Francesco Sansone

SIIT Group @ IFC

CTS-IDEM


@khekkosan

DATA SOURCE MULTIPLI

VI CHIEDO SOLO UNA COSA...

```
ansible-playbook playbook.yml -i hosts -e '{"pre_t":3}'
```

DI COSA PARLIAMO?

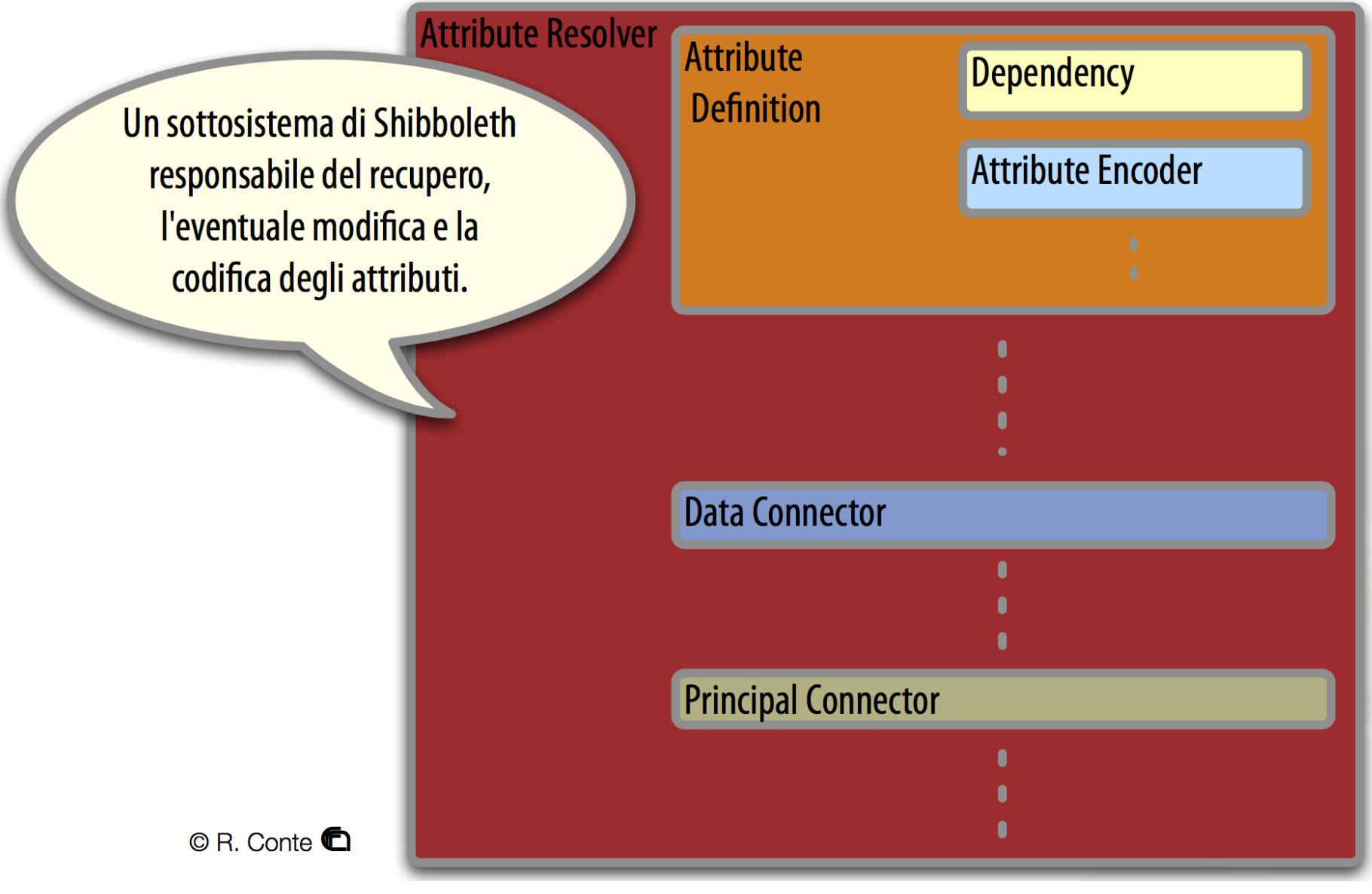
come l'IdP recupera i dati *raw*

come l'IdP produce gli *Attributi*

DOVE SONO IMPIEGATI (?)

- definzione dei data source
  - ad es. in ldap.properties
  - come bean (di solito in global.xml)
- uso dei data source per funzioni specialistiche
  - autenticazione (check username/pwd)
  - persistentId (storicizzazione persistentId)
- definzione dei data connector in
  - attribute-resolver.xml

ATTRIBUTE RESOLVER



ATTRIBUTE DEFINITION

Definizione di uno o più attributi e degli *Encoders* associati

hanno un TIPO!!!!

Simple, Scoped, Mapped, Template, ScriptedAttribute, RegexSplit, Prescoped, PrincipalName

ATTRIBUTE DEFINITION - SIMPLE

```
<AttributeDefinition
        xsi:type="Simple"
        id="cn"
        sourceAttributeID="cn"
>
     <Dependency ref="myLDAP" />
     <AttributeEncoder
        xsi:type="SAML2String"
        name="urn:oid:2.5.4.3"
        friendlyName="cn"
        encodeType="false"
     />
</AttributeDefinition>
```

ATTRIBUTE DEFINITION - MAPPED

```
<AttributeDefinition id="eduPersonAffiliation"
        xsi:type="Mapped" xmlns="urn:mace:shibboleth:2.0:resolver:ad"
        dependencyOnly="true"sourceAttributeID="employeeType"
>
<!-- Essendo solo per uso interno l'Encoder non serve -->
  <Dependency ref="myLDAP" />
  <DefaultValue>affiliate</DefaultValue>

  <ValueMap>
    <ReturnValue>member</ReturnValue>
    <SourceValue>tempo indeterminato</SourceValue>
    <SourceValue>tempo determinato</SourceValue>
    <SourceValue>assegno di ricerca</SourceValue>
  </ValueMap>

  <ValueMap>
    <ReturnValue>staff</ReturnValue>
    <SourceValue>tirocinio extracurriculare</SourceValue>
    <SourceValue>distaccato da altri organi cnr</SourceValue>
    <SourceValue>borsa di studio</SourceValue>
        <SourceValue>associato di ricerca</SourceValue>
  </ValueMap>
</AttributeDefinition>
```

ATTRIBUTE DEFINITION - SCRIPTED

```
<AttributeDefinition xsi:type="ScriptedAttribute" id="schacHomeOrganization" >
    <Dependency ref="myLDAP" />
    <AttributeEncoder
                xsi:type="SAML2String"
                name="urn:oid:1.3.6.1.4.1.25178.1.2.9"
                friendlyName="schacHomeOrganization"
                encodeType="false" />
    <Script>
    <![CDATA
      logger = Java.type("org.slf4j.LoggerFactory").getLogger("net.shibboleth.idp.attribute.resolver.schacHomeOrganizationBuilder");
      valueType = Java.type("net.shibboleth.idp.attribute.StringAttributeValue");
      // This implementation composes the value of the attribute schacHomeOrganization from the values of the properties idp.scope inside idp.properties.
      if (typeof schacHomeOrganization === 'undefined' || schacHomeOrganization.getValues().size() < 1) {
         logger.info("No schacHomeOrganization in LDAP found, creating one");
         // compose value from 'schacHomeOrganization'
         schacHomeOrganization.addValue(new valueType("%{idp.scope}"));
         logger.info("schacHomeOrganization final value: " + schacHomeOrganization.getValues().get(0));
      } else {
         logger.info("schacHomeOrganization has value: " + schacHomeOrganization.getValues().get(0));
      }
    ]]>
    </Script>
</AttributeDefinition>
```

DATA CONNECTOR

producono set di *IdPAttribute* consumati da *Attribute resolver*

definiscono le connessioni alle varie sorgenti dati

definibili interamente in attribute-resolver.xml

```
<DataConnector id="myDatabase" xsi:type="RelationalDatabase">
  <FailoverDataConnector ref="BackupDataseConnector"/>
  <ApplicationManagedConnection
   jdbcDriver="org.hsqldb.jdbc.JDBCDriver" jdbcURL="jdbc:hsqldb:mem:RDBMSDataConnectorStore"
   jdbcUserName="SA" jdbcPassword="secret" />
  <dc:QueryTemplate>
     <![CDATA[
     SELECT * FROM people WHERE userid='$resolutionContext.principal'
     ]]>
  </QueryTemplate>

  <Column columnName="homephone" attributeID="phonenumber" />

  <ResultCache elementTimeToLive="PT10S"/>
</DataConnector>
```

oppure definibili richiamando un bean già esistente

```
<DataConnector
        id="myDatabase"
        xsi:type="RelationalDatabase"
        mappingStrategy="MappingBeanId"
>
  <BeanManagedConnection>DataConnectorBeanId</BeanManagedConnection>
  <QueryTemplate>
  [...]
</DataConnector>
```

Ad esempio può essere riutilizzato il *bean* definito per il *persistentId* e lo storage general-purpose (sessione e consenso).

hanno un TIPO!!!!

Static, ScriptedDataConnector, ComputedId, StoredId,
RelationalDatabase, LDAPDirectory

StaticDataConnector

```
<DataConnector id="staticAttributes"xsi:type="Static">
  <Attribute id="eduPersonAffiliation">
    <Value>CORSO IdP SUPERSONICO</Value>
  </Attribute>
</DataConnector>
```

RelationalDatabaseConnector

TOCCA A VOI!

- configura un *RelationalDatabase DataConnector*
  - file: */opt/shibboleth-idp/conf/attribute-resolver-myRdbms.xml*
  - connessione: <BeanManagedConnection> = MyDataSource
- riavvio del servizio
  - curl -sk https://idp.example.org/idp/profile/admin/reload-service?id=shibboleth.AttributeResolverService

RuoliOrganizzativi

```
Tabella shibboleth.RuoliOrganizzativi
(
  id MEDIUMINT NOT NULL,
  uid VARCHAR(255) NOT NULL,
  ruolo VARCHAR(255) NOT NULL,
  PRIMARY KEY (id)
);
```

id	uid	ruolo
0	Mario	Tecnico
1	Pino	Amministrativo
2	Gino	Docente

RUOLO: DA DB A IDP

- configura una query con chiave *uid*
  - tabella: *RuoliOrganizzativi*
  - attributo da referenziare: `myRuolo`

ESTRAGGO I DATI DA DB

```
  <BeanManagedConnection>MyDataSource</BeanManagedConnection>
  <QueryTemplate>
<![CDATA[
    SELECT * FROM shibboleth.RuoliOrganizzativi WHERE uid = '$resolutionContext.principal'
]]>
  </QueryTemplate>

  <!-- Ref myRuolo -->

  <Column columnName="ruolo" attributeID="myRuolo" />
```

ATTRIBUTE DEFINITION

```
<AttributeDefinition xsi:type="Simple" id="myRuolo" sourceAt
      <Dependency ref="myRdbms" />

      <AttributeEncoder xsi:type="SAML2String" name="urn:oid
</AttributeDefinition>
```

Per poter rilasciare l'attributo dobbiamo configurare anche *attribute-filter.xml*.
E' una AttributeRule come le altre:

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

ULTIMO STEP: RILASCIO

```
<util:list id ="shibboleth.AttributeResolverResources">
      <value>%{idp.home}/conf/attribute-resolver.xml</value>
      <value>AGGIUNGERE QUI IL NOSTRO RESOLVER CUSTOM</value>
</util:list>
```

RIAVVIO DEL SERVIZIO

```
curl -sk
https://idp.example.org/idp/profile/admin/reload-
service?id=shibboleth.AttributeResolverService
```