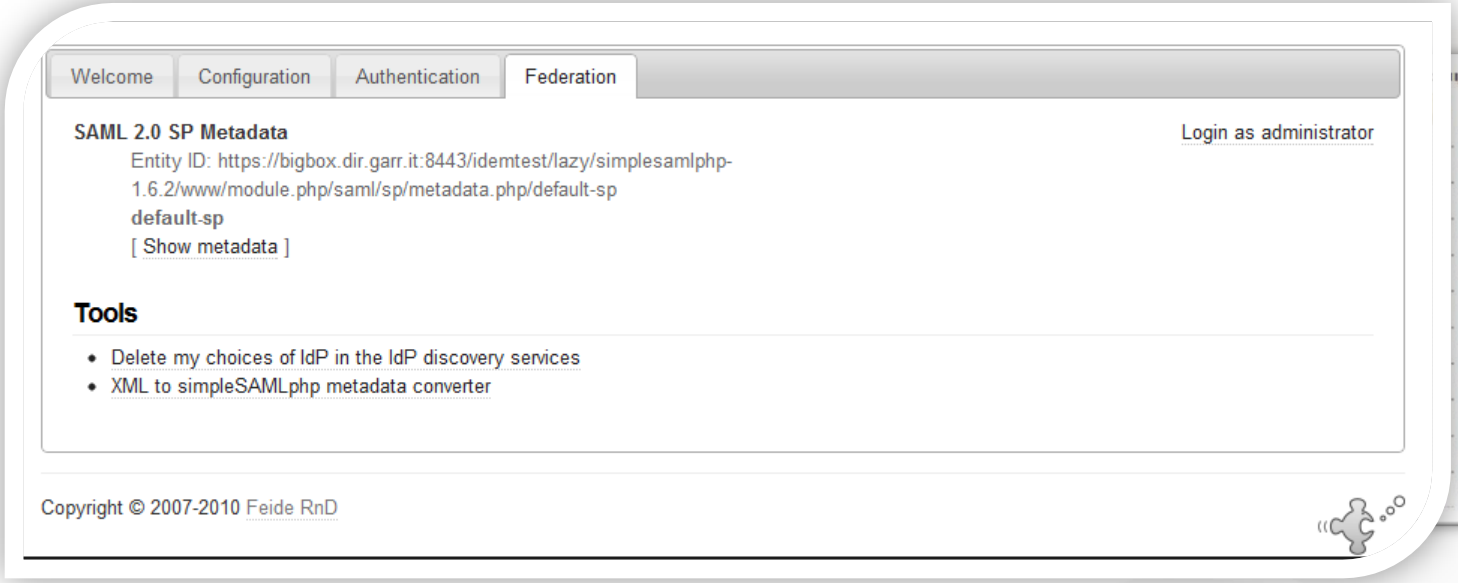


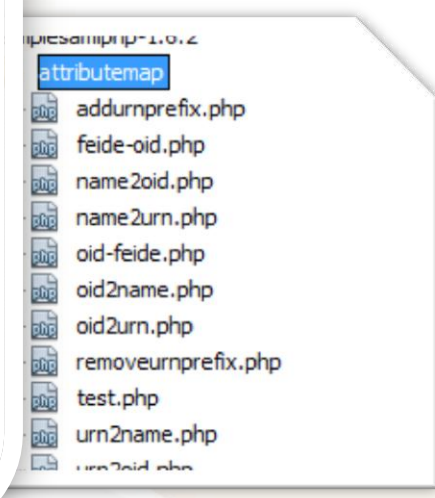
Federare un'applicazione

Proteggere un'applicazione con SimpleSAMLphp: Configurazione SP

- accedere a https://myappurl/simplesamlphp_directory/www: l'interfaccia di amministrazione di simplesamlphp che permetterà varie operazioni come la verifica dell'installazione, test delle funzionalità e recupero dei metadati dell'SP da passare alla federazione

E' già tutto pronto! l'attribute mapping è possibile (vedere in attributemap/*), ma non serve dato che di default simpleSAML accetta qualsiasi attributo riceva dall'IdP, quindi potremo reperirli nell'applicazione semplicemente usando il loro nome urn:oid di federazione.





Federare un'applicazione

Proteggere un'applicazione con SimpleSAMLphp: Uso SP

```
$userrole = $_SERVER['eduPersonScopedAffiliation'];
```

```
require_once('simplesamlphp-1.6.2/lib/_autoload.ph

/* Get a reference to our authentication source. *
$as = new SimpleSAML_Auth_Simple('default-sp');

/* Require the user to be authentcated. */
if (isset($_GET["login"])) {
    $as->requireAuth();
}
```

Rimuovendo questa condizione si avrebbe un login richiesto in maniera forzata (non lazy)

```
$attributes = $as->getAttributes();
$userrole = $attributes['urn:oid:1.3.6.1.4.1.5923.1.1.1.9'][0];
```


Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Federare un'applicazione

Proteggere un'applicazione con SimpleSAMLphp: Uso SP

```
if (isset($_GET["login-strong"])) {
    $as->login(array(
        'saml:AuthnContextClassRef' => 'urn:oasis:names:tc:SAML:2.0:ac:classes:X509',
    ));
}

if (isset($_GET["login-ip"])) {
    $as->login(array(
        'saml:AuthnContextClassRef' => 'urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol',
    ));
}
```

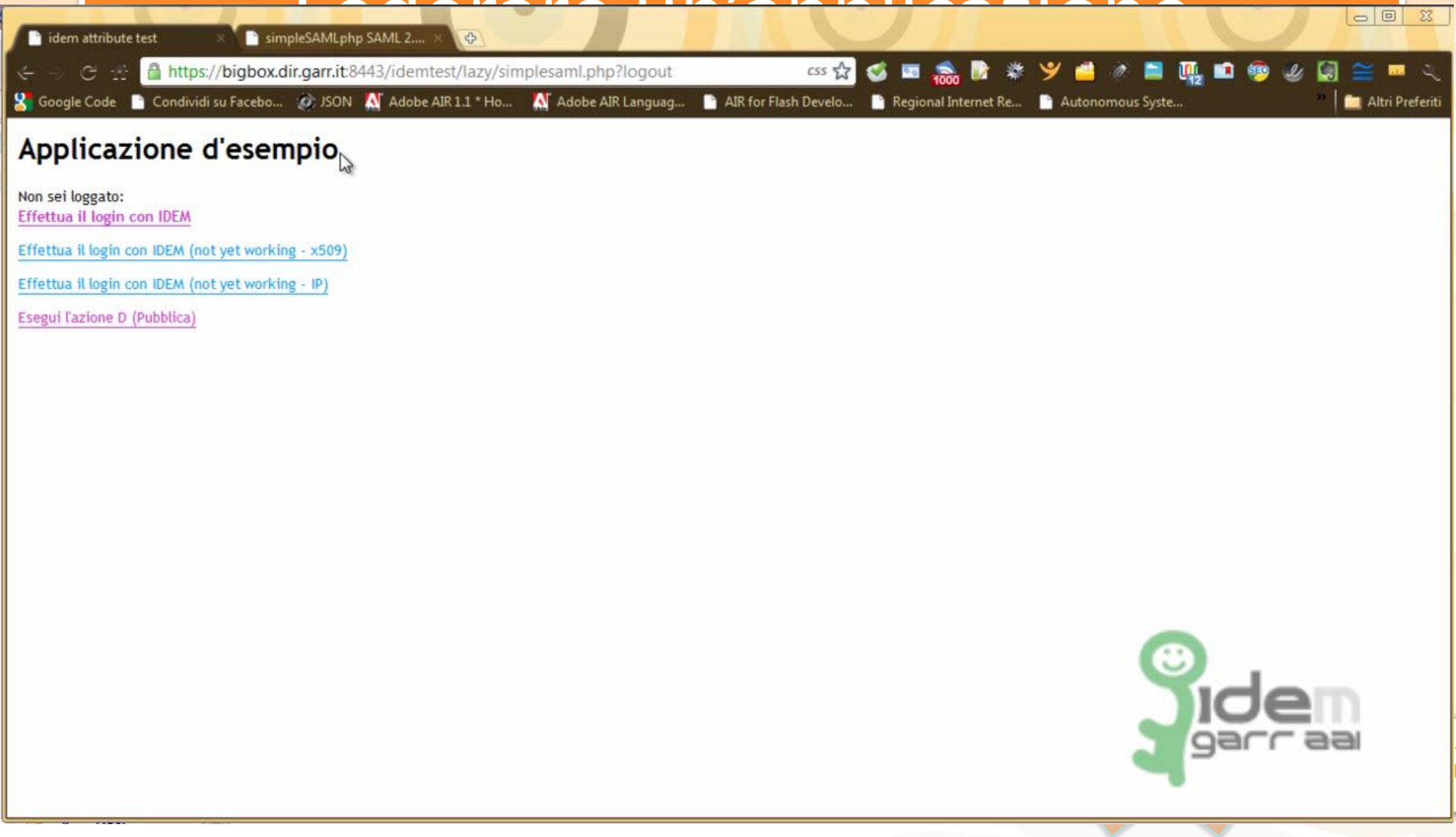
3.1.1 Authentication method

The <AuthnContext> element (required) indicates how that authentication was done. Note that the authentication statement does not provide the means to perform that authentication, such as a password, key, or certificate. This element will contain an authentication context class reference.[5]
Available authentication methods and their corresponding URNs are provided in the following table:

Authentication Method	URN
Internet Protocol	urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol
Internet Protocol Password	urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocolPassword
Password	urn:oasis:names:tc:SAML:2.0:ac:classes:Password
Password Protected Transport	urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
Kerberos	urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos
Previous Session	urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession
Secure Remote Password	urn:oasis:names:tc:SAML:2.0:ac:classes:SecureRemotePassword
SSL/TLS Certificate	urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient
X.509 Public Key	urn:oasis:names:tc:SAML:2.0:ac:classes:X509
PGP Public Key	urn:oasis:names:tc:SAML:2.0:ac:classes:PGP
SPKI Public Key	urn:oasis:names:tc:SAML:2.0:ac:classes:SPKI
XML Digital Signature	urn:oasis:names:tc:SAML:2.0:ac:classes:XMLDSig
Unspecified	urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified

Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Federare un'applicazione



Federare un'applicazione

Altri frameworks SAML2

Pure Java:

- OIOSAML (Basato su OpenSAML, la stessa base di Shibboleth IdP)
- OpenSSO/Fedlet/OpenAM
- Spring Security SAML Module

.NET:

- OIOSAML.net
- ADFS

Federare applicazioni esistenti

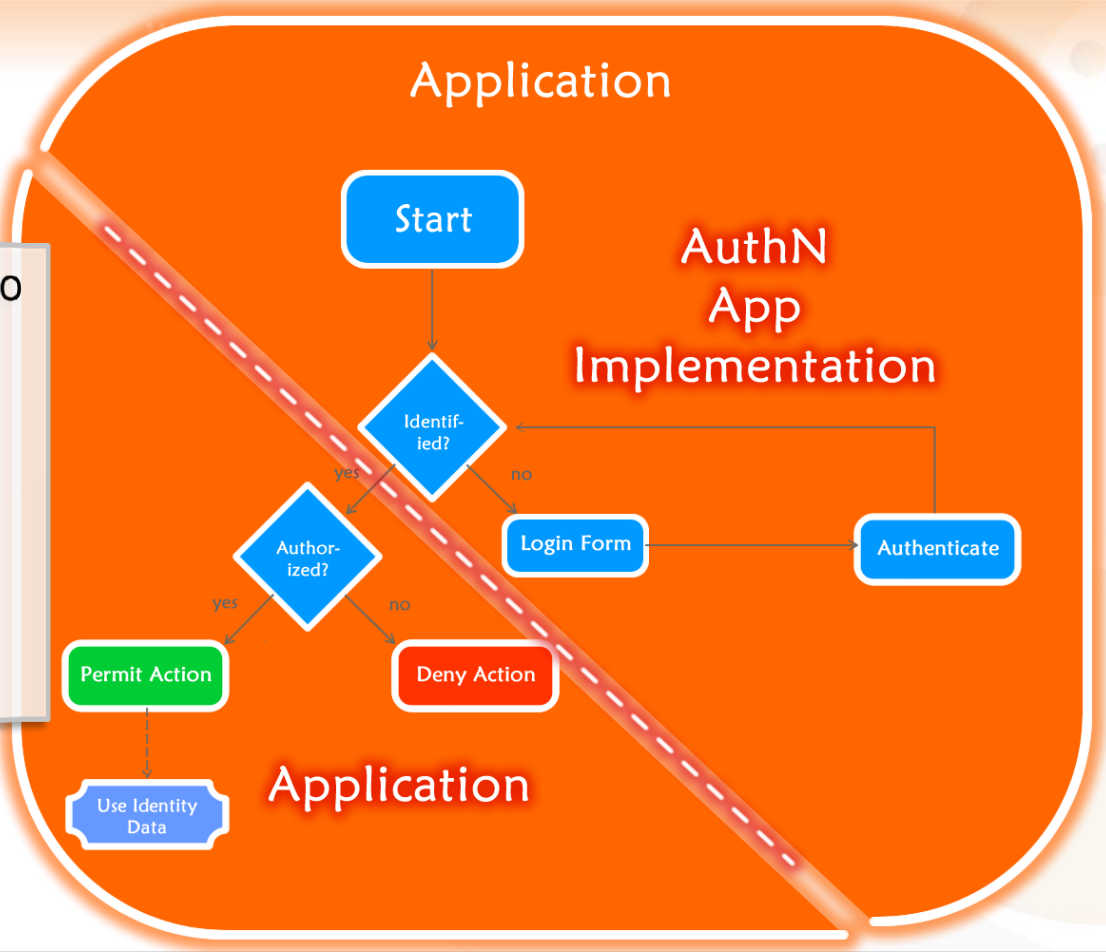
O terze parti.

Federare un'applicazione

Introduzione

Torniamo al modello generico di applicazione, ed immaginiamola come la nostra applicazione da federare

Cos'è necessario fare per renderlo federabile?



Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Federare un'applicazione

Modularizzare

Innanzitutto bisogna tener presente che è buona norma rendere modulare la parte che si occupa dell'autenticazione e dell'inizializzazione della sessione utente.

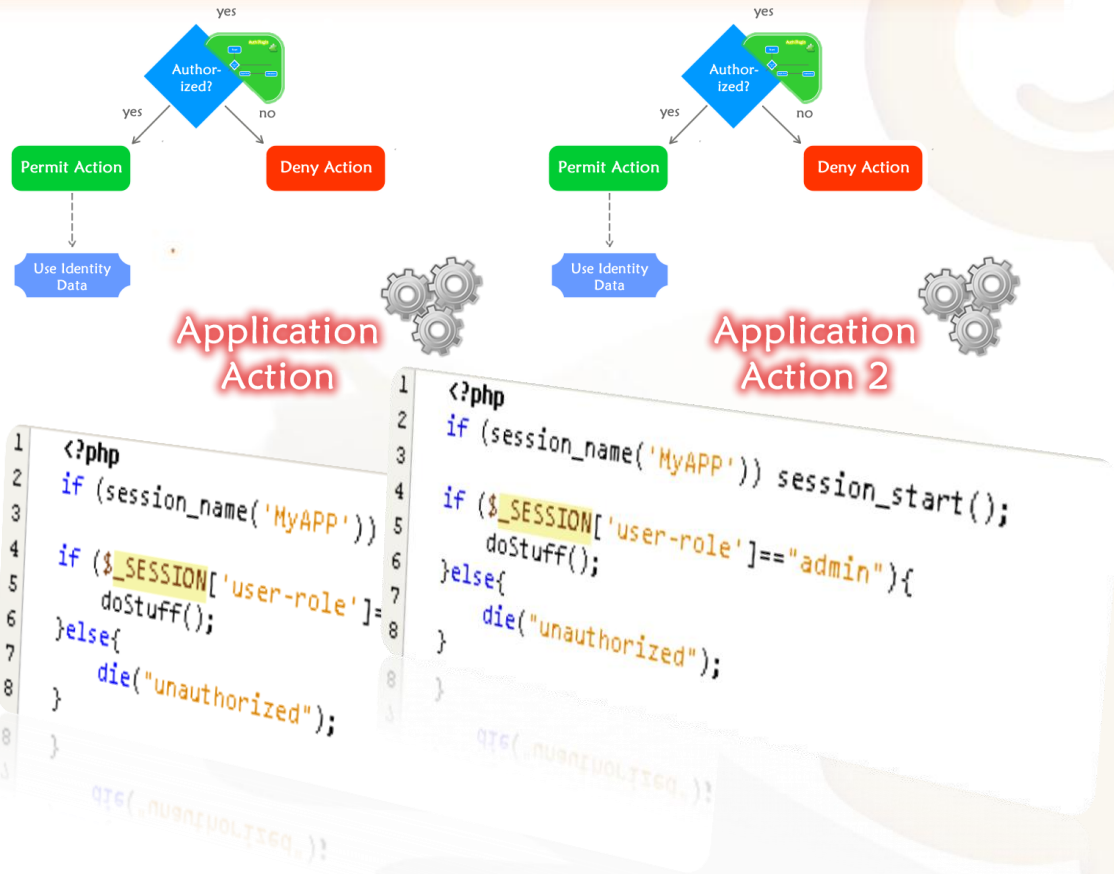
Molte applicazioni moderne lo fanno di default, prevedendo plugins per l'autenticazione.

Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Federare un'applicazione

Modularizzare

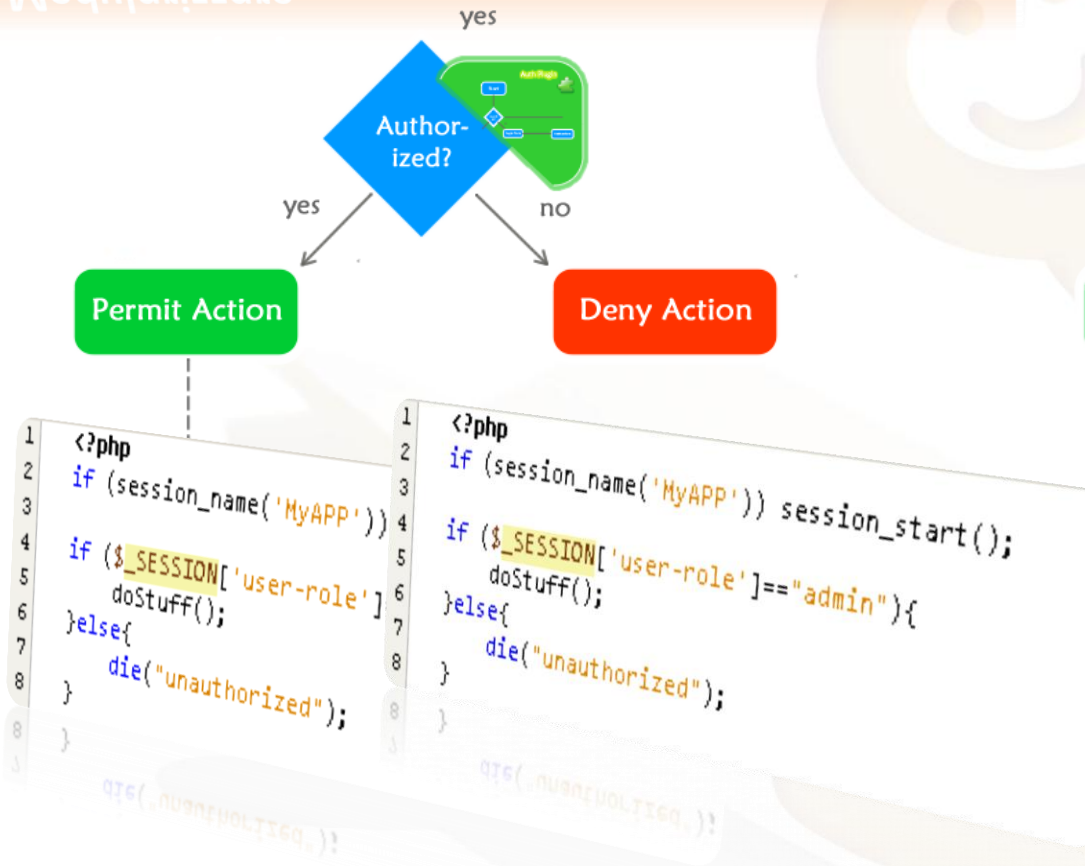
Ovviamente potremmo trovarci di fronte ad applicazioni sviluppate in maniera non modulare, che originariamente prevedevano un solo sistema di autenticazione ed autorizzazione, e che quindi operano con la sessione utente in più punti punti del codice.



Federare un'applicazione

Modularizzare

Anche se ciò non è molto elegante, ne manutenibile (quindi se ne consiglia il *refactoring)*, non rappresenterà un problema per la nostra applicazione da federare, solitamente infatti, questi controlli distribuiti in più parti del codice riguardano solo l'autorizzazione: non l'autenticazione. Quindi potrebbero anche restare invariati



Federare un'applicazione

Punti critici

I punti da individuare con priorità sono invece quelli in cui l'applicazione scrive nella sessione i dati relativi all' identità dell'utente, ed i relativi punti che li invocano (le pagine di login e gli endpoint dei POST di login)

Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Federare un'applicazione

SSO Init (SimpleSAMLphp)

Una volta individuati questi punti vanno eliminate le ridondanze e sostituiti (o affiancati) da chiamate a funzioni di inizializzazione del flusso SSO (lazy session)

Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Federare un'applicazione

SSO Init (Shibboleth)

Una volta individuati questi punti vanno eliminate le ridondanze e sostituiti (o affiancati) da chiamate a funzioni di inizializzazione del flusso SSO (lazy session)

Federare un'applicazione

Session Bridge

Va anche implementata una parte di «ricezione» dell'assertion SAML, che mappi i valori degli attributi con gli oggetti della session preesistente (con SimpleSAML il conflitto di sessione può essere un problema, aggirabile mediante session bridging) ad ogni modo si consiglia il refactoring e l'introduzione di wrappers laddove possibile)

```
if($_SERVER['eduPersonPrincipalName'] =="member@garr.it" ||
        in_array("urn:mace:dir:idem.it:myapp:admin-role",$_SERVER['eduPersonEntitlement'])
    ){
        $SESSION["user-role"]="admin";
    }else{
        $SESSION["user-role"]="user";
    }
```

```
<?php
if (session_name('MyAPP')) session_start();

if ($_SESSION['user-role']=="admin"){
    doStuff();
}else{
    die("unauthorized");
}
```


Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Federare un'applicazione

Session Wrapper

Wrapper che estende un oggetto generico User

```
 * @author Stefano
 */
@DataTransferObject
public class GarrSSOUser extends User{

    public final String GARRSSO_NAME_ATTROID = "urn:oid:2.5.4.42";
    public final String GARRSSO_SN_ATTROID = "urn:oid:2.5.4.4";
    public final String GARRSSO_MAIL_ATTROID = "urn:oid:0.9.2342.19200300.100.1.
    public final String GARRSSO_PROJECTS_ATTROID = "x-garr-ProjectUserMembership
    public final String GARRSSO_SEX_ATTROID = "urn:oid:1.3.6.1.4.1.25178.1.2.2";
    public final String GARRSSO_IDEM_EDUPERSONENTITLEMENT="urn:oid:1.3.6.1.4.1.

    @RemoteProperty
    private String username;
    @RemoteProperty
    private String email;
    @RemoteProperty
    private String name;
    @RemoteProperty
    private String sn;
```

Login Endpoint (Istanzia il wrapper nella sessione locale)

```
<%@page import="dk.itst.oiosaml.sp.UserAttribute"%>
<%@page import="dk.itst.oiosaml.sp.UserAssertion"%>
<%
        UserAssertion ua = dk.itst.oiosaml.sp.UserAssertionHolder.get();

        for (UserAttribute a : ua.getAllAttributes()) {
            Logger.getAnonymousLogger().log(Level.INFO, "Attributi SAML Ricevuti: ", a.getName() +
        }
        Logger.getAnonymousLogger().log(Level.INFO, "Provo a creare una sessione locale ");

        User gu = new GarrSSOUser(ua);
        session.setAttribute("loggedUser", gu);

        Logger.getAnonymousLogger().log(Level.INFO, "Sessione inizializzata per: ", gu.getEmail())
        response.sendRedirect("../index.jsp");

%>
```

```
public GarrSSOUser(UserAssertion ua) {
    this.name = ua.getAttribute(this.GARRSSO_NAME_ATTROID).getValue();
    this.sn = ua.getAttribute(this.GARRSSO_SN_ATTROID).getValue();
    this.username = ua.getAttribute(this.GARRSSO_MAIL_ATTROID).getValue();
```

Costruttore (setta gli attributi nel wrapper, elaborandoli se necessario)

L'applicazione usa sempre la superclasse per l'AuthZ

```
if (user.isProjectEnabled("SCARR")) {
```


Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Federare un'applicazione

Scegliere il livello di trust

E' possibile decidere il grado di fiducia nella fed
scegliere di usare un m
sugli attr
soluzione
automatis
IDEM, ov
autorizzat

si puo`
direttamente
una
utonomia ed
agli attributi

Provisioning the Application's Database or Session

Alternatively, an application can retain its existing authentication handling mechanism, but you can place something alongside the application that checks the variables presented by Shibboleth and transforms them into whatever session mechanism the application expects. This means the user's first destination after Shibboleth authentication is this entry script, and then once the script does its thing, the user is sent further on into the application. The application believes the user is authenticated per usual and remains oblivious to the SSO infrastructure. There are two common ways this is done.

1. Many applications check user identities against databases. A popular integration strategy is to create a script that provisions user information from the SP-supplied variables into the application's database. The user may be presented additional tokens to present to the application in the redirect, such as a one-time username/password, but sometimes implicit information like the IP address is used.
2. An application session can be created directly by creating or binding to the session cookie o whatever else the application relies on.

ID	Name		
1	Stefano	Gargiulo	stefano.gargiu

https://spaces.internet2.edu/display/SHIB2/NativeSPEnableApplication

Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Federare un'applicazione

Federare? Un gioco da ragazzi!

Se avete una di queste applicazioni, potete federarla a costo zero: esistono infatti già decine di applicazioni note per le quali sono disponibili estensioni SAML opensource

moodle

Joomla! ...because open source matters

DSPACE

Google Apps

Sakai

WordPress

DokuWiki

MediaWiki Because ideas want to be free.

Drupal

sympa

https://spaces.internet2.edu/display/SHIB2/ShibEnabled

https://rnd.feide.no/fed_software/

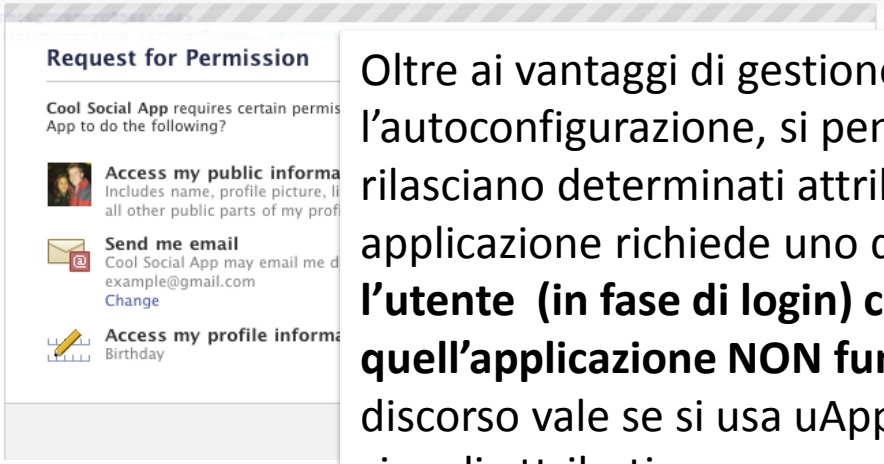
Stefano Gargiulo – GARR (stefano.gargiulo@garr.it)

Requested Attribute

La ciliegina sulla torna

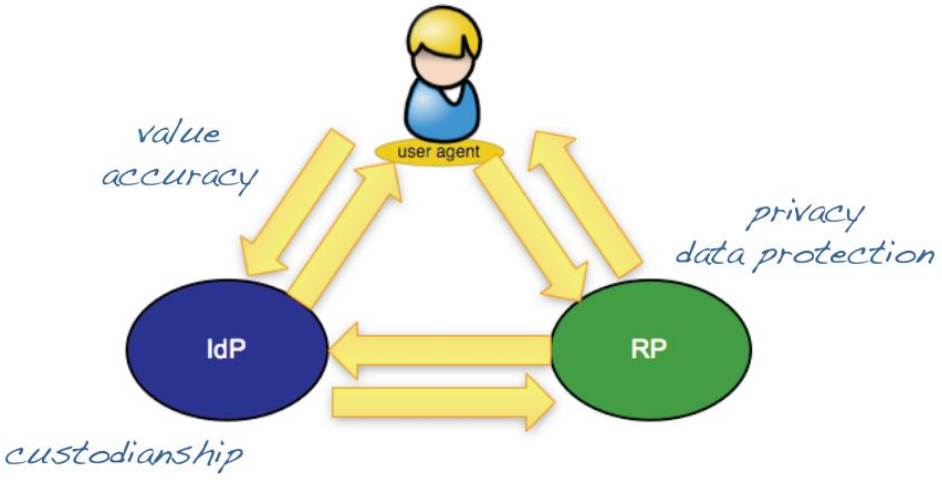
Come consigliato dal wiki ufficiale shibbleth2 è buona norma compilare i metadati dell'SP con gli attributi richiesti dall'applicazione ai fini di documentazione, ma in futuro, (simpleSAMLphp già lo fa) gli IdP si autoconfigurerranno per rilasciare solo gli attributi richiesti ad ogni SP (niente più oneri di gestione dei filtri, che

```
<md:AttributeConsumingService index="1">
  <md:ServiceName xml:lang="en">Sample Service</md:ServiceName>
  <md:ServiceDescription xml:lang="en">An example service that requires a human-readable identifier and optional name and e-mail address.</md:ServiceDescription>
  <md:RequestedAttribute FriendlyName="eduPersonPrincipalName" Name="urn:mace:dir:attribute-def:eduPersonPrincipalName" NameFormat="urn:mace:shibboleth:1.0:attributeNamespace:uri" isRequired="true"/>
  <md:RequestedAttribute FriendlyName="mail" Name="urn:mace:dir:attribute-def:mail" NameFormat="urn:mace:shibboleth:1.0:attributeNamespace:uri"/>
  <md:RequestedAttribute FriendlyName="displayName" Name="urn:mace:dir:attribute-def:displayName" NameFormat="urn:mace:shibboleth:1.0:attributeNamespace:uri"/>
</md:AttributeConsumingService>
```



Oltre ai vantaggi di gestione per l'autoconfigurazione, si pensi agli IdP che non rilasciano determinati attributi opzionali, se un applicazione richiede uno di essi, **come informiamo l'utente (in fase di login) che con il nostro IdP quell'applicazione NON funzionerà mai???** Stesso discorso vale se si usa uApprove con checkbox sui singoli attributi..

Multilateral trust:
who is willing to promise what?



Sviluppi futuri

Benefici del SSO Federato

Collaborazione internazionale

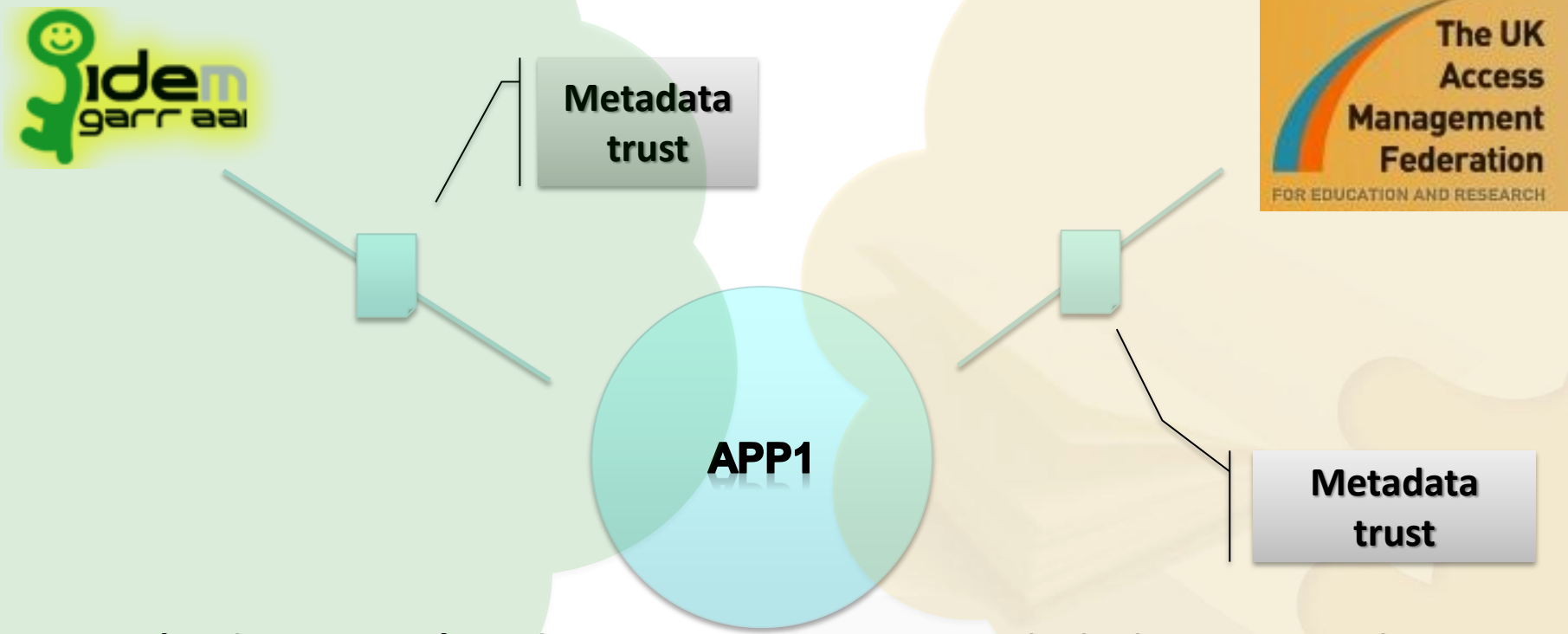
Bacini di utenza in espansione, Collaborazione sui progetti resa facile!

Some methods for facilitating interoperability between federations:

- Federation Bridging
- Confederations: federation of federations, highly hierarchical control
- Peering: bi-/multi-lateral agreements
- Virtual Organizations (VOs): not under single hierarchical control

Collaborazione internazionale

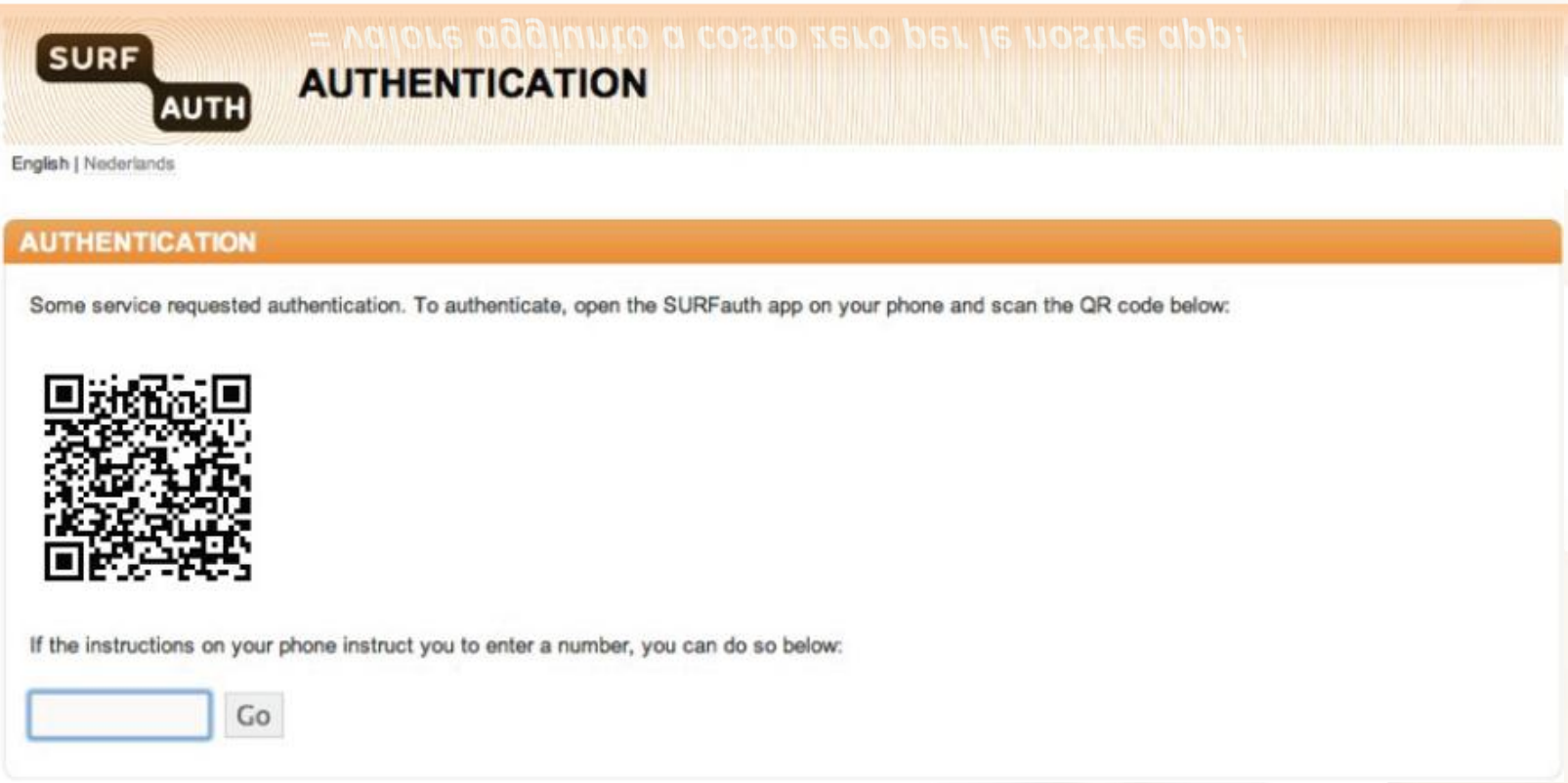
Utenti a portata di metadati



L'applicazione può anche instaurare trust con singoli IdP di enti o Virtual Organizations esterne (anche solo in qualità di Attribute Authorities) – si pensi a progetti comuni tra più università, enti ed aziende

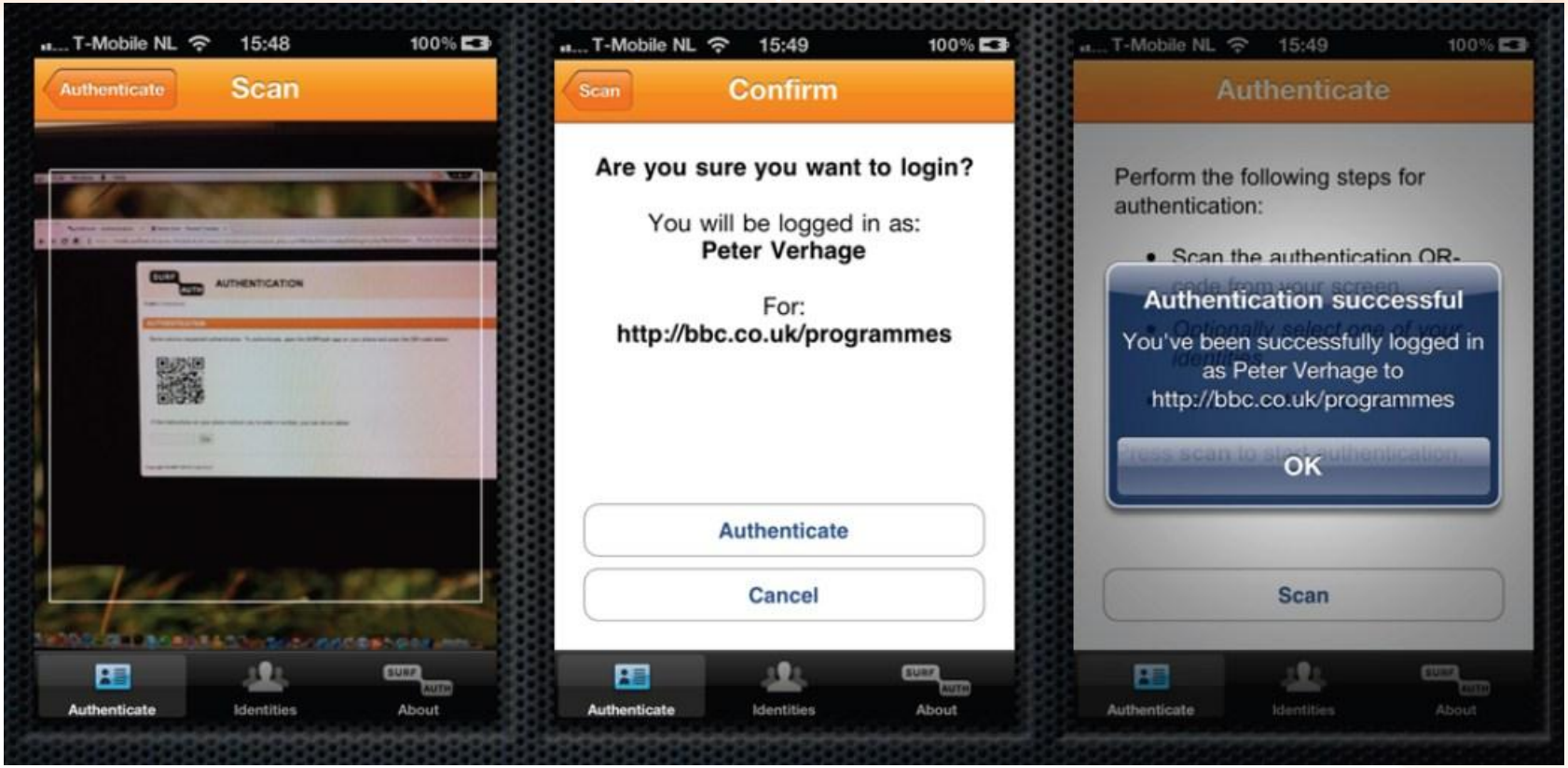
Sviluppi futuri lato IdP

= valore aggiunto a costo zero per le nostre app!



Sviluppi futuri lato IdP

= valore aggiunto a costo zero per le nostre app!



Ringraziamenti

- Stefano Zanmarchi (Universita` degli Studi di Padova)
  - Slides introduttive sul concetto di identita` digitale
    - 4 , 6…10 e 83
    - Leggermente rielaborate e tradotte dalla sua versione in inglese presentata all'EUMEDCONNECT AAI INFODAY

Q&A

Any (Attribute) Query?

Q&A

Any (Attribute) Query?