idem day
24-25 Novembre 2009
Roma, Sede centrale ENEA

L'Identity Provider: il nostro primo attore

Raffaele Conte, CNR - IFC • Barbara Monticini, GARR

Cos'è una Federazione per l'AA

- È un insieme di regole tecniche e procedure condivise su cui si costruiscono relazioni di fiducia
- Regole di "convivenza" sono necessarie per ottenere l'interoperabilità fra i partecipanti
- Quali regole? Protocolli, applicativi, sintassi e semantica delle informazioni scambiate ecc.

Agenda

- Shibboleth
- Logging
- Autenticazione
- Metadati
- Aderire alla Federazione IDEM
- Attributi: risoluzione e filtraggio
- uApprove

Applicativi

- Shibboleth 1.3 deprecato (Internet2 non aggiungerà più nuove funzionalità e non lo supporterà più da giugno 2010)
- Shibboleth 2.x è indicato (e supportato) per tutte le nuove installazioni

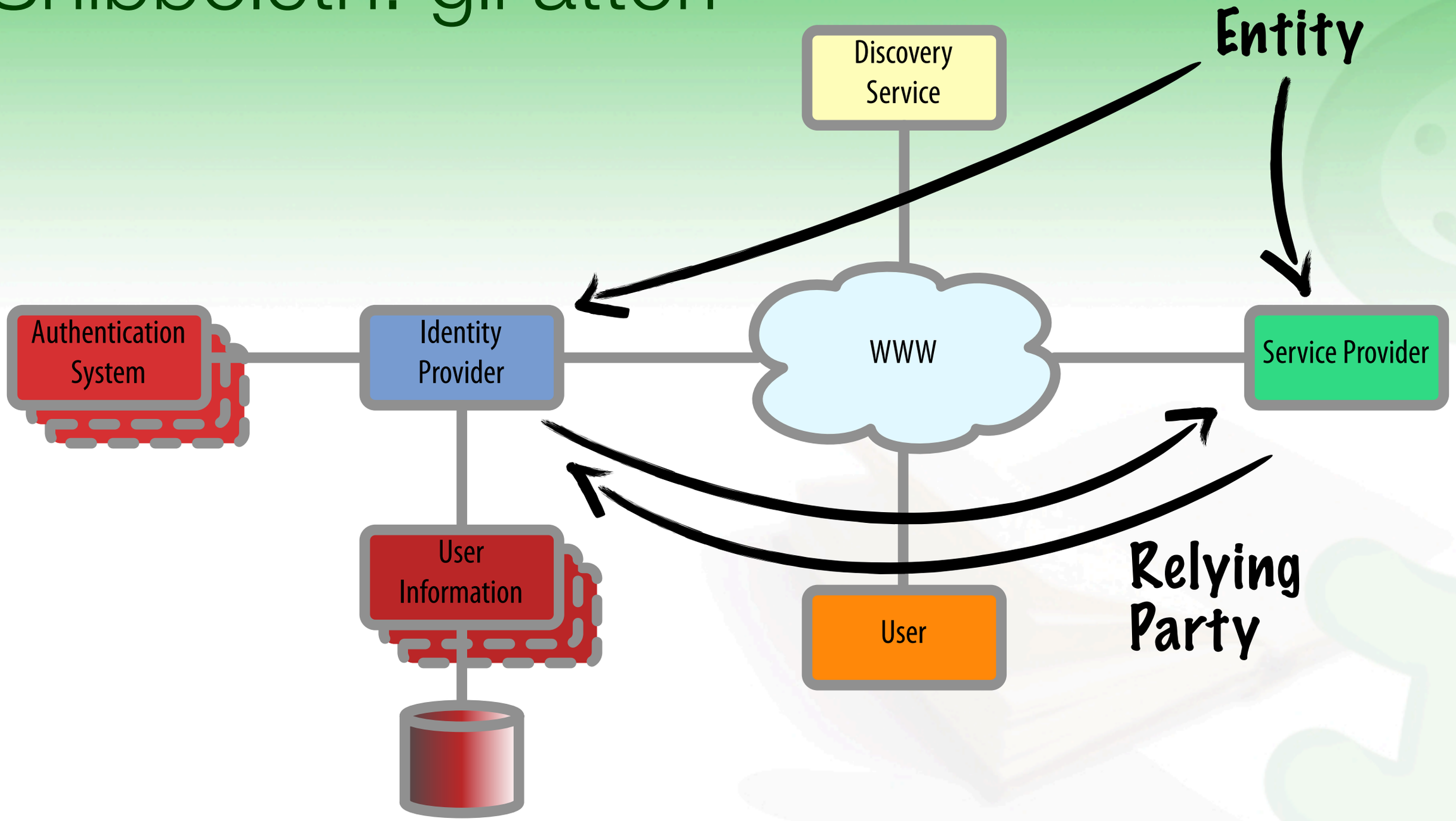
(e possibilmente anche per le vecchie!! ;-)

+ semplice da installare e configurare
+ informazioni nei log
migliore gestione metadati
IdP Tomcat-only
...

Shibboleth

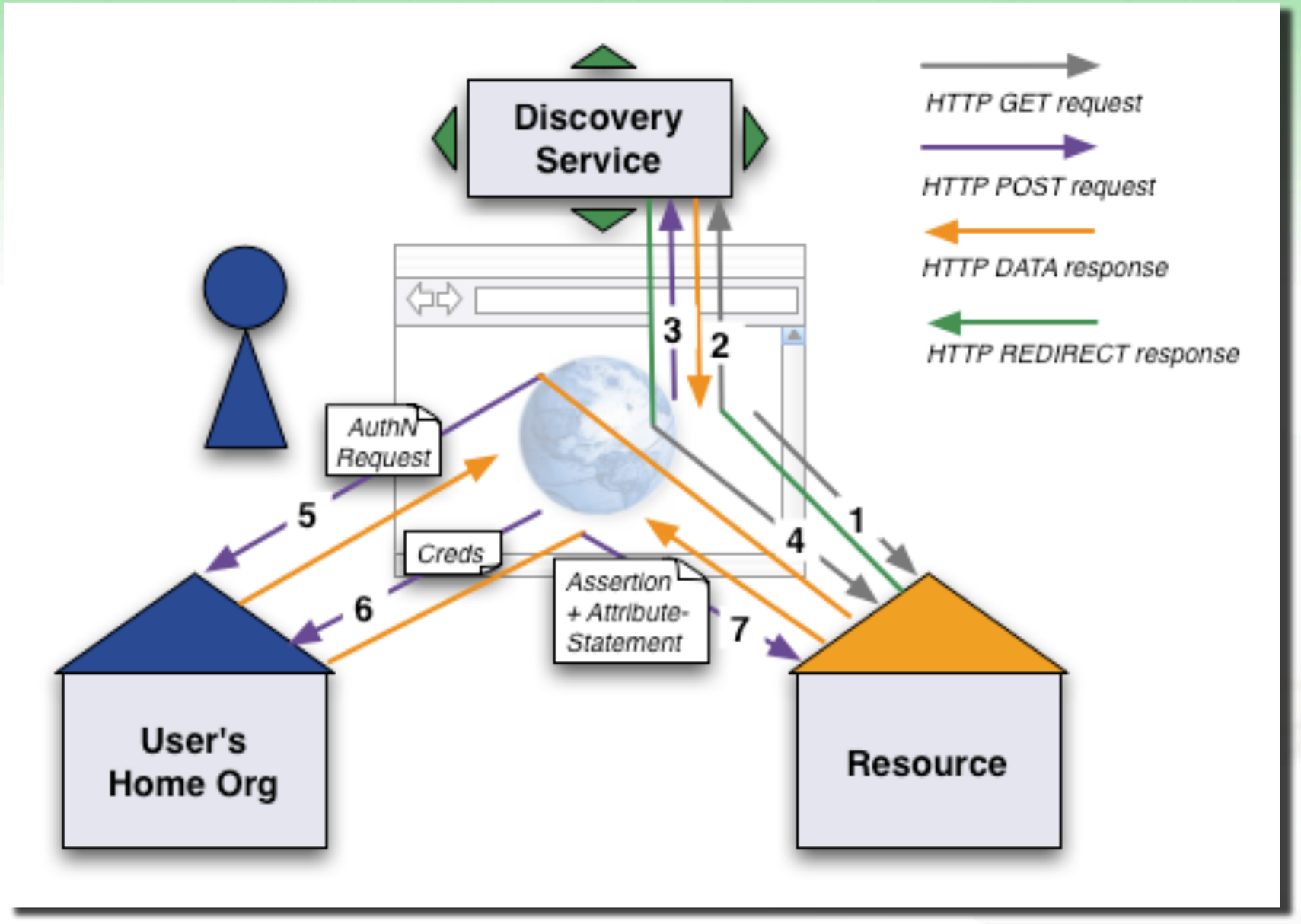
- "*The Shibboleth System is a standards based, open source software package for web single sign-on across or within organizational boundaries*" *(shibboleth.internet2.edu)*.
- Implementazione di SAML
- Progetto ufficiale di Internet2
- Rilasciato con Apache Software License

Shibboleth: gli attori

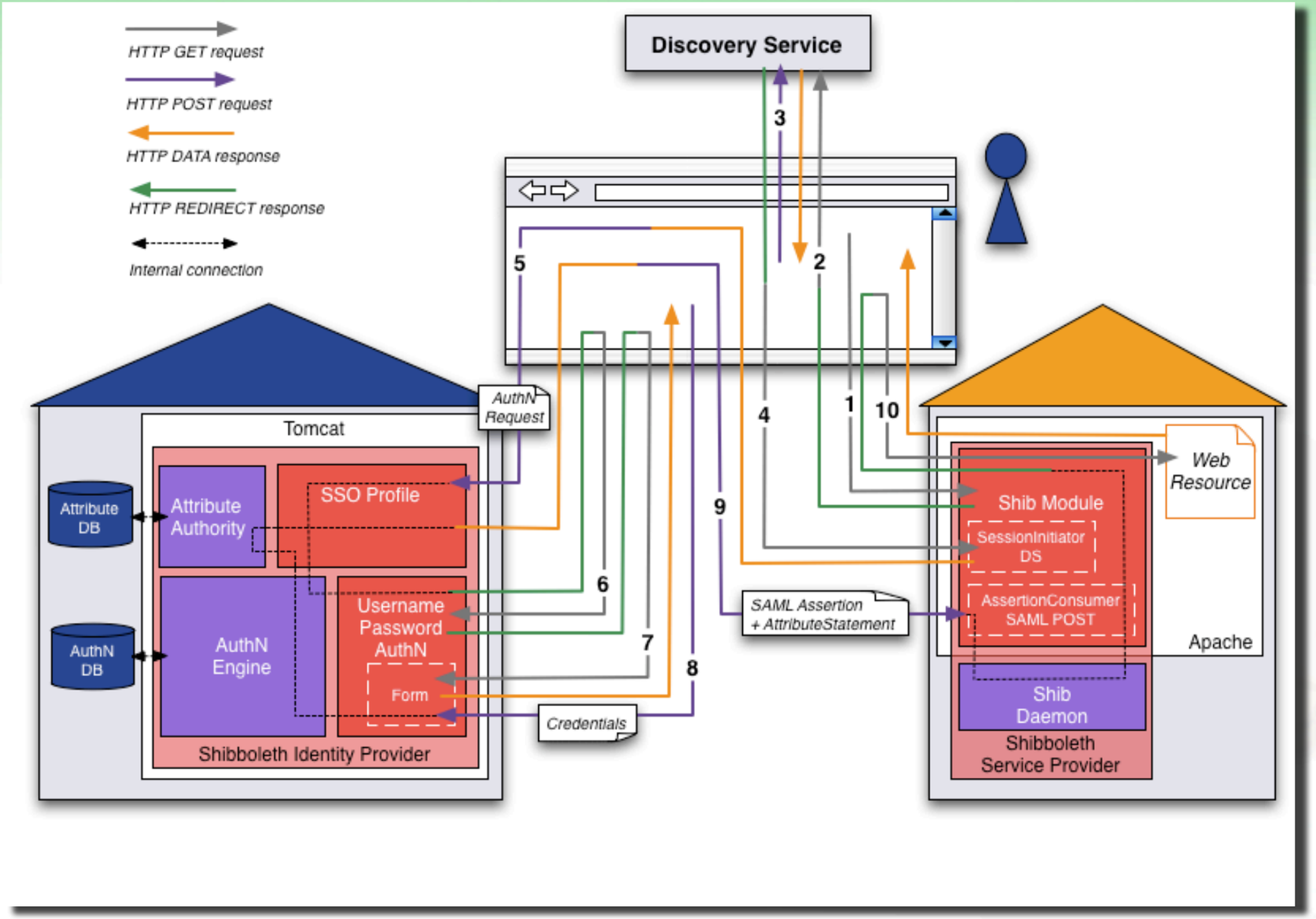


© R. Conte

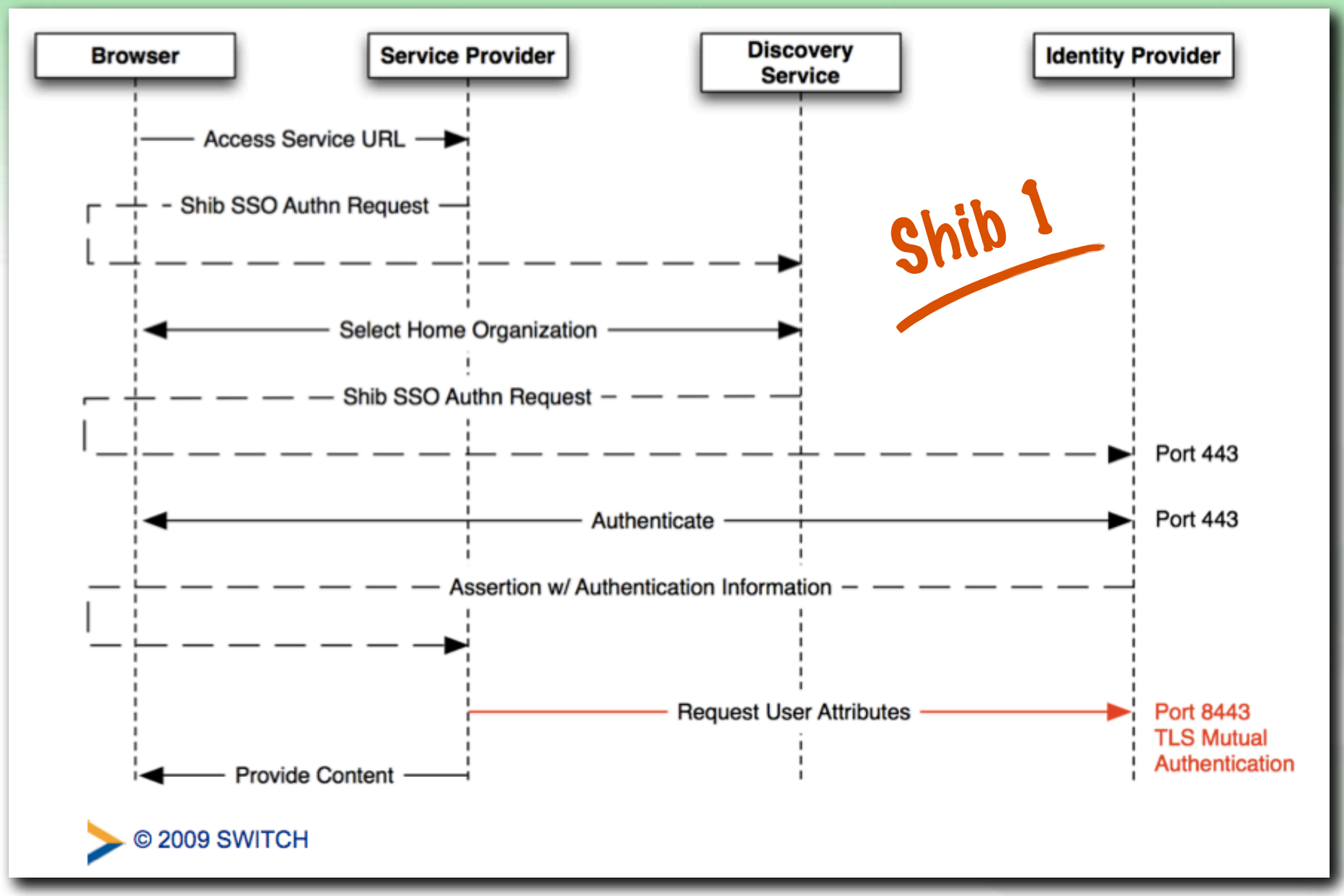
Le relazioni nella Federazione



Le relazioni nella Federazione

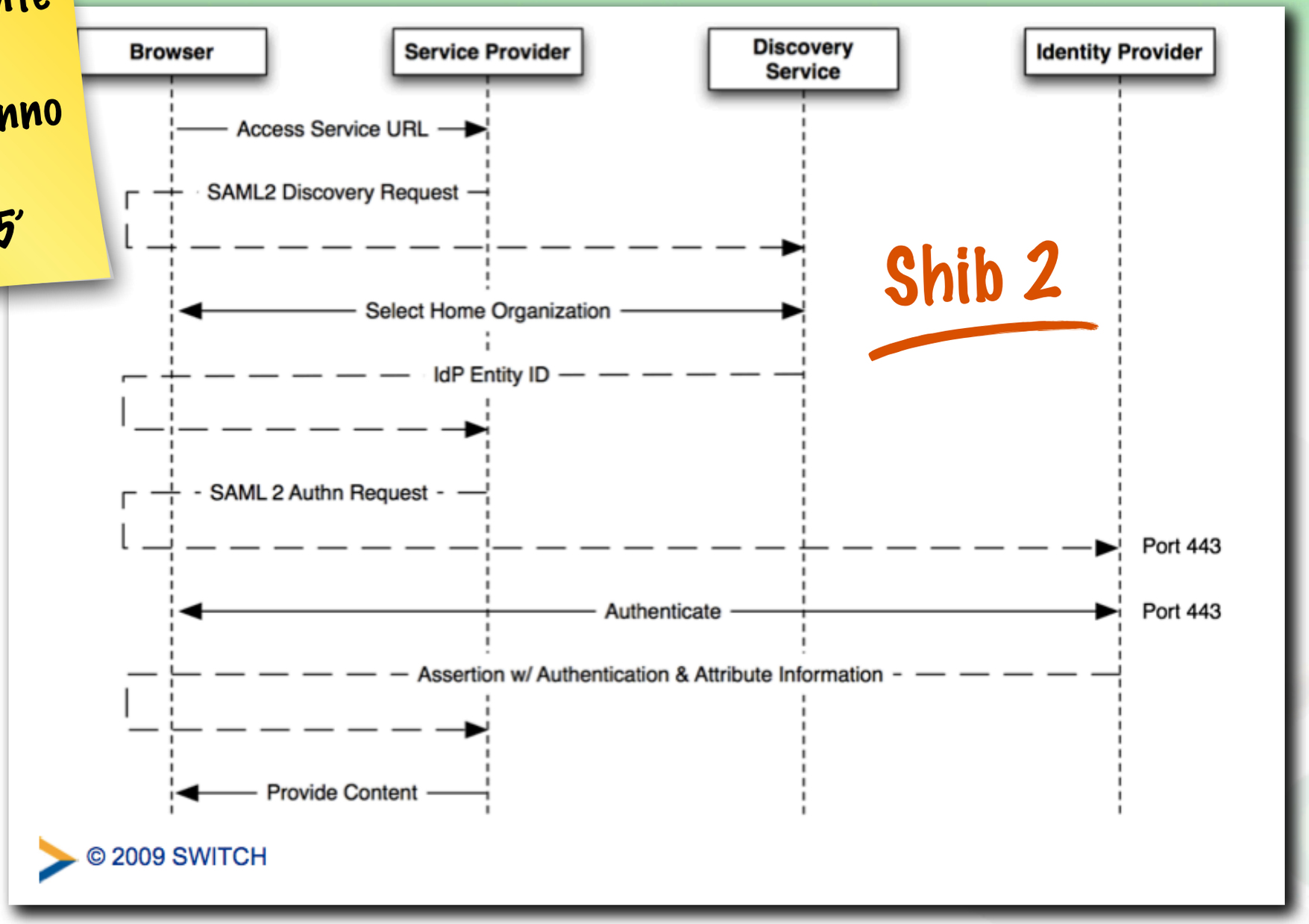


Shibboleth Communication Flow



Shibboleth Communication Flow

Nota: è fortemente consigliato ntp le asserzioni hanno una validità tipicamente di 5'



Organizzazione dei file

- SHIB_HOME contiene:
  `./bin`

Contiene dei command line tools
aacli: Attribute authority command line interface: permette di simulare un attribute query/release
version: Fornisce la versione dell'IdP

Organizzazione dei file

- SHIB_HOME contiene:
  - `./bin`
  - `./conf`

File di configurazione dell'IdP. Molti dei quali verranno analizzati oggi.

Organizzazione dei file

- SHIB_HOME contiene:
  - `./bin`
  - `./conf`
  - `./credentials`

Le credenziali usate dall'IdP. Shibboleth genera di default la chiave (idp.key), il sertificato (idp.crt) e un keystore (idp.jks) contenenti entrambe.

Organizzazione dei file

- SHIB_HOME contiene:

```
./bin
./conf
./credentials
./lib
```

Le librerie (jars) che implementano l'IdP.
Sono copie di quelle presenti nei file WAR dell'IdP e sono utilizzate solo dai command line tools.

Organizzazione dei file

- SHIB_HOME contiene:
  - `./bin`
  - `./conf`
  - `./credentials`
  - `./lib`
  - `./logs`

Contiene i log file di Shibboleth:
process log: descrizione dettagliata delle IdP processing requests
access log: registrazione dei client che accedono all'IdP
audit log: registrazione di tutte le informazioni mandate fuori dall'IdP

Organizzazione dei file

- SHIB_HOME contiene:

```
./bin
./conf
./credentials
./lib
./logs
./metadata
```

La posizione di default dove tenere i file dei metadati reperiti con diverse modalità e caricati nell'IdP.

Organizzazione dei file

- SHIB_HOME contiene:

```
./bin
./conf
./credentials
./lib
./logs
./metadata
./war
```

WAR file creati dall'installer. Si fa puntare Tomcat a questi file, piuttosto che copiarli in Tomcat per evitare di dover ripetere l'operazione in caso di rebuild dell'IdP.

Logging

Logging

- File di configurazione dei log: `logging.xml`
  - Basato sul sistema Logback
  - Non riavviare il server dopo le modifiche

- 5 livelli possibili per i logger definiti: TRACE, DEBUG, INFO, WARN, ERROR
  - TRACE, DEBUG non usarli in produzione!

- https://spaces.internet2.edu/display/SHIB2/IdPLogging

Logging

- Cartella dei log: `$IDP_HOME/logs`:
  - idp-process.log da consultare sempre in caso di errori o problemi
- Logger utili:
  - edu.internet2.middleware.shibboleth
  - edu.internet2.middleware.shibboleth.common.attribute [solo in test]
  - org.opensaml
  - edu.vt.middleware.ldap
  - PROTOCOL_MESSAGE

Autenticazione

Autenticazione

- Shibboleth 2 offre **meccanismi** basati su REMOTE_USER, username/password su LDAP / Kerberos e indirizzo IP
- Ogni meccanismo è gestito da un **Login Handler**
- Un IdP supporta l'uso di più **metodi di autenticazione** contemporaneamente
- https://spaces.internet2.edu/display/SHIB2/IdPUserAuthn

username/password - LDAP

- Basato su **JAAS** - Java Authentication and Authorization Service
- Definire il login handler: **`conf/handler.xml`**

username/password - LDAP

```
<LoginHandler xsi:type="UsernamePassword"
  jaasConfigurationLocation="file:///opt/shibboleth-idp/conf/
  login.config">
  <AuthenticationMethod>
    urn:oasis:names:tc:SAML 2.0:ac:classes
    :PasswordProtectedTransport
  </AuthenticationMethod>
</LoginHandler>
```

username/password - LDAP

- Modulo di login LDAP: **`conf/login.config`**
  - edu.vt.middleware.ldap.jaas.LdapLoginModule
  - Campi accettati: host, base, port, serviceUser ...
- Configurazioni possibili
  - 1 o più ldap nel campo Host
  - Stacking Login Modules
  - Configurazione failover
- https://spaces.internet2.edu/display/SHIB2/IdPAuthUserPass

username/password - LDAP

```
ShibUserPassAuth {
   edu.vt.middleware.ldap.jaas.LdapLoginModule
required
      host="ldap://example.org:636"
      base="dc=example,dc=org"
      serviceCredential="password"
      serviceUser="cn=myUser,dc=example,dc=org"
      ssl="true"
      userField="uid"
      subtreeSearch="true";
};
```

Pagina per Login

- `src/main/webapp/login.jsp`
- Deve essere personalizzata tranne:
  - j_username, j_password (input)
  - /Authn/UserPassword (action form)
- Per il deploy:
  - Rilanciare l'installazione dell'IDP

```
The directory '/opt/shibboleth-idp' already
exists.  Would you like to overwrite this
Shibboleth configuration? (yes, [no]) NO
```

Metadati

Metadati: contenuto

- Certificati
- Scope degli IdP *(es. ifc.cnr.it)*
- Posizione (url) e tipologia dei componenti per lo scambio e l'utilizzo delle *assertion* dei partecipanti
- Eventuale descrizione testuale dei partecipanti

Metadati: contenuto

lo scope deve corrispondere a quello utilizzato per gli attributi

- Certificati
- Scope degli IdP *(es. ifc.cnr.it)*
- Posizione (url) e tipologia dei componenti per lo scambio e l'utilizzo delle *assertion* dei partecipanti
- Eventuale descrizione testuale dei partecipanti

Metadati: i certificati

- È consentito l'utilizzo di certificati self-signed per la comunicazione SP-IdP (back-channel)
  - Il ruolo di Garante, affidato a una CA in una PKI, qui è svolto dalla Federazione
  - Equivale ad inserire la chiave pubblica, quindi minore tempo di verifica della controparte
  - Può essere rigenerato velocemente, quindi minore tempo di downtime in caso di compromissione del certificato

Aderire alla Federazione

Aderire alla Federazione

- Inviare il frammento dei metadati dell'idp
  - idp-metadata.xml
- Scaricare i metadati di idem aggiornati
  - signed-metadata.xml
- Scaricare il certificato con cui verificare la firma dei metadati
  - signer-bundle.pem

Relying-party.xml

- Metadata Configuration
  - MetadataProvider per IDEM
    - File Backed HTTP Metadata Provider
    - Id, MetadataURL, backingFile
    - Metadata Filter per verificare la firma

Relying-party.xml

```
<!-- *** IDEM *** -->
<MetadataProvider id="URLMD-idem"
    xsi:type="FileBackedHTTPMetadataProvider"
    xmlns="urn:mace:shibboleth:2.0:metadata”
    metadataURL="https://www.idem.garr.it/docs/conf/signed-
    metadata.xml"
    backingFile="/opt/shibboleth-idp/metadata/signed-
    metadata.xml">

    <MetadataFilter xsi:type="ChainingFilter"
        xmlns="urn:mace:shibboleth:2.0:metadata">
    <MetadataFilter xsi:type="SignatureValidation"
        xmlns="urn:mace:shibboleth:2.0:metadata"
        trustEngineRef="shibboleth.MetadataTrustEngine"
        requireSignedMetadata="true" />
    </MetadataFilter>
</MetadataProvider>
```

Relying-party.xml

- Security Configuration
  - **`security:TrustEngine`** (usato in Metadata Filter)

```
<security:TrustEngine id="shibboleth.MetadataTrustEngine"
    xsi:type="security:StaticExplicitKeySignature">

    <security:Credential id="IDEMCredentials"
        xsi:type="security:X509Filesystem">
        <security:Certificate>
            /opt/shibboleth-idp/credentials/signer-bundle.pem
        </security:Certificate>
    </security:Credential>
</security:TrustEngine>
```

Metadati: gestione

- È necessario aggiornare i metadati al più ogni 24h
- È necessario comunicare il proprio frammento con messaggio firmato
- Il file è scaricabile solo con HTTPS, è fortemente consigliata la verifica della firma
- È consigliabile mantenere il file con diritti tali da non consentirne la modifica

Attributi

Attributi

- Attributo: una “piccola” informazione riguardo un utente. Ogni attributo ha un unico ID ed ha zero o più valori.
- Attributi Shibboleth: strutture dati “*protocol-agnostic*”.
- Attributi SAML: attributi rappresentati tramite notazione SAML.
- Shibboleth trasforma i propri attributi in attributi SAML mediante un processo denominato “***encoding***”.

Attributi nella Federazione

- È compito della Federazione:
  - Standardizzare gli attributi scambiati fra i partecipanti alla Federazione. In particolare:
    - denominazione
    - sintassi
    - semantica
  - Limitare l'uso degli attributi ai soli effettivamente necessari per l'erogazione del servizio
- Spetta comunque all'organizzazione (*Attribute Filter Policy*) ed eventualmente all'utente (***uApprove***) limitarne il rilascio

Denominazione e sintassi

- Sono state utilizzate denominazioni e sintassi degli schemi LDAP
  - LDAPv3 (RFC 4519)
  - Cosine
  - inetOrgPerson
  - eduPerson
  - SCHAC

Notazione e metadati

- Necessari per comprendere le modalità di utilizzo dell'attributo
- È riportato l'identificativo dell'attributo, in forma di urn, come indicato da SAML1 e SAML2,

(necessario per l'encoding)

- es.

  (SAML 1) urn:mace:dir:attribute-def:sn

  (SAML 2) urn:oid:2.5.4.4

Notazione: classificazione

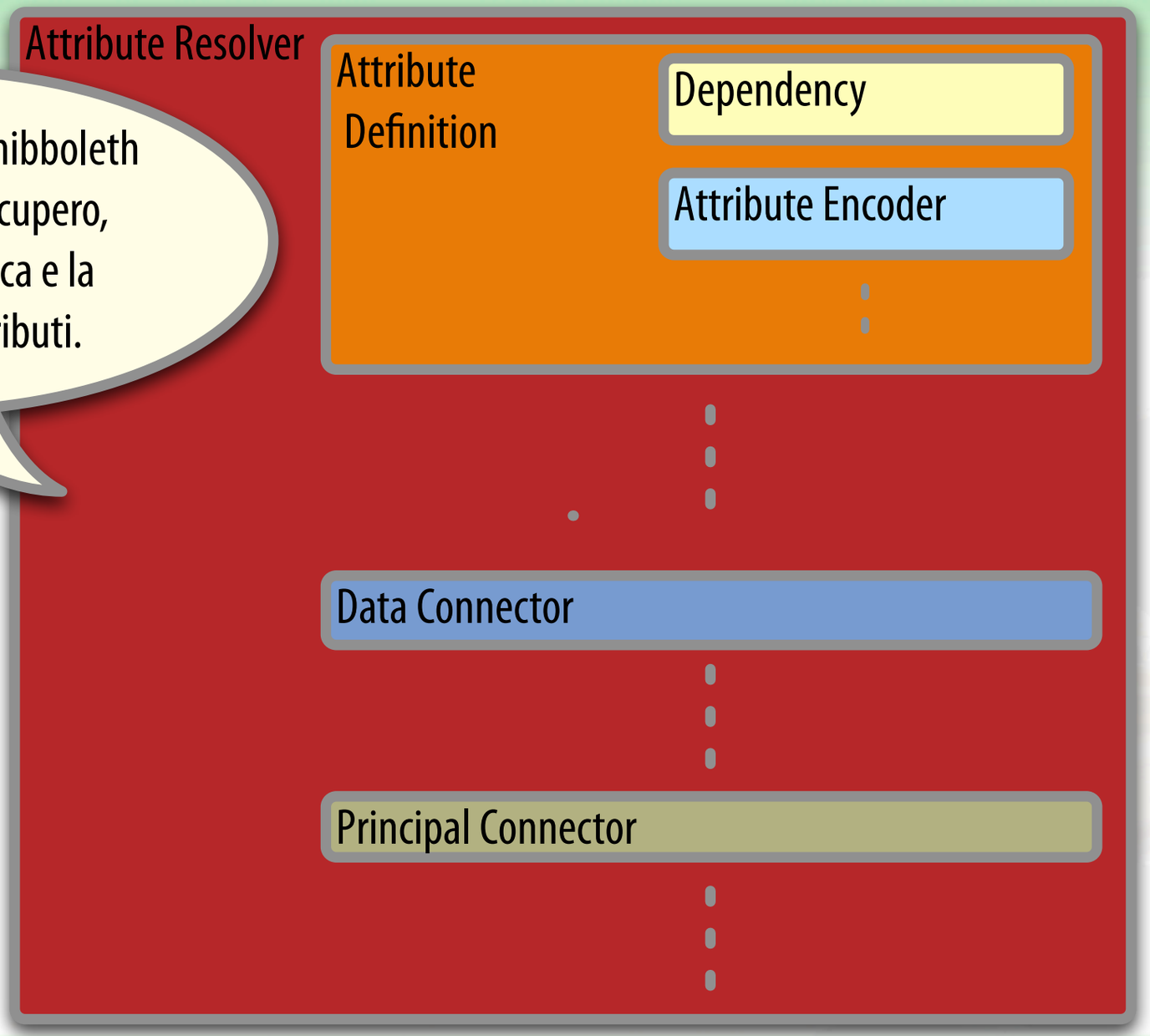
- Gli attributi sono classificati come:
  - **obbligatori**: un IdP deve fornire questi attributi per poter fare parte della federazione
  - **raccomandati**: è fortemente consigliato che un IdP fornisca questi attributi
  - **opzionali**: alcuni SP potrebbero richiedere questi attributi

L'insieme degli attributi

- Gli attributi sono suddivisi in:
  - **caratteristiche personali**: sn, givenName, cn, preferredLanguage ecc.
  - **contatti**: mail, telephoneNumber, mobile ecc.
  - **autorizzazione e accounting**: eduPersonScopedAffiliation, eduPersonTargetedID, eduPersonPrincipalName, eduPersonEntitlement

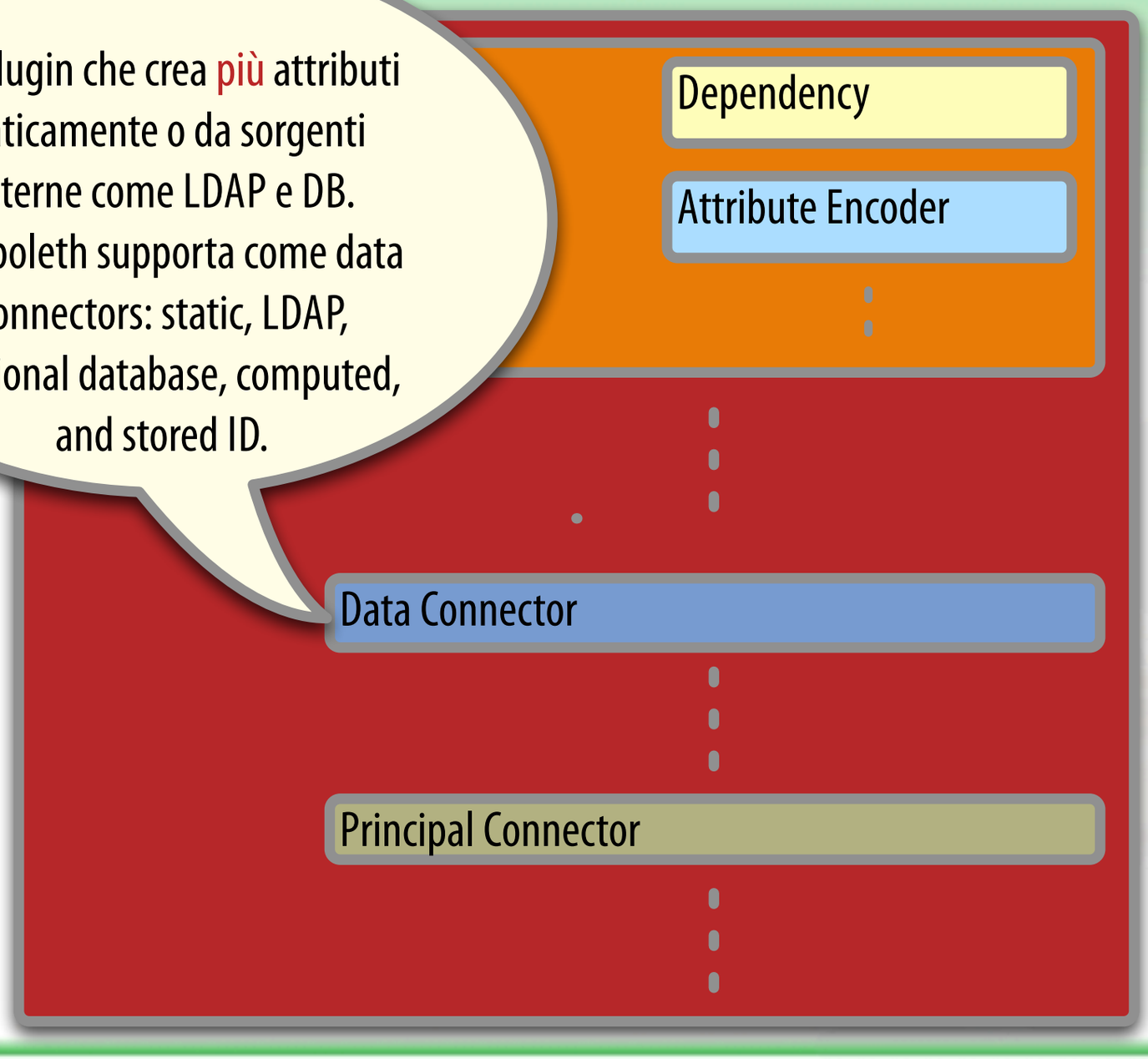
Configurazione: `attribute-resolver.xml`

Un sottosistema di Shibboleth responsabile del recupero, l'eventuale modifica e la codifica degli attributi.

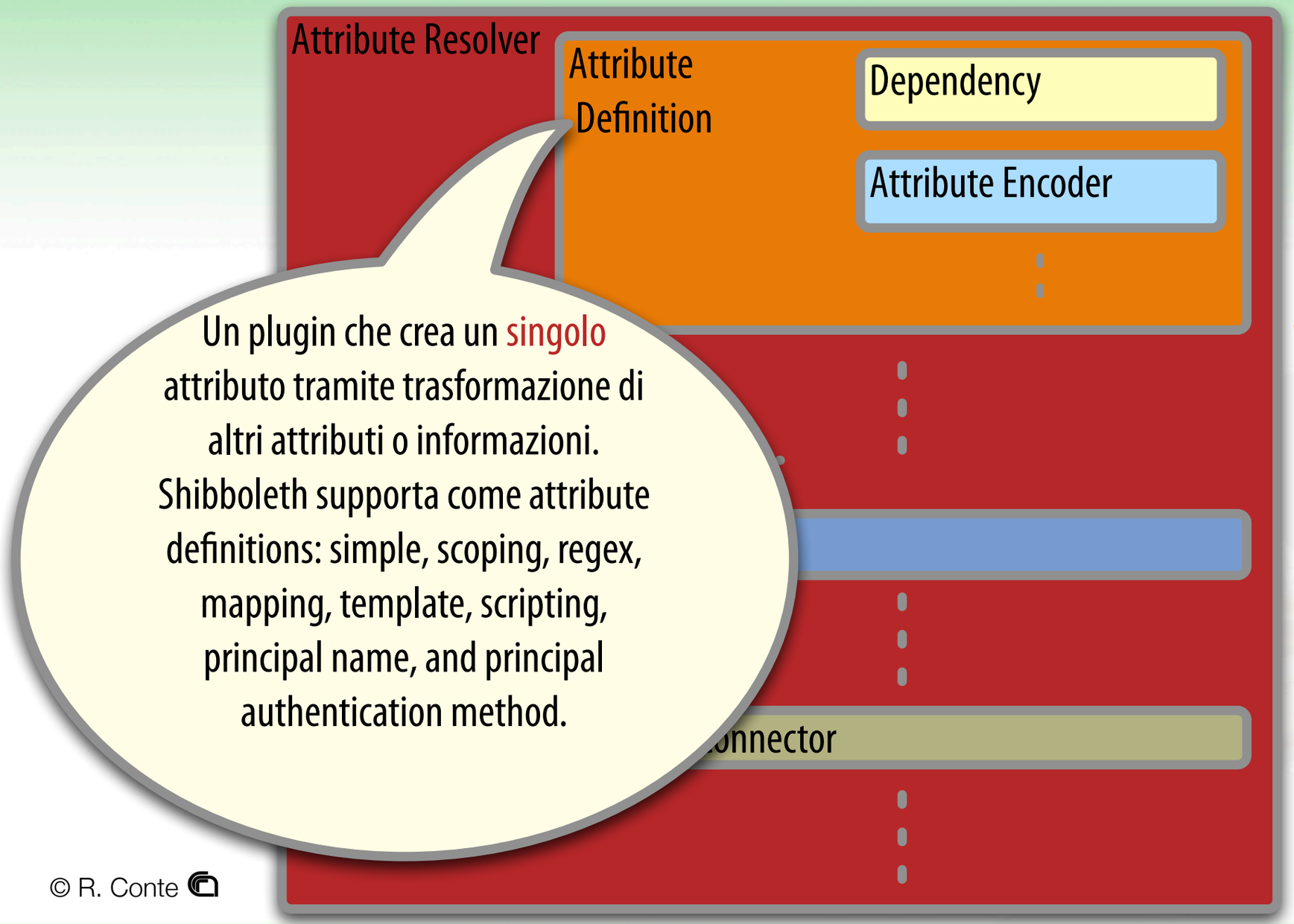


Configurazione: `attribute-resolver.xml`

Un plugin che crea più attributi staticamente o da sorgenti esterne come LDAP e DB. Shibboleth supporta come data connectors: static, LDAP, relational database, computed, and stored ID.

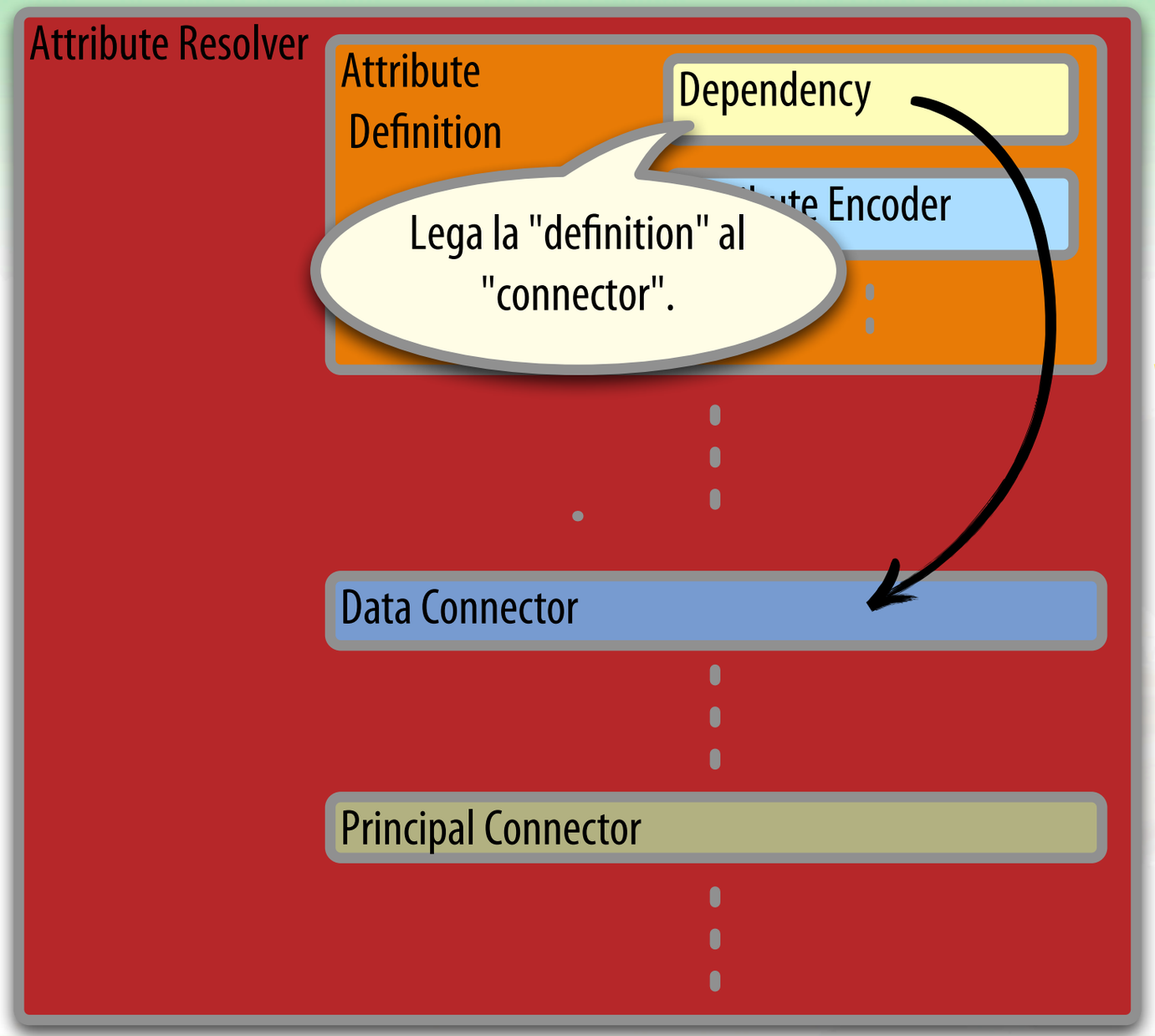


Configurazione: `attribute-resolver.xml`



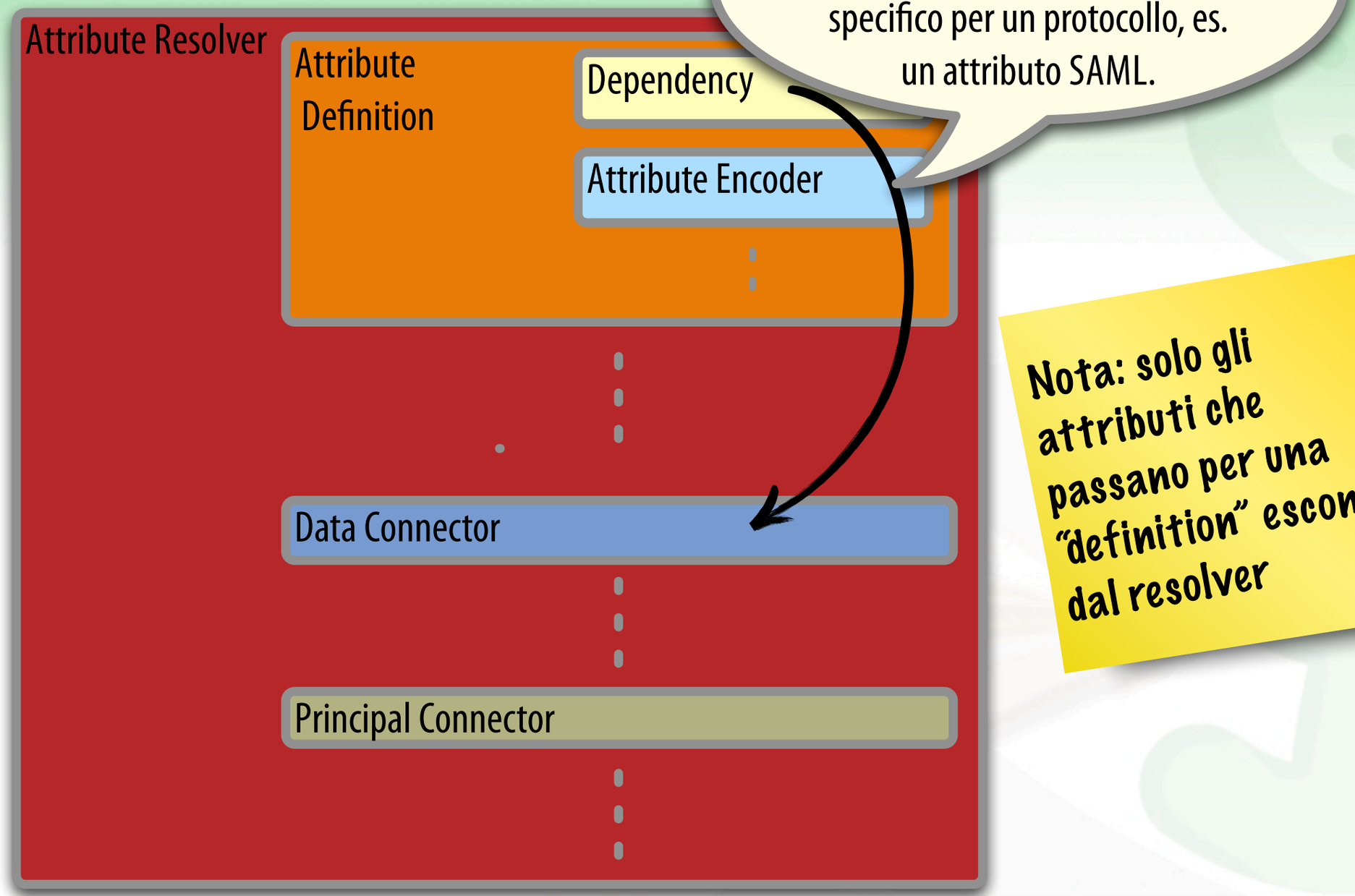
Nota: solo gli attributi che passano per una "definition" escono dal resolver

Configurazione: `attribute-resolver.xml`



Nota: solo gli attributi che passano per una "definition" escono dal resolver

Configurazione: `attribute-r` `ml`



Un attributo “static”: eduPersonOrgDN 1/2

```
<resolver:DataConnector id="staticAttributes"
  xsi:type="Static" xmlns="urn:mace:shibboleth:2.0:resolver:dc">

    <Attribute id="eduPersonOrgDN">
        <Value>o=Istituto di Fisiologia Clinica,o=CNR</Value>
    </Attribute>

</resolver:DataConnector>
```

Ogni Data Connector ha un unico id
Esistono diversi tipi di Data Connector

Ogni tipo ha un proprio set di parametri di configurazione

Un attributo “static”: eduPersonOrgDN 2/2

```
<resolver:AttributeDefinition id="eduPersonOrgDN"
    xsi:type="Simple" xmlns="urn:mace:shibboleth:2.0:resolver:ad"
    sourceAttributeID="eduPersonOrgDN">

    <resolver:Dependency ref="staticAttributes" />

    <resolver:AttributeEncoder xsi:type="SAML1String"
        xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
        name="urn:mace:dir:attribute-def:eduPersonOrgDN" />

    <resolver:AttributeEncoder xsi:type="SAML2String"
        xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
        name="urn:oid:1.3.6.1.4.1.5923.1.1.1.3"
        friendlyName="eduPersonOrgDN" />
</resolver:AttributeDefinition>
```

Un attributo “static”: eduPersonOrgDN 2/2

```
<resolver:AttributeDefinition id="eduPersonOrgDN"
    xsi:type="Simple" xmlns="urn:mace:shibboleth:2.0
    sourceAttributeID="eduPersonOrgDN">

    <resolver:Dependency ref="staticAttributes" />

    <resolver:AttributeEncoder xsi:type="SAML1String
        xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
        name="urn:mace:dir:attribute-def:eduPersonOrgDN" />

    <resolver:AttributeEncoder xsi:type="SAML2String"
        xmlns="urn:mace:shibboleth:2.0:attribute:encode
        name="urn:oid:1.3.6.1.4.1.5923.1.1.1.2"
        friendlyName="eduPersonOrgDN"
</resolver:AttributeDefinition>
```

La dipendenza è dichiarata prima di ogni altro parametro di configurazione

Ogni Attribute Definition ha un unico id
Esistono diversi tipi di Attribute Definition

Ogni tipo ha un proprio set di parametri di configurazione

Un “semplice” attributo ricavato da LDAP: cn

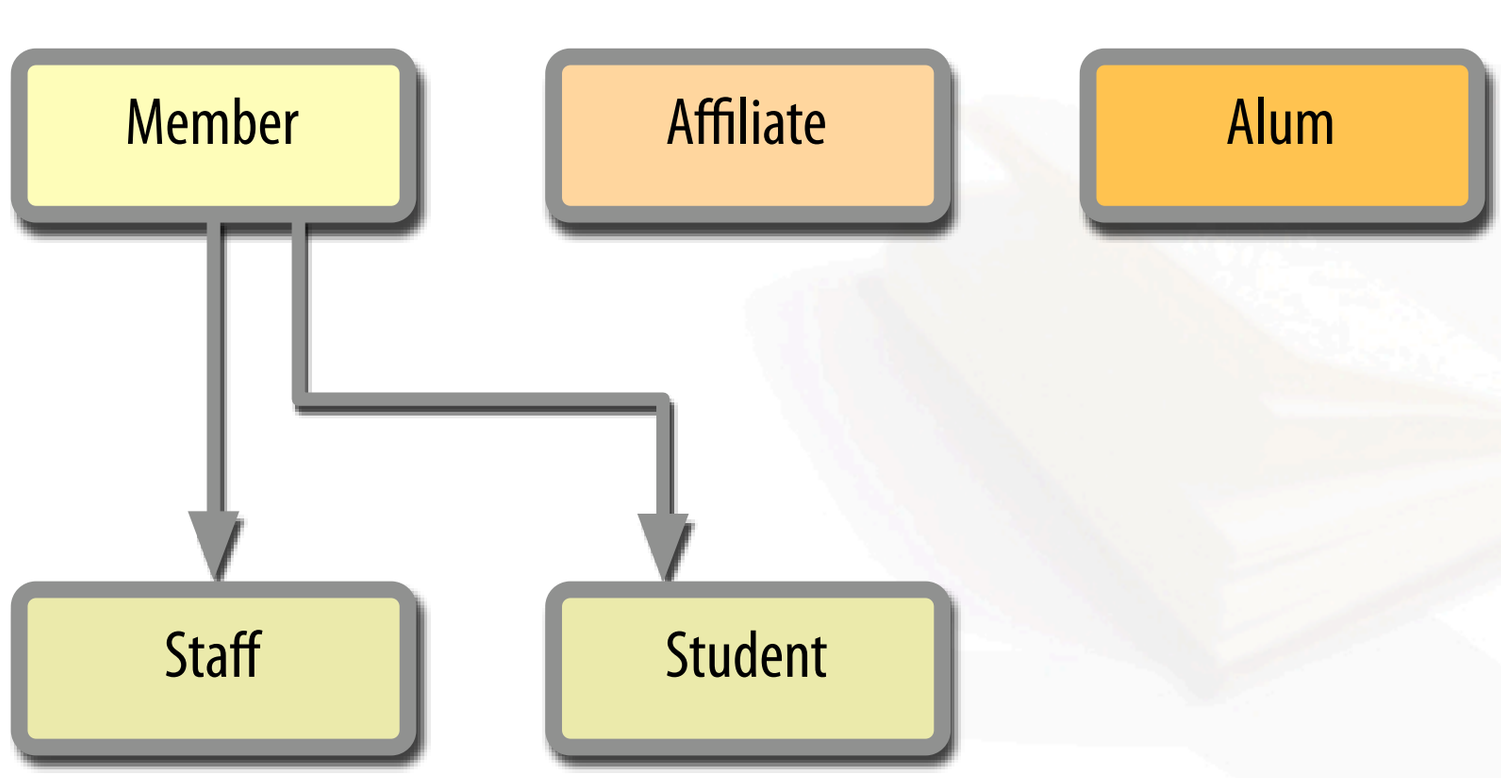
```
<resolver:AttributeDefinition id="cn" xsi:type="Simple"
    xmlns="urn:mace:shibboleth:2.0:resolver:ad"
    sourceAttributeID="cn">

  <resolver:Dependency ref="myLDAP" />

  <resolver:AttributeEncoder ... />

</resolver:AttributeDefinition>

<resolver:DataConnector id="myLDAP" xsi:type="LDAPDirectory"
    xmlns="urn:mace:shibboleth:2.0:resolver:dc"
    ldapURL="ldap://ldap.example.org"
    baseDN="ou=people,dc=example,dc=org"
    principal="uid=myservice,ou=system"
    principalCredential="myServicePassword">
  <FilterTemplate>
    <![CDATA[(uid=$requestContext.principalName)]]>
  </FilterTemplate>
</resolver:DataConnector>
```

Molti degli attributi IDEM possono essere definiti in Shibboleth 2 con il tipo "Simple"

Attenzione a non confondere la configurazione per questo LDAP con quella nel relying-party.xml

Rimappare un attributo: eduPersonAffiliation 2/2

- Definisce la relazione fra l'utente e la propria Organizzazione
- L'affiliazione prevede (al momento) come valori possibili:



Rimappare un attributo: eduPersonAffiliation 2/2

```
<resolver:AttributeDefinition id="eduPersonAffiliation" xsi:type="Mapped"
     xmlns="urn:mace:shibboleth:2.0:resolver:ad"
        dependencyOnly="true" sourceAttributeID="employeeType">

     <resolver:Dependency ref="myLDAP" />

     [...]

     <DefaultValue>affiliate</DefaultValue>

     <ValueMap>
        <ReturnValue>staff</ReturnValue>
        <SourceValue>dirigente tecnologo</SourceValue>
        <SourceValue>dirigente di ricerca</SourceValue>
[...]
        <SourceValue>ricercatore</SourceValue>
        <SourceValue>personale tecnico-amministrativo</SourceValue>
        <SourceValue>specializzando</SourceValue>
     </ValueMap>
[...]
     <ValueMap>
        <ReturnValue>member</ReturnValue>
        <SourceValue>direttore</SourceValue>
        <SourceValue>dirigente tecnologo</SourceValue>
     [...]
```

Un attributo “scoped”: eduPersonScopedAffiliation

- Espone la relazione fra utente ed Organizzazio
  formato affiliation@organization

ePSA dipende da un altro attributo: ePA

- Organizzazione nel formato DNS

```
<resolver:AttributeDefinition
    id="eduPersonScopedAffiliation"
    xsi:type="Scoped"
    xmlns="urn:mace:shibboleth:2.0:resolver:ad"
    scope="example.org">

  <resolver:Dependency ref="eduPersonAffiliation"

        [...]
</resolver:AttributeDefinition>
```

Attenzione!! lo scope DEVE corrispondere a quello dichiarato nei metadati

Non è necessario il "sourceAttributeID", l'AD individua un unico attributo

Un attributo particolare: eduPersonTargetedID

- Implementa il *persistent identifier* di SAML 2
- Permette la gestione di sessioni in forma anonima *(pseudonomizzazione)*
- IDEM utilizza la versione 2006 (conforme a SAML 2)
- Prevede *un* valore diverso (max 256 char) *per ogni* SP
- *NON* deve essere riassegnato
- Dovrebbe essere mantenuto più a lungo possibile
- Valori nel formato:

  `nameQualifier!SPNameQualifier!stringa_opaca`

Generare eduPersonTargetedID

- È generato direttamente da Shibboleth in maniera:
  - Algoritmica

    `ComputedID Data Connector`
  - Per memorizzazione

    `StoredID Data Connector`

Generare dinamicamente ePTID

- Gestione algoritmica (`ComputedID`)
  - È generato ogni volta che serve
  - SHA-1 di un attributo + salt
  - Più semplice da trattare
  - Variando l'attributo sorgente variano tutti i valori con conseguente perdita personalizzazioni
  - NON può essere identificativo utente
  - Deprecato in Shibboleth 2.x

Generare dinamicamente ePTID

```
<resolver:DataConnector
   xsi:type="ComputedId"
   xmlns="urn:mace:shibboleth:2.0:resolver:dc"
   id="computedID"
   generatedAttributeID="persistentID"
   sourceAttributeID="SOME_ID"
   salt="<stringa casuale>">

   <resolver:Dependency ref="myLDAP" />
</resolver:DataConnector>
```

Generare e memorizzare ePTID

- Gestione per memorizzazione (`StoredID`)
  - Primo valore è generato come per ComputedID
  - Richiede una tabella in DB
  - Consente la revoca e rigenerazione
  - Può essere usato come identificativo

Generare e memorizzare ePTID

```
<resolver:DataConnector
   xsi:type="StoredId"
   xmlns="urn:mace:shibboleth:2.0:resolver:dc"
   id="storedID"
   generatedAttributeID="persistentID"
   sourceAttributeID="SOME_ID"
   salt="<stringa casuale>">

   <resolver:Dependency ref="myLDAP" />
   <ApplicationManagedConnection
      jdbcDriver="DRIVER_CLASS"
      jdbcURL="DATABASE_URL"
      jdbcUserName="DATABASE_USER"
      jdbcPassword="DATABASE_USER_PASSWORD" />

</resolver:DataConnector>
```

Definire eduPersonTargetedID

```
<resolver:AttributeDefinition
   id="eduPersonTargetedID"
   xsi:type="SAML2NameID"
   xmlns="urn:mace:shibboleth:2.0:resolver:ad"
   nameIdFormat="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"
   sourceAttributeID="persistentID">

   <resolver:Dependency ref="computedID" />

   <resolver:AttributeEncoder xsi:type="SAML1XMLObject"
      xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
      name="urn:oid:1.3.6.1.4.1.5923.1.1.1.10" />

   <resolver:AttributeEncoder xsi:type="SAML2XMLObject"
      xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
      name="urn:oid:1.3.6.1.4.1.5923.1.1.1.10"
      friendlyName="eduPersonTargetedID" />
</resolver:AttributeDefinition>
```

oppure "storedID"

Notare gli encoder: entrambi usano la notazione con OID per usare ePTID nella vers. 2006

Ricevere eduPersonTargetedID

Attenzione!! Questa configurazione riguarda il SP (attribute-map.xml)

```
<!-- First, the deprecated version: -->
<Attribute name="urn:mace:dir:attribute-def:eduPersonTargetedID"
    id="targeted-id">

    <AttributeDecoder xsi:type="ScopedAttributeDecoder"/>
</Attribute>

<!-- Second, the new version (note the OID-style name): -->
<Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.10" id="persistent-id">

<AttributeDecoder xsi:type="NameIDAttributeDecoder"
    formatter="$NameQualifier!$SPNameQualifier!$Name"/>
</Attribute>

<!-- Third, the SAML 2.0 NameID Format: -->
<Attribute name="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"
    id="persistent-id">

<AttributeDecoder xsi:type="NameIDAttributeDecoder"
    formatter="$NameQualifier!$SPNameQualifier!$Name"/>
</Attribute>
```

Autorizzazione esplicita: eduPersonEntitlement

- es. “hanno diritto di accedere alla risorsa x gli utenti a cui assegno la uri y in ePE”

  (y potrebbe essere l’identificativo di x)

  - es.:

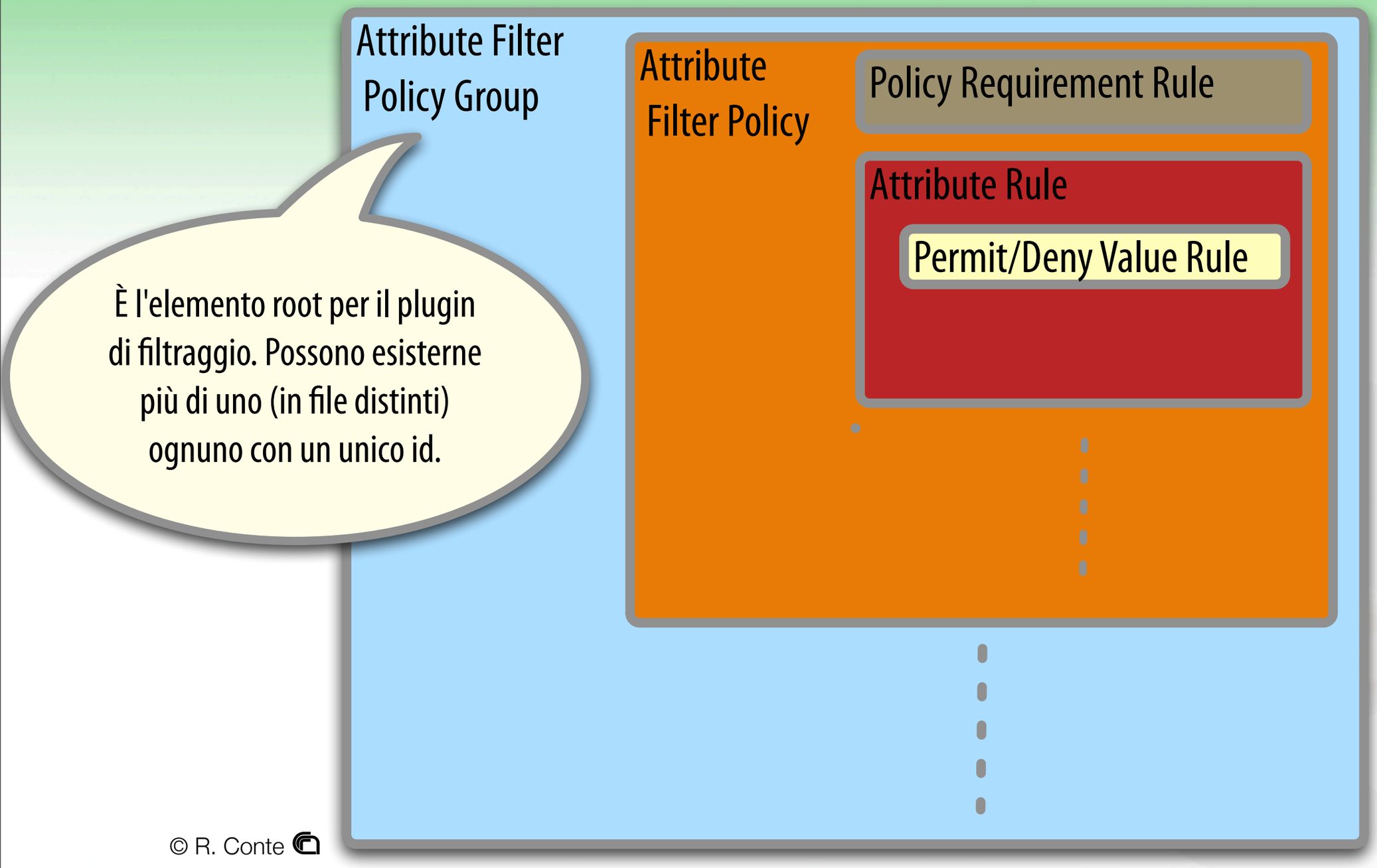
    http://nilde.bo.cnr.it
    urn:mace:cnr.it:services:puma:docs:1234

- E l’IdP che autorizza l’accesso alla risorsa
- Discriminatory Access Control (DAC) vs Attribute Based Access Control (ABAC)
- Equivale a definire dei gruppi

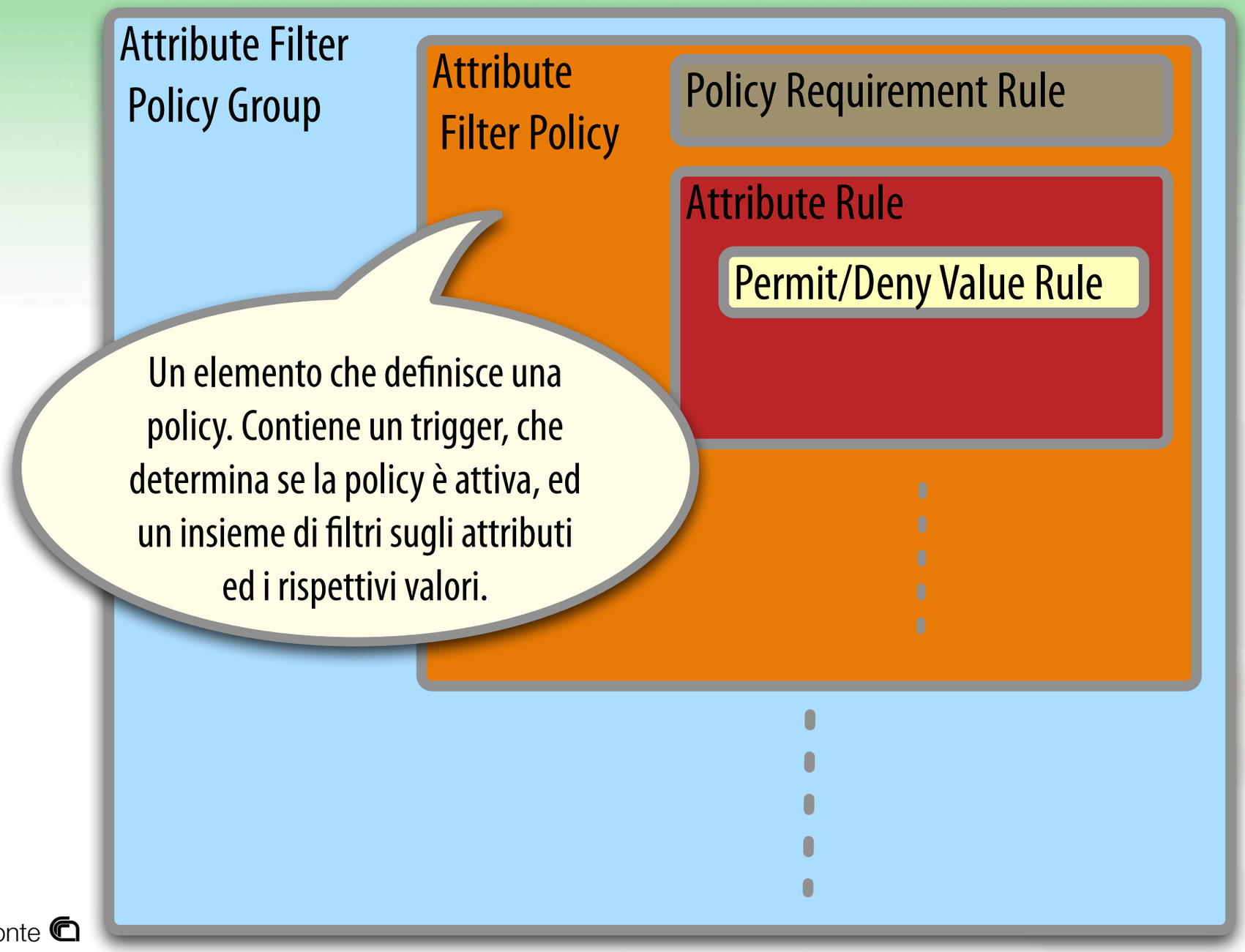
Attribute Filter Policy

- Insieme di regole per SP/attributo (o gruppi)
- È possibile utilizzare più file configurando service.xml

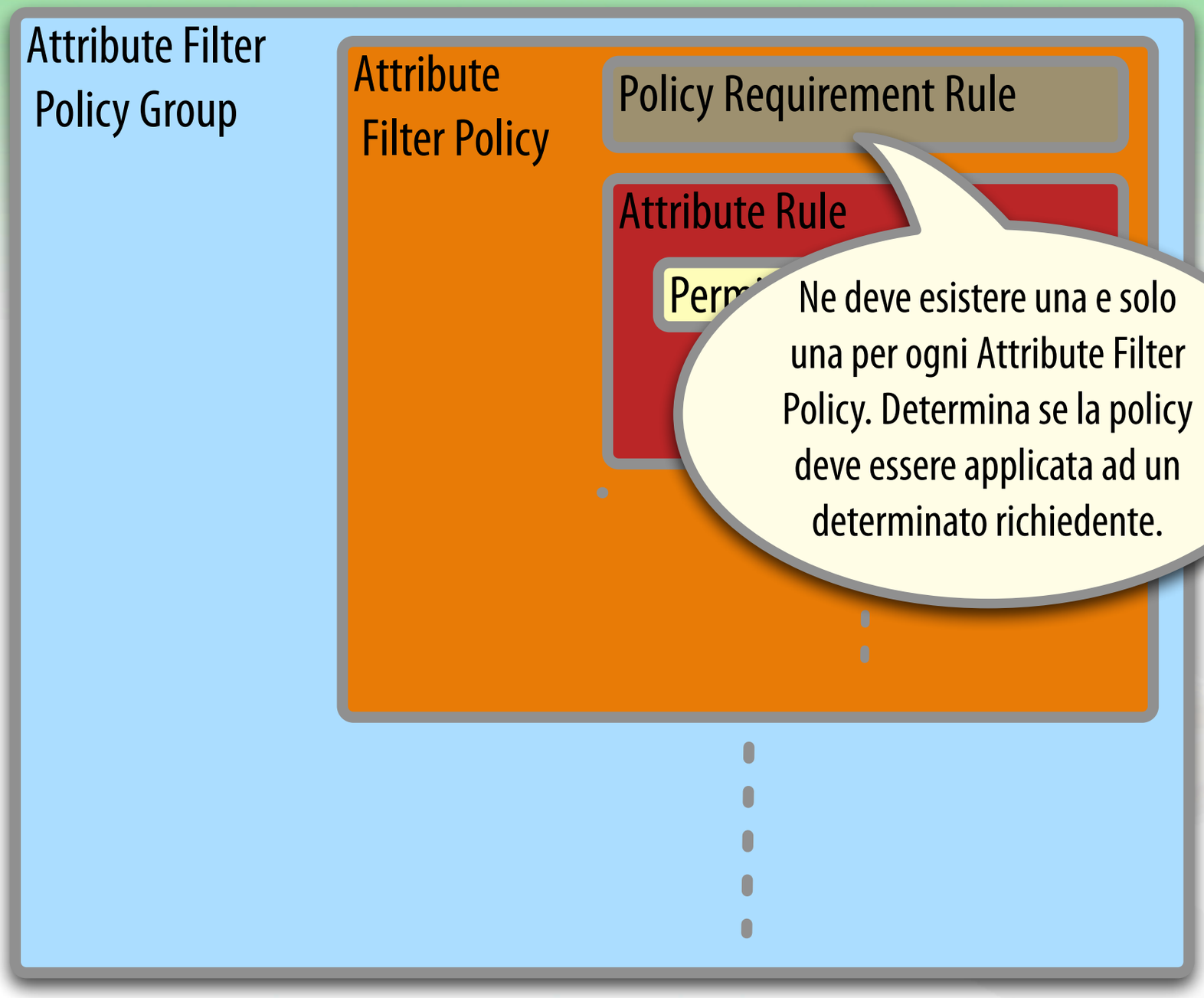
Configurazione: `attribute-filter.xml`



Configurazione: `attribute-filter.xml`

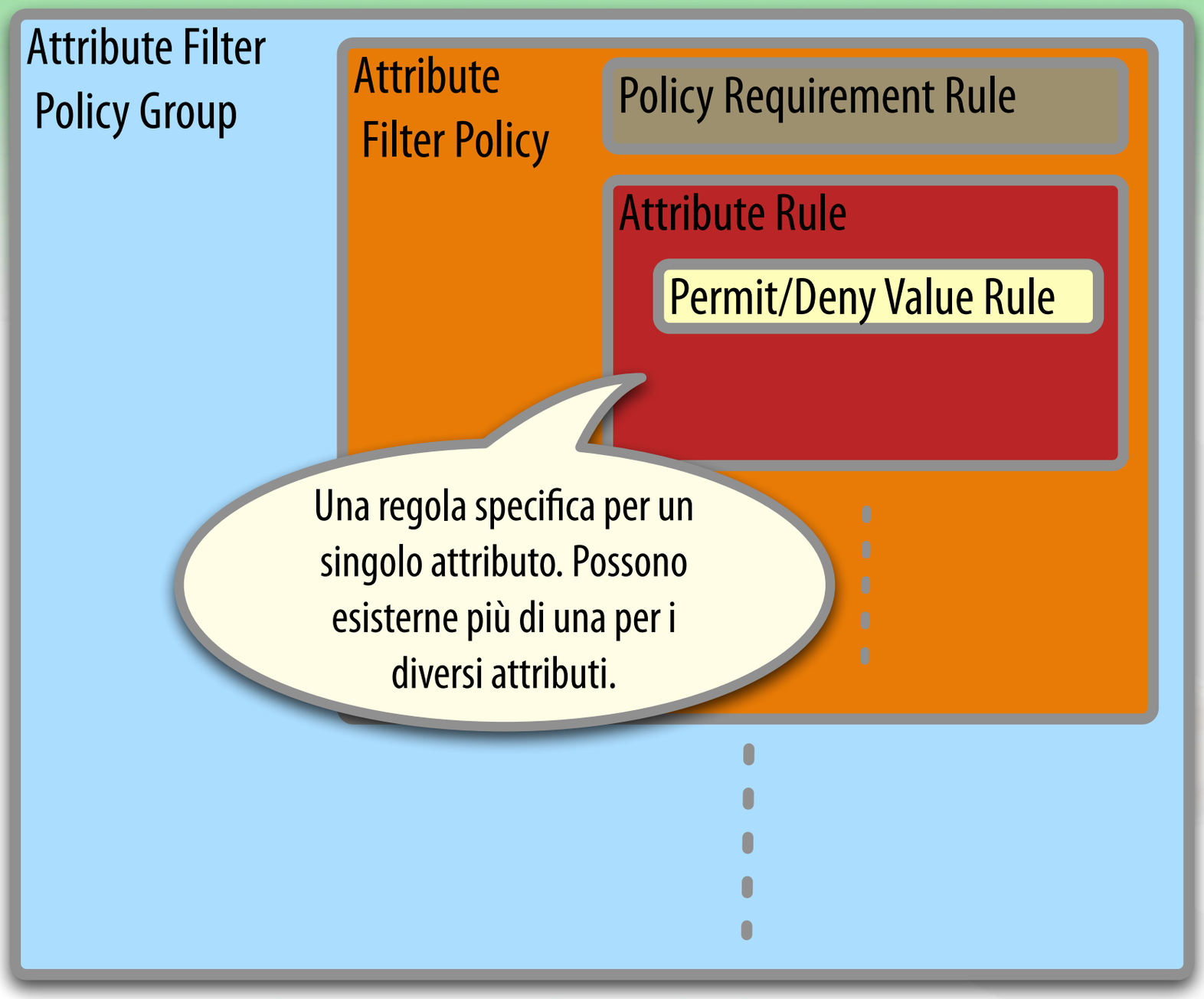


Configurazione: `attribute-filter.xml`

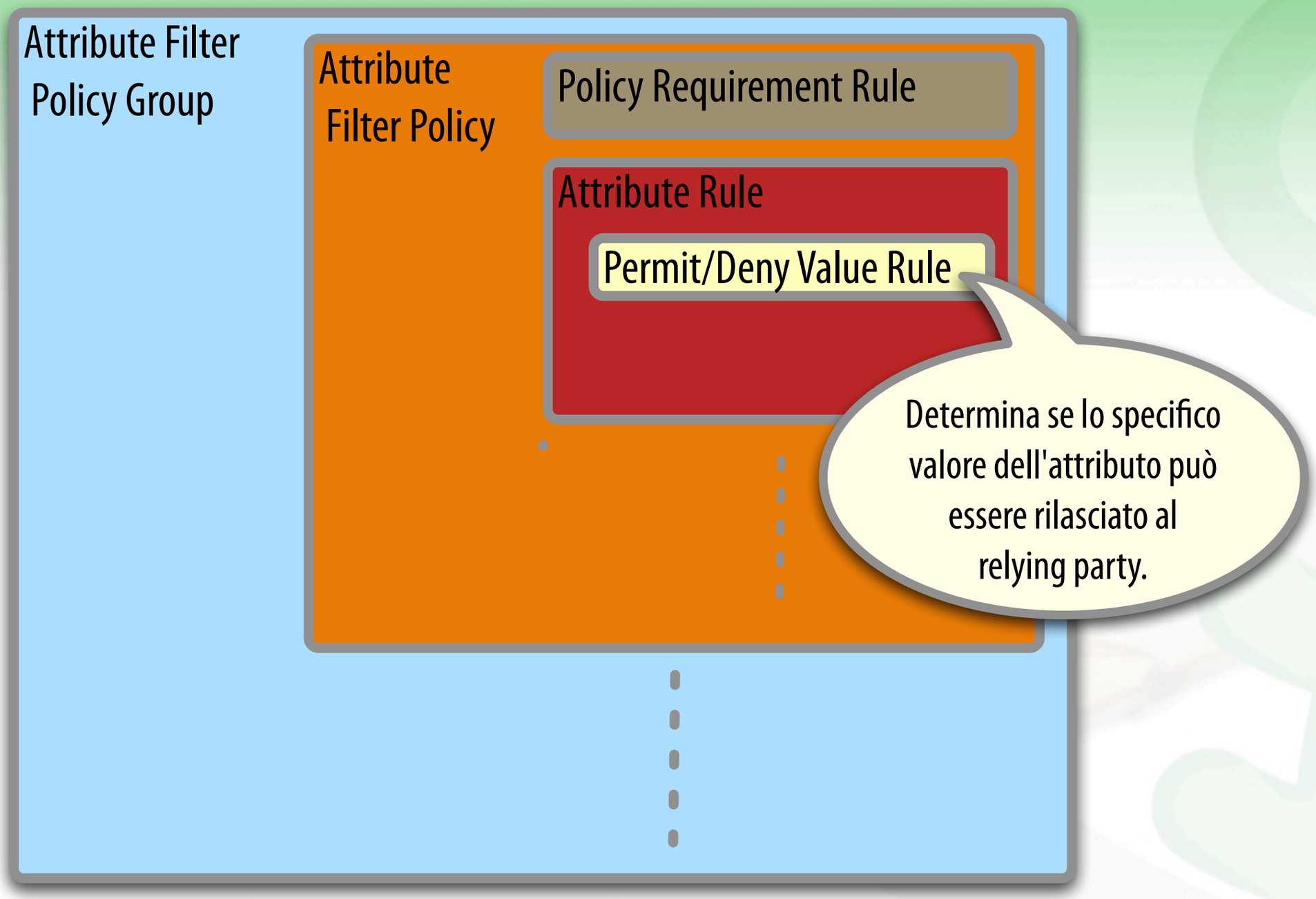


© R. Conte

Configurazione: `attribute-filter.xml`



Configurazione: `attribute-filter.xml`



Rilasciare un attributo a tutti: transientID

```
<AttributeFilterPolicy id="releaseTransientIdToAnyone">
    <PolicyRequirementRule xsi:type="basic:ANY" />
    <AttributeRule attributeID="transientId">
        <PermitValueRule xsi:type="basic:ANY" />
    </AttributeRule>

</AttributeFilterPolicy>
```

Ogni "rule" ha un unico id. Esistono diversi tipi di PolicyRequirementRule e Permit/ DenyValueRule

Ogni tipo ha un set di parametri di configurazione diverso

Rilasciare un attributo solo a qualcuno

```
<AttributeFilterPolicy id="releaseToSpExampleOrg">

   <PolicyRequirementRule
      xsi:type="basic:AttributeRequesterString"
      value="http://sp.example.org"/>

   <AttributeRule attributeID="email">
       <PermitValueRule xsi:type="basic:ANY" />
   </AttributeRule>

</AttributeFilterPolicy>
```

Negare un attributo in funzione di un altro

```
<AttributeFilterPolicy id="denyOnPrivacyAttr">

   <PolicyRequirementRule xsi:type="basic:AttributeValueString"
      attributeID="privacyAttr"
      value="true" />

      <AttributeRule attributeID="firstName">
         <DenyValueRule xsi:type="basic:ANY" />
      </AttributeRule>

      <AttributeRule attributeID="surname">
         <DenyValueRule xsi:type="basic:ANY" />
      </AttributeRule>

</AttributeFilterPolicy>
```

Consenso Informato

Consenso informato: uApprove

- Plug-in per Shibboleth Identity Provider
- Per l'utente:
  - Indica gli attributi rilasciati dall'Idp
  - Tiene traccia del consenso indicato
- Per il gestore dell'IdP
  - Obbliga l'accettazione dei termini di uso
  - Registra l'accesso alle risorse

uApprove: caratteristiche

- Software sviluppato da Switch in java
  - Storage in MySQL o xml flat file
- Termini d'uso (cambio di versione)
- Digital ID Card con attributi rilasciati
- Gestisce il consenso globale
- Permette di azzerare il consenso dato (mysql)

uApprove: ID Card

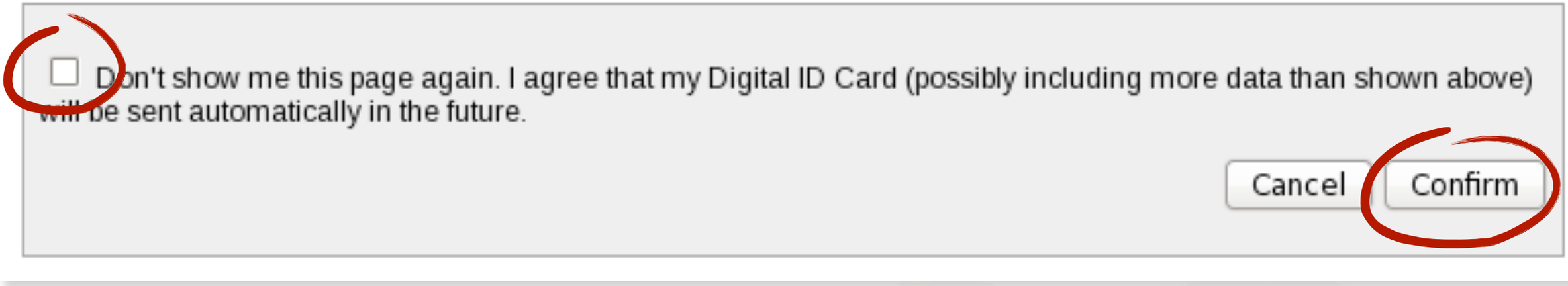
This is the Digital ID Card to be sent to '**https://sp-test.garr.it**':

Digital ID Card	
surname	**Monticini**
givenName	**Barbara**
email	**monticini@garr.it** **monticini@fi.infn.it**
uid	**bmonticini**
preferredLanguage	**it**
eduPersonAffiliation	**member** **staff** **employee**
eduPersonOrgUnitDN	**o=garr,ou=firenze**
eduPersonEntitlement	**urn:mace:rediris.es:entitlement:wiki:tfemc2**
eduPersonPrincipalName	**bmonticini**
organizationName	**GARR**
eduPersonScopedAffiliation	**member** **staff** **employee**
organizationalUnit	**people**
commonName	**Barbara Monticini**

uApprove: come funziona

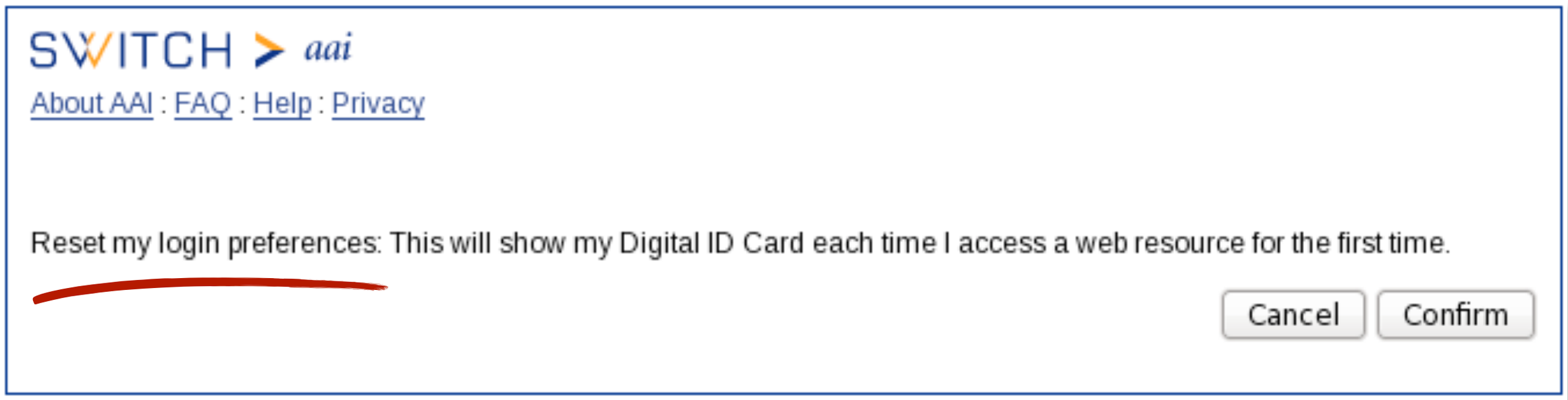
- L'utente che accede ad una risorsa:
  - Accettazione dei Termini d'uso (1a volta)
  - Consenso al rilascio degli attributi (che resta valido finche' non cambia il set)
  - Consenso globale (anche in caso di cambiamenti al set)
  - Reset del consenso globale

uApprove: Consenso



- Consenso per SP
  - Solo bottone “Confirm”
- Consenso Globale
  - Check “Dont't show me this page again”

uApprove: Reset consent



Grazie