Bridging the gap between user attributes and service policies with attribute mapping

Davide Cerri and Francesco Corcoglioniti

11th IEEE Conference on Commerce and Enterprise Computing
Vienna, 20-23 July 2009
Outline

Scenario and motivation

• A world of interactions
• A matter of point of view

The attribute mapping proposal

• Concept and properties
• Attribute mapping steps
• Example
• Implementation

Outlook

• Potential extensions and future work
A world of interactions

• The Web hugely multiplied the interactions between different actors
  – People, companies, public authorities...
• Many independent parties exchange information and provide services
  – Many users, many services, many providers
• Many interactions, often sporadic
  – Interacting parties are often strangers
  – No central authority: it’s an open world!

• So... who’s who?
We need attributes!

• Services need **information about users**...
  – Is this user entitled to use this functionality?
• ...but they cannot always rely on a-priori knowledge
  – The world is large and open!

• Services need to rely on **third parties that vouch for user information** (acting as trusted authorities)
  – Such authorities already exist in the “real world”: governments, universities, associations, companies...
• **Attribute-based** rather than identity-based paradigm
  – *Tell me something about you, not just your name*...
  – ABAC models
It’s a matter of point of view

• But, again, the world is large and open...
  – Different authorities may certify the same or similar information in different ways

• Attributes certified by external parties tend to be general purpose
  – External authorities exist independently of specific services
  – General purpose attributes tend to be close to the user point of view

• Not necessarily coincide with what is relevant for the service policy
  – Service point of view can be different
It’s a matter of point of view

A few examples...

<table>
<thead>
<tr>
<th>User point of view</th>
<th>Service point of view</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACME employee</td>
<td>Partner company employee</td>
</tr>
<tr>
<td>Cardiologist at “Niguarda” hospital in Milan</td>
<td>Non-local doctor</td>
</tr>
<tr>
<td>Italian aged 25</td>
<td>Qualified to buy alcohol</td>
</tr>
</tbody>
</table>

Some external knowledge may be involved (e.g., legal age to buy alcohol in different countries in 3rd example)
Service policy vs. user point of view

• The input user attributes tend to reflect the user point of view

• Writing the service access policy w.r.t. the user point of view may not be the best option
  – Not straightforward to write and understand
  – The policy needs to deal with concepts which are not the most appropriate for the service needs
  – The policy can be unnecessarily long, complex, prone to errors and difficult to maintain
Let’s decouple!

- Our proposal: **Decoupling the user point of view (input attributes) and the service point of view (service policy)**

- The service access control policy can focus on the concepts which are relevant for the service logic

- A separate **attribute mapping policy** establishes the bridge
The attribute mapping process acts before the regular access control logic of the service. It is a sort of adapter, and does not substitute or interfere with the access control logic of the service. It is governed by the attribute mapping policy.
Our attribute mapping process is monotonic: if the user submits a superset of some set of user attributes, he will get a superset of the corresponding set of mapped attributes.

This preserves a monotonicity property of the “downstream” access control policy:
- If this is the case, additional disclosed attributes can only result in additional privileges.
The attribute mapping process is conceptually organised in **three steps**:
- UP – UP
- UP – SM
- SM – SM

Attributes are **name-value pairs** (value is optional)
- SAML standard
- Namespaces disambiguate homonym attributes by different authorities
The UP – UP step uses a **user attribute taxonomy** to derive new attributes “on the user side”

The user attribute taxonomy allows to **model the input of the mapping process**
- External **existing knowledge** can also be imported and reused
  - e.g. standard classifications or taxonomies published by attribute authorities

This step is used to simplify the input to subsequent steps, e.g. **exploiting generalisations**
- E.g. specify “doctor” as a generalisation of “cardiologist”
- Similar attributes from different authorities can be reduced to the same concept
The UP – SM step maps from user provided to mapped attributes by means of **mapping rules**.

This step is the core of the mapping process.

Mapping rules are first-order definite Horn clauses:

- **body** $\Rightarrow$ **head**: $(\text{atom} \land \text{atom} \ldots \land \text{atom}) \Rightarrow \text{atom}$
- The **body** states a set of conditions over *user provided attributes*.
  - Existence, value, and relational predicates are supported.
- The **head** states the *mapped attributes* that are derived if all the conditions in the body are satisfied.
  - Existence and optionally value.
- All variables in the head must be present also in the body.
SM – SM step

- The SM – SM step uses a mapped attribute taxonomy to derive new attributes “on the service side”

- This step is similar to the UP – UP step
  - It allows to *model the output* of the mapping process

- Generalisation relationships between SM attributes can be specified here
  - No need to explicitly consider them in mapping rules
Recognised drivers have held a driving licence for at least one year

\[
\text{ittr:licence} \land \text{ittr:issueDate} : ?d \land (\text{daysBetween} (?d, \text{today}()) > 365) \Rightarrow \text{srv:driver}
\]

Expert drivers are at least 25 years old and have held a driving licence for at least 5 years

\[
\text{ittr:licence} \land \text{ittr:issueDate} : ?d \land \text{ittr:age} : ?a \land (?a > 25) \land (\text{daysBetween} (?d, \text{today}()) > 5 \times 365) \Rightarrow \text{srv:expertDriver}
\]

Partner employees are gold users

\[
\text{:partnerEmployee} \Rightarrow \text{srv:goldUser}
\]

Fly1 frequent flyers are silver users, for which we keep track of program name and ID

\[
\text{fly1:freqFlyerno} : ?n \Rightarrow \text{srv:silverUser} \land \text{srv:fidelityProgramId} : ?n \land \text{srv:fidelityProgramName} : \text{"fly1"}
\]
• **Semantic technologies** can be used to realise the attribute mapping
  – Reusable tools, reusable knowledge, further evolution
• From a performance point of view, we have limited requirements
  – We do not require persistence (attributes discarded after computation)
  – We do not require complex reasoning capabilities (definite Horn clauses)
  – We do not operate on a large knowledge base (at least at runtime)
  – Rules and taxonomies are basically static
Extensions and future work

• Further exploitation of semantic technologies
  – Extend the possibility to import external knowledge
  – Specify constraints on input attributes with ontology languages

• Trust and privacy
  – The world is large and open, so do I trust this authority?
  – If an authority can “delegate” another one we may need attribute mappings between authorities
  – With a negotiation approach, think about a backward mapping from the service point of view to the user point of view
Thank you!

Questions?

Davide Cerri
Semantic Technology Institute (STI) Innsbruck
University of Innsbruck, Austria
davide.cerri@sti2.at

This work has been carried out at CEFRIEL – Politecnico di Milano, Italy