Roma, 2-3 Dicembre 2010
Ministero dell'Istruzione, dell'Università e della Ricerca

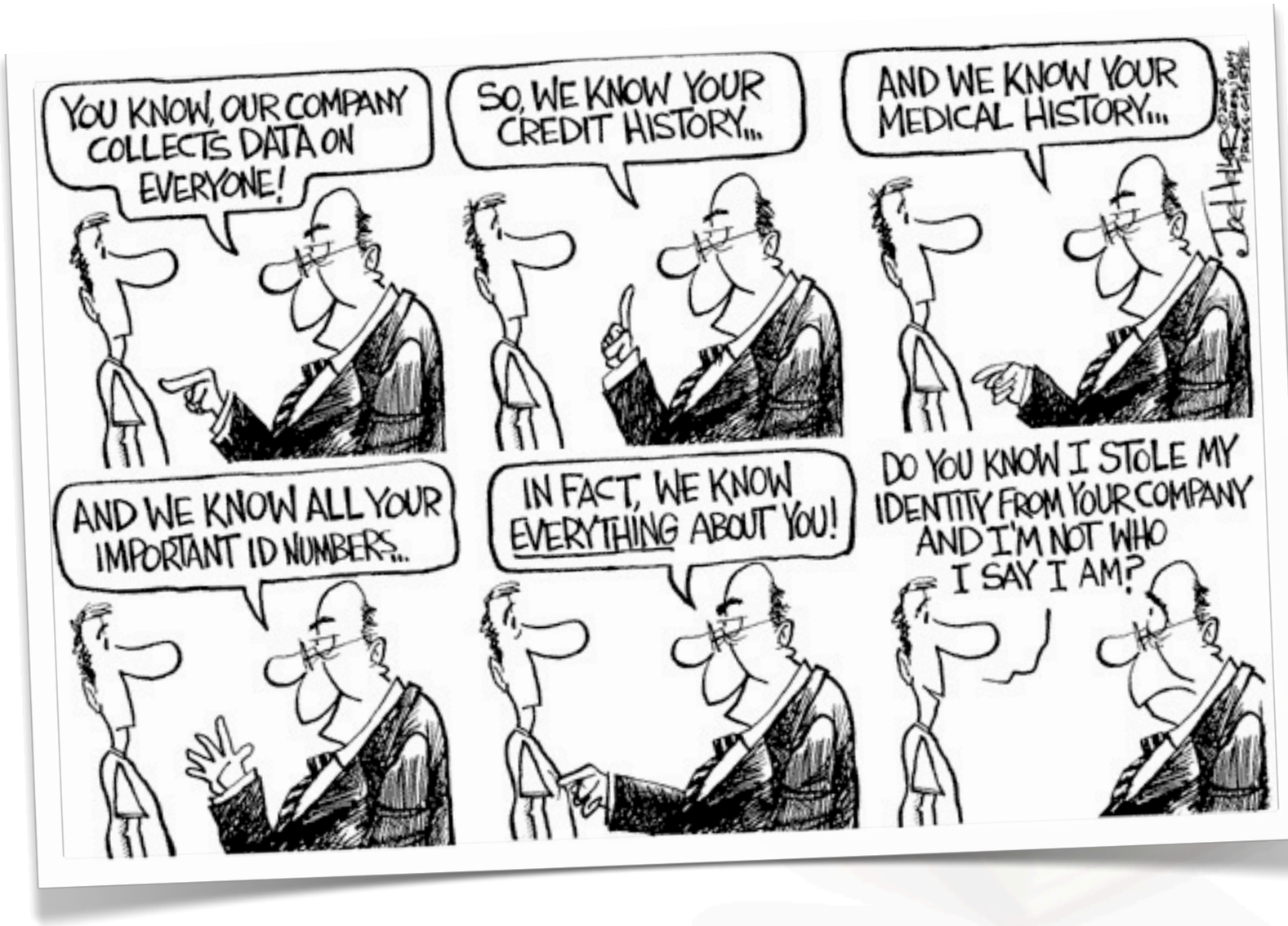
# Anatomia di un Identity Provider

Raffaele.Conte@cnr.it
*Istituto di Fisiologia Clinica del CNR*
*Comitato Tecnico Scientifico - Federazione IDEM*

# Agenda

- Identità digitali & Identity Management Systems
- Shibboleth
- Logging
- Autenticazione
- Metadati
- Aderire alla Federazione IDEM
- Attributi: risoluzione e filtraggio

# Identità digitali & Identity Management Systems



# Perché gestire le identità digitali? È necessario?

Un'organizzazione il cui sistema informativo comprende più sistemi e più utenti, anche non avendo procedure formali e/o sistemi dedicati, comunque gestisce le identità digitali

Sono ben gestite?
Quanto costa gestirle?

# Gestire le identità digitali è fondamentale

Per:

- ridurre i costi e migliorare l'efficienza della propria organizzazione
- consentire operazioni più agevoli sul proprio sistema informativo
- limitare i rischi
- rispettare la legge

# Rispettare la legge!

**"Codice in materia di protezione dei dati personali"** (D.L. 30/6/2003, n

Art. 3 (Principio di necessità nel trattamento dei dati)

1. I sistemi informativi e i programmi informatici sono configurati ridu
minimo l'utilizzazione di dati personali e di dati identificativi, in modo
escluderne il trattamento quando le finalità perseguite nei singoli cas
possono essere realizzate mediante, rispettivamente, dati anonimi o
opportune modalità che permettano di identificare l'interessato solo
di necessità.

Pseudonomizzazione

Art. 34 (Trattamenti con strumenti elettronici)

1. Il trattamento di dati personali effettuato con strumenti elettronici è consentito solo se sono adottate [...] le seguenti misure minime:

a) autenticazione informatica;

b) adozione di procedure di gestione delle credenziali di autenticazione;

c) utilizzazione di un sistema di autorizzazione;

[...]

# Rispettare la legge!!

D. L. 196/2003, ALLEGATO B - **DISCIPLINARE TECNICO IN MATERIA DI MISURE MINIME DI SICUREZZA**

Sistema di autenticazione informatica

[...]

5. La parola chiave, quando è prevista dal sistema di autenticazione, è composta da almeno otto caratteri oppure, nel caso in cui lo strumento elettronico non lo permetta, da un numero di caratteri pari al massimo consentito; essa non contiene riferimenti agevolmente riconducibili all'incaricato ed è modificata da quest'ultimo al primo utilizzo e, successivamente, almeno ogni sei mesi. In caso di trattamento di dati sensibili e di dati giudiziari la parola chiave è modificata almeno ogni tre mesi.

6. Il codice per l'identificazione, laddove utilizzato, non può essere assegnato ad altri incaricati, neppure in tempi diversi.

7. Le credenziali di autenticazione non utilizzate da almeno sei mesi sono disattivate, salvo quelle preventivamente autorizzate per soli scopi di gestione tecnica.

[...]

# Rispettare la legge!!!

D. L. 196/2003, ALLEGATO B - **DISCIPLINARE TECNICO IN MATERIA DI MISURE MINIME DI SICUREZZA**

[...]

Sistema di autorizzazione

12. Quando per gli incaricati sono individuati profili di autorizzazione di ambito diverso è utilizzato un sistema di autorizzazione.

13. I profili di autorizzazione, per ciascun incaricato o per classi omogenee di incaricati, sono individuati e configurati anteriormente all'inizio del trattamento, in modo da limitare l'accesso ai soli dati necessari per effettuare le operazioni di trattamento.

[...]

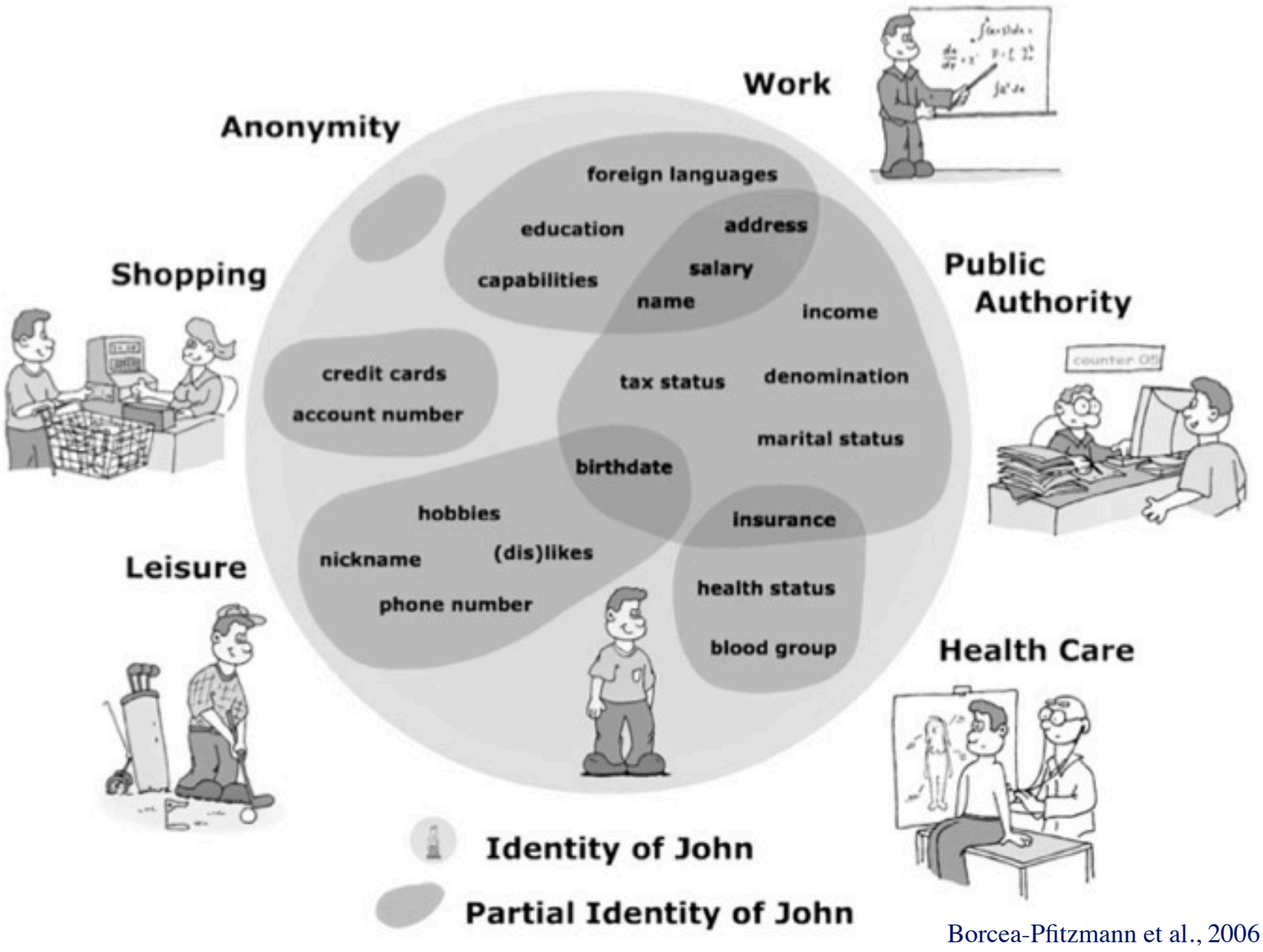
# Identità: concetti di base

Può essere rappresentata come un insieme di *attributi* che descrivono l'individuo

L'insieme contiene più sottoinsiemi *(identità parziali)*

È sufficiente un'identità parziale per rappresentare un individuo in differenti contesti o situazioni

# Identità parziali




Borcea-Pfitzmann et al., 2006

# Evoluzione nella gestione delle identità digitali

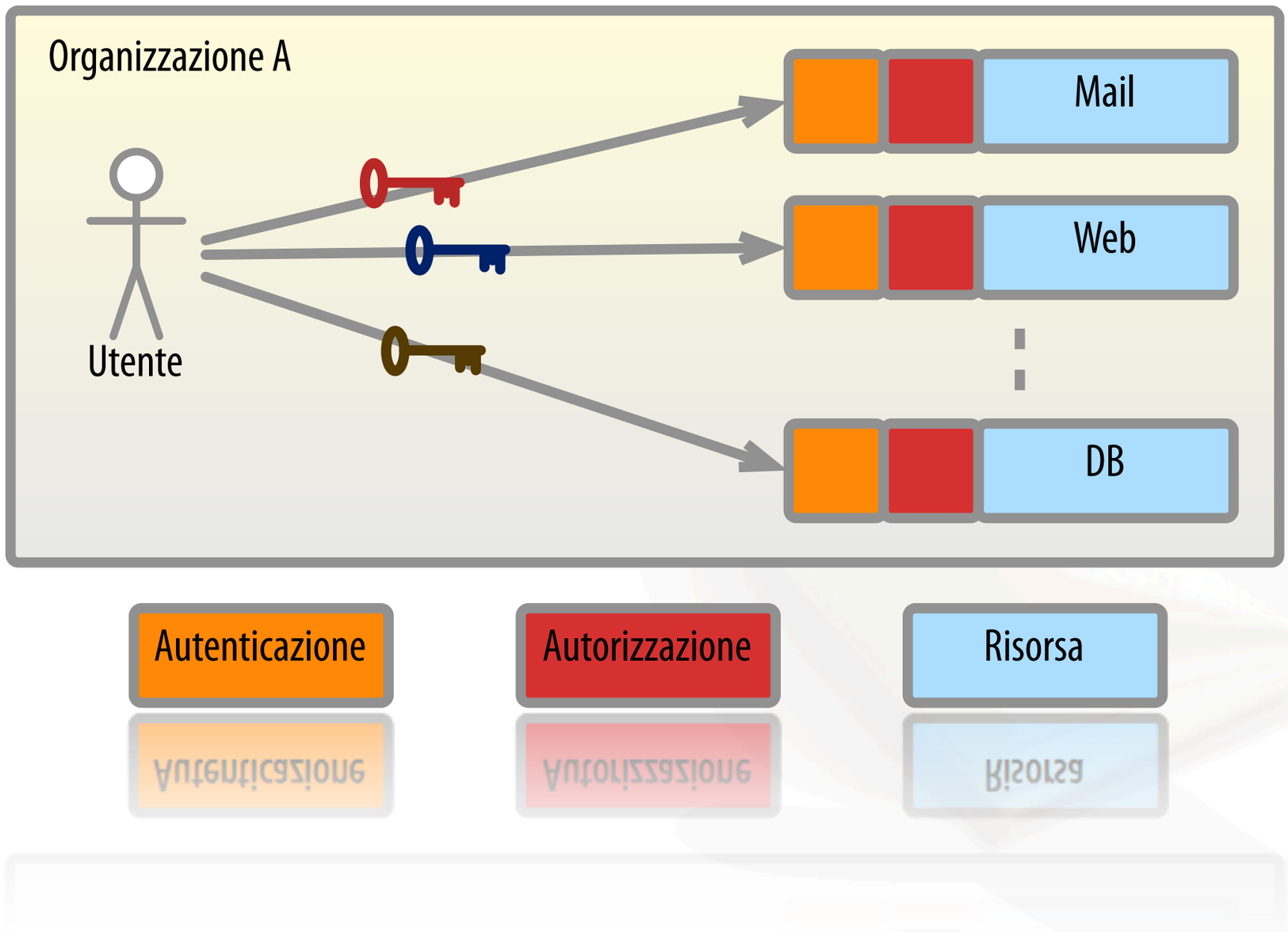
**Stone Age:** Le applicazioni gestiscono le credenziali e le informazioni per ogni utente

**Bronze Age:** Le credenziali sono centralizzate (es. Kerberos, LDAP) ma le applicazioni gestiscono tutte le informazioni sulle identità degli utenti

**Iron Age:** Le credenziali ed un nucleo di informazioni sugli utenti sono centralizzate (Identity Management System); le applicazioni gestiscono solo "app-specific user data"

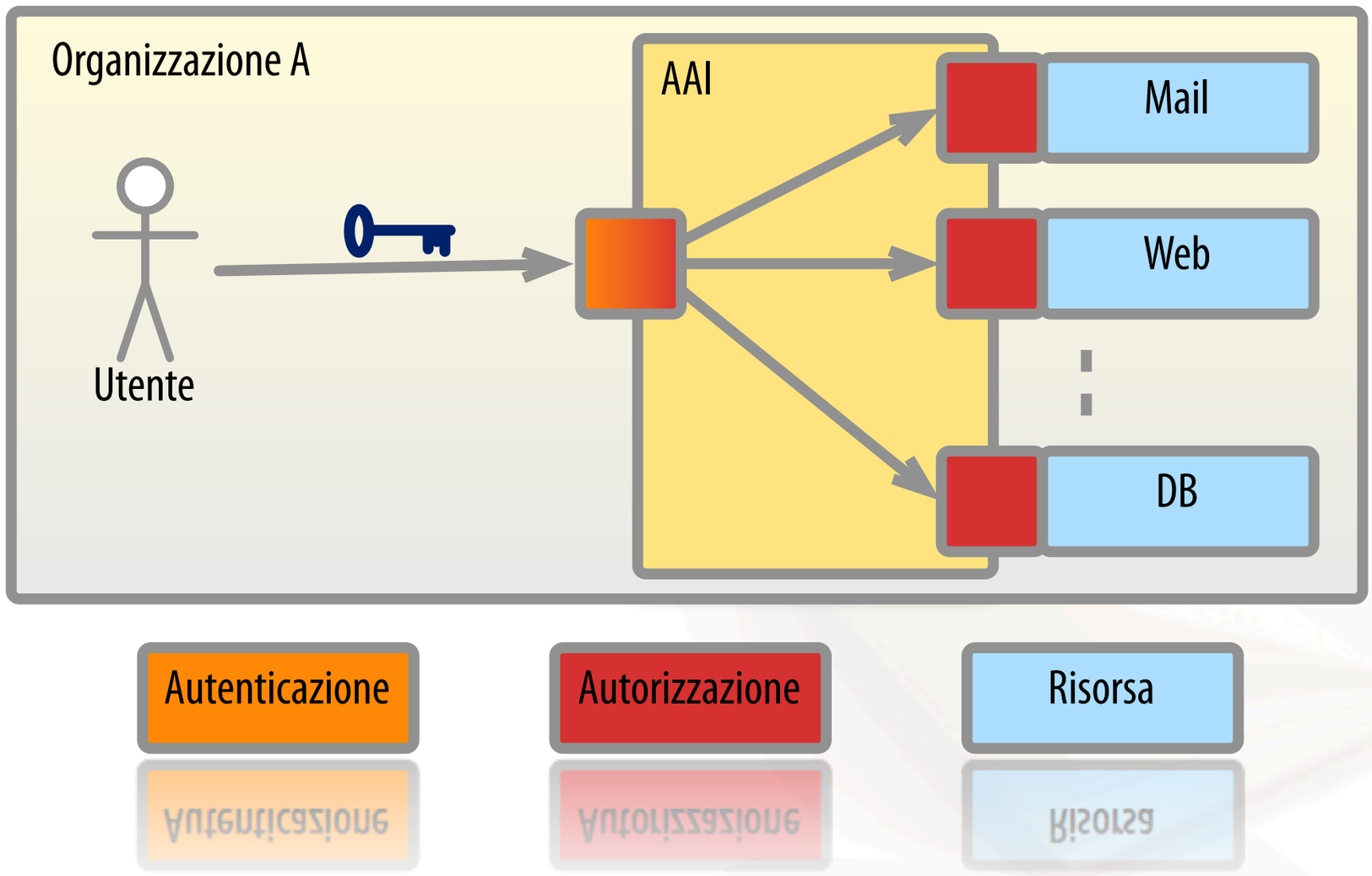
# IdM: Authentication and Authorisation Infrastructure

*senza Infrastruttura di AA*



# IdM: Authentication and Authorisation Infrastructure

*con Infrastruttura di AA*



# Identity Management (IdM): vantaggi

Riduzione del numero di credenziali lato utente

Single Sign-On

Eliminazione delle incoerenze relative ai dati dell'utente

Autorizzazione semplificata, basata su attributi/ruoli (ABAC/RBAC)

# Identity Management (IdM): vantaggi

Profili utente gestiti *esclusivamente* dall'"Ufficio del personale"

È possibile soddisfare (più facilmente) alcune misure richieste dal DL 196/'03

- scadenza password (art. 5, all.B)
- assegnazione univoca degli userid (art. 6, all. B)
- disabilitazione account per inutilizzo (art. 7, all. B)

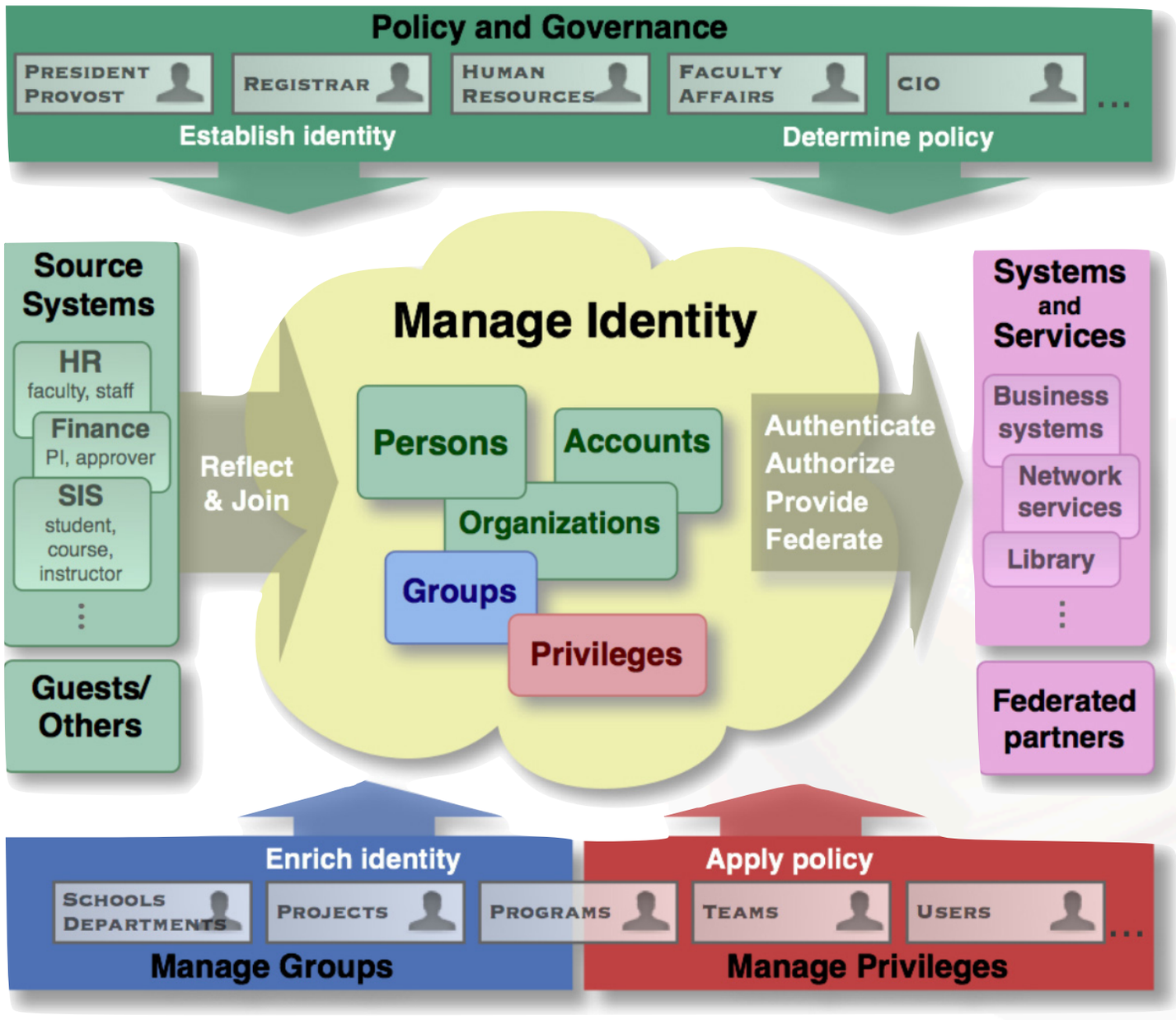
# Identity Management (IdM): vantaggi

Le informazioni sugli utenti centralizzate ma automaticamente replicate

Il personale tecnico cura gli aspetti tecnologici piuttosto che amministrativi

La gestione delle autorizzazioni è effettuata dal responsabile del servizio o particolare trattamento dati

# Identity and Access Management




http://www.internet2.edu/pubs/20070703-IS-MW.pdf

# Identity Management System

Deve supportare:

- la definizione e la rappresentazione di attributi e relativi valori
- la gestione di subset di attributi (identità parziali)
- la possibilità di decidere quali attributi e valori rivelare agli altri
- la pseudonomizzazione

# Strumenti per gli IMS

LDAP:

- Standard (organizzazione dei dati)
- Cross-platform
- Ampiamente supportato dalle applicazioni
- Ottimizzato per le operazioni di ricerca e lettura
- Meccanismi di replica built-in
- Sofisticati meccanismi controllo accessi (ACL)

# Strumenti per gli IMS

## Kerberos

- protocollo per l'autenticazione ed il SSO

## CAS (Central Authentication Service)

- protocollo (e sw) per l'autenticazione ed il SSO sul web intra-organizzazione

*ed altri ancora...*

# ...e se l'utente è esterno all'organizzazione che offre il servizio?

# Problematiche

## Per l'utente:

- nessun controllo sulla riservatezza delle informazioni fornite
- uso di password diverse per servizi diversi (per evitare l'uso inappropriato delle stesse su altri servizi)

## Per il fornitore del servizio

- scarse informazioni sull'utente
- possibilità di variazione dei dati a sua insaputa

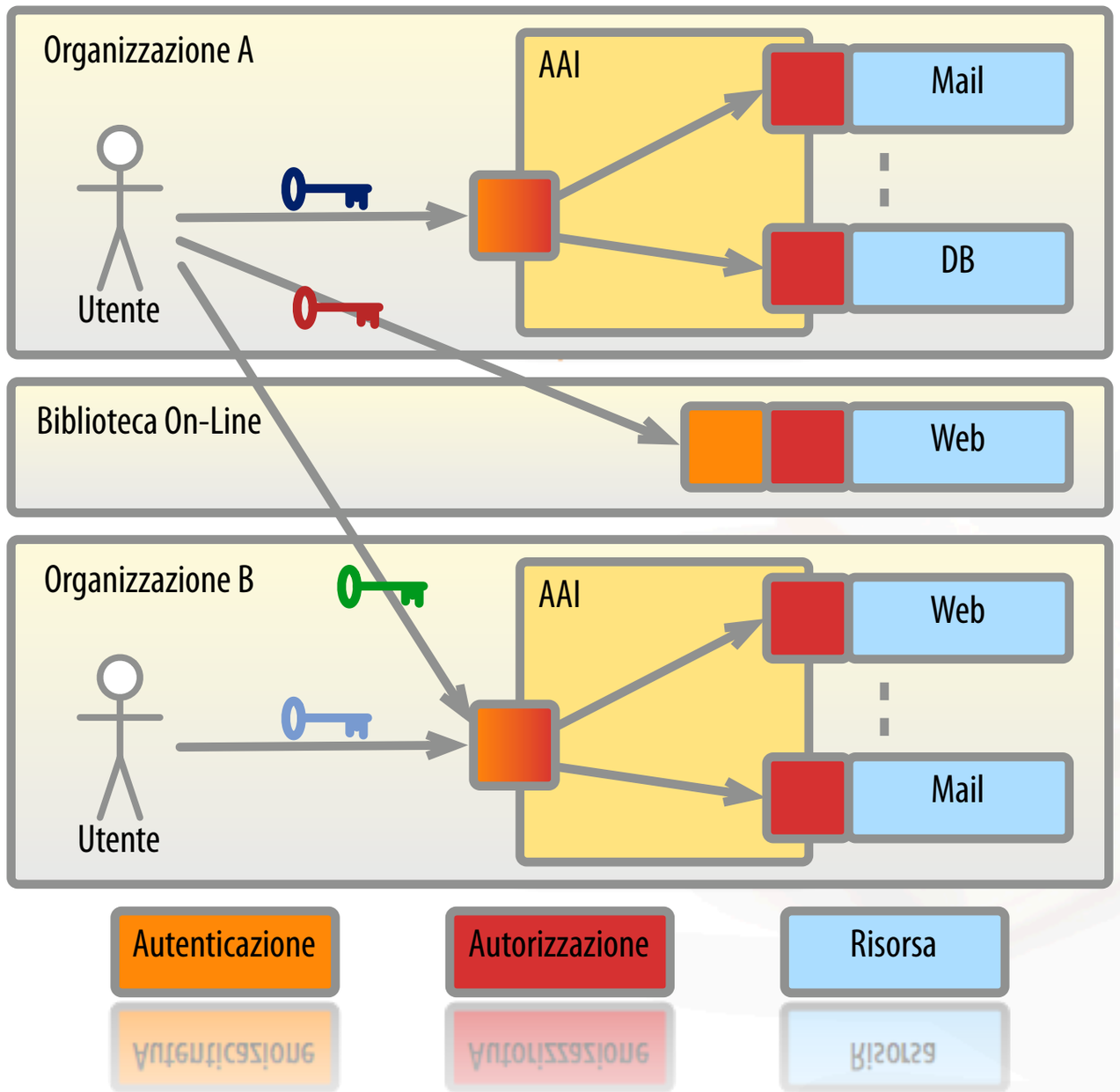
# Problematiche

## Per l'organizzazione:

- aggiungere un utente esterno comporta la creazione di un nuovo account sul proprio IMS con il rischio di assegnargli più diritti di quelli voluti

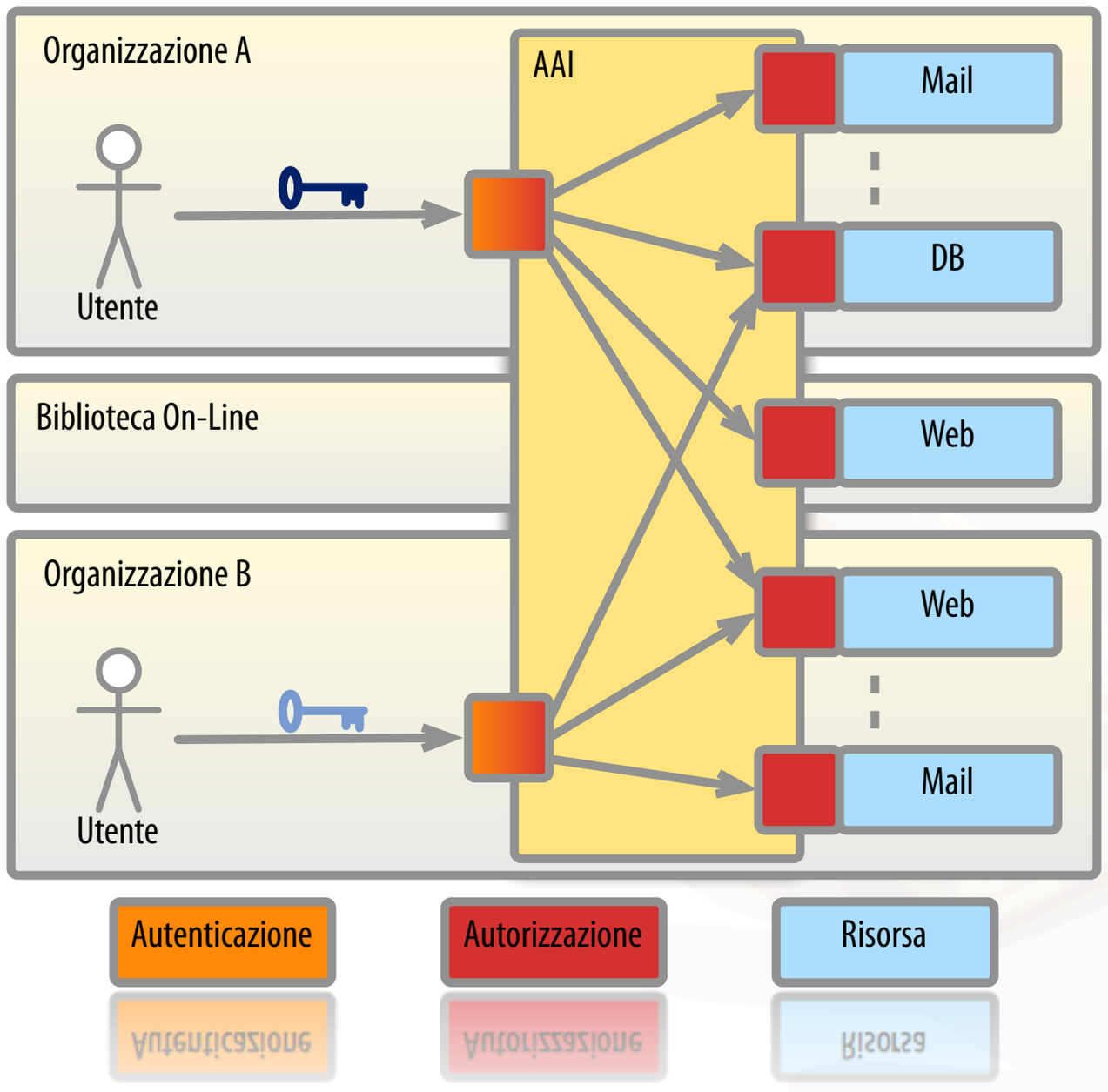
# IdM intra-organizzazione

*senza AAI Federata*



# IdM inter-organizzazione

*con AAI Federata*



# AAI federata

Autenticazione locale, presso l'organizzazione di appartenenza

Accesso remoto a servizi di altre organizzazioni senza ulteriori autenticazioni

# AAI federata: SAML

*Security Assertion Markup Language*

- framework XML-based
- sviluppato da OASIS
- standard per lo scambio di informazioni *(assertion)* relative all'identità e all'affiliazione di un utente nei confronti di un'organizzazione

# AAI federata: SAML

Sviluppato per rendere sicuri i servizi web-based

Consente Web-SSO

Oltre al *"tradizionale"* Service Provider (**SP**) introduce i concetti di:

- Identity Provider (**IdP**)
- Metadati: "coordinate" dei partecipanti

# AAI federata: privacy

L'obiettivo principale nella gestione federata delle identità *(Federated IdM)* è la protezione **attiva** delle informazioni sugli utenti:

- proteggere le credenziali degli utenti: solo l'IdP le può trattare
- proteggere le informazioni sugli utenti (identificativi inclusi) tramite il rilascio di subset predefiniti di informazioni, diversi per ogni SP

# AAI federata: vantaggi

Per l'**utente finale**

- utilizzo di credenziali uniche e SSO
- controllo sulle informazioni diffuse
- rilascio dati personali solo se necessario e solo per i servizio voluti
- accesso alle risorse on-line dall'esterno della propria organizzazione
- più servizi

# AAI federata: vantaggi

Per l'**organizzazione**:

- controllo sul processo di autenticazione e autorizzazione
- riduzione nel carico di lavoro per la gestione degli account
- isolamento rispetto a possibili compromissioni del servizio
- riduzione della area da proteggere *(solo l'IdP può accedere ai dati degli utenti)*
- più semplice introduzione di nuovi servizi

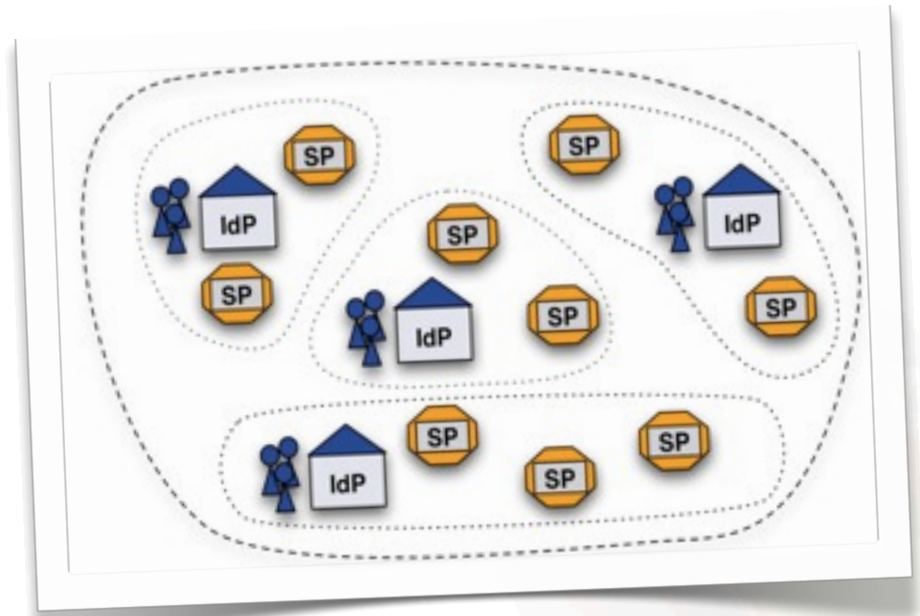
# AAI federata: vantaggi

Per il **fornitore del servizio**:

- riduzione oneri gestione credenziali
- utilizzo di informazioni aggiornate e affidabili *(maggiore sicurezza)*
- autorizzazione concessa in base al ruolo (RBAC) nell'organizzazione di provenienza
- più utenti per il servizio

# Le federazioni per l'AA

Sono un gruppo di IdP e SP che condividono regole tecniche e procedure su cui si costruiscono relazioni di fiducia




Ogni partecipante può appartenere a più federazioni

Sono possibili inter-federazioni

# Le federazioni per l'AA: quali regole

## Tecniche

protocolli, applicativi, attributi utenti, certificati accettati

## Legali

accordi fra i partecipanti, regole operative e sull'accreditamento degli utenti

# Ruolo della Federazione

Definisce le regole e gli accordi

Gestisce e pubblica i metadati

Gestisce il Discovery Service

Rende disponibile una federazione di test

# Shibboleth

# Shibboleth

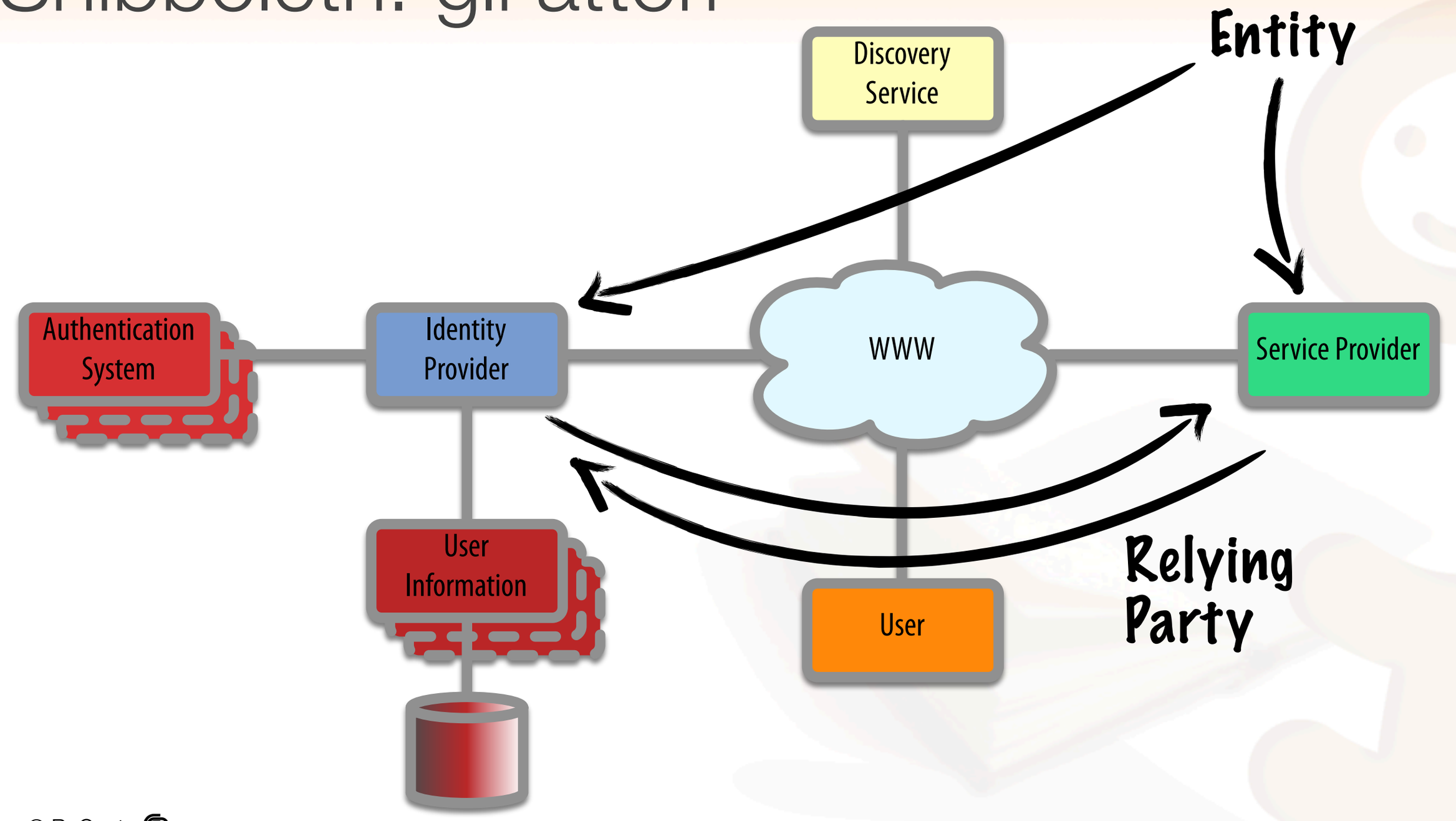

"*The Shibboleth System is a standards based, open source software package for web single sign-on across or within organizational boundaries*" *(shibboleth.internet2.edu)*.

Implementazione di SAML

Progetto ufficiale di Internet2

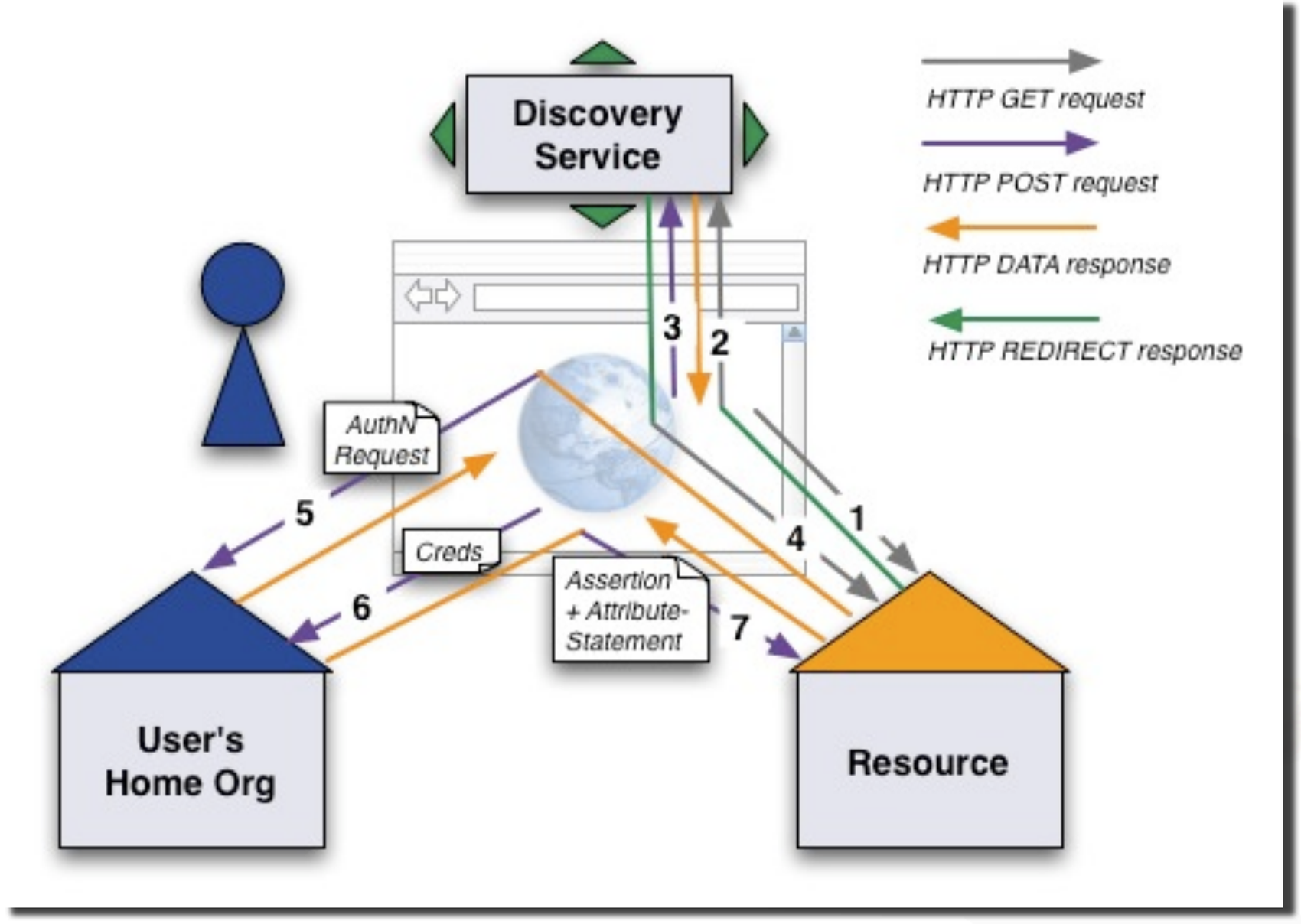
Rilasciato con Apache Software License

# Shibboleth: gli attori

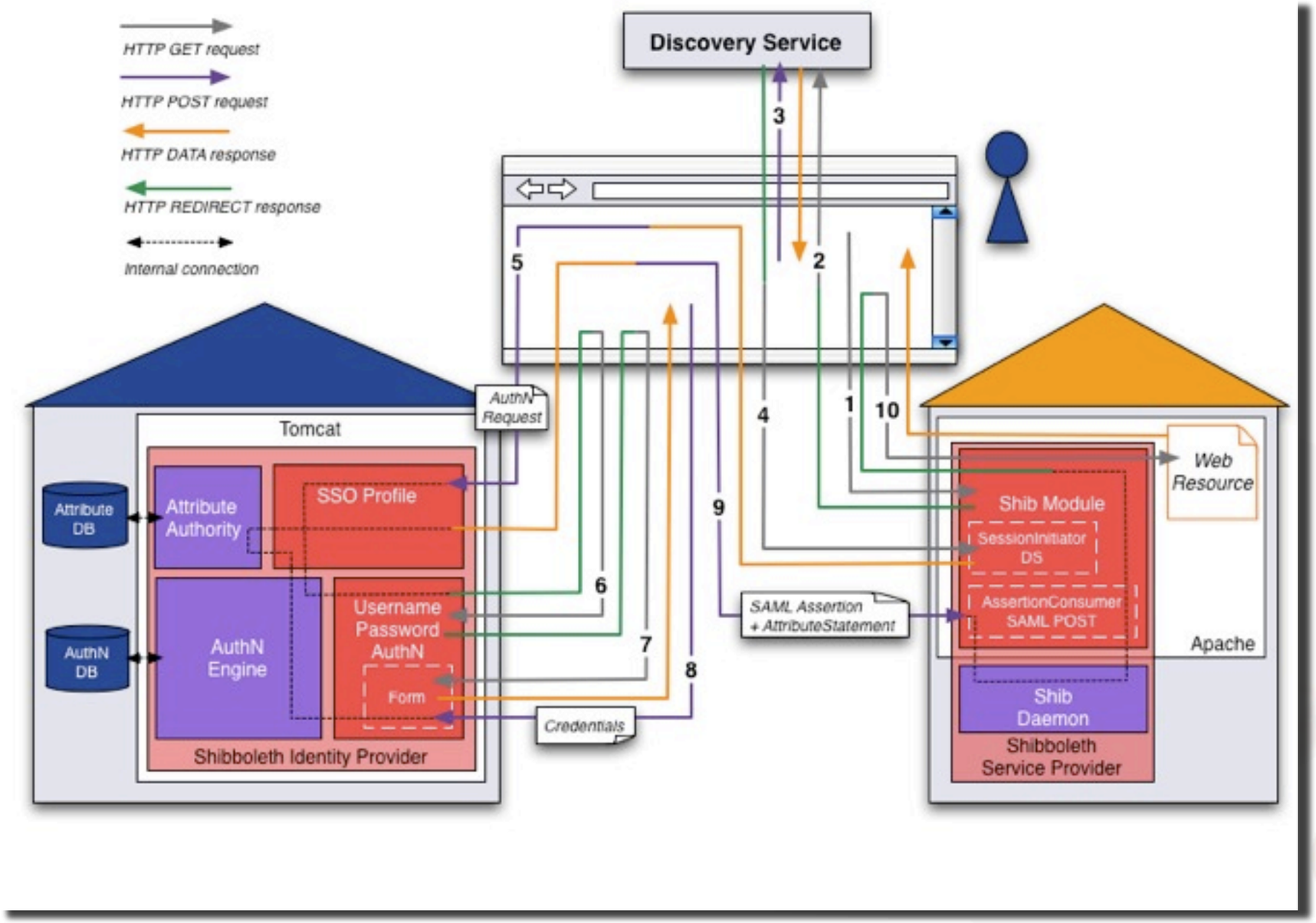



© R. Conte

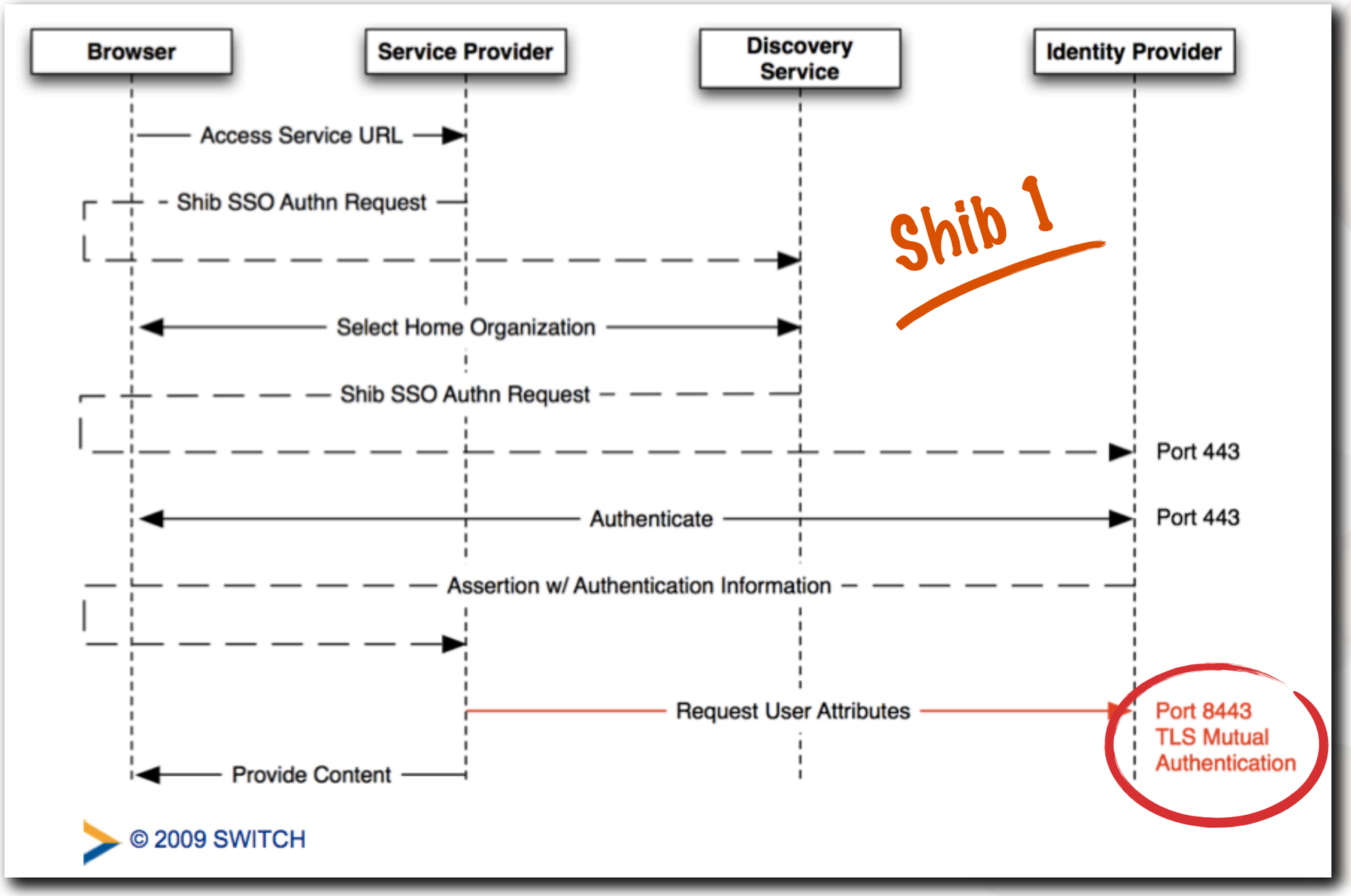
# Le relazioni nella Federazione



# Le relazioni nella Federazione

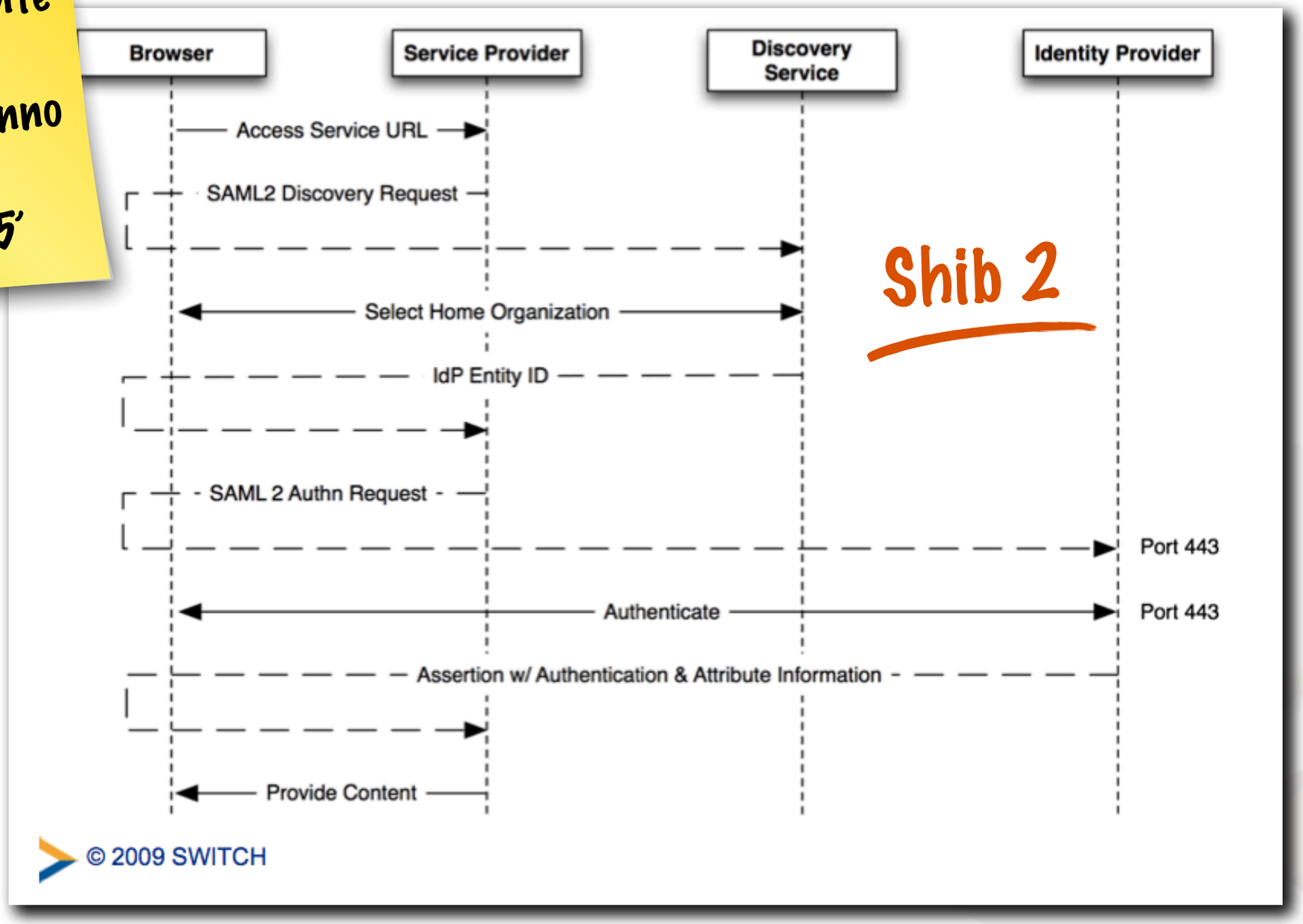


# Shibboleth Communication Flow



# 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 certificato (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 richiede riavvio del server dopo le modifiche

5 livelli possibili per i logger definiti:

- TRACE, DEBUG, INFO, WARN, ERROR
  - *(non usare in produzione TRACE, DEBUG!)*

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

# Logging

## Logger presenti:

edu.internet2.middleware.shibboleth

edu.internet2.middleware.shibboleth.common.attribute [solo in test]

org.opensaml

edu.vt.middleware.ldap

PROTOCOL_MESSAGE

Es.

```
<!--
<logger name="edu.internet2.middleware.shibboleth.common.attribute">
        <level value="DEBUG" />
</logger>
-->
```

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

# Logging

**`SHIB_HOME/logs`:**

idp-access.log

contiene un record per ogni chiamata all'IdP

idp-audit.log

contiene un record per ogni volta che l'IdP invia dati ad un relying party

idp-process.log

registra le normali operazioni dell'IdP

# Autenticazione

# Autenticazione

Shibboleth 2 offre **meccanismi** basati su

- REMOTE_USER
- username/password su LDAP/Kerberos
- 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: **`handler.xml`**

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

# username/password - LDAP

## Modulo di login LDAP: **`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"
        base="ou=People,dc=example,dc=org"
        serviceCredential="password"
        serviceUser="cn=myUser,dc=example,dc=org"
        ssl="true"
        userField="uid"
        subtreeSearch="true";
};
```

# Pagina per Login

**`PACK_DIR/src/main/webapp/login.jsp`**

Deve essere personalizzata *(vedi Specifiche Tecniche)* tranne per:

- 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: cosa e quando

Un SP parla solo con un IdP noto (i cui dati siano nel file dei MD)

Il file contiene le indicazioni (in XML) su come e dove contattare un "relying party"

Ogni partecipante, per verificare l'identità della controparte e comunicare con esso, utilizza il relativo certificato contenuto nei metadati

È lo strumento con cui si costruiscono le relazioni di fiducia

Non utilizzare altri metodi (verifica CRL ecc.)

# Metadati: contenuto

Certificati

Scope degli IdP *(estensione di Shib, es. ifc.cnr.it)*

lo scope deve corrispondere a quello utilizzato per gli attributi

Posizione (url) e tipologia dei componenti (*binding*) per lo scambio e l'utilizzo delle *assertion* dei partecipanti

Eventuale descrizione testuale dei partecipanti

# Metadati: i certificati

È consentito (e anche consigliato) 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

# 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 ed è fortemente consigliata la verifica della firma

È consigliabile mantenere il file con diritti tali da non consentirne la modifica

# 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

# Configurare i Metadati: 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="/usr/local/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>
```

# Configurare i Metadati: `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>
            /usr/local/idp/credentials/signer-bundle.pem
        </security:Certificate>
    </security:Credential>
</security:TrustEngine>
```

# Attributi: risoluzione e filtraggio

# Attributi

Attributo: una "piccola" informazione riguardo un utente. Ogni attributo ha un unico ID ed ha zero o più valori.

Attributi SAML: attributi rappresentati tramite notazione SAML.

Attributi Shibboleth: strutture dati "*protocol-agnostic*".

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 raccomandato che un IdP fornisca questo attributo poiché esistono SP che ne fanno richiesta

**opzionali**: sebbene non esistano SP in Federazione che ne facciano richiesta esplicita, l'attributo potrebbe risultare utile

# L'insieme degli attributi

Gli attributi sono suddivisi in:

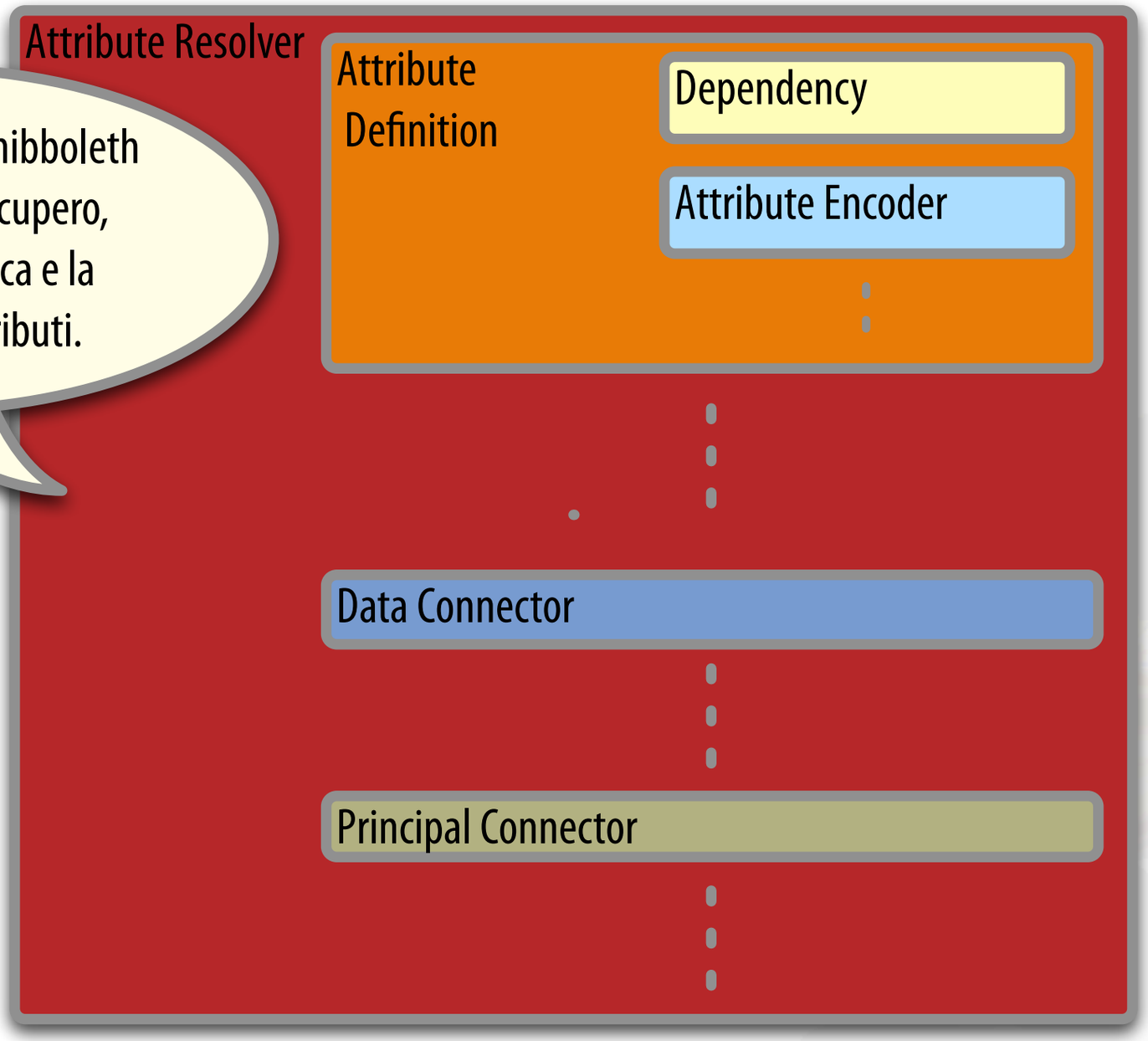
**caratteristiche personali**: sn, givenName, cn, preferredLanguage ecc.

**contatti**: mail, telephoneNumber, mobile ecc.

**autorizzazione e accounting**: eduPersonScopedAffiliation, eduPersonTargetedID, eduPersonPrincipalName, eduPersonEntitlement

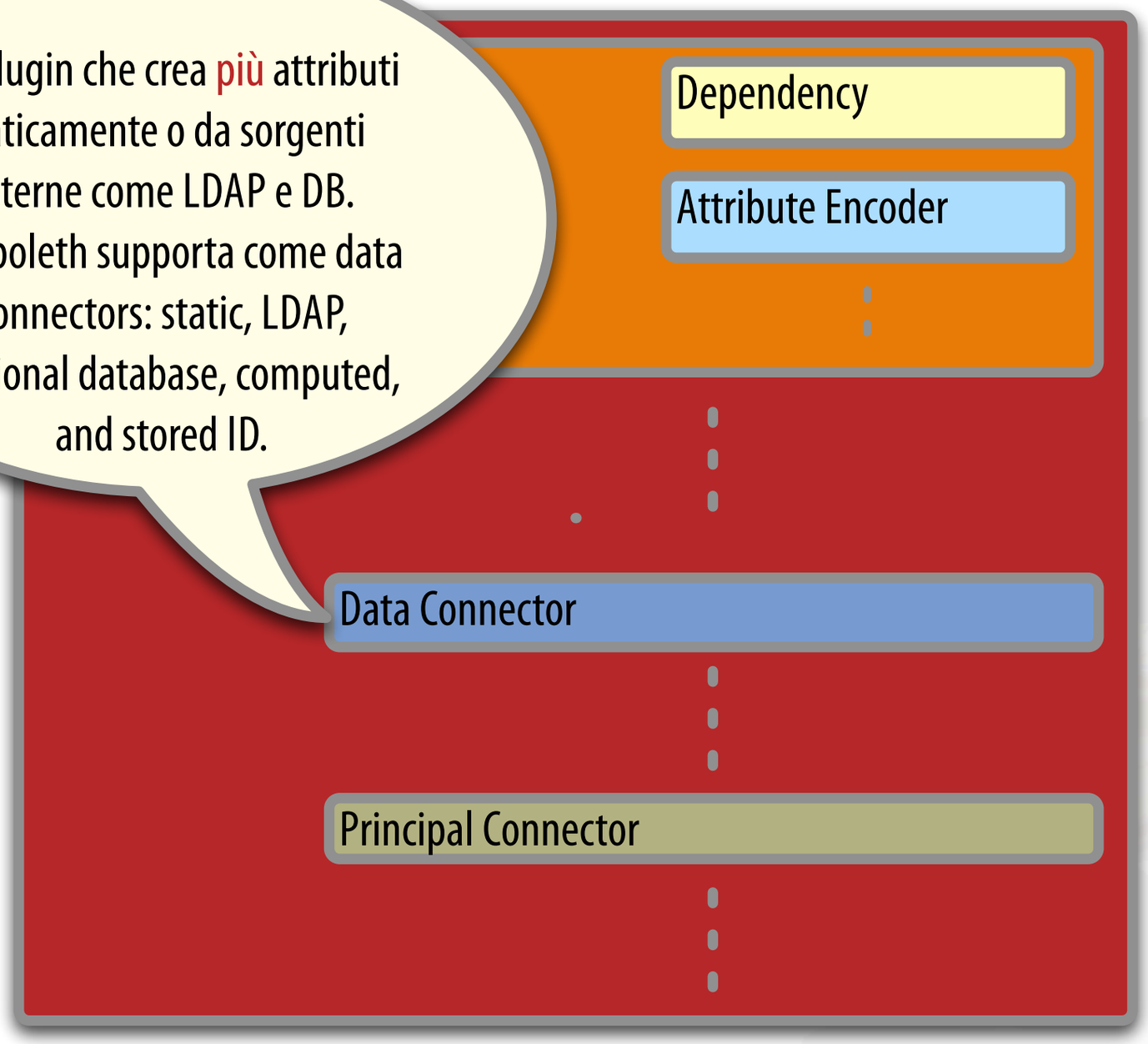
# Configurazione: `attribute-resolver.xml`

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

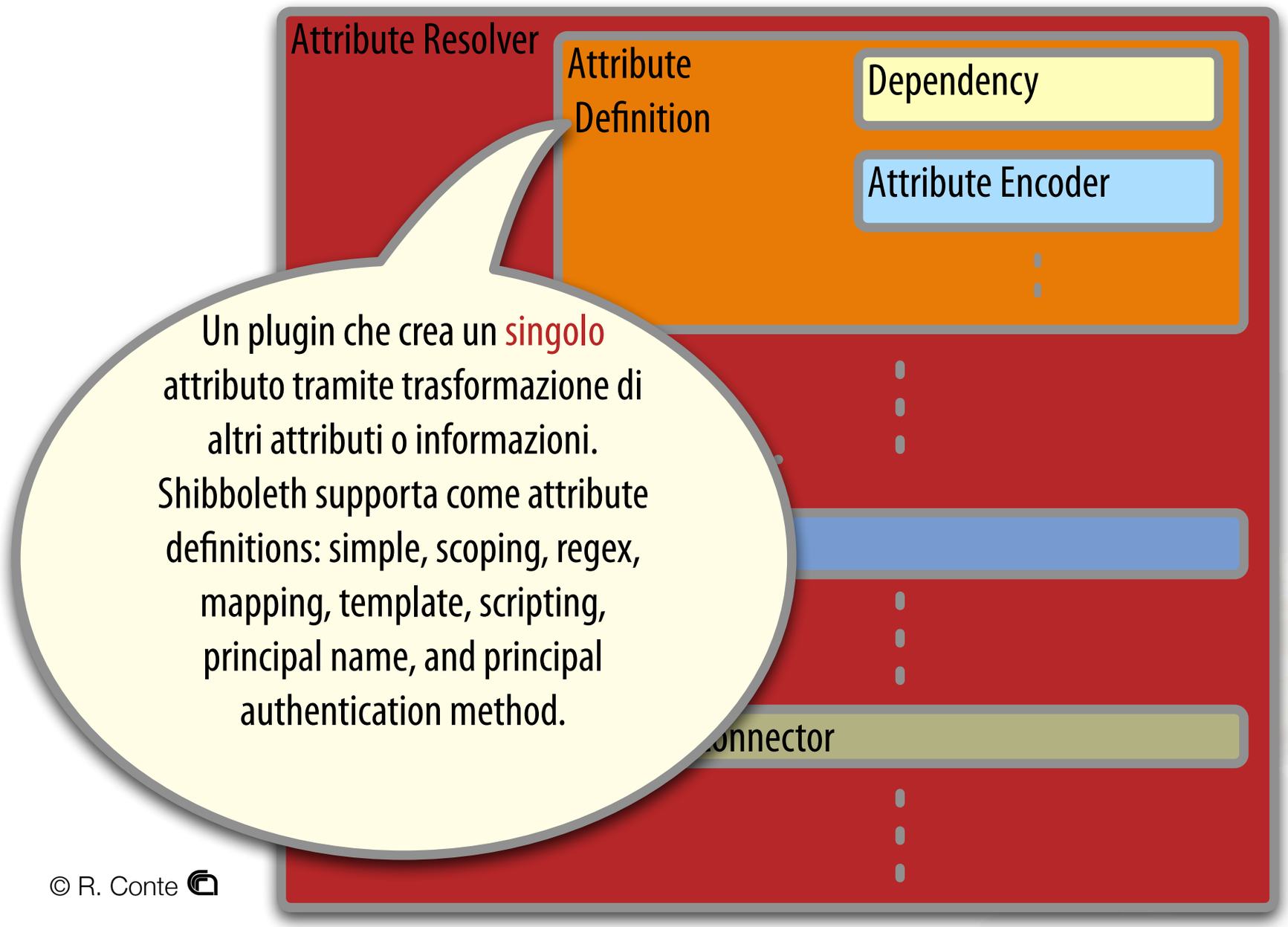


# Configurazione: `attribute-resolver.xml`

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.

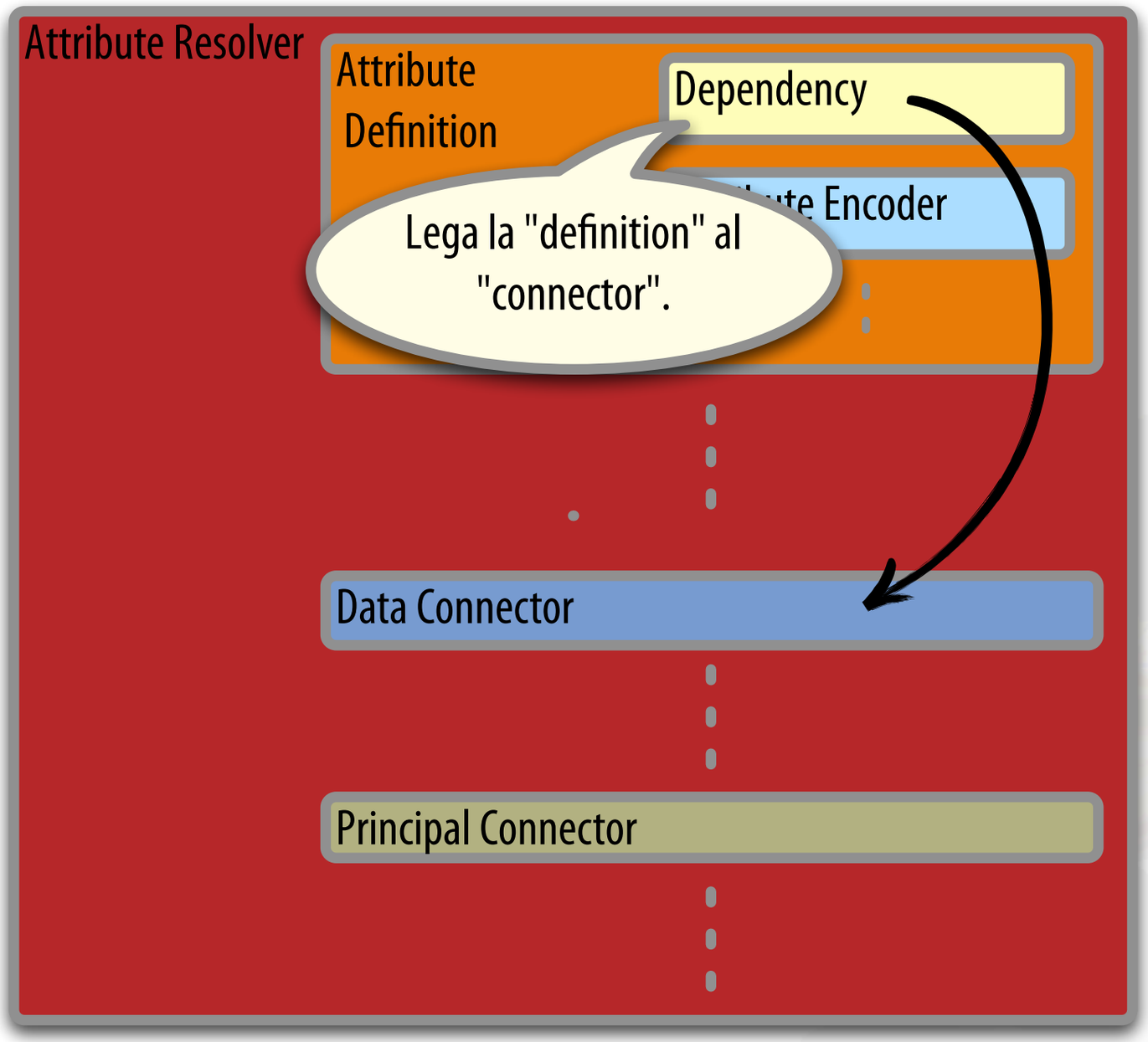


# Configurazione: `attribute-resolver.xml`



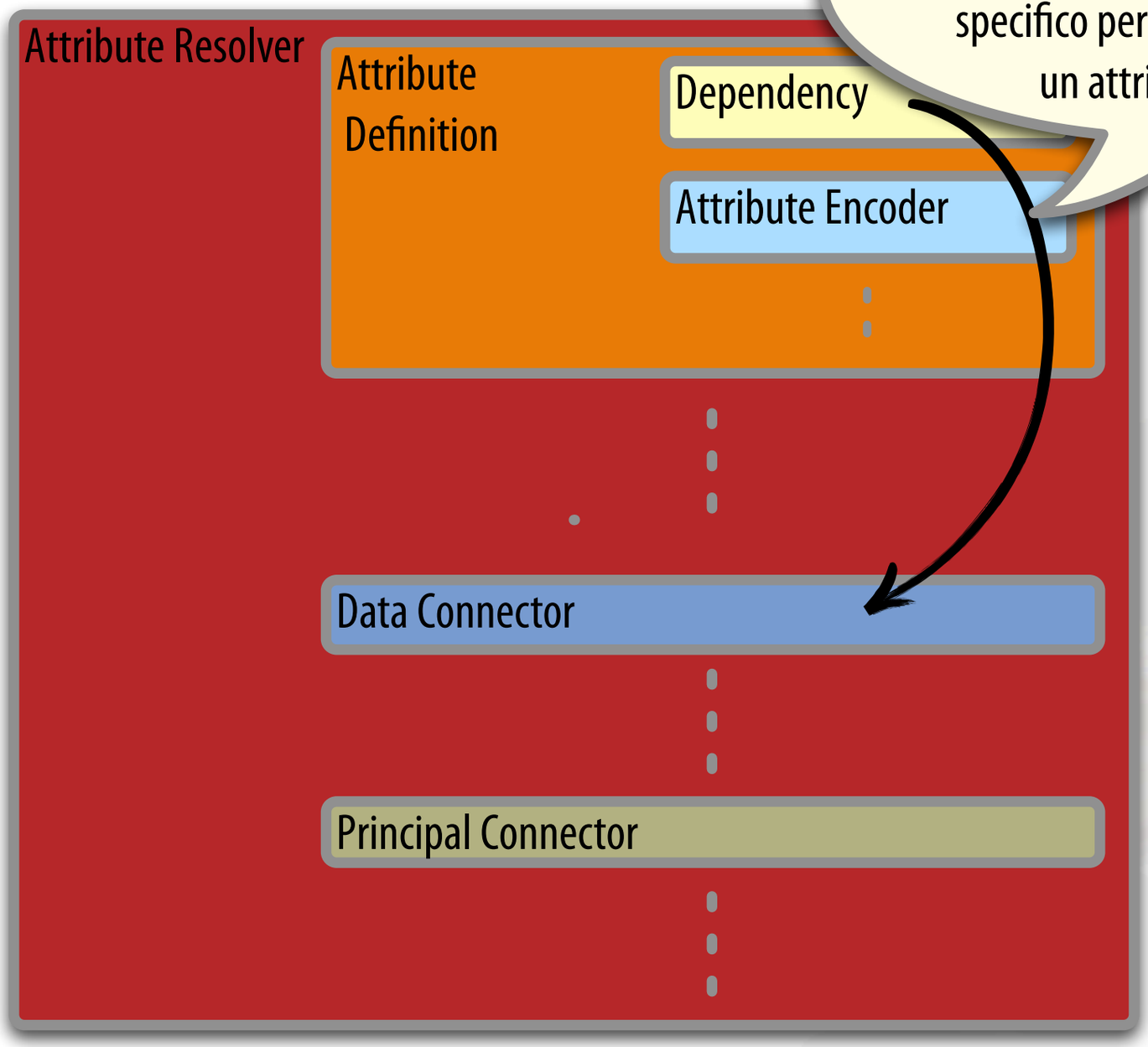

Nota: solo gli attributi che passano per una "definition" escono dal resolver

# Configurazione: `attribute-resolver.xml`




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

# Configurazione: attribute-r…ml




Un plugin per convertire un attributo in un formato specifico per un protocollo, es. un attributo SAML.

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

# 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:enc
        name="urn:mace:dir:attribute-def:eduPersonOr

    <resolver:AttributeEncoder xsi:type="SAML2String"
        xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
        name="urn:oid:1.3.6.1.4.1.5923.1.1.1.2"
        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

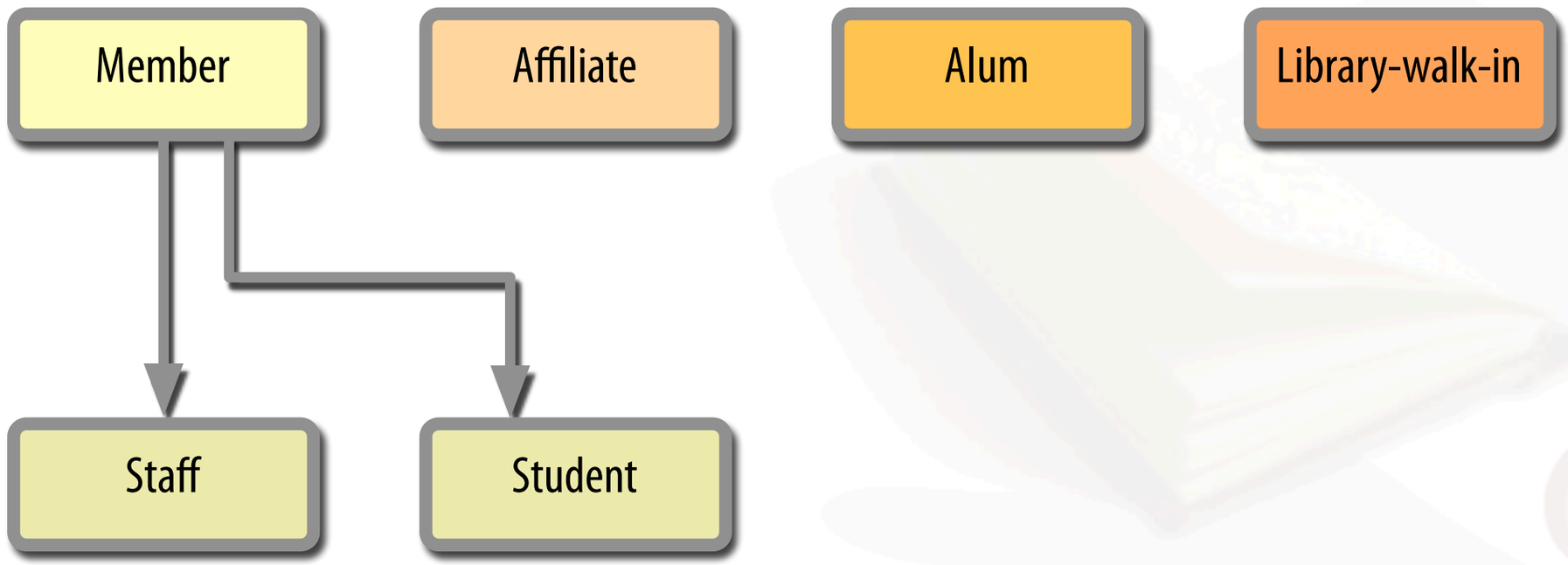
```
<resolver:AttributeDefinition id="cn" xsi:type="Simple"
    xmlns="urn:mace:shibboleth:2.0:resolver:ad"
    sourceAttributeID="cn">

  <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



# 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>PERSONALE IN ORGANICO</SourceValue>
[...]
        <SourceValue>PERSONALE IN FORMAZIONE</SourceValue>
     </ValueMap>
     <ValueMap>
        <ReturnValue>member</ReturnValue>
        <SourceValue>PERSONALE IN ORGANICO</SourceValue>
        <SourceValue>ASSEGNISTA</SourceValue>
        <SourceValue>CONTRATTISTA</SourceValue>
     [...]
     </ValueMap>
     <ValueMap>
        <ReturnValue>student</ReturnValue>
        <SourceValue>PERSONALE IN FORMAZIONE</SourceValue>
     [...]
```

# Un attributo "scoped": eduPersonScopedAffiliation

Espone la relazione fra utente ed Organizzazione nel formato affiliation@organisation

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>
```

ePSA dipende da un altro attributo: ePA

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

È l’IdP che autorizza l’accesso alla risorsa

Discriminatory Access Control (DAC) vs Attribute Based Access Control (ABAC)

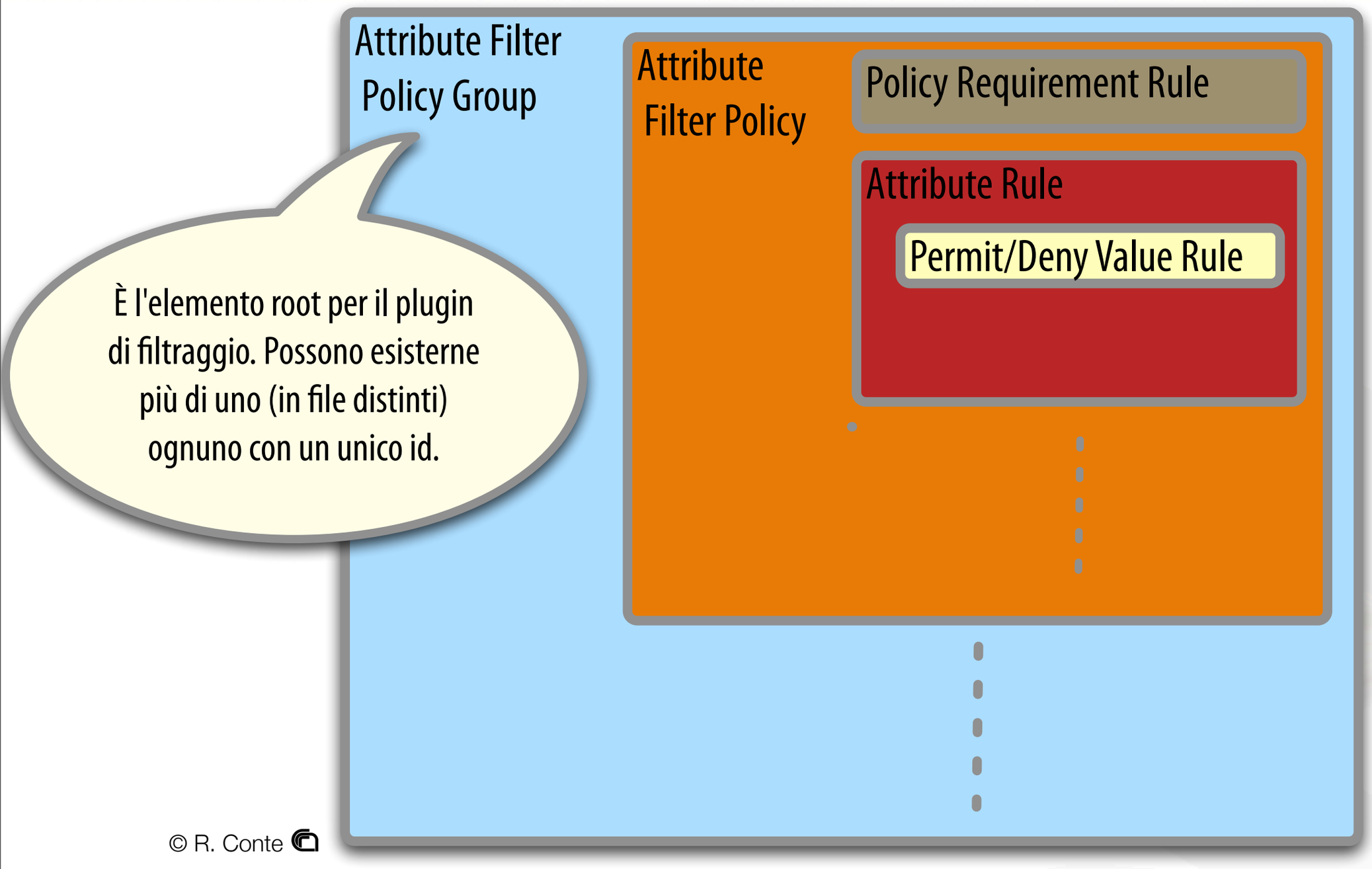
Equivale a definire dei gruppi

# Attribute Filter Policy

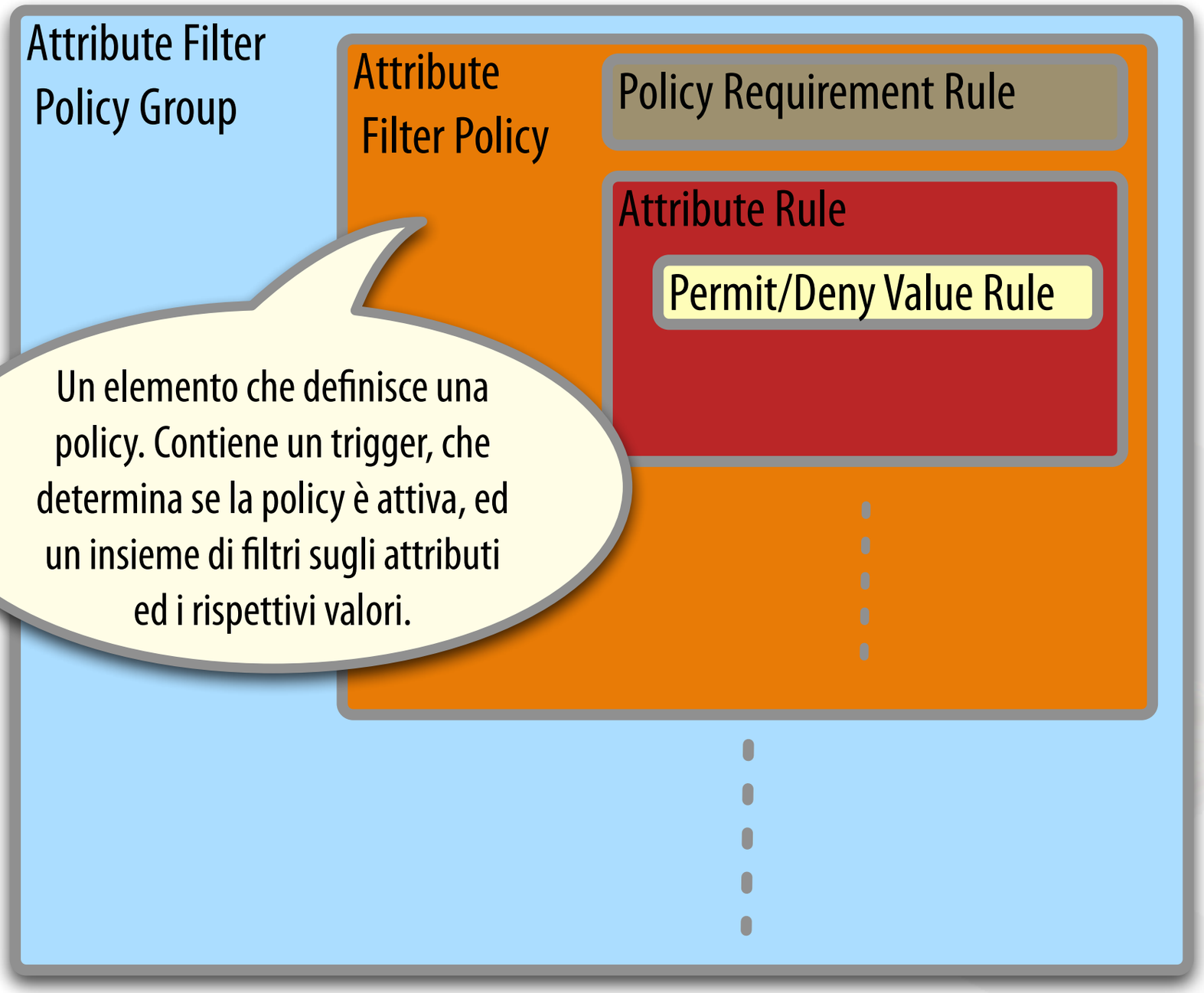
Insieme di regole per SP/attributo (o gruppi)

È possibile utilizzare più file configurando service.xml

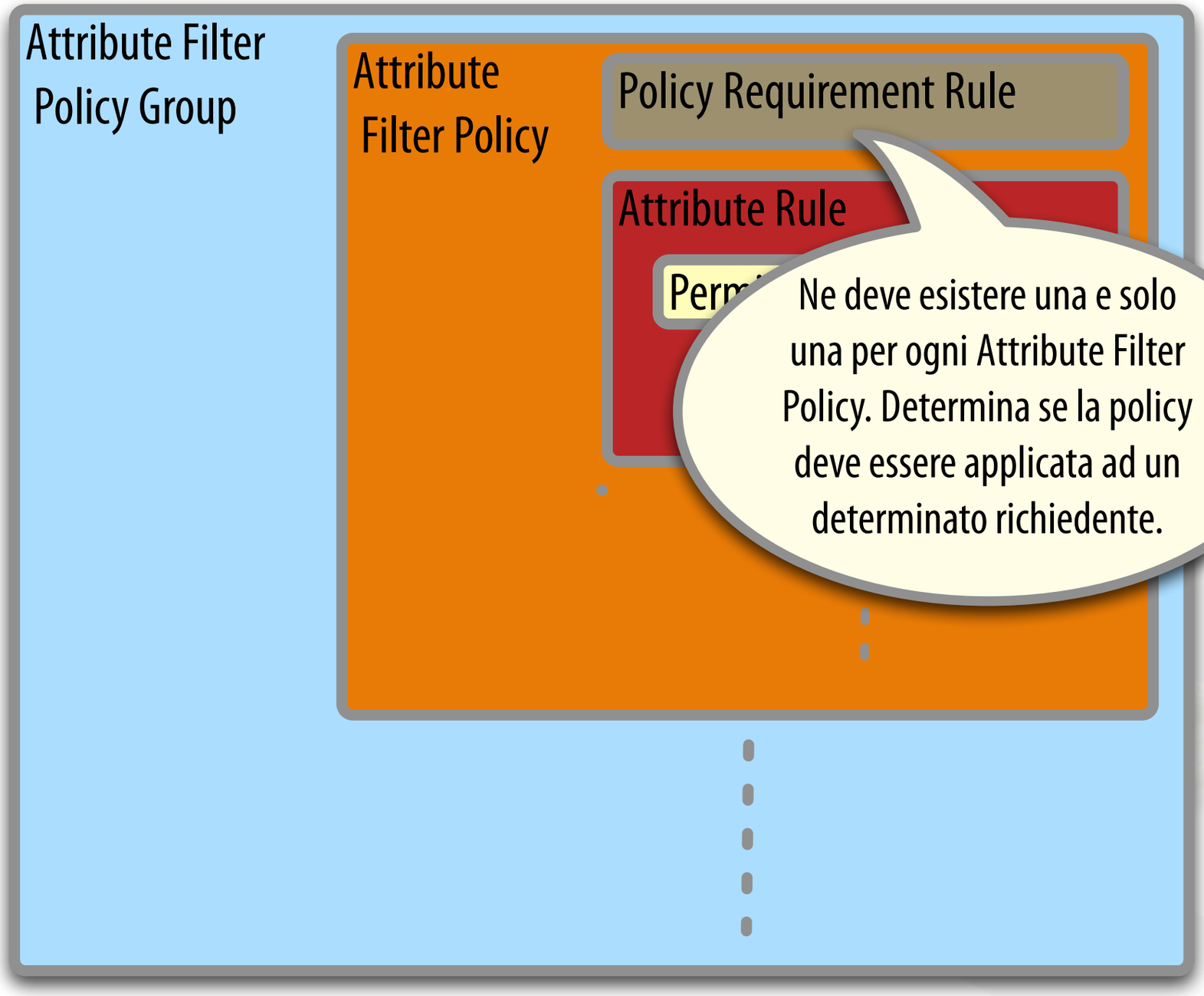
# Configurazione: `attribute-filter.xml`



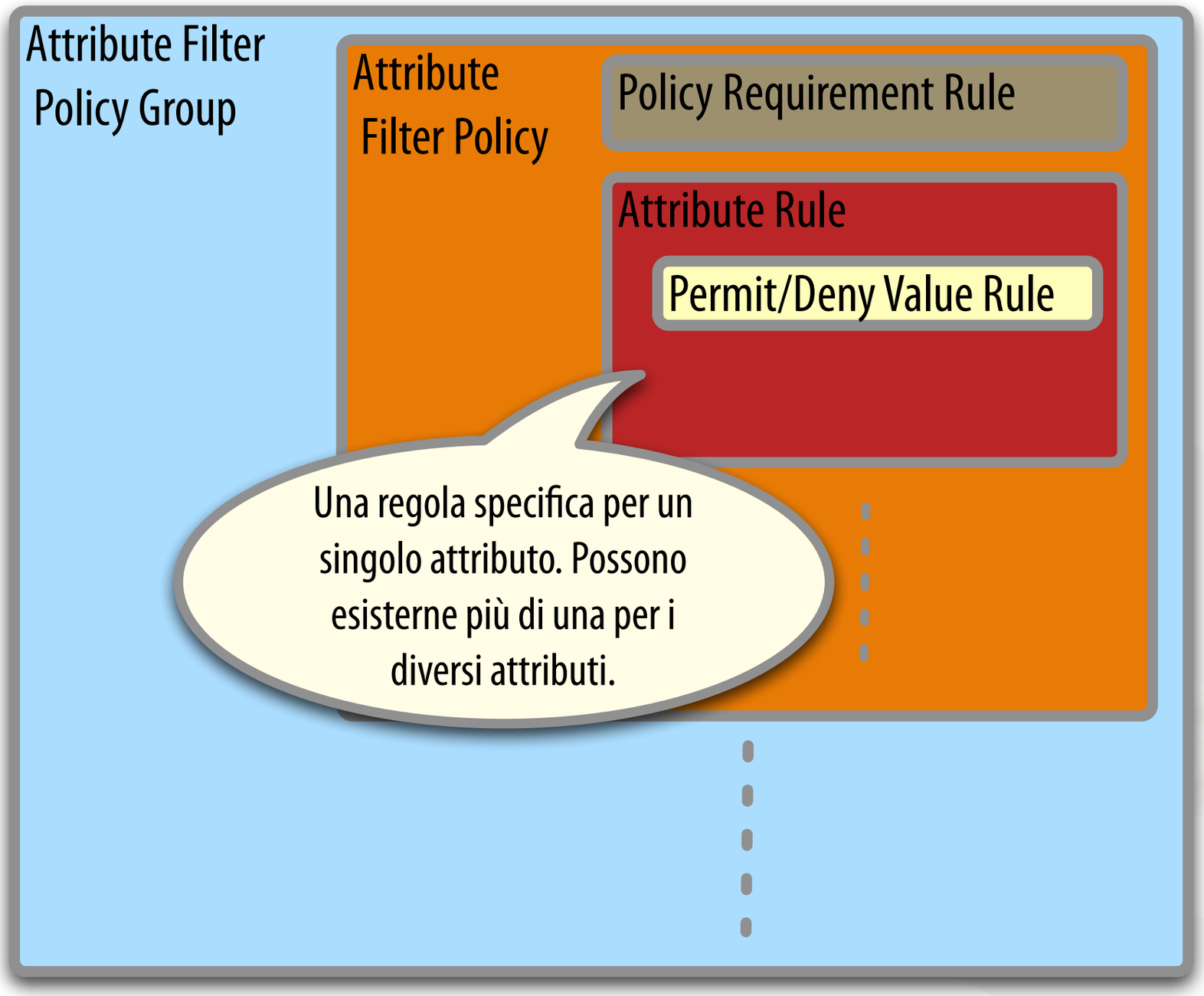
# Configurazione: `attribute-filter.xml`



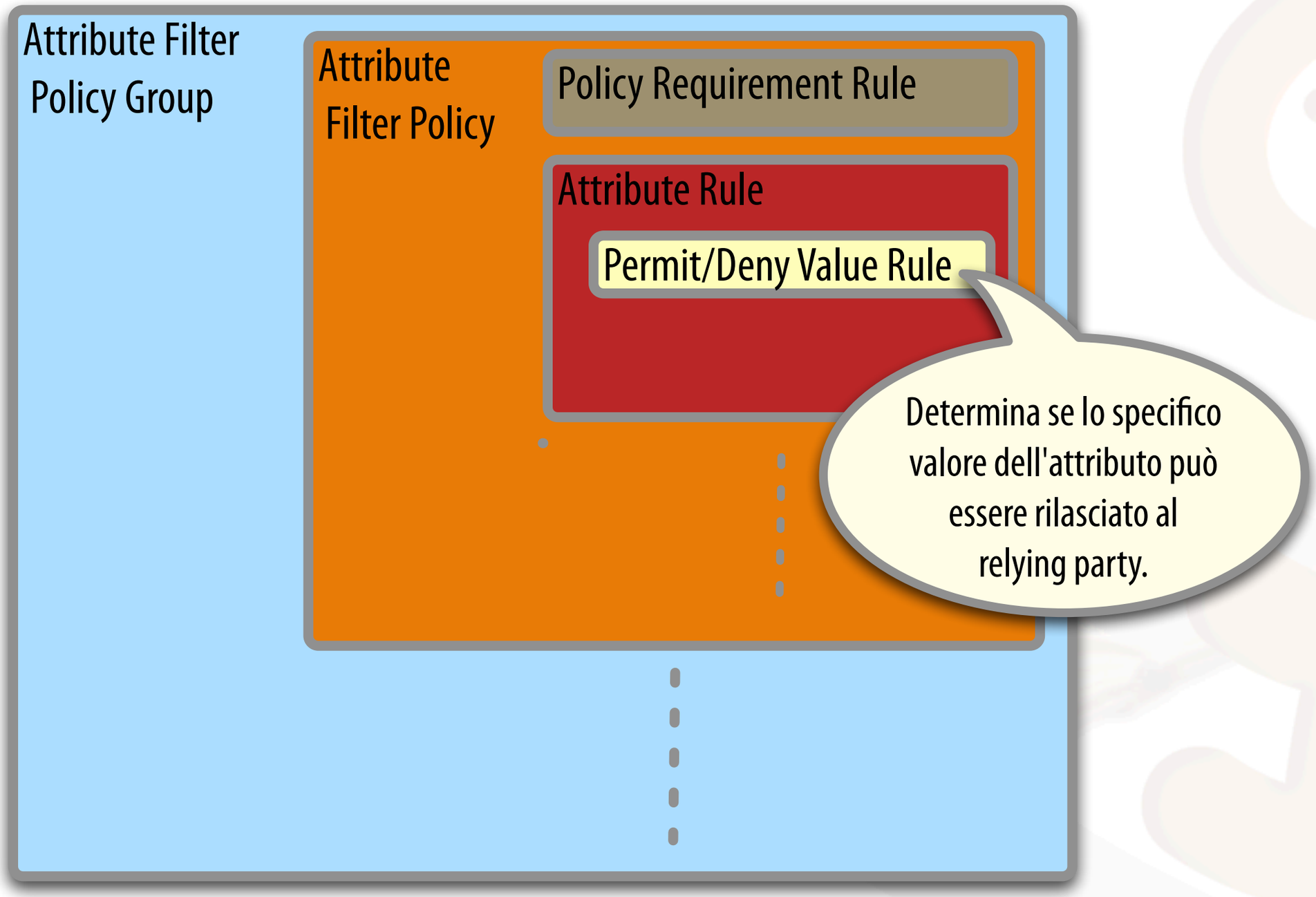
# Configurazione: `attribute-filter.xml`



# 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>
```

# Grazie!