Document	gmai
Identifier	1.2.752.104.2.1.1.1
Editor	Pål Axelsson Uppsala universitet, Torbjörn Wiberg Umeå universitet
Version	1.0.0
Last Modified	2006-06-09 14.45
Status	Published
URL	http://www.swami.se/space/GMAI

General Model for Authorization Information

Table of Contents

Introduction and Summary

The two extreme approaches to authorization information and authorization are: (1) Give the precise position/role of the user within the organisation and let the authorization system draw the conclusions of what authority this gives. (2) Describe as careful as possible what authorities the user has in a particular system. This document presents a General Model for Authorization Information, for short GMAI, that can be used for both these cases. The model suggests that most authorization decisions can be based on a tuple with two or more elements of authorization information. The two first elements contain information about Application/Application Area, and Role/User Type respectively. If applicable there may, in addition, be one or more elements defining restrictions on the Scope of Authority. These tuples can be explicitly stored in for example an LDAP directory or generated as requests for authorization information is received.

It is further suggested that for federated authorization within the Swedish higher education sector, the following roles should be used: Self Reporter; Handling Officer; Reviewer; Certifier; Controller; Reader.

These suggestions are based on the result of the work done by a working group in SWAMI - the Swedish Alliance for Middleware Infrastructure (see www.swami.se), whose task was to suggest a small set of nationally harmonised roles to be used for federated authorization among Swedish higher education institutions. The working group members were selected from both the human resource and IT area, to get a wider perspective.

Background

There is a clear trend in Swedish higher education towards systematically introducing increased IT-support for internal business processes to increase efficiency and quality. There is a corresponding interest to support the interinstitutional processes appearing in multiinstitutional research projects; sharing of eScience resources and in cooperation and competition in education around courses and IT-systems (the Bologna Process).

These are the main reasons for the centralisation of user accounts, and for the ongoing implementation of Authentication and Authorisation Infrastructures in our higher education institutions. In the SWAMI LDAP schema best common practice document, it is recommended that, where applicable, the Internet2 edu* and the GNOMIS norEdu* object classes shall be used to structure the identity information for the institution and the individuals when releasing attributes in a federated environment.

The norEduPerson objectclass is one of the norEdu* object classes, initially defined by Uninett in Norway and later modified in the GNOMIS cooperation (the Greater Nordic Middleware Initiative) to as well satisfy the needs in other countries in Norden. The norEdu* object classes are based on edu* object classes defined by EDUCAUSE and Internet2.

The focus on development of our business processes to move ahead in the eGoverment area, potentially makes every student and everyone in the personnel a user in all involved systems. To, in these circumstances, make the expected increased efficiency happen we have to limit the amount of system specific authorization information, and instead try to find ways of using existing general information about users for authorization purposes.

Need for Authorization

First an observation: In a situation where user accounts are managed internally in an application, the authorization to use that application is given by registering the user. When you instead centralise the user accounts in an organisation and introduce an infraservice for authentication you have at that very moment created a need to initially authorize an authenticated user before allowing her to enter a particular system. For this initial authorization, the general information defined by eduPerson and norEduPerson attributes, and perhaps only the attribute eduPersonAffiliation, may well be sufficient since that gives information about which type of affiliation you have to the institution. But, normally, the general information attributes in eduPerson and norEduPerson are far too general to be useful for authorization once you have been granted access to the applications - so we need something else.

The Organisational Role and Authority Tuple

Organisational Role Tuple

Our original task in the work group was to identify a number of organisation wide roles that all universities without further specification would assign the same meaning and that thus could be used for federated authorization. There unfortunately does not seem to exist such roles. The names and the authority associated with such a role varies a lot between universities. For example roles such as ”Director of Studies”, ”Student Advisor” and ”Project Manager” appears to have different meanings in different institutions. We concluded that this approach was not feasible.

For local use, in particular with an authorization system like SPOCP, the approach still is feasible. Thus, persons can be assigned organisational roles. Roles that gives the person certain authorities in systems that understand the meaning of the role. There is no application element in these tuples (see below), only the role and the scope. And - the organisational roles are normally local.

The Authority Tuple

Due to the problem of identifying roles with the same meaning in the higher education institutions, we introduced the A*uthority Tuple* whose elements contain an Application/Application Area, a Role/UserType and possibly several dimensions of Scope. The advantage is that we may define a few standard roles a user can have visa vis an application in general. In an authority assertion the role is combined with the application and the scope.

We assume that the local information, released in connection with federated systems or services, follows the recommendations in the SWAMI LDAP schema best common practice document and that eduPerson and norEduPerson is used.

The tuple can be represented on the form (application, role, scope). Later in the document we will present an LDAP representation of the tuple. To avoid application name conflicts in a federated environment, applications and application areas should be prefixed in the same way as LDAP attribute names. We so far mean that the value space for the Role element is a union of three component spaces.

First the role value may be one of the general User Types – defined below and introduced primarily for federated use. They certainly are of interest for local use in applications locally as well. This approach is partly inspired by classification of operations in the theory of abstract data types – the system may be viewed as an abstract data type and its functions as operations on that data type. Secondly, it certainly will be of interest at a university, to introduce organisational roles, as for example ”head of department”, ”student advisor” and ”webmaster”, and use them as authorization information. Therefore the controlled term ”*gmaiAssertion*” is defined, to describe these organisational roles, as an application area. Thirdly, roles may be introduced locally for use in one specific application. This corresponds to maintaining an application-internal authorization system.

Some further comments on these three approaches. It is extremely important to find ways of managing and maintaining authorization information that does give us some advantage of scale as the number of systems grows. Or phrased in another way: We want the marginal cost for running a new system to be as low as possible. With the third approach, the cost for maintaining the authorization information is the same for a setup, where we retrieve the authorization information from an enterprise directory as for a traditional setup, with an application internal authorization management. This would be extremely unsatisfactory – no advantage of scale, not a single datum of authorization information in our “directory” can be reused. If approach three is the approach of choice for each type of user in every application we have achieved nothing. But it is not a problem, if the third approach instead is used to add extra and fine-grained control of the authority given to small groups of users with specific needs in an application. The goal is that the first two approaches, with a limited number of organisation-wide roles can provide the applications with the major part of the required authorization information. Role data has to be carefully managed in order to be useful as authorization information. Also with this perspective, approach three shall be avoided if possible. It is difficult to define and follow routines for managing data in approach three. In our experience it is quite common in particular to forget to remove authorities – this is a quality and security risk.

The tuples can either be derived as they are needed, or derived from a person's attributes when they are changed and then stored explicitly as authorizations in for example an enterprise directory. They can also be explicitly given as a privilege to somebody. There is a lot more to be said about management of authorities, for example about appointments and delegations, but that is out of the scope of this document.

The User Type for Federated Use

The purpose of the introduction of the concept User Type is to introduce a more abstract role concept, for federated use, that is possible to standardise independently of the particular system or application we are talking about. The goal is to settle for a limited number of user types or roles and to qualify the user type with an application/application area and often also by giving a scope, sometimes combined from several scope elements in the tuple, that sets boundaries for the authority given to a user with a certain user type. The user types we suggest are:

SelfReporter

This user type shall typically be given the authority to view, enter, change and perhaps even delete information about himself. Typical examples are: changing your contact information, claiming compensation for travel expenses. An alternative way to introducing the user type SelfReporter is to use the scope Self in combination with any other user type.

HandlingOfficer

This user type is involved in the work flow in a certain business area. The user typically should be given the same authorities as the SelfReporter, but with a wider scope reflecting the position in the organisation.

Reviewer

This user type is assigned the task to check certain facts concerning a step in a business process and to make an assertion about that in some system. Typical examples are asserting that the books on an invoice from the book store really have been received.

Certifier

This user type takes decisions in some business process, often with economic consequences, but not always. Examples are: deciding the grade of a student on a course, deciding to pay the claimed amount of money for travel expenses.

Controller

This user type is involved with analysing data and producing reports to be used to support decisions and shall be given the right to read and request reports from systems.

Reader

This user type is given the authority to read information in some system. The Dean should for example normally be given the authority to read anything concerning her faculty in any system.

The Scope

The scope shall define the extension of an authority in dimensions of relevance for the particular authority. Considering for instance an organisational scope. Giving a unit in the organisational hierarchy as the scope, means that the user is given the authority to operate on data belonging to part of the organisation in the organisational subtree, starting at the given unit. If there is a need for more than one scope, extra elements are added to the tuple. Furthermore, if there are no scope restrictions, there are no scope elements in the tuple. Example of other scopes are that a user can use an application during business hours or that the user can order IT-equipment for up to a specific amount of money.

LDAP Representation of the Tuple

The most obvious selection of GMAI LDAP representation should have been eduPerson-Entitlement in the eduPerson object class. This is not suitable due to that there could be a large number of authority tuples and that eduPersonEntitlement does not support substring matching. This makes it impossible to search for all tuples associated with one specific user in one application or application area. Therefore there is an auxiliary object class swamiGMAI defined with the single attribute swamiGmaiAssertion. The GMAI LDAP schema definition is published in the schema file http://www.swami.se/space/GMAI/gmai.schema.

Name	**swamiGmaiAssertion**
OID	1.2.752.104.2.3.1
Description	Used to store a set of GMAI authorization tuples
Format	urn:mace:swami.se:gmai:<application>:<role>(:<scopeDenomiator>=<scopeValue>)* ● <application> is the application or application area that the assertion is valid for. The controlled term “*gmaiAssertion*” is used for organisational role assertions. ● <role> can either be one of the above mentioned general User Types or role names known or defined by the application or application area. ● There can be zero or more scope-pairs where all pairs is combined together with an 'and' to a resulting scope. The <scopeDenomiator> defines the area of the scope-pair and <scopeValue> defines the limitations of the scope-pair. ● The NSS-part of the urn shall be case-insensitive which imply that the application urn preprocessing for swamiGmaiAssertion must be complemented with case normalisation of the NSS. In short this means that the whole urn is case-insensitive.
# of values	Multi
Reference	● RFC 2141 – URN Syntax
RFC 2252 definition	(gmaiAttributeType:1 NAME 'swamiGmaiAssertion' DESC 'Used to store a set of GMAI authorization tuples' EQUALITY caseIgnoreMatch SUBSTR caseIgnoreSubstringsMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.15)
Security consideration	swamiGmaiAssertion has substring matching due to the need to search for all defined roles for a user in a specified application or application area. This can be considered a security risk which implies that the attribute shall be protected from unintended usage.

Examples of Authorities and Derivations

Organisational Role Tuples

Persons can be assigned organisational roles. Roles that gives the person certain authorities in systems that understand the meaning of the role. That is the application element is represented with the controlled term “*gmaiAssertion*” in the tuple. The organisational roles are normally local. Two examples are:

A person can be assigned the organisational assertion “Webmaster” in an organisational unit:

swamiGmaiAssertion: urn:mace:swami.se:gmai:gmaiAssertion:Webmaster: norEduOrgUnitID=4823198

The CIO for the institution can be assigned the organisational assertion “CIO” for the whole organisation:

swamiGmaiAssertion: urn:mace:swami.se:gmai:gmaiAssertion:CIO

Authority Tuples

An application system or a general authorization system, like SPOCP, typically associates a set of authorities with a specific organisational role. We mean that these authorities often can be specified by authority tuples as we have defined them above.

For example, the organisational assertion “Webmaster” for “norEduOrgUnitID= 4823198” may generate at least the following authority tuple:

swamiGmaiAssertion: urn:mace:swami.se:gmai:WebSystems:Certifier: norEduOrgUnitID=4823198

Certifier may here be interpreted as deciding what web pages may be published. If all employees at the department shall have the authority to write web pages, they should be given the authority:

swamiGmaiAssertion: urn:mace:swami.se:gmai:WebSystems: HandlingOfficer:norEduOrgUnitID=4823198

And again, an application with an advanced authorization module or an application that uses an advanced authorization system like SPOCP does not require that the explicit authority tuple shall be stored in for example an enterprise directory. It instead concludes that a person with the affiliation attribute “employee” at the department has that authority.

Another example: any person is a student record (Ladok) reader. This information is public.

swamiGmaiAssertion: urn:mace:swami.se:gmai:Ladok:Reader

There is also possible for an tuple to have multiple scopes. A person can order IT-equipment for up to 50.000 Swedish Krona for an organisational unit:

swamiGmaiAssertion: urn:mace:swami.se:gmai:ITprocurment: HandlingOfficer:norEduOrgUnitID=4839458:upperLimit=50000 SEK

Finally, an example of a general role for a specific application. The authority following from that role is decided by the application. A person can be assigned the role of portal administrator in an organisational unit:

swamiGmaiAssertion: urn:mace:swami.se:gmai:Portal:Administrator: norEduOrgUnitID=3749234

The Contributing Group

Pål Axelsson, Uppsala universitet
Johan Ekman, Lunds universitet
Daniel Erlandsson, Örebro universitet
Annika Fröberg, Kungliga tekniska högskolan i Stockholm
Holger Henningsson, Uppsala universitet
Sara Jacobsson, Göteborg Universitet
Maxi Lubian, Göteborg Universitet
Henrik Lövendahl-Nyrén, Karolinska institutet i Stockholm
Torbjörn Wiberg, Umeå universitet

References

Swedish Alliance for MIddleware (SWAMI), http://www.swami.se

SWAMI LDAP schema best common practice document, http://www.swami.se/space/LDAPBCP

Greater NOrdic MIddleware Symposium (GNOMIS), http://www.gnomis.org

EDUCAUSE/Internet2 edu*, http://www.educause.edu/eduperson/

FEIDE/GNOMIS norEdu*, http://www.feide.no/dokumenter/feide-schema-current.html or http://gnomis.cvs.sourceforge.net/gnomis/norEdu/

SPOCP, http://www.spocp.org/