USE OF MOBILE AGENTS IN A FEDERATED IDENTITY STRUCTURE

Farah LAYOUNI, Laboratoire CEDRIC (CNAM), layoun_m@editeur.cnam.fr
Yann POLLET, Laboratoire CEDRIC (CNAM), pollet@cnam.fr

ABSTRACT

This work will try to join two axis of research which concepts are in vogue. The first concerns the federated identity. The studies in this topic will allow the interconnection of information systems, access to different resources and, above all, a secure and controlled sharing data. The implementation of such architectures required several exchanges of requests and responses which can be costly in terms of traffic data. So the second axis, namely mobile agents, intervenes to solve these problems. They offer the advantage of reducing the network load, to move the code to the data, to provide more fault tolerance...
So this work seeks to take advantage of the benefits that can offer mobiles agents to improve the architecture of federated identity.

Keywords:
Federated identity, Federated authentication, Circle of trust, Identity provider, Service provider, Mobile agents.

INTRODUCTION

The individual, its digital existence, profiles, trace, avatars, blogs, social networks, pseudonyms: the question of identity in the numeric network is still one of the key topics of the coming years. Although the technologies are old, the topic has not reached maturity. Moving beyond the framework of security and the protection of privacy, "Federated Identities" approach is a way for individuals to control their own lives, and for organizations, a source of innovation and value creation. Identity federation dramatically streamlines and simplifies the process of sharing with trusted partners the identity data associated with users who share electronic access to information and resources across domains. Using examples, digital identities allow us access to a universe of information such as:
- The different states (coordinates, consumption, invoices), which can be viewed online, which service operators (telecommunications, water, electricity, ISPs, banks, insurance) have about us,
- Different data held by retailers (loyalty systems, purchases historicization...),
- Administrative information (data held by gover-nments or local authorities, made gradually available online) on the subject of marital status, tax information, criminal records, education,
- Medical information held by all the health professionals with whom we interact (insurance, mutual companies, hospitals, pharmacies, general practitioners, specialists, laboratories...), and with whom we are going to be required to communicate more increasingly via the Internet,
- All traces that I leave voluntarily or not online, through various accounts (messaging, forums, blogs, personal pages, professional web sites) - information ranging from marital status data to financial data (CB number) or public or private multimedia elements (photo albums..), as well as various services of Microsoft, Google and Yahoo spheres.

To share this type of information, a system of identity provides the following services: Assigns an identifier, Authenticates the user or resource claiming to be appointed by this identifier, Serves data on that user or his resource (his name, age, country, language), Serve external data from other resources and Serves authorizations on what the user has the right to do or not. There are various identification systems. It is hard to compare them because most do not have the same specifications and do not address the same issues. First we eliminate centralized systems like Microsoft's Passport[1]. These systems do not fit all needs and, even if they could, it would be very dangerous to have a single identity provider. Among the decentralized systems, we note two leaders, in the freedom world, Liberty Alliance and Shibboleth: Liberty Alliance[2] is a consortium of companies, founded in 2001, has produced several specifications on the management of identities and whose primary aim was to establish a standard free federation of identities. Shibboleth[3] is a solution-oriented university, and reprinted by several publishers. Its primary objective is to facilitate the sharing of online resources between different schools. Many of these systems use SAML (Security assertion markup language), OASIS standard for expressing assertions security with XML.

In the future, these solutions are likely to be an element among others in new architectures. These solutions can be mixed together to provide new architectures for future decentralized, pluralistic identification system allowing users to keep or regain control over their identities. This is particularly the case of Shibboleth which will serve as a basis for our work.

Through this work and as part of a great European project "FC2"[1], we will seek to propose a new execution model taking advantage of similarities between these previous solutions and allowing a dynamic cooperation between multiple, independent and heterogeneous information systems, with different levels of involvement. The implementation of such architectures required several exchanges of requests and responses which can be costly in terms of traffic data. So the mobile agents intervene to solve these problems. They offer the advantage of reducing the network load, to move the code to the data, to provide more fault tolerance...

This paper is structured as follows. Sec. II describes the federated identity concepts. Sec. III presents mobile agent system. Sec. IV exposes our solution. Sec. V presents our conclusions and perspectives for future work.

FEDERATED IDENTITY

The notion of federated identities arises from the need to want to share information without centralizing the data in an unique repository. In this context, it was imperative to introduce the concept of "circles of trust". Every circle represents an establishment which manages a set of users and these various circles try to interconnect their authentification services and to use common set of users' attributes. This policy presents the following advantages: managing in a global and coherent way the users and their habilitations, reducing administrative costs, facilating the opening of several information systems by implementing new ways of communication and facilating secure access to digital shared resources among institutions based on user profiles. We note two key services of this solution which are the authentification delegation and the user attributes propagation.

The authentification delegation: consists to use the authentification service offered by the attachment institution of the user, even when the application requiring this authentification is a service outside the establishment.

The user attributes propagation: consists in collecting attributes relative to the user. We define two different types of attributes: those who allow customizing the service (name, e-mail, address) and those required to perform access control (user category, training, roles).

Architecturally, the various solutions of federated identities like liberty alliance, Higgins[4], Infocard or Cardspace[5] regroup two fundamental elements:

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. In case of several identity providers, a protocol such as OAI (OAI-PMH = Open Archives Initiative Protocol for Metadata Harvesting)[6] centralizes metadata leaving them to their original location. So the attributes remain divided and accessible.

It is possible that the same organisation plays the role of an IdP in a context and an SP in another. IdP and SP, each deploys interoperable technical solutions for exchanging assertions authentication and attributes. Thus, the two partners are establishing a relationship of trust. The IdP ensures that the attributes of its users are used only for legitimate needs. Conversely, the SP trusts the IdP, especially on realised authentication and the quality of disseminated attributes.

MOBILE AGENTS

The purpose of this section is to present our vision of a mobile agent system, its various components and the benefits that can be learned from. We define a mobile agent as software module that is able to move from one host to another in the network; it can transport his state and its code of an environment to another in the network where it pursued his execution. A mobile agent is not related to the system in which it begins its execution. Its basic life cycle passes by three states[7] like illustrated in the following figure:

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

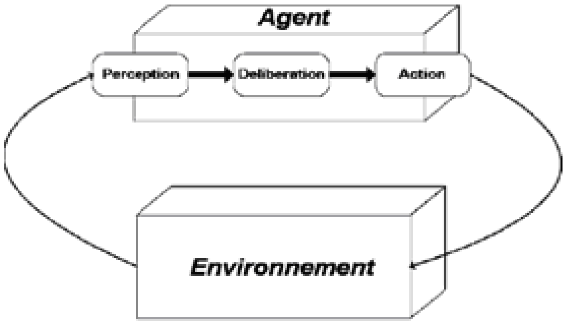

Figure 1. Life cycle of a mobile agent

Perception: recognition of the objects in the environment as well as the interpretation of received messages.
Deliberation: expresses all the means used by the agent to accomplish its action.
Action: describes the operations that an agent performs as well as communications which is the issuer.

These states are necessarily sequential, but the agent may repeat the cycle as many times as necessary. If during its action, the agent discovered a new perception of the network, it can redo the cycle. This can be useful in cases of collaboration between agents which we will present in the following part. By passing into "action" state, the agent acquires the following characteristics[8]; first, agents are able to operate: the action is based, in a Multi-Agent system, on the fact that the agents perform tasks that are affecting the environment and subsequently modify their decision-making. Second, agents are autonomous: agents are independent and not subject to order. They are led by their individual objectives which they seek to satisfy. Each agent has the freedom to act and to respond to requests from other agents. For this, it requires a number of resources or a private local memory which makes it independent of its environment and other agents. Third, agents have a partial representation of the environment, an agent knows generally only some agents who form its knowledge conscripts also relations. It is with this group of agents that it communicates and exchanges information.
Such concepts have several advantages[9]. We found that these benefits can be classified into four classes that we define here:
Flexibility: It is possible to adjust the number of agents to the size of the information system and to train agents according to the monitored system.
Efficiency: Agents affect at a minimum the performance of each machine because they are content to work on targeted resources. It should be noted that the gain in network traffic is particularly important.
Reliability: If an agent is out of service, other agents can be reproduced.
Portability: Agents bear more easily distributed systems. Mobile agents have the ability to dynamically adapt to changes and can thus react more quickly.
We turn now to a description of the components of a model for mobile agents. Firstly, we have the agent, which is an entity that has five attributes [10]: its status, its implementation, its interface, its identification and its authority. When an agent moves through the network, it carries its attributes:
Status: enables the agent to resume his execution when it arrives at its destination. The state of an agent can be seen as a snapshot of his execution.
The implementation: Like any other program, the mobile agent requires a code to run. When it moves through the network, the agent can either take its code or go to destination to see what code is available on the remote machine. The implementation of agent must be both enforceable and without risk to the host destination.
The interface: An agent provides an interface that allows other agents and other systems to interact with it. This interface can be a set of methods which enables agents and other applications access to methods of the agent by a messaging system.
The identifier: Each agent has a unique identifier during its life cycle, which enables it to be identified and located. Because the identifier is unique, it can be used as the key in transactions that require a means to reference a particular instance of agents. Secondly, we have the agent system (also called a server agent [9]), it is an environment that is able to create, interpret, implement, and stop a transfer agent. In the same way that an agent, an agent system is associated with an authority that identifies the person or organization for which it works, like presented in the following schema:

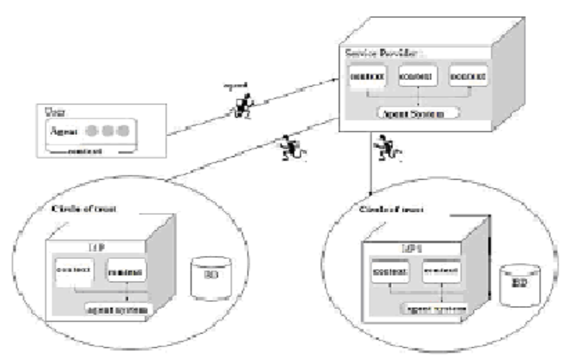

Figure 2. Agent System

A host machine contains a unique agent system but several contexts of agents. Four elements play an important role in the system of agent, which can be used to identify an agent:

The place: is a context under an agent system, in which an agent runs. This context can provide a uniform set of services on which the agent relies regardless of its specific location.

The type of an agent system: it is used to define the profile of an agent. For example, if the type of an agent system is "AGLET" [10], then the agent system is implemented by IBM and supports Java as a programming language.

Resources: the agent system and the place provide controlled access to local resources and services (database, processors, memory, and disks).

Location: is an important concept for mobile agents. It defines the location of an agent as the combination of the place in which it works and the network address of the system where an agent starts up. Specifically, it is defined by the IP address, listening port of the agent system and the name of the place. But in our case, we will add other elements to this location like the identifier of the circle of trust, the identifier of the user and also remove the listening port because it don't have any signification in our topology.

After introducing the various components of the federated identity architecture and agent systems, we develop in the next part an execution model for the distributed queries of this platform.

PROPOSED CHOREGRAPHY

The authentication techniques can be simple techniques where users directly provide passwords to applications or hosts, to much more complex techniques using advanced cryptographic mechanisms to protect identifiable information from user applications and potentially malicious hosts. Provide a password in plain text to an application or a host is regarded as the most mediocre authentication technique, due to the risk of interception of the sequence of authentication. More effective authentication techniques can protect authentication information. This is usually done by data cryptographic signature, with the secret password that only the user and a trusted third party know. A computer authenticates the user via the presentation of cryptographic signature data to a trusted third party. The third compares the signature with the known data of the user and tells the computer whether it thinks the user is or isn't who he claims to be. This mechanism allows keeping the totally secret character of passwords.

There are two general categories for key-based encryption[11] symmetric and asymmetric.

Symmetric encryption uses a single key to encrypt and decrypt the message. This method is easy and fast to implement but has weaknesses; for instance, if an attacker intercepts the key, they can also decrypt the messages. Asymmetric encryption, also known as Public-Key encryption, uses two different keys a public key to encrypt the message, and a private key to decrypt it.

We have to note that we have used a specific authentification technique. It is based on the principle of asymmetric encryption. The asymmetric encryption relies on a couple of keys, a public key and a private key, but in our case it will be two private keys that IdP will generate for each user. One will be send to the user that we note "Cu" and the IdP will keep the other, noted "Cp". Any message to be encrypted by Cu can be decrypted by Cp and reciproquely. In our model, this is the user's query that would be encoded as explains the following scheme:

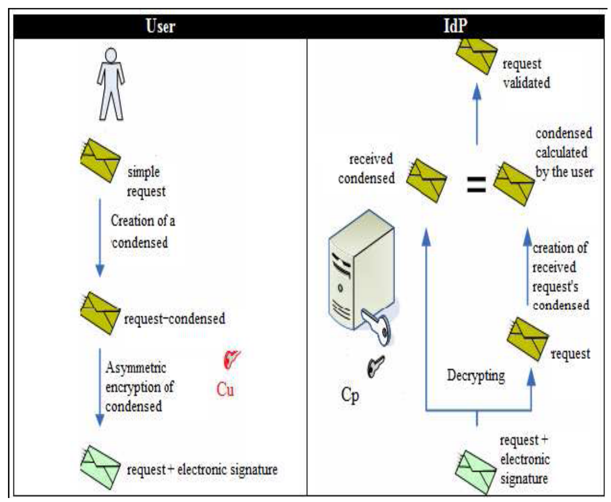

Figure 3. Encryption technique

So, if a user decides to use an application provided by a SP, his request must be encrypted with his Cu then headed to this provider.

In the following, we explain the interaction between different actors in this system under different scenarios.

4.0.1 Service Provider and identity provider in the same circle of trust

In the first case, we suppose that the user and SP are part of the same circle:

1) The browser or user application sends a mobile agent, whose message reflected an encrypted request (with the Cu) to the service provider,

2) The service provider, without authentication information and unable to decrypt the request, will redirect it to the IdP which is located in the same circle, and he will specify the list of attributes he needs to control access and grant access to resources for the specific user,

3) The IdP will decrypt the request using the Cp associated with the Cu of this user, so it ensures the user's identity. Then he sends the users attributes demanded by the service provider,

4) Having received the decrypted request and the necessary attributes, the SP will execute the query and returns the result to user.

In previous architectures, the user should establish a session with the IdP to ensure its authentication. In our scenario, we combine authentication with request sending through the mechanism of encryption, like noted in the following figure:

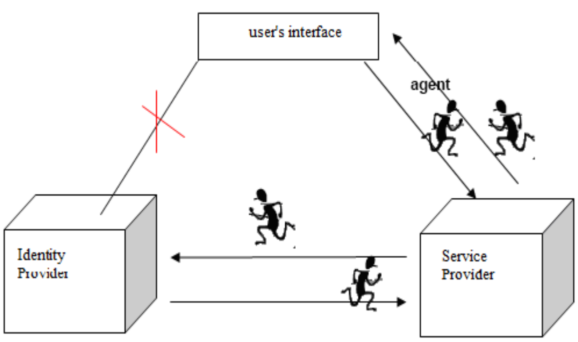

Figure 4. SP and IdP on the same circle of trust

We also optimize exchange between the SP and IdP since the confirmation of authentication and attribute propagation are in one exchange. During the first exchange, the SP remembers that the user has been authenticated. In this way, no encryption or redirection to IdP will be required for next requests.

4.0.2 Interaction between different circles of trust

We are in the case where the SP is available for users attached to different circles. The problem that arises is that the SP does not know how to redirect agent transporting request, he did not know which IdP he has to ask to identify the user. This trouble can be solved in different ways, for example, we can use a discovery service like a WAYF (Where Are You From): central component in previous federated architecture allowing user to select his own identity provider. Other way consists on cloning agents and directing them to each existing IdP, also agents can collaborate to facilitate the location of IdP.

First request to the SP without agents collaboration: To locate the IdP to which the user refers, we will use one of the properties of mobile agents namely "cloning": its ability to copy itself and multiply in the network via a cloning operation. The first IdP which will decrypt the request and send back a response to the SP will be the server in the charge of the identity of this user.

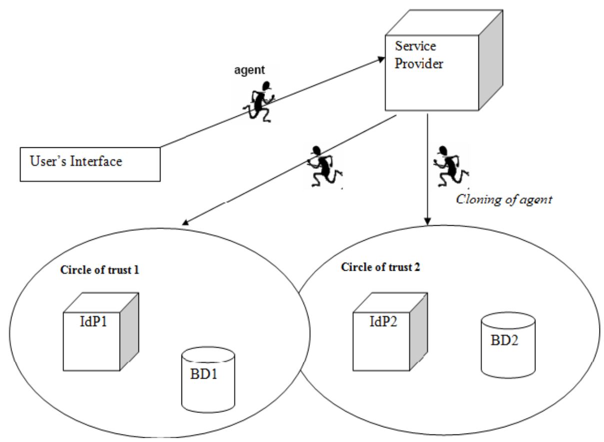

Figure 5. SP and IdP on different circle of trust

It is certainly impressive to have a significant number of agents generated on the network. To overcome this disadvantage, we can organize the circles of trust by neighbourly relations and choose an average number of neighbours to contact. Many studies[12] were performed to estimate the average number, for our case, we estimate it to three; this is linked to the number of the main circles of trust in our platform which will be enumerated later. A user has at least one account in one of these three basic circles.

First request to the SP with agents' collaboration: Agents can cooperate with each other and exchange data, which facilitates and shortens the localisation of IdP. In our case an agent can guide another in locating the IdP attached to the user. This exchange of information is based on user ID. An agent can guide another agent if their both user IDs are in the same range.

We assume that the user identifier is composed of two parts: ID-IdP+a random number. So as illustrated by the following figure, if two agents succeed in matching their Id-IdPs, they can exchange the location of the IdP, which is very probable to occur because usually IdP manages an important number of users. So the agent can often meet another agent who has already appealed to the IdP it seeks. This collaboration will considerably reduce the number of clones.

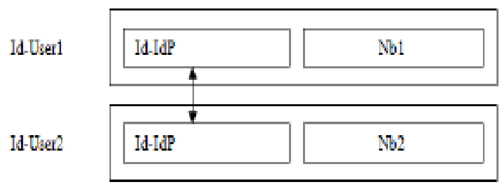

Figure 6. Comparison of ID-IdP

Cooperation may also be extended by combining trace visitation and neighbourliness relations. When visiting a site, an agent may find or submit information in the memory space of the site, for example couples (pre-Id, target direction). So it may help another agent visiting the same site.

4.0.3 Attribute divided among several circles of trust

The use of WAYF will greatly facilitate this task since the user can choose not only an IdP but rather a set of identity servers with which the agent has to communicate in order to collect the different attributes of the user. Also the system allows the user to enter one or more data sources by specifying a plan, a sequence of displacement to guide agent between the various components. That is what we call "reactive migration" where the system initiates the movement without the need for an explicit request of the agent without limiting its autonomy.

To resume, all exchanges will be conveyed via the agent's body who offers a great value and a high level of security for exchanges. Indeed, the number of queries and answers will be at least halved in comparison with other architectures of federated identities. Traditionally, other exchanges required before four pairs of requests and answers: the first pair between the user and SP for sending the request, the second pair between SP and IdP to ask for users' authentication, a third exchange between user and IdP for authentication and then a fourth exchange between SP and IdP for the spread of attributes. So the introduction of agent will minimise the first two pairs of exchanges by removing the opening session between the user and IdP since authentification information will be conveyed at the beginning with the request. Also the exchange between IdP and SP about sending attributes is eliminated since IdP will transmit it directly by the agent who will confirm authentification. Moreover, this platform will take advantage of agents' properties: portability will help them to better function in a heterogeneous environment, autonomy will foster collaboration between agents and help to a better localisation of the IdP and finally the encryption of the body of the agent offers a higher level of reliability.

CONCLUSION AND FUTUREWORK

In this paper, we have presented a scenario in which the mobile agent would greatly reduce the management complexity and costs, as well as improve the security regarding to user authentication in a federated identity topology. After outlining the three cases where the agent can intervene, we demonstrated that they offer more autonomy and adaptability.

In the future we will be working on the last case, when the attributes are divided among several circles. We have presented in this paper a centralized solution based on the WAYF but it would be interesting to avoid the involvement of the user for this task by proposing a protocol which manages the set of attributes: defining the constraints of type, a pattern of data movement between the circles of trust. We can also define a semantic and common frame-work for shared attributes. A large part of the proposed choreography is based on the collaboration of agents. But this collaboration can not take place if agents don't share the same vocabulary, the same syntax. We will try to describe the ontology necessary to our agents who need content interpretable in a unique way by all components of federated identity. It can be formalized for example by using OWL, UML, XML Schema, RDF graphs or ontologies.

REFERENCES

[1] K. Kang, «Network identity.” Seminar on Networking” Business, 2004.

[2] T. Candia, “Benefits of federated identity to government.” White paper, Liberty Alliance Project, March, 2004.

[3] N. Dors, “Shibboleth architecture.”JRES, 2005.

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

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

[6] S. Oulahal, "Accès unique à des ressources numériques distribuées." JRES, 2003.
[7] M. Wooldridge, "Intelligent agents." MIT Press, 1999.

[8] A. Jansen, "Authorization and delegation of privileges in mobile agent systems." ACSAC 2001. Proceedings 17th Annual, 2002.

[9] M. P. H. Hexmoor and N. Suri, "A distributed content-based search engine based on mobile code and web service technology", ACM, 2006.

[10] S. G. Gupta, S. Ganeriwalla and S. Nautiyal, "Virtual internet pets based on java-enabled mobile agents", The International Workshop on Agent Technologies over Internet Applications, 2001.

[11] S.Mazaher and P.Røe, "A survey of state of the art in public key infrastructure", August 2003.

[12] J. G. W.M. Farmer and V. Swarup, "Security for mobile agents: Issues and requirements", in 19th National Information Systems Security Conference, pages 591597, Baltimore, October 1996.