Access Control in Federated Systems

Sabrina De Capitani di Vimercati   Pierangela Samarati

Dipartimento di Scienze dell'Informazione
Università di Milano
via Comelico 39/41 Milano 20135, Italy
Phone: +39-2-55006257
Fax: +39-2-55006253
{decapita,samarati}@dsi.unimi.it

Abstract

One of the new emerging technology for data management is today represented by federated systems. The success of this technology, which has been receiving increasing attention from researchers and developers, comes from the need to integrate and work on different existing systems independently developed and evolved. The necessity of making them available to users as if they were a single system while at the same time not affecting their independent working arises several issues with respect to authorization management and specification and to access control enforcement. In this paper we outline some of these issues and illustrate the basic ideas of possible authorization model for the protection of information in federated systems.

1 Introduction

Traditional distributed database systems were designed and organized with distribution in mind [7]. When a system had to be developed the global database schema was defined and split into different pieces which were then stored at different, geographically distributed, sites. Most distributed systems today do not fall in this framework. Very often organizations are faced with the need to integrate and cooperate in accessing different databases which have been independently developed and evolved, in order to allow users to access them as if they were a single system. The traditional distributed database design approach, which would require redesigning from scratch the whole schema, is in this case not applicable. Moreover, the integration process must not affect the single databases, which may need to continue working independently to satisfy the requests of their local users. This problem has introduced a new distributed architecture, that of *federated systems*.

A federated system integrates existing, possible heterogeneous, databases while preserving their autonomy [21]. The main difference between the federated system concept and the traditional distributed system concept is that in federated systems each component remains autonomous. Autonomy of a component system means that the local administrator maintains some control over his system. Different types of autonomy have been distinguished, namely: *design autonomy*, *communication autonomy*, *execution autonomy*, and *association autonomy*. Design autonomy refers to the ability of a component system to choose how information it stores is to be organized and accessed, for example, with reference to the data model and query language to be used. Communication autonomy refers to the ability of a component system to decide whether to communicate with (i.e., to respond to the requests of) other component systems. Execution autonomy refers to the ability of a component system to execute local operations without interference from external operations. Association autonomy refers to the ability of a component system to decide whether and how to share its functionalities (operations and resources) with other systems.

We believe that a particular type of association autonomy is worth independent investigation, which we call *authorization* autonomy [8]. By authorization autonomy we mean the ability of a site to specify which accesses are to be allowed or denied on objects stored at the site.

Authorization and access control issues have not been receiving, if not for few exceptions, much attention by the research, mainly focused on the various problems related to data access and communication [21]. However, many are the security issues that need investigation. The need to share data in the federation on one side and to maintain site autonomy on the other side raise several protection requirements which traditional authorization models [6, 20] do not address. Security en-

1996 ACM New Security Paradigm Workshop Lake Arrowhead CA
Copyright 1997 ACM 0-89791-878-9/96/09 ..$3.50

forcement at the federation level must take into consideration the protection requirements and the protection policies of each participating site. This task can be further complicated by the heterogeneity of the constituent systems, which may enforce protection policies either difficult to combine or inconsistent with each other. Moreover, *local autonomy* impacts the ability of the federation to acquire and replicate data or to make them available to others. A major problem in this context is also the establishment of administrative policies that determine the authority of the different federation participants for the specification of access authorizations. As a matter of fact, while in a centralized or distributed system ownership or centralized administration may be satisfactory solutions, federated systems call for more flexible approaches [19]. Enforcing complete strict ownership would put on the data owner the burden of specifying authorizations for federated users and therefore to maintain information on who can access the federation. Applying a centralized administration approach at the federation level may imply a loss of control, and therefore of autonomy, for the data owner. Moreover, even traditional problems, such as authentication, require careful reconsideration in the federated context.

Although recent research has addressed the problem of protecting federated systems [4, 12, 13, 16, 17, 23, 24, 25] and few federated systems, like Mermaid [22], Orion-2 [15], or the one proposed by Heimbigner and McLeod [9] support some form of authorization specification and access control, several issues still remain to be investigated.

In this paper we discuss some of the protection issues which arise in federated systems and discuss possible approaches to their solution. Moreover, we present an authorization model on which we are currently working targeted to the protection of information in federated systems based on a tightly coupled architecture. The model we have in mind supports authorizations to build and maintain the federation as well as authorizations to control access to federated data. At each component site users declare the objects they wish to export and the access modes executable on them by federated users. Inclusion of objects into the federation requires their subsequent import by the federation administrator. Different degrees of authorization autonomy are supported, whereby users can retain or delegate to the federation administrator the task of specifying authorizations. A site can require to authenticate the user at each access or accept his identity as communicated by the federation. The remainder of this paper is organized as follows. Section 2 illustrates some security issues that must be considered in ensuring protection to federated systems. Section 3 proposes an authorization model for authorization management and enforcement in federated systems. Section 4 compares our model with previous proposals. Finally, Section 5 presents the conclusions.

2 Research issues

In this section we introduce some of the security problems which arise in federated systems together with possible solution to them. The major issues of the discussion are summarized in Table 1.

2.1 Authentication and access control

A good user's authentication is a prerequisite for a correct access control. The identity of a user determines the groups to which the user belongs, the roles he can play (if applicable), and ultimately the privileges he is allowed to exercise. Even in mandatory systems, where clearances are used in access controls instead of identifiers, the user's identity is needed to determine the security level with which the user can connect to the system. In any case therefore, on a user's identity depends whether his requests to access the data will be allowed or denied.

In federated systems, access to data can be seen at two different levels: at the federation level, where users explicitly require to access the federated data, and at the local level, where the local requests corresponding to the global requests must be processed. Access control may possibly be executed at both levels. The question therefore arises of what identity should be used in the two access controls, i.e., against whose authorizations should the access controls be enforced. A first decision to be taken concerns whether users should connect and authenticate themselves to the federation in order to access federated data. A possible approach consists in requiring that in order to access the federated objects, a user should own an account at the "federation site". Access control at the federation can then be performed with respect to the identity with which the user connected to the federation. An alternative approach consists in leaving the federation freely available to everybody (without any identification and authentication procedure). Access control at the federation can in this case be enforced on the basis of the user's remote identity or of the site where the connection originated. For instance, access to some federated data can be allowed to all users connected from site `site1` or to the user remotely connected to the federation and with local identifier `tom@site2`. However, this approach would require access authorizations to be specified only with respect to remote identities. The approach of always requiring explicit connection to the federation is preferable in general, since it allows authorizations on federated data to be specified against identifiers established and managed by the federation

Problem		Solutions
Authentication	federation	• can require users to explicitly identify themselves • can allow access to everybody without authentication
	local components	• can require users to identify themselves at the site • can trust identities as communicated by the federation
Access control	at the federation	• uses remote identity of the user • uses identity of the remote site • uses federated identity of the user
	at each local component	• uses remote identity of the user • uses identity of the remote site • uses federated identity of the user • uses identity of the federation • uses local identity of the user
Population of the federation		• direct creation (global objects) • import of objects (imported objects): - by the federation administrator - by local sites administrators - through negotiation of the federation and local site administrators
Administration of authorizations for imported objects		• federation administrator • local administrator • cooperative
Specification of authorizations		• global and local authorizations are independent • **bottom-up derivation** - global authorizations are derived from local ones • **top-down derivation** - local authorizations are derived from global ones
Authorization state consistency		• **immediate** - changes are propagated as soon as they happen • **access time** - changes are discovered and propagated during access control • **periodic** - changes are propagated at specific time instants • **explicit request** - changes are propagated upon explicit request of the federation or the local administrator

Table 1: Security problems and possible solutions

administrator. Note that an alternative approach would be to enforce no access control at the federation. In this case access control is enforced only locally on the local requests corresponding to the global access. However, this approach has the drawback of always allowing access to the system and its schema (although, notice, not to the information stored in the objects). Moreover, it may result in unnecessary sending requests to the local systems, which can instead be avoided by access control at the federation level.

Let us therefore assume that each user needs to identify himself at the federation. The question that arises now is what happens when his requests are forwarded to the local sites. What identity should each local site consider, i.e., against whose authorizations should the local site enforce access control? Again, there are two different approaches that can be taken with respect to this, which we can distinguish as *local* versus *global* authentication. Each of them has some pros and cons.

In the local authentication [22] users are required to re-authenticate themselves at each local site. Upon reception of the requests by the federation, the local site asks the user to identify himself and, after authenticating him, performs access control and possibly returns the data to the federation. This approach has the advantage that local access decisions can be taken with respect to identifiers known at the site and therefore does not require the local site to be informed about remote or federation identities of users. However, it may make the access control process very heavy. Indeed, each access request on federated data can be split into several access requests on local data, possibly stored at different sites. If local authentication is to be applied, the user will have to type in login and password for each site involved in the transaction. Moreover, this approach may compromise the transparency of the system according to which the user should exercise access to a federated object without worrying about the specific local objects from which it has been obtained or about their location.

In the global authentication [14], users are not required to authenticate themselves at each local site. Their identity (and/or other information needed for access control) is passed to the site by the federation together with the request. Access control at the local level can therefore be enforced by considering: *i)* the federation from which the request arrives, *ii)* the identity of the user at the federation, or *iii)* the remote identity of the user at the site from which he connected. In the first case, access decisions are taken only with respect to the federation and not to the identity of the specific user requiring access. For instance, at the local site, an authorization can specify that object `o1` can be accessed by federation `f1`. Every request on the object coming by any user through the federation will therefore be allowed: the system delegates identity-based access decision to the federation. In the latter two cases instead identity-based access decisions are taken by the local site but with respect to the user's identities communicated to it by the federation. The local system therefore needs to put some trust on the remote or federation identity communicated by the federation. This requires that some form of certified communication of identities be applied [11, 18, 26]. This approach however has the drawback that authorizations at the local site need to be specified with respect to identities not administered by the local site itself.

2.2 Population of the federation

Populating the federation means defining the objects that are part of the federated schema. Population of the federation can be done in two ways: by directly creating objects in the federated database, or by importing objects from the local sites taking part into the federation. Direct creation of objects in the federated database can be executed by either the federation administrator or any user explicitly authorized for that.

Import of objects from local sites is instead more complex, since it requires agreement between the local administrators of the objects and the federation administrator. The local administrator of an object must be willing to share the object with the federated users. The federation administrator must be willing to include the object among the federated data. This negotiation process can be required only at the time a site enters in a federation. In this case, users can then be allowed to directly insert their objects and federation administrators direcly allowed to import the objects. Alternatively, the negotiation process can be carried out through different steps as follows. First, local users declare the objects they wish to share with the federation, thus defining a sort of export schema from which the federation administrator can get data. This operation allows simply to declare data which are available to the federation but it does not include them in the federated schema and does not have any effect on it. Second, the federation administrator imports objects into the federation by getting them from the export schemas. This approach has the advantage that negotiation can be enforced at the granularity of each single object, and even for each specific access mode. It therefore allows users to selectively share their objects and federation administrators to selectively import objects in the federation. The fact that both the object's and the federation's administrator must agree in order for an object to be inserted in the federation also represents a guarantee to both of them with respect to the protection of the information they manage.

2.3 Administration of authorizations

A major issue that arises after the federation has been populated is who should administer access on the federated objects, i.e., who should specify authorizations to exercise privileges on them. As for objects directly created in the federation, classical administrative policies applied in centralized systems can be considered. For instance, the administration can rest with the federation administrator (centralized administration) or with the user who created the object (ownership).

Administration of objects imported from local sites is instead more complex. Should it be left to the federation administrator or to the administrator of the local objects imported? It is desirable that a balance be maintained between the necessity of avoiding the local users the complete burden of specifying authorizations on federated data and the necessity of assuring the local users some form of control over their objects. Three different approaches can be taken with respect to administration of federated objects. A first approach consists in delegating the administration of the object to the federation administrator. The federation administrator specifies authorizations to access the federated objects and the access control decision is taken only with respect to these authorizations. A second approach consists in leaving the privilege of specifying authorizations to the administrator of the local object. No authorizations need to be specified by the federation administrator and access control decisions are taken only with respect to authorizations specified by the local administrator. A third approach is to allow both the federation administrator and the local administrator to specify authorizations. The two administrators therefore cooperate in specifying accesses to be permitted. The different approaches obviously provide different degrees of control of the local user on his data which imply different degrees of administrative burden on the user. We believe that the indiscriminate application of any of the approaches may result too rigid. Indeed, different situations can be found where any of the approaches can be preferred over the other ones.

It would therefore be desirable that the authorization mechanism be able to enforce the different options. The choice of the specific administrative policy to be applied with respect to an object can be the result of a negotiation between the administrator of the object/site and that of the federation involved.

2.4 Authorizations specification

In federated database systems authorizations can be specified at two different levels: at the federation level (on the federated data) and at the local level (on the objects exported to the federation). Different approaches can be taken for the specification and coexistence of global and local authorizations.

The first approach consists in considering the two sets of authorizations as independent. The federation administrator specifies global authorizations to access the federated data. The local administrator specifies authorizations to access the local objects. Global and local authorizations are specified independently. However, it is desirable that the two administrators coordinate and cooperate in order to avoid inconsistent specifications.

The other two approaches are based on the assumption that global and local authorizations are related and that they can be derived from each other. The two approaches differ in the direction of the derivation.

The first approach consists in applying top-down derivation. Access authorizations are specified at the global level and then derived at the local level [13]. This approach works as follows. At the global level, the federation administrator specifies authorizations for users to access global objects. Then, the accesses to the local data needed for successful execution of the accesses authorized globally are determined. Hence, a request to grant the authorizations for each of these accesses is sent to the corresponding local site. At the local site the administrator of the interested object can decide whether to accept or reject the request, i.e., whether to specify the required authorization. If consistency between the global and local authorizations is required, the global authorization will be granted only if all local grant requests have been accepted. Consistency means that a request permitted according to the global authorizations cannot fail due to access rejection at the local level. A main drawback of the top-down derivation approach is that it might be very difficult, if not impossible, to derive the local authorizations corresponding to a global authorization. We know that each global request can be mapped by the data management system onto corresponding local requests on local data. Therefore, local privileges necessary on local objects in order to satisfy a global request are known. The problem is with subjects. As we have already discussed in Section 2.1 subjects of authorizations can be different at the global and at the local level. Global authorizations will use identifiers established by the federation administrator. By contrast, in case of local authentication, local sites will use local identifiers. Moreover, at both the local as well as the global level user groups or roles can be supported. The knowledge about their configuration (i.e., which users are part of a groups or can play given roles) is limited to the site/federation where they have been defined and is not known at other sites. As a consequence, authorizations specified at the global level for groups/roles cannot be mapped onto authorizations at the local level for the same groups/roles, since these do not have any meaning outside the federation site. Therefore the problem arises of maintaining the correspondence between global subjects and local identifiers at each site. An authorization for a group/role specified at the global level will most probably need to be mapped into several authorizations at the local sites, one for each user in the group or allowed for the role. Even if this mapping can be enforced (and it may not always be so), the problem arises of maintaining the correspondence between authorizations upon changes in the configuration of groups/roles.

The second approach is based on bottom-up derivation. Authorizations at the global level are derived from authorizations at the local level [5]. In this approach, when an object is imported in the federation, global authorizations are derived from the authorizations specified by the administrator of the object being imported. If the federated object is composite, i.e., obtained from more local objects, the authorizations on all the local objects must be considered to derive the global authorizations. The case may arise where according to the authorizations at some site a given access should be granted while according to the authorizations at another site the same access should be denied. In this case, if consistency is required, no global authorization will be derived for the access on the federated object. The global authorizations will then mirror the intersection of the privileges allowed by the local authorizations. Like the previous, this approach has the problem of dealing with the different subjects against whom authorizations are specified at the global and at the local level. Moreover, it also has the problem of maintaining the consistency of authorizations upon changes occurring at the local level, either to the authorizations (new authorizations granted or existing authorizations revoked) or to the subjects' configuration (membership for groups or roles) which may imply changes of accesses to be allowed.

2.5 Consistency between global and local authorizations

If derivation of authorizations, either bottom-up or top-down, is enforced, the problem of maintaining the correspondence between the authorizations upon changes arise. Different propagation strategies, similar to those used for update propagation in distributed databases may be applied for this. In particular, changes to authorizations can be propagated immediately, at the access time, periodically, or upon explicit request. In the first case, whenever a change occurs at the global level the corresponding changes required at the local levels are immediately required (the direction is inverse in case of bottom-up derivation). This approach has the advantage of always providing full consistency between the authorizations. However, it has the disadvantage of requiring synchronous changes to authorizations stored at different sites, that may not always be possible (due for instance to site or communication failures). The other three approaches overcome this drawback by allowing temporary inconsistencies of the authorizations. In the second approach inconsistencies to authorizations will be found out at the access time and only then they will be fixed. For instance, suppose a global authorization has been specified corresponding to several local authorizations at some sites and that after some time one of these authorizations has been revoked. Suppose now a user requires access to a federated object and that the access is authorized by the global authorization above. When the corresponding requests on local objects are sent to the sites[1] the negative response from one of them will inform the federation that the global authorization is no longer supported by local authorizations and should therefore be removed. In the third approach changes to authorizations are enforced periodically, for instance every day or every week. Finally, in the latter approach changes to authorizations are propagated upon explicit request by either the federation administrator or the local administrator.

2.6 Aggregation and inference

Aggregation and inference problems, which are not easy to control in centralized environment, become even more difficult in federated systems, where data from different, autonomous systems are collected together to form the federated data.

The aggregation problem arises because federated data so constructed may be more sensitive than each single component. The situation can therefore be where users who are authorized to access each single component should not be given access (or should be given only partial access) to the federated data. The increased sensitivity of the federated data may be due to global policies which are unknown to the single components. For example, federal laws exist that control the computer matching of data among the different federal agencies [17]. Although users can access separately the databases at the different federal agencies, they must not be allowed to match data among them. The federated system must therefore enforce this global policy if local sites may not even be informed about it.

Inference refers to the ability to withdraw information about some data by observing other data. Inference obviously violates the protection requirements of the system when a user can infer data he is not allowed to access by accessing data for which he is authorized. In a centralized system, protection from inference requires the consideration of the semantic dependencies between data and the analysis of the authorizations on them. This control becomes very complicated, if not impossible, in a federated system, where data stored at different sites are under the administrative control of different authorities. The different authorities may also use different user identifiers in the specification of the authorizations. It becomes therefore difficult to keep track of all the accesses for which a user (who may be identified differently at the different sites) is authorized.

2.7 Heterogeneity

The systems participating in a federation can be heterogeneous. Heterogeneity can occur in various forms ranging from hardware heterogeneity, to differences in operating systems and networking protocols, to variations in database management systems. This variety of situations makes it impossible to realize an approach capable of addressing all aspects of heterogeneity. Heterogeneity is caused, or increased, by the fact that generally each system taking part in the federation has been constructed independently and not keeping in mind the possibility of its inclusion in a large system.

[1] Note that also if global authorizations are derived from local authorizations (or vice versa) it is always necessary to send the request to the site in order to retrieve the data stored at the site (the federation only provides a view and does not actually stores data). Moreover, since temporary inconsistencies can arise it is always necessary to perform access control at the local level.

Heterogeneity can also occur at a higher level when systems: use different data models, use different query processing systems or query languages, represent same data in different forms, assign different meaning to the same or on related data (semantics heterogeneity). For example, a database can represent a date by using a string, while another database may represent the data as triple of integers day-month-year, and another again as a triple month-day-year. In such a situation a heavy burden is put on the management system at the federation level which must take care of heterogeneity while at the same time maintaining the autonomy and independence of the component systems. The data model and semantics heterogeneity complicate the definition of the federated objects and of the mapping between the global and the local operations. Moreover, also the enforcement of integrity constraints, which local systems may independently require on their data becomes complicated at the global level, where the different constraints need to be translated and to coexist.

2.8 Access control policy heterogeneity

Besides the different forms of heterogeneity at the system or at the data model level, which may impact enforcement of security measures, a further kind of heterogeneity may need to be consider: access control policy heterogeneity. With this expression we refer to the case where the different local sites enforce different access control policies.

A possible difference concerns the types of policy, mandatory versus discretionary, that a site can apply. Suppose objects taken from a site enforcing the discretionary policy and from a site enforcing the mandatory policy are to be combined to form a federated object. In this case, deriving or specifying authorizations on the federated object in such a way that the requirements at both sites are satisfied may result complicated.

Even in the case where all sites enforce the same type of policy, either discretionary or mandatory, heterogeneity problems may arise.

As for mandatory policies, heterogeneity may occur if different sites: use a different granularity of classification (for instance, with reference to a relational system, a site can assign labels to whole relations, to each single tuple in a relation, or to each single element in it); refer to different classification lattices; or give different meaning to the same security levels. If the federation also applies a mandatory policy, the classification lattice considered at the federation should be obtained by merging the local ones. Moreover, the classification assigned to the federated objects should reflect the classification of the local objects from which they were obtained.

As for discretionary authorizations different kinds of heterogeneity can occur. A possible heterogeneity concerns the situation where different sites allow different types of authorizations to be specified [3]. For instance a site may enforce a closed policy, where only positive authorizations are specified and only accesses explicitly authorized are to be allowed. Another site can enforce an open policy, where only negative authorizations can be specified and all accesses not explicitly denied are to be allowed. Other sites can enforce a hybrid policy where both negative as well as positive authorizations can be specified. In this case, again, different sites can enforce different policies to regulate coexistence of positive and negative authorizations and to resolve possible conflicts between them. For instance a site can require complete absence of conflicts while another site can allow them and resolve them according to priority rules on the authorizations. Another possible type of heterogeneity consists in the granularity of objects on which authorizations are specified. For instance, considering an object oriented system, a site can allow only authorizations at the class level, another site can allow authorizations for single object instances, another site can allow authorizations for single attributes inside the objects. A further type of heterogeneity concerns the subjects of the authorizations. For instance, sites can specify authorizations with respect to single users, to groups of users, to roles, or even with references to applications.

Beside heterogeneity in the specific elements of the authorization model, heterogeneity can also concern the regulation policies governing access to the data at the different sites. "Metapolicies" may then need to be defined that coordinate the enforcement of the different security policies [10].

2.9 Other problems

In order to ensure the co-operation between different local systems, several other issues need to be addressed, which if not directly impacting the authorization mechanism, affect the correctness of the operations on the data [2, 7]. We briefly summarize them here.

- *Network security.* Users, through the federation, access data distributed at the component database systems via some type of network. It is necessary protect to all information transferred over the global communication network and standardize the communication methods.

- *Integration of different concurrency control mechanisms.* Different components can use different algorithms for transaction processing. The various concurrency control mechanisms need therefore to be integrated.

- *Distributed query processing.* Distributed query processors automatically decompose a global query into subqueries to be processed at different local systems and select an execution plan which will minimize the execution cost. This process must take into account two important requirements. First, the decomposition must be a correct transformation of the input query. Second, the execution plan must be optimal.

- *Replication.* It is necessary to analyze replication and its impact on the federated system in order to develop an efficient replica control protocol that will improve system availability.

- *Accountability.* Accountability may result complicated in federated systems, where different identifiers are used, possibly managed by different administrators and at different physical locations.

3 A proposal for an authorization model for federated systems

In this section we illustrate an authorization model, on which we are currently working, targeted to the protection of information in a federated system. The model addresses mainly the issues related to the population of the federation and the specification and management of authorizations.

3.1 Federation's organization

Before discussing the proposed solution with references to the research issues presented in section 2, we introduce the basic elements of our model. In particular, we illustrate the architecture of the federated system and give the basic assumptions on objects recognized by the system as needing protection and on subjects that can access them.

Architecture

We consider a federated system based on a tightly coupled architecture [21]. At the global level, we assume a *federation administrator* is in charge of maintaining the federated schema and the authorizations on its objects. At the local level, we assume at each site a *local site administrator* is responsible of maintaining the relationship of the site with the federation.

Sites taking part in a federation must be explicitly registered as such. A site can be registered at a federation as a customer, as a provider, or as both. A customer is a site whose users can be authorized to connect to the federation and access its objects. A provider is a site whose users can make their data available to the federation. Registration of a site at a federation as belonging to one of the categories above is the result of negotiation between the federation and the site administrators.

Subjects and objects

At each site a set of local users is assumed and a set of local objects is stored. We do not make any assumption on the data model used at the each specific site or at the federation. Moreover, our model is independent of the administrative policy that is applied at the local level. We assume each object o is associated with a set of administrators. This set contains: the object's owner, if ownership is applied; the system's administrator, if a centralized policy is applied; and all users owing an administrative authorization on the object if decentralized administration is applied.

At the *global level*, the federated schema can contain three kinds of objects: *global* objects, created directly in the federation, *imported* objects, stored and defined at some participating site, and *composite* objects, obtained by aggregating other global or imported objects.

As for subjects, at the federation level we consider two kinds of subjects: users and groups. Users are entities allowed to connect to the federation and submit requests on its data. Groups, which are sets of users, can be defined with reference to the users identities at the federation or at the site from which their connection originates (for instance a group can be defined as containing all users connecting to the federation from a specific site).

3.2 Authentication

Although the model focus on problems connected to the access control of information managed by DBMSs and does not deal with authentication issues, we need to make some assumptions on authentication to establish the identities againsts which access control is enforced.

As for the global level, only users of sites registered as customer of a federation can access the federation. To access a federation, a user must explicitly open a working session by connecting to the federation site. Connection requires identification of the user and corresponding authentication of his identity by the federation. This identity will be used by the federation for enforcing access control.[2] Besides the identity with which he connected at the federation, a user has also associated an attribute containing his identity at the local site of origin. In this way, authorizations can refer to both the identity of the subjects as connected to the federation and the remote identity of the subjects at the site where the connection originates. Moreover, we also allow subject patterns to be used instead of

[2] Note that this assumption does not rule out the possibility of anonymous connection. Anonymous connection may be treated with a special user identifier, anonymous.

specific identifiers for the subjects in the authorizations. Subject patterns allow the use of the wild character "*" meaning any identity. For example, subject pattern "*@site1" indicates all user identifiers at site1. The use of this authentication policy together with subject patterns allows to specify authorizations in a flexible and powerful way. For example, an authorization can specify that all users connecting to the federation from site site4 can execute certain operations.

At the local level, each site registered as a provider of the federation can, during the negotiation phase, choose between two different authentication policies to be applied for federated users needing to access local objects through the federation: *global* and *local* authentication (see Subsection 2.1). In the *global* authentication, federation's users do not need to identify themselves at the site, their identity as communicated to the site by the federation will be used for access control (of course the federation will have to authenticate itself). In the *local* authentication, before processing each access request coming by a user through the federation, the participating site will require the user to identify himself at the site. This identity will be used in the access control process. Note that communication of identities between federation and sites requires some form of trust in the federation/site that enforced the authentication and in the communication system. Different certification forms can be used to provide this, such as those illustrated in [1, 18, 26].

3.3 Authorizations for populating the federation and administration of federated objects

We allow population of the federation through direct creation of objects as well as through import of objects from the local component sites. Authorizations regulate the population in both cases.

Authorizations for the create operation on a federation can be granted and revoked only by the federation administrator. Direct creation of objects is allowed only by users explicitly authorized for that.

Import of objects is carried out in two steps: 1) export of objects from their local sites and 2) import of objects in the federation (see Figure 1).

At each local site, the site administrator can authorize users for the export operation. Users so authorized can make their objects available (export) to a given federation. Objects can also be made available only for specific access modes. This is an important characteristic since it allows sharing of objects to be confined to specific operations. For example, a user may wish to export an object, i.e., allow access to the federation's users, only for reading and another object for both reading and writing.

Users can also delegate the site administrator to export their objects, with reference to specific access modes. Reasons for delegation can be various. On one hand, users may not want to worry about federations in which the site participates and about authorizations for users of the federation. On the other hand, the administrator himself may wish to not allow direct export of objects by users thus retaining the control of what the site exports (a sort of centralized administration).

When exporting an object, a user can also decide the administrative policy establishing who can specify authorizations to access the object once imported in the federation. Three kinds of policies are supported:

- *site retained* - **SR** - Access authorizations can be specified only by the local administrators of the object;
- *federation controlled* - **FC** - The object is freely available to the federation. Access authorizations defined by the federation administrator establish who can access the object.
- *cooperative* - **C** - Access authorizations are granted by both the local administrators and the federation administrator.

A federation administrator can import in a federation all objects made available to him by the provider sites. No authorization is needed for the import operation.

3.4 Specification of access authorizations

In subsection 2.4 we have described three approaches for the coexistence of global and local authorizations. In this model we have adopted the approach where global and local authorizations are independently specified. This approach has the advantage of not requiring the definition of mapping and derivation rules between authorizations and enforcement of maintenance operation upon changes. Moreover, the fact that local grant and revoke operations are independent of whether a site participates in a federated system ensures the autonomy of the site with respect to the specification of authorizations.

In our model, access authorizations are specified at both the global level (on federated objects) and the local level (on local objects exported to the federation).

At the global level authorizations are specified for federated users/groups to access the federated data.

At the local level authorizations can be specified with reference to local user identities (in case a local authentication policy is enforced) or with reference to federated groups. Federated users are not considered.

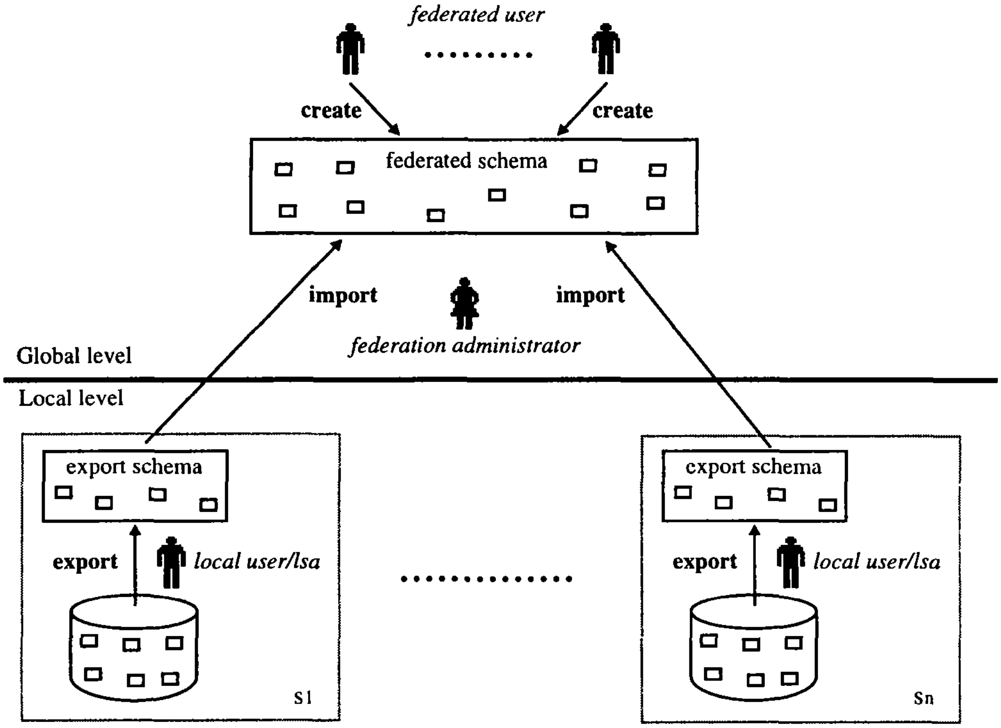

Figure 1: Populating the federation

The reason for this is that requiring each single component site to be informed of the specific user identities at the federation level would make the system very heavy.

For instance, an authorization can specify that group `junior-member` of federation `fed1` can read a given object. Authorizations can also refer to the remote identity of the user or to the site where the connection has originated. For instance, an authorization can specify that all users in group `senior-member` of federation `fed1` and connected from `site3` can read a certain object. At the local level we allow the specification of both positive and negative authorizations. The reason for this is to give exporters a means of retaining control on who can access their objects. This characteristic is very important for two main reasons. First, authorizations are specified for groups of users. However, an exporter may wish to grant a whole group an access but at the same time make sure that some specific user will not be able to exercise it. Since user's groups are defined at the federation site, and therefore the exporter has no means of controlling their configuration (for instance by excluding the specific user), the specification of negative authorizations may be the only means to enforce this. Second, in the case of federation controlled administration, the exporter delegates the federation administrator the task of specifying access authorizations on his object once imported. This means that federated users will not need to have local privileges in order to access the object. Negative authorizations allow the exporter to specify that somebody should be denied for an access even if authorized at the federation level. As a matter of fact no access, even if authorized by the administrator, will be allowed if a negative authorization for it exists at the local level.

3.5 Access Control

In order to determine whether an access request must be granted or denied, authorizations at both the federation as well as at the local sites involved must be controlled. Specific controls and additional authentication processes required depend on: the type of object (global versus imported or composite), the kind of administrative policy of the component object(s), and the authentication policy required by each site involved. Each request on an imported or composite object is translated into a request or set of requests on the corresponding local objects. Each of these requests must be communicated to the appropriate site for both access control, since local authorizations must be present for the data in the local objects to be released, as well as for data retrieval, since data are not replicated at the federation but must be obtained upon each request. The mapping of operations on federated objects onto operations on the corresponding local objects is enforced by the data management system of the federation. Then, the federation sends each site storing a local object involved in the transaction an access request for the groups to which the user belongs together with the remote identity of the user. In case of local authentication the user will need to re-authenticate himself at the local site. Each local site will check the local authorizations and allow or deny the access according to the policy established for the object. In particular, in case of site-retained or cooperative policy access will be granted if an authorization exists for the access and no negative authorization exists denying it. In case of federation controlled administration, access will be granted if no negative authorization denying it exists. Note that in this case no positive authorization is necessary because administration has been delegated to the federation. The final reply of the federation to the user is the result of the replies to the local requests received by the sites. So far we have considered that the global access is granted if all local sites accept the local request; it is denied otherwise. We are investigating different approaches to determine whether access can be successfully granted even if not all the local access requests are satisfied. To illustrate, suppose that a federated transaction requires a read operation on an object **savings-account**, stored at **site1** and a read operation on an object **checking-account** stored at **site2** and that it returns the data read in the objects. Consider now a user who asks to execute the global transaction and suppose the first read operation to be allowed and the second one to be not. The federation can complete in any case the transaction and return to the user the data retrieved from **savings-account** possibly informing the user that data have been released only partially.

4 Comparison with other models

In this section we shortly describe the main characteristics of other models, and then compare them to our model.

Wang and Spooner [25] propose an approach to enforces content-dependent access control in a heterogeneous federated system where authorizations can be specified at both the local and the global level. The approach is based on the use of views and enforce ownership based administration. Content-dependent access control is enforced by materializing views and treating them as protection objects. This approach allows the local systems to preserve authorization autonomy since the local administrator decides whether the local authorization needed for performing the view materialization should be granted or not. However, in [25] authorizations can be specified only for users and a user must be registered at any local system he needs to access, i.e., he has to be known to the local system. In our model, we avoid this by considering as subjects of local authorizations groups defined at the global level. In this way, local systems do not need to keep track of identifiers of each single user of the federation. Moreover, in [25], local systems are influenced by the federated operations because for every view that is created at the global level and using local data, the description of an auxiliary structure must be kept. In our model, instead, authorizations are independently specified.

In Mermaid [22], a front-end system for the integration of multiple homogeneous DBMSs, an authorization model enforcing access control at both global and local level is considered. In order to use Mermaid a user must be registered for it. Access authorizations are specified both at the global level, in the Mermaid system, and at the local level, at each site. Access control at a site is always carried out with respect to the identity of the user at the site. The advantage of Mermaid is that it preserves authorization autonomy and supports different degrees of authentication autonomy. However the approach of Mermaid also suffers from some drawbacks. First, Mermaid does not support decentralized authorization at the global level. The federated users need to negotiate with the local authorities for the specification of the required local authorizations. In our model federated users are not burdened with this responsability. Second, if a user wants to work with Mermaid, he must be registered with Mermaid as well as with any involved local system. Third, access control is based on access control lists which are associated with external and federated schemas. A user can access an object belonging to a schema if an authorization for this exists in the ACL associated with the schema. Therefore, if a fine-grained access control is required then many external schema may need to be defined.

Another model allowing the specification of authorizations at both the local and global level has been proposed by Jonscher and Dittrich in [13]. In this model a global security administrator specifies the local identities corresponding to each global identifier. Authorizations can be positive or negative. The grantor of an authorization at the global level can require consistency of the authorizations. Consistency means that a request permitted according to the global authorizations cannot fail due to access rejection at the local level. Consistency is enforced by propagation of authorizations: every time a global authorization is granted, local sites are required to grant the corresponding necessary authorizations. The global authorization is inserted only if all the corresponding local grants can be enforced. This model has several advantages: it supports different degree of authentication autonomy; it supports decentralized administration based on an ownership paradigm; it uses a set of pre-defined rules to infer implicit authorizations from explicit authorizations; and it preserves autonomy of the local systems. However it also suffers from some limitations. First, the model does not allow local systems to share their objects with reference to specific privileges, which is instead possible in our model. Thus, for example, it is not possible to allow access to the federated's users only for the read access mode on a local object. Second, the model provides a limited form of authorization administration. Each object is associated with an owner who is either a concrete user or a pre-defined user "SYSTEM". A user can access an object only if an authorization is granted to him by both the global owner and the local administrator. This type of administration coincides with the cooperative administrative policy of our model. However, we also support other administrative policies, by allowing different degrees of authorization autonomy. Third, the problem of populating a federation is not considered.

Blaustein et al. [4] propose an approach to control access in federated database systems based on agreements established among the different sites of the federation. Agreements are rules regulating the access to the cooperating database systems by users connected from the different sites. Two kinds of agreements are considered: action agreements and access agreements. Action agreements describe the action to be taken in response to database requests, while access agreements allow to enforce exceptions to prohibitions otherwise in effect. The identity of users at the remote site from which they submit the request is used in access control. This approach is very flexible since there is neither global control nor restriction regarding local autonomy at all. However, it seems to put a heavy burden on users responsible for negotiation at each site, who have to specify agreements with each other single site in the federation.

5 Conclusions

Federated systems represent one of the new emerging technology for distributed database management and organization. These systems are characterized by the fact that while the component systems cooperate and share their resources they also must maintain their autonomy and a good degree of control over their data and resources. Moreover, component systems can be heterogeneous with respect to different aspects of the system. These characteristics raise several interesting issues regarding the specification and management of authorizations and the enforcement of access control. In this paper we have outlined some of these issues together with possible solutions to them. We have also briefly described an authorization model for the protection of federated systems on which we are currently working. We note that the model cover only some issues discussed while some other stile need to be investigated Interesting issues which will need also to be addressed concern trusting measures for communication of identities between sites and the management of users, distributed groups, and credentials [11].

References

[1] M. Abadi, M. Burrow, B. Lampson, and G. Plotkin. A Calculus for Access Control in Distributed Systems. Technical Report 70, DEC, System Research Center, Palo Alto, February 1991.

[2] P. Bernstein, V. Hadzilacos, and N. Goodman. *Concurrency Control and Recovery in Database Systems.* Addison-Wesley, 1988.

[3] E. Bertino, S. Jajodia, and P. Samarati. Supporting Multiple Access Control Policies in Database Systems. In *Proc. IEEE Symp. on Security and Privacy*, Oakland, CA, May 1996.

[4] Barbara T. Blaustein, Catherine D. McCollum, Amon Rosenthal, and Kenneth P. Smith. Autonomy and Confidentiality: Secure Federated Data Management. In *Proceeding of the 2nd International Workshop on Next generation Information Technologies and Systems*, Naharia, Israel, June 1995.

[5] S. Castano. An Approach to Deriving Global Authorizations in Federated Database System. In *Proc. IFIP WG11.3 Working Conference on Database Security*, Como, Italy, July 1996.

[6] S. Castano, M.G. Fugini, G. Martella, and P. Samarati. *Database Security.* Addison-Wesley, 1995.

[7] S. Ceri and G. Pelagatti. *Distributed Databases-Principles and Systems.* McGraw-Hill, New York, 1984.

[8] S. De Capitani di Vimercati and P. Samarati. An Authorization Model for Federated Systems. In *Proc. ESORICS'96*, Rome, Italy, September 1996.

[9] D. Heimbigner and D. McLeod. A Federated Architecture for Information Management. *ACM Transactions on Office Information Systems*, 3(3):253–278, 1985.

[10] Hilary H. Hosmer. Multipolicy Paradigm II. In *Proceedings of the New Security Paradigms Workshop*, Little Compton, R.I., September 1992.

[11] V. E. Jones, N. Ching, and M. Winslett. Credentials for Privacy and Interoperation. In *Proc. New Security Paradigms Workshop*, pages 93–100, La Jolla, California, U.S.A, August 1995.

[12] Dirk Jonscher and Klaus R. Dittrich. Access Control for Database Federations a discussion of the state-of-the-art. In *Proceeding DBTA Workshop on Interoperability of DBSs and DB Applications*, October 1993.

[13] Dirk Jonscher and Klaus R. Dittrich. An Approach for Building Secure Database Federations. In *Proceedings of the 20th VLDB Conference, Santiago, Chile*, 1994.

[14] Dirk Jonscher and Klaus R. Dittrich. Argos — A Configurable Access Control Subsystem Which Can Propagate Access Rights. In *Proc. 9th IFIP Working Conference on Database Security*, Rensselaerville, New York, U.S.A, August 1995.

[15] Wom Kim, Nat Ballou, Jorge F. Garza, and Darrel Woelk. A Distributed Object-Oriented Database System Supporting Shared and Private Databases. *ACM Transactions on Office Information Systems*, 9(1):31–51, January 1991.

[16] J. McHugh and B. Thuraisingham. Multilevel Security Issues in Distributed Database Management Systems. *Computers & Security*, 7:387–396, 1988.

[17] Matthew Morgenstern, Teresa F. Lunt, Bhavani Thuraisingham, and David L. Spooner. Security Issues in Federated Database Systems: Panel Contributions. In C. E. Landwehr and S. Jajodia, editors, *Database Security, V:Status and Prospects, IFIP*, pages 131–148, 1992.

[18] B. Clifford Neuman and Theodore Ts'o. Kerberos: An Authentication Service for Computer Networks. *IEEE Communications Magazine*, 32(9):33–38, 1994.

[19] P. Samarati, E. Bertino, and S. Jajodia. An Authorization Model for a Distributed Hypertext System. *IEEE Transactions on Knowledge and Data Engineering*, 8(4):555-562, August 1996.

[20] R.S. Sandhu and P. Samarati. Access control: Principles and Practice. *IEEE Communications*, pages 2–10, September 1994.

[21] Amit P. Sheth and James A. Larson. Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. *ACM Computing Surveys*, 22(3):183–236, 1990.

[22] M. Templeton, E. Lund, and P. Ward. Pragmatics of Access Control in Mermaid. In *IEEE-CS TC Data Engineering*, pages 33–38, September 1987.

[23] B. Thuraisingham. Multilevel Security Issues in Distributed Database Management Systems II. *Computers & Security*, 10:727–747, 1991.

[24] B. Thuraisingham and Harvey H. Rubinovitz. Multilevel Security Issues in Distributed Database Management Systems III. *Computers & Security*, 11:661–674, 1992.

[25] Ching-Yi Wang and David L. Spooner. Access Control in a Heterogeneous Distributed Database Management System. In *IEEE 6th Symp. on Reliability in Distributed Software and Database Systems, Williamsburg*, pages 84–92, 1987.

[26] Edward Wobber, Martin Abadi, Michael Burrows, and Butler Lampson. Authentication in the Taos Operating System. *ACM Transactions on Computer Systems*, 12(1):3–32, 1994.