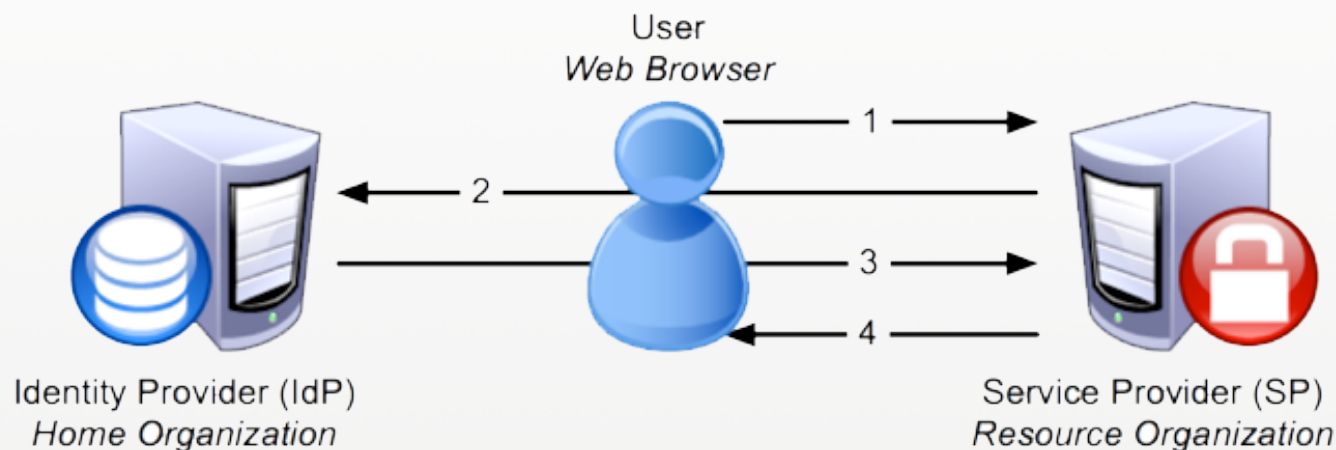


# Agenda

- Come è fatta la sessione utente: gli **attributi!**
- Come leggere i dati di sessione
  - Esempio di **codice PHP**
  - Esempio di **codice Python**
  - Esempio di **codice Java** su Apache+Tomcat
- La sessione "passiva": **lazy session**

# Una login federata, cosa succede?



1. L'utente richiede la risorsa federata.
2. L'utente è rediretto sulla pagina di login dell'IdP.
3. **L'IdP scambia con l'SP gli attributi dell'utente autenticato (nome, cognome, email, ...)**
4. L'IdP redirige l'utente autenticato sulla risorsa.

# La sessione utente, come è fatta?

- Una sessione utente Shibboleth, in Apache HTTPd, consiste di elementi distinti:
  - Il **REMOTE\_USER**, variabile di ambiente speciale di Apache che contiene lo username dell'utente autenticato.
  - Un insieme di attributi utente, inseriti in **variabili di ambiente** di Apache.

# Quali attributi?

- Gli attributi utente sono descritti tecnicamente all'interno degli **schemi LDAP**, quelli più comuni sono:
  - LDAPv3 rfc4519
  - inetorgPerson
  - Schac
  - eduPaerson
- La federazione IDEM rilascia delle linee guida riguardo agli attributi utilizzabili dagli IdP e SP della federazione:
  - **Caratteristiche personali** (*nome, cognome, titolo, ...*)
  - **Contatti** (*email, telefono, organizzazione, unità, ...*)
  - **Autorizzazioni ed accounting** (*affiliazione, entitlement, ...*)

<https://www.idem.garr.it/informazioni-tecniche/attributi>

# Esempio, nome e cognome!

- Alcune **informazioni** possono essere **contenute in diversi attributi**: prendiamo l'esempio del nome e cognome.
- Queste informazioni possono essere contenute in:
  - **displayName**
  - **commonName**
  - **givenName** e **sn**
- Quando si devono usare questi attributi, quindi, è buona prassi "**ricercare**" **le informazioni necessarie nei vari attributi** che potrebbero contenerle.



# Configurazione del REMOTE\_USER

- Sull'SP nel file **/etc/shibboleth/shibboleth2.xml** vengono specificati quali attributi (della sessione Shibboleth) devono essere usati per valorizzare il REMOTE\_USER. Il primo attributo trovato nella sessione Shibboleth viene usato a questo scopo.

```
<ApplicationDefaults entityID="https://sp1.local/shibboleth"  
    REMOTE_USER="eppn persistent-id targeted-id">
```

- Il REMOTE\_USER non è quindi altro che un attributo di sessione Shibboleth un po' speciale.

# Configurazione degli attributi

- Sull'SP nel file **/etc/shibboleth/attribute-map.xml** decommentare la parte:

```
<!-- Some more eduPerson attributes, uncomment these to use them... -->  
<Attribute name="urn:mace:dir:attribute-def:eduPersonPrimaryAffiliation"  
  id="primary-affiliation">    <AttributeDecoder xsi:type="StringAttributeDecoder"  
  
  caseSensitive="false"/>  
</Attribute>  
  
<Attribute ...  
<Attribute ...  
<Attribute ...
```

# Leggere gli attributi: PHP

- Le variabili di ambiente di Apache vengono lette da PHP dal dizionario **`$_SERVER`**:

```
<?php
function getName() {
    if (array_key_exists("displayName", $_SERVER)) {
        return implode(" ", explode(";", $_SERVER["displayName"]));
    } else if (array_key_exists("cn", $_SERVER)) {
        return implode(" ", explode(";", $_SERVER["cn"]));
    } else if (array_key_exists("givenName", $_SERVER) && array_key_exists("sn", $_SERVER)) {
        return implode(" ", explode(";", $_SERVER["givenName"])) . " " .
            implode(" ", explode(";", $_SERVER["sn"]));
    }
    return "Unknown";
}

$username = $_SERVER["REMOTE_USER"];
$name = getName();
print "<h1>Ciao " . $username . "!!!</h1>";
print "<p>Il tuo nome &grave; " . $name . ".</p>";
?>
```



# Leggere gli attributi: Python

- Le variabili di ambiente di Apache vengono lette da Python dal dizionario **os.environ**:

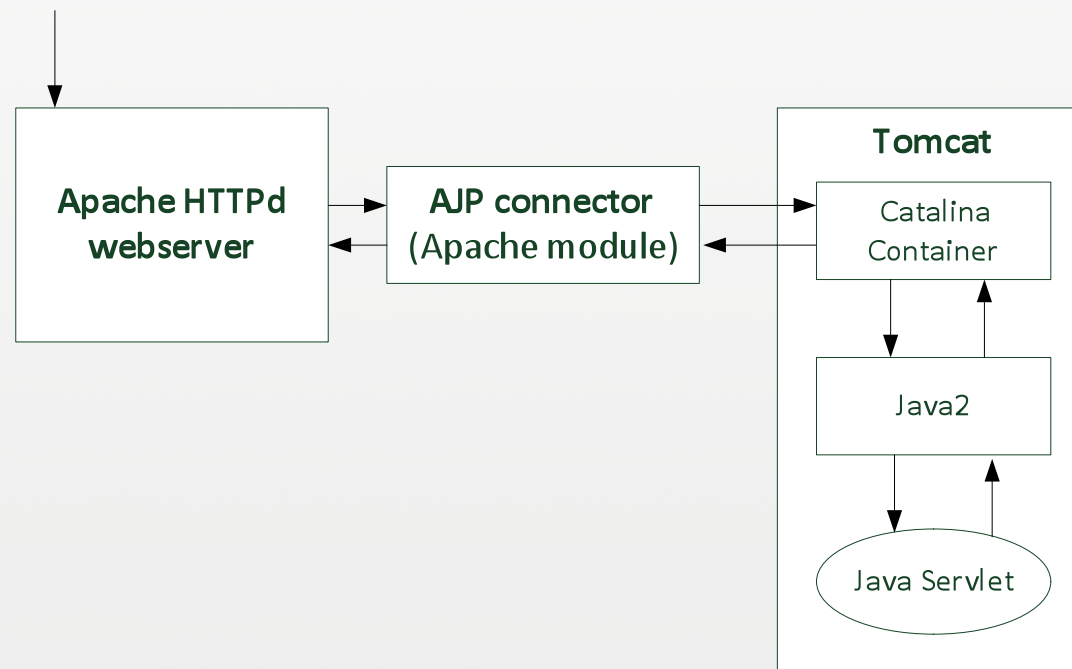
```
#!/usr/bin/python2.7
import cgi
from os import environ
cgi.enable()

def get_name():
    if "displayName" in environ: return " ".join(environ["displayName"].split(";"))
    elif "cn" in environ: return " ".join(environ["cn"].split(";"))
    elif "givenName" in environ and "sn" in environ:
        return " ".join(environ["givenName"].split(";")) + " " + \
            " ".join(environ["sn"].split(";"))
    return "Unknown"

print "Content-Type: text/html;charset=utf-8\n"
username = environ.get("REMOTE_USER", None)
name = get_name()
print "<h1>Ciao %s!!!</h1>" % username
print "<p>Il tuo nome &grave; %s.</p>" % name
```

# Leggere gli attributi: Java, schema

- La configurazione preferibile per applicazioni Java prevede di utilizzare Apache HTTPd davanti a Tomcat e far dialogare i due tramite **connettore AJP**.



# Leggere gli attributi: Java, conf. 1

- Il connettore AJP passa automaticamente le **variabili di ambiente** di Apache HTTPd a Tomcat, ma passa SOLO le variabili che hanno il **prefisso AJP\_**.

- Nel file /etc/shibboleth/shibboleth.xml

```
<ApplicationDefaults  
  entityID="https://sp1.local/shibboleth"  
  REMOTE_USER="eppn persistent-id targeted-id"  
  attributePrefix="AJP_">
```

# Leggere gli attributi: Java, conf. 2

- Il connettore AJP passa automaticamente il REMOTE\_USER, ma perché tomcat lo legga è necessario **disabilitare l'autenticazione di Tomcat**.

- Nel file /etc/tomcat7/server.xml:

```
<Connector port="8009"  
    protocol="AJP/1.3"  
    redirectPort="8443"  
    tomcatAuthentication="false" />
```

# Leggere gli attributi: Java, codice

- A questo punto in una servlet è possibile leggere le variabili dall'**oggetto request** come segue:

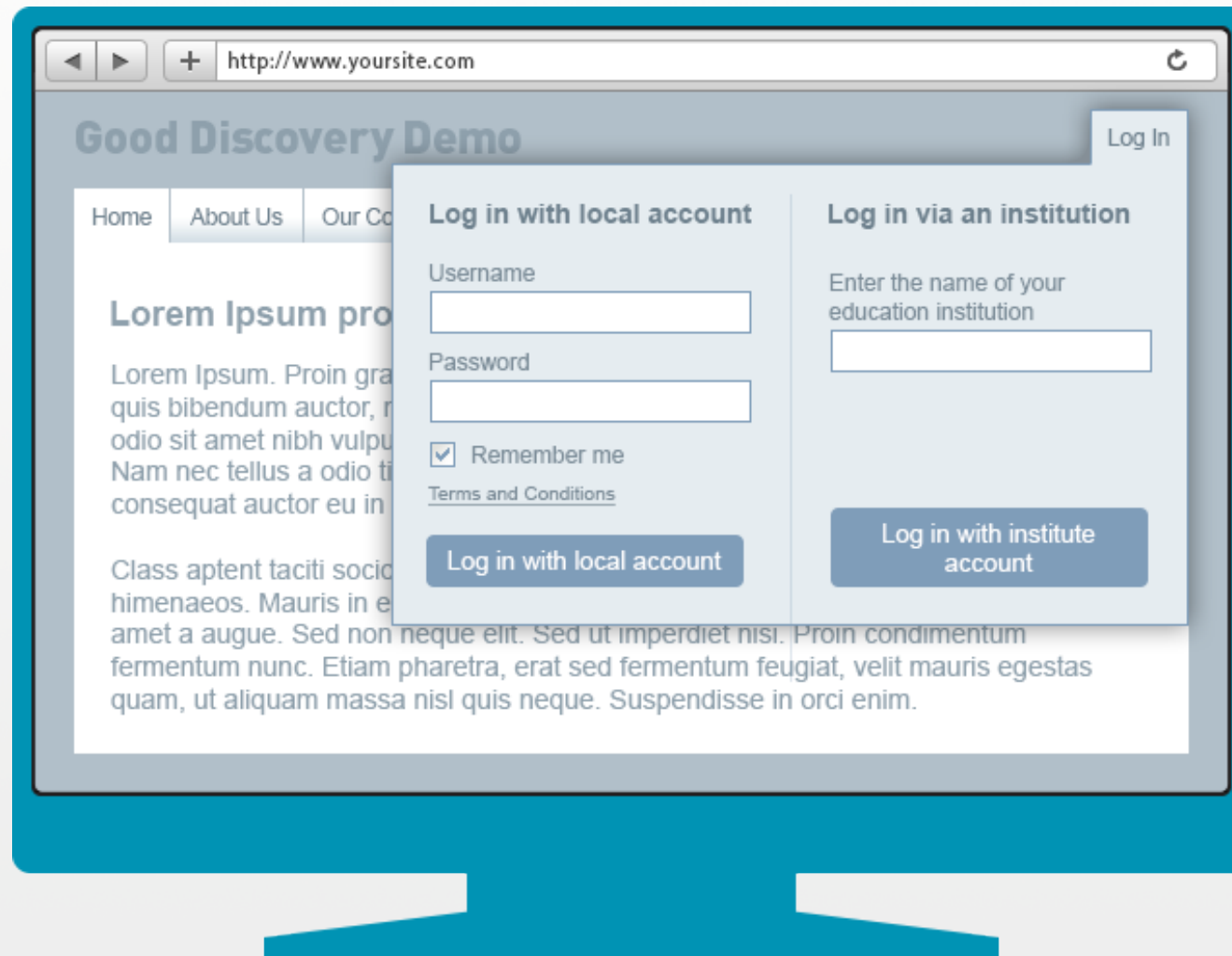
```
private String getName(HttpServletRequest request) {  
    if (request.getAttribute("givenName") != null) {  
        return (String) request.getAttribute("givenName");  
    } else if (request.getAttribute("cn") != null) {  
        return (String) request.getAttribute("cn");  
    } else if (request.getAttribute("givenName") != null && request.getAttribute("sn") != null) {  
        return (String) request.getAttribute("givenName") + " " + (String) request.getAttribute("sn");  
    }  
    return "Unknown";  
}  
  
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws  
    ServletException,  
    IOException {  
    PrintWriter pw = response.getWriter();  
    String username = request.getRemoteUser();  
    String name = getName(request);  
    pw.println("<h1>Ciao " + username + "!!!!</h1>");  
    pw.println("<p>Il tuo nome &egrave " + name + "</p>");  
}
```



# La sessione "passiva": lazy session

- Per alcune applicazioni potrebbe avere senso offrire **qualche funzionalità ad utenti anonimi** e permettere agli **utenti autenticati** di effettuare **operazioni aggiuntive**.
- Altre applicazioni potrebbero essere accessibili **sia tramite utenza federata che tramite utenza locale** (o altri tipi di login, federati o non).
- In questi casi Shibboleth offre una funzionalità chiama "lazy session".

# Lazy session, un esempio!



The screenshot shows a web browser window with the address bar displaying `http://www.yoursite.com`. The page title is "Good Discovery Demo". A "Log In" button is visible in the top right corner. The main content area features a navigation menu with "Home", "About Us", and "Our Co". Below the menu, there is a section titled "Lorem Ipsum pro" followed by a paragraph of Lorem Ipsum text. A modal login form is overlaid on the page, containing two columns:

- Log in with local account:** Includes fields for "Username" and "Password", a "Remember me" checkbox, a link to "Terms and Conditions", and a "Log in with local account" button.
- Log in via an institution:** Includes a field for "Enter the name of your education institution" and a "Log in with institute account" button.

# Lazy session: Apache configuration

- La configurazione della Location in Apache deve quindi diventare come segue.

- In `/etc/apache2/sites-enabled/service_provider.conf`

```
<Location /lazy.php>  
    AuthType shibboleth  
    ShibRequestSetting requireSession false  
    Require shibboleth  
</Location>
```

# Lazy session, una pagina di esempio

- A questo punto possiamo creare una pagina così fatta (ad esempio la pagina /lazy.php in PHP):

```
<?php
$l = "/Shibboleth.sso/Login?target=/lazy.php";
$username = $_SERVER["REMOTE_USER"];

if (!isset($username) || empty($username)) {
    print "<p>Utente anonimo ";
    print "<a href=\"".$l."\">Login</a></p>";
}
else {
    print "<p>Utente autenticato: ".$username."</p>";
}
?>
```

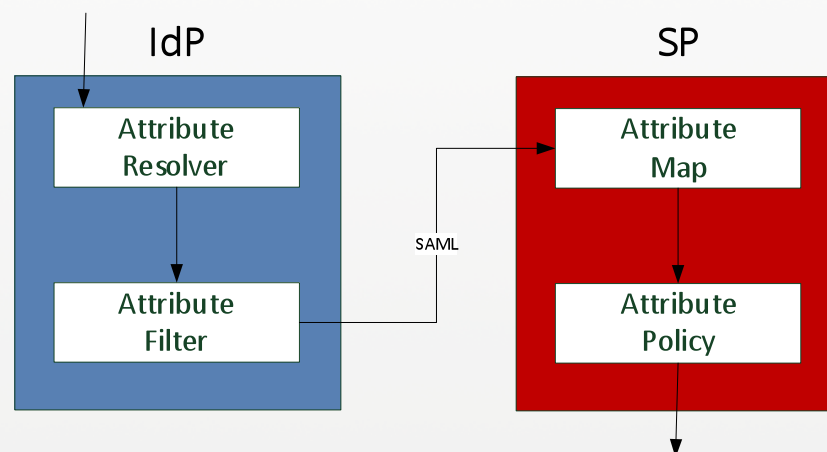
THANK YOU

GRACIAS ARIGATO SHUKURIA GOZAIMASHITA EFCHARISTO JUSPAXAR DANKSCHEEN TASHAKKUR ATU YAQHANYELAY SUKSAMA EKHMET BOLZIN MERICI BIYAN SHUKRIA TINGKI MAAKE GRAZIE MEHRBANI PADDIES KOMAPSUMANDA HEKSYENY SAKCE SALUTTO AGUYE FARKALU MAGAL



# Configurare gli attributi di sessione

- Gli attributi di sessione Shibboleth devono essere **configurati** sia lato **IdP** che lato **SP**.



- In entrambi i casi si hanno due fasi:
  - Mappatura e recupero** dei valori degli attributi di sessione (attribute-resolver e attribute-map).
  - Filtro ed eliminazione** degli attributi non conformi o indesiderati (attribute-filter e attribute-policy).

# Esempio configurazione, IdP

- /opt/shibboleth-idp/conf/attribute-resolver.xml

```
<resolver:AttributeDefinition id="givenName"
  xsi:type="ad:Simple" sourceAttributeID="givenName">
  <resolver:Dependency ref="myLDAP" />
  ...
  <resolver:AttributeEncoder xsi:type="enc:SAML1String"
    name="urn:mace:dir:attribute-def:givenName" />
  <resolver:AttributeEncoder xsi:type="enc:SAML2String"
    name="urn:oid:2.5.4.42" friendlyName="givenName" />
</resolver:AttributeDefinition>
```

- /opt/shibboleth-idp/conf/attribute-filter.xml

```
<afp:AttributeRule attributeID="givenName">
  <afp:PermitValueRule xsi:type="basic:ANY" />
</afp:AttributeRule>
```

# Esempio configurazione, SP

- /etc/shibboleth/attribute-map.xml

```
<Attribute name="urn:mace:dir:attribute-def:givenName" id="givenName"/>  
<Attribute name="urn:oid:2.5.4.42" id="givenName"/>
```

- /etc/shibboleth/attribute-policy.xml

```
<afp:AttributeRule attributeID="givenName">  
  <afp:PermitValueRule xsi:type="ANY"/>  
</afp:AttributeRule>
```