An Ontology-based Architecture for Federated Identity Management

Farah Layouni and Yann Pollet
Laboratoire CEDRIC
Conservatoire National des Arts et Métiers
Paris, France
farah.layouni@cnam.fr, pollet@cnam.fr

***Abstract*—Federated identity platform on the basis of digital content require sophisticated descriptions of that content, as well as service-oriented carrier architectures that allow negotiating and enforcing contract and license schemes in heterogeneous digital application environments. So this paper gives ontology in the Web Ontology Language (OWL) for federated identity systems, giving explicit, formal, and general specifications of a conceptualization of such architecture. OWL ontologies are standardized, machine-readable formats that support automated processing with Semantic Web applications. Intermediate concepts, concepts between base-level concepts and higher level concepts, are central in a federated identity system.**

***Index Terms*—Federated identity, Circle of trust, Identity provider, Service provider, Ontology, OWL**

I. Introduction

The benefits of using ontologies have been recognised in many areas such as knowledge and content management, electronic commerce and recently the emerging field of the Semantic Web. These new applications can be seen as a great success of research in ontologies. On the other hand, moving into real application comes with new challenges that need to be addressed on a principled level rather than for specific applications. This special issue will be devoted to less well-explored topics that have come into focus recently as a response to the new problems we face when trying to use ontologies in heterogeneous distributed environments. These environments include the use of ontologies in federated identity systems.

The identity federation[1] aims to facilitate the sharing of digital assets between institutions and organisations by connecting their authentication services. It becomes possible to open access to digital resources (business application, scientific, educational service, etc.) for an identified population without having to manage local registration of users. The federated identity has a high impact on information systems and several points of vigilance must be observed: we must start by services expected by the user and not by the technical solution; we should carefully identify the partners and the proposed services. Having regard to these requirements for the implementation, these concepts remain potentials values to develop. Indeed, the applications based on federated identity are not unanimous in the public sphere. Local communities and administrations do not mutualize data for the moment. Information systems would exceed the boundaries that still mark the national and local "administrative chain". And this is one of the main objectives of the project "FC^{2}"[1]. Eighteen companies are mobilizing around this project (Gemalto, Amadeus, Atos Worldline, CEV Group, Cnam, Constructive Card, Dictao, EADS, Ensi Caen, Entrouvert, Ephi Formation, GIE-CB, INT Evry, Leirios, nCryptone, NTX Research, France Telecom RD, Sagem). The aim of this project is to create architectures, infrastructure and ontologies platforms for the development of new services based on transparent management and federated identities, both in the field governmental applications that electronic commerce. It is intended to define an overall architecture of identity federation managed by a global ontology.

This paper is structured as follows. Sec. 2 exposes our system architecture. Sec. 3 describes the proposed ontology. Sec. 4 presents our conclusions and perspectives for future work.

II. Overall system architecture

The functional architecture describes the structure of the system in terms of components, modules with how the modules interact with each other. Our system is modelled in three layers. We distinguish the following layers:

First layer: this is a layer comprising actors of the platform. These actors can be classified into two categories:
- users or physical elements (organizations, server): they are represented in the form of singles actors
- other components (service, agent) are represented as stereotyped actors

Among these actors, we name:

- User (Employees, interns, service providers, partners, customers ...) is any individual who needs to use the service of the federated identities to run an application. These users are associated with roles. These roles define a hierarchy of user profiles that are attached various rights of access to functionalities of the system.

[1] http://www.fc2consortium.org/index.html

-Servers [2][3] :

- Identity Provider, IdP: It manages the numeric identity of a set of users (creation, deletion, maintenance of their identifying information). IdP offers an authentification service to its users, allowing them to authenticate on the network. When a user wants to reach a service offered within the federation, he uses the authentification service of his attachment institution. Also the IdP can define the users attributes that it auto-authorizes the propagation to service provider.

- Service provider, SP: represents the applications that require authentification, thus consuming meta-data of users. These metadata is a structured set of data used to describe the user. They are descriptive metadata and management metadata. They can be held by one or more identity providers.

- Application server: is a program that provides services to external users.

-Organization: can be an identity provider, a service provider, or both. An identity provider is an organization that issues and manages identities. A service provider provides and controls access[4] to resources of its kind databases and files. Most organisations combine the two: they provide resources to their partner organizations and issue and manage accounts for their own employees.

-Services: these are electronic services based on transparent management and federated identities. These services can be classified into government services (health, interior, e-democracy, public), financial services or telecommunications services. These services will be decrypted in four aspects: a pragmatic, a semantic aspect, a system and a physical aspect.

Second layer: this functional module consists of four sub-modules[5] that perform the following functions:

A "use" module: responsible interface that allows actors to interact with the system, it also helps to interpret the actions generated through interfaces, to involve corresponding treatment and ensure data processing in the system.
A "cooperation" module that ensures the transfer of information from one module to another and the knowledge base.
An "update" module: to make the process of updating the platform especially the database manager organizations joining the federation of identity.
A "supervising" module that will continuously check the functioning of the infrastructure. It also helps to identify malfunctions and inform concerned administrators.

Third Layer or "Ontological" module[6]: it is a module that lets managing the domain ontology. This block is to define the area and scope of ontology. Developing such ontology will allow sharing the common understanding of the structure of information, reuse of knowledge about the field, explaining what is regarded as implied. In identity federation, the combination of multi-agents with the declarative knowledge, leading to the use of ontologies, is relevant to the development of our architecture.

This combination is justified firstly because of the classic benefits of declarative solutions reporting on the procedural. The declarative solutions provide a more integrated approach with an ontological more direct translation of knowledge domain. With this declarativity of knowledge, changes and evolutivity can be easily taken into account, without recompiling code or stopping execution and this constitutes a significant advantage of scalability.

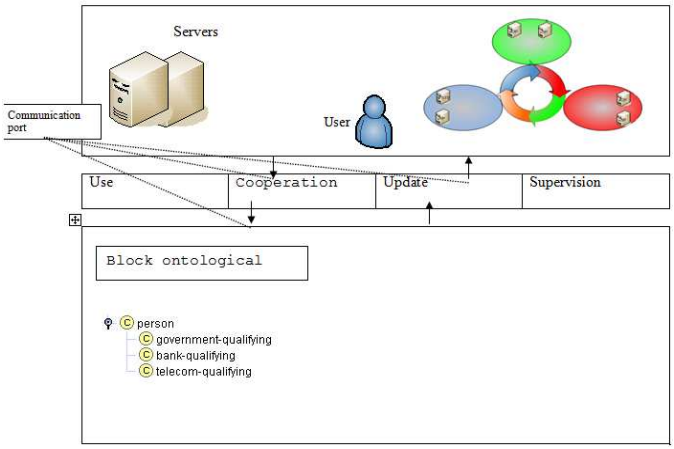

Figure 1. overall architecture

The figure above shows the overall architecture of the system broken down into several modules. You just have to note that this architecture facilitates weaving relationships of trust between those involved in the structure while ensuring adequate level of security.

III. Proposed Ontology

In this section, we detail the ontological layer of the proposed architecture. For this, we use the Web Ontology Language (OWL)[7] [8] to represent semantic models of the user, the user-interface, the work domain, and the information content relevant to work in federated identity domain.

A. Step 1

The first step consists in defining the domain and scope of ontology, this is facilitated by the answers to some basic questions:

What is the domain that the ontology will cover?
For what we are going to use the ontology?
For what types of questions the information in the ontology should provide answers?
Who will use and maintain the ontology?

In our case the individual is the cornerstone of our domain[9], we will also join its attributes and its authentication

information. Our ontology is intended to cover all elements that surround him from near or far in the context of using a shared informatics service. Many questions may concern him, hence the purpose of this ontology which seeks to demarcate, build a framework for this sphere and especially help share these concepts by a federated identity community. Among these questions, we can include the following:

What are the users? Employees, trainees, providers services, partners, customers, any individual seeking computer service proposed by the federated identity community, an inter-partners application.

Who is behind the identity of users? Human Resources, Management Skills, General Services, ISD. This will be modelled by the identity provider of the circle of trust to which the user belong. The IdP will take charge this identity.

How the lifecycle of user identity is it managed? Creation, modification, deletion...

How are managed the user authorization? It is the service provider who verifies these authorizations after receiving the user attributes.

What are the identities strictly Internal and External? The user can have an identity within its organization called "internal identity", and must also acquire another external identity to share with other partners contributing to provide the service.

How to integrate this platform? The company to which belongs the user must open its information system to strangers in order to take advantage of the many external applications to which its users can access (hosted applications, partners applications,...).

Who will use the ontology? Any organization wishing to improve the experience of its users beyond its information system by extending the authentication mechanism.

B. Step 2

The second step verifies existing ontologies for possible reuse of these ontologies. Many previous works on ontologies [10] propose reusable frameworks, we can cite assembly, extension and alignment of ontologies, by establishing links between the concepts. The existing ontologies will persist and will be part of the new ontology. In most existing public ontologies, we have not found one that can contribute to that we want to create. While many refer to a person as the Aristotle's ontology, but none deal the socio-organizational aspect as we hope, hence the contribution of this work.

C. Step 3

This step consists of making a list of important terms, more precisely capture the words related to the domain and precise theirs senses.

In our case we are dealing primarily with three circles of trust, namely "public circle of trust", "bank circle of trust", "telecom circle of trust". This will facilitate our work to list all the concepts relating to our domain by classifying them according to these three circles:

Public circle: we will find the data identifying a person: name, date of birth, place of birth, country, nationality, sex, address, number of the piece identity, telephone, function...

Bank circle: we distinguish two types of data, those relating to the user as the number of his credit card, her expiration date,... but also data relating to the bank as its identifier, address.

Telecom circle: is the most important sphere in our domain, it includes communication on the network, the communication techniques between the different circles. The following figure illustrates the interconnection of the three circles:

Figure 2. Interconnection of the three circles

Each circle represents a significant number of concepts, functionalities, but since the functionalities of our system do not have the same values and have different objectives, it is interesting to make an abstract sort. To do this we propose a design based on two layers of abstraction:

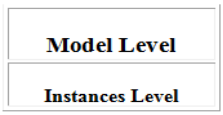

Figure 3. abstraction layers

These layers of abstraction reduce the complexity associated to interconnecting open systems.

-The first layer will define the elements unchanged or immutable referenced to our context.

-A second layer will provide more specifications by adding details concerning variable elements of the platform or a particular instantiation.

When the system is defined, it positions itself in a specific environment. The actors, objects and static interactions of the system will be placed in the first layer of the model. Moreover, the overall vision provided by this layer is meant to be described by our ontology in a unified context. Thus our developments will obey the same internal logic, logic which will ensure an automated, stereotyped augmentability of the system. This architecture allows a very good separation of layers and makes it easy to add or change components.

Let's enumerate elements that will be part of each layer:

The more abstract layer whose data are invariably and identically printed:
Among actors, we name :
-User
-Servers: An IdP, for example, might be an enterprise that manages accounts for a large number of users who may need secure Internet access to the Web-based applications or services of customers, suppliers, and business partners. An SP might be a Software-as-a-Service (SaaS) or a business-process outsourcing (BPO) vendor wanting to simplify client access to its services.
-Organization,Services.

D. Step 4

This step classifies and hierarchies classes. For this, we will rely on the development process "generalization/ specialization" that we began above. In this way, we create a hierarchy of classes more and more specialized. This has the major advantage of not having to start from scratch when one wants to specialize an existing class. So the super-classes are the concepts of the models layer, so we will have the following classes:

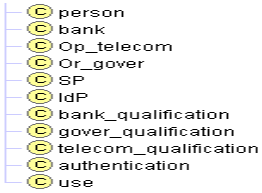

Figure 4. The super-classes

The class "person" is the key of this modelling; it represents the individual who will use the services of the federated identity platform where the SP class represents the application to consume. This person must authenticate hence "IdP" class and "authentication" class which ensures the link between person and IdP. Then we have a class that corresponds to each circle of trust; therefore "bank" class for the banking circle, "Op_telecom" class for the telecom circle and "Or_govern" class for the government circle. And relatively, we will have the "qualification_gover" class involving attributes of a person vis-a-vis his government affiliation, "banking_ qualification" class assemblies bank qualifiers of the person and finally "telecom_qualification" class collects the attributes of the telecommunications sphere.

E. Step 5

This step defines the properties of classes because the hierarchy above does not specify details. So for class "person", we will have the following attributes: Name, address. These attributes will be the only default conveyed by the mobile agent, all those whom we cite after will be exchanged as needed service.

Classes	Attributes
Op_telecom	Operateur_name, coverage, address
Or_gouver	Organisation_name, function, address
Authentication	Login, pwd, fingerprint, signature
Qualification_gover	Date and place of birth, haveFather, haveMother, nationality, family status, haveBrother
Bank_qualification	Credit card number, expiry date
Telecom_qualification	Fixed telephone, mobile phone, subscription duration
SP	URL, service, access control
IdP	ID, Circle membership, Location

Figure 5. classes-attributs

These attributes can have several facets describing the value-type, domain and range of a slot, the cardinalities.

F. Step 6

This step sets restrictions on properties mentioned before. We will describe some whose cardinalities and restrictions are important in federated identity architecture. The ID of the person must be an attribute with single cardinality to identify the person, locate it and also can be used as key of operations. We choose for this attribute a numeric value. "Circle of trust" is an attribute that can define a circle of trust which helps locating IdP and regrouping organizations belonging to the same circle, it will have a numeric value.
There are various ways of attaching characteristics to the properties, thus greatly refine the quality of arguments related to this property. Among the main features of properties, we find transitivity, symmetry, functionality, the reverse.... In our case this is useful for the following properties: We have specified the asymmetry characteristic for the

```
<owl:ObjectProperty rdf:ID="haveFather">
<rdfs:domain rdf:resource="#Person" />
<rdfs:range rdf:resource="#Person" />
<owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID=" haveMother">
<rdfs:domain rdf:resource="#Person" />
<rdfs:range rdf:resource="#Person" />
<owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID=" haveBrother">
<rdf:type rdf:resource="&owl;SymmetricProperty" />
<rdfs:domain rdf:resource="#Person" />
<rdfs:range rdf:resource="#Person" />
</owl:ObjectProperty>
```

Figure 6. Properties of family ties

property "haveFather" and also for the property "haveMother" because these are properties operating at one-way, however "haveBrother" must be valid in both directions. The two first properties must have a restriction on cardinalities because a person can have only one father and one mother. We also use transitive characteristic specifically for "Localisation" property which is used to determine the position of the IdP in the platform.

```
<owl:ObjectProperty rdf:ID="location">
  <rdf:type rdf:resource="&owl;TransitiveProperty" />
  <rdfs:domain rdf:resource="&owl;Thing" />
  <rdfs:range rdf:resource="#Region" />
</owl:ObjectProperty>
```

Figure 7. Location Property

Thus we avoid redundancy cases, the relations city->region-> country will be well taken into account. The other attributes have more or less standard values, this is why we are not going to describe them.

This ontology allowed to study the project in three ways: it gave a diachronic view of system: how the structure will evaluate over time under the effect of adding or removing components? A synchronic view, focusing only on elements that characterize it in a precise moment, more precisely in this developing phase of project regardless of what it will become after. And an achronic view where each object has been provisionally separated from organizational procedure.

IV. Conclusion and Future work

The system we present is designed to assist a monitoring and interactive exploration of the identity federation area through ontology. The ontology has the advantage of making explicit what is considered implicit in the field, to use language understandable by all stakeholders in the learning, use and develop this vocabulary. Thus, we have created and implemented an ontological model, which addresses the problem of data sharing and interoperability in a federated identity environment. He solved the problem by providing a unified interface for the semantics of data, sharing and reuse of knowledge among information resources that can be dynamic. The formalization of widely shared attributes facilitates interaction and cooperation of mobile agents who seems to be the best solution for nomadic access to distributed information in the platform.

References

[1] F. Layouni and Y. Pollet, "Use of mobile agents in a federated identity structure." 48th Annual IACIS International conference,Savannah Georgia, USA, september 2008.

[2] J. Oltsik, "Services-oriented architecture (soa) and federated identity management (fim)." IBM White paper, November 2006.

[3] K. Cameron and B. Jones, "Design rationale behind the identity metasystem architecture." White paper, January, 2006.

[4] F. Layouni and Y. Pollet, "Fi-orbac: a model of access control for federated identity platform." IADIS International Conference Information Systems Barcelona, Spain, February 2009.

[5] M.Tiado, R. Dhaou, and A-L.Beylot, "Multilevel network modelling to achieve cross-layer mechanisms." 4th Annual Mediterranean Ad Hoc Networking Workshop, Med'Hoc Net'05, Iles de Porquerolles, June 2005.

[6] M. K. H. Stuckenschmidt, "Structurebased partitioning of large concept hierarchies." 3rd International Semantic Web Conference, Hiroshima, Japan, 2004.

[7] M. Smith, C. Welty, and D. McGuinness, "Owl web ontology language guide." W3C Recommendation, February 2004.

[8] N. F. Noy and D. L. McGuinness, "Ontology development 101: A guide to creating your first ontology." Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, 2001.

[9] R.Bhatti, E.Bertino, and A.Ghafoor, "Towards improved federated identity and privilege management in open systems." Purdue University, 2004.

[10] G. Falquet, C. M. Jiang, and JC.Ziswile, "Intégration d'ontologies pour l'accès à une bibliothèque d'hyperlivres virtuels." 14ème Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle, 2004.