Available online at www.sciencedirect.com
SCIENCE DIRECT®
Future Generation Computer Systems 21 (2005) 549–558
FGCS FUTURE GENERATION COMPUTER SYSTEMS
www.elsevier.com/locate/fgcs

# From gridmap-file to VOMS: managing authorization in a Grid environment

R. Alfieri[a], R. Cecchini[b], V. Ciaschini[c], L. dell'Agnello[d,*], Á. Frohner[e], K. Lőrentey[f], F. Spataro[g]

[a] *INFN Parma and University of Parma, Italy*
[b] *INFN Firenze, Italy*
[c] *INFN CNAF, Italy*
[d] *INFN CNAF v.le Berti Pichat 6/2 I-40100 Bologna (Italy)*
[e] *CERN, Switzerland*
[f] *ELTE, Hungary*
[g] *INFN CNAF, Italy*

Available online 1 February 2005

**Abstract**

Grids are potentially composed of several thousands of users from different institutions sharing their computing resources (or using resources provided by third parties). Controlling access to these resources is a difficult problem, as it depends on the policies of the organizations the users belong to and of the resource owners. Moreover, a simple authorization implementation, based on a direct user registration on the resources, is not applicable to a large scale environment. In this paper, we describe the solution to this problem developed in the framework of the European DataGrid [M. Draoli, G. Mascari, R. Piccinelli, Project Presentation, DataGrid-11-NOT-0103-_1] and DataTAG [http://www.datatag.org/] projects: the Virtual Organization Membership Service (VOMS) [R. Alfieri, et al., Managing Dynamic User Communities in a Grid of Autonomous Resources, TUBT005, in: Proceedings of the CHEP 2003, 2003]. VOMS allows a fine grained control of the use of the resources both to the users' organizations and to the resource owners.


*Keywords:* Grids; Authorization; Attributes

## 1. Introduction

In the previous generation of Grids, where middleware such as Globus 1.x [4] was developed from a single project, very simple authorization schemes were possible, largely due to their use for demon-

* Corresponding author.
*E-mail address:* luca.dellagnello@cnaf.infn.it (L. dell'Agnello).

0167-739X/$ – see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2004.10.006

strations of the technology rather than production. In that scenario, a small number of users was involved and authorization could be managed by hand at each resource.

In the framework of Globus, a simple mechanism, the "gridmap-file", was used: a simple list, resident on resources, of all authorized grid users expressed as Distinguished Names and associated with the corresponding the local credentials (e.g. usernames on Unix systems).

In the present generation of Grids, such as European DataGrid (EDG) [1] and DataTAG [2] testbeds, middleware has been developed from several sources, with published and stable interfaces. In this scenario, with a larger number of users (hundreds) and sites (tens), a strong requirement is that authorization at each resource must be managed by some automated procedure, which derives local policy from one or more central, manually managed source(s) of authorization.

Hence, users in a Grid are usually organized in entities called Virtual Organizations (VOs). A Virtual Organization is a collection of individuals and institutions that is defined according to a set of resource sharing rules [5]. The VOs generally share resources and establish agreements with general facilities called Resource Providers (RPs) offering resources (e.g. CPU, network, storage).

In a potentially large environment like the Grid, the problem of access control to resources, i.e. the authorization, is conveniently simplified by adopting the above schema based on VOs and RPs.

The security infrastructure of present incarnations of the Grid concept, e.g. the testbeds deployed by the EDG and DataTAG projects, is based on Public Key Infrastructure (PKI) [6,7], both for authentication and authorization.

While authentication is usually delegated to external trusted entities, the Certification Authorities, the authorization process is managed by both VOs and RPs with different roles. More specifically, the general information regarding the relationship of the user with his VO (the attributes he can be granted) is managed by the VO itself, while the RPs evaluate locally this information taking into account the local policies and the agreements with the VO, eventually mapping from the grid credentials (i.e. X.509 certificates) onto local ones (e.g. Unix credentials, eventually through the pool account mechanism[1]).

In this paper, we discuss the authorization requirements of a Grid, focusing on the frameworks of the DataGrid and DataTAG projects and illustrate the architecture of the new service we have developed, the Virtual Organization Membership Service (VOMS) [3] to manage authorization information in the scope of the VO.

VOMS, having been developed as part of the joint efforts of the EDG and DataTAG projects, has been fully tested and integrated into the EDG and DataTAG testbeds. However, its flexibility and portability have led to its use in a different Grid middleware distribution (still based on Globus middleware) as Grid3 [9] and the inclusion in the VOX system[10]. Moreover, it is now considered for acceptance into the LHC Computing Grid (LCG) [11] and Enabling Grids for E-science in Europe (EGEE) [12] projects.

## 2. Basic authorization requirements

Authorization is the granting or denial of permission to carry out a given action; this applies not just to users but also to all processes involved in a security operation (e.g. a process may need authorization to access confidential data). As seen above, authorization in present Grid applications is based on the concept of VO; VOs administer users grant them permissions and establish agreements with RPs. RPs, in turn, enforce local authorization.

To examine the concept of VO and see how to manage a VO, we will make a preliminary analysis of the main authorization requirements [13].

The first condition for a user to access the Grid is to be a member of a VO. In general, a user may be a member of any number of VOs with any number of roles, and his membership in a VO must not be relevant to

[1] The pool account mechanism [8] is an EDG extension to standard Globus enabling the dynamic allocation of local Unix usernames to Grid users. The account is allocated to the Grid user only for the time needed to access the local resources; then it can be deallocated and reassigned to the pool of available accounts. It is possible to define multiple pools in order to implement policies at local level. This mechanism makes negligible the probability of exhausting the set of standard Unix credentials (normally, the maximum number of possible user accounts on a Unix system is 65,536).

other VOs. Once the user has authenticated himself, he must be able to select and deselect VOs and roles. Permissions must be granted automatically, and security requirements must be adhered to automatically.

On the other hand, the owner of a resource (i.e. the RP) should be able to enforce local user authorization based on various user characteristics such as his membership in a VO, roles he can have or his identity. The authorization mechanism must preserve the identity of the user, i.e. the Distinguished Name of the user.

Access to resources can be differentiated according to the VOs, the user is member of and the roles he has in it.

The authorization requirements on file access should hold regardless of replication and should not depend on any other site.

It must be possible to confirm that a user has the VO membership(s) and role(s) they are claiming to have.

### 2.1. VO structure

A VO can have a complex, hierarchical structure with groups and subgroups. This structure is needed to clearly divide VO users according to their tasks and home institutions. From an administrative point of view, the management of each group can be independently delegated to different administrators. The administrators of each group can create subgroups and grant administration rights to these subgroups; they cannot modify memberships in any other subgroup.

Thus, in general, we can represent the VO structure with a Direct Acyclic Graph, where the groups are the vertexes of the graph and the subgroup–group relationships are the oriented edges. As a special case, the VO is a group containing all other groups.

### 2.2. User attributes

As seen above, a group is basically a set of users, which may also contain other groups. In general, a user can be a member of any number of groups contained in the VO hierarchy.

We assume that membership in a group implies the membership in all ancestor groups up to the root (i.e. the VO itself). This assumption comes from the fact that only the VO manager must have the permission to include new users in the VO. Moreover, in order to allow RPs to map correctly VO members, a user can be a member of some ‘mandatory group’; to implement this feature, all groups are always returned in the credentials.

To allow for more flexibility, in our model, users are also characterized by two other sets of credentials: roles and capabilities.

Roles are used to further specify users’ properties, properties they have as members of some groups. For this reason, the scope of roles is limited to specific groups; a user may hold a specific role only in a certain group and not in others. The main difference between groups and roles is that the user can choose which of his roles are to be listed in his credentials, while all his groups are always specified. This makes it possible to have groups that lessen the user’s privileges.

Capabilities are expressed as free-form strings of characters, which can be used to describe the user’s special characteristics.

An attribute is a string that consists of a group membership, roles and capabilities in the scope of the VO; these attributes can have an indefinite or temporary validity (i.e. on a scheduled time basis or on a periodic basis).

The enforcement of these VO-managed policy attributes (group memberships, roles, capabilities) at the local level must reflect the agreements between the VO and RPs. However it should be possible for RPs to override the permissions granted by VOs (e.g. to ban unwanted users). As a consequence, to permit traceability at user level (and not only at VO level), users must present their credentials to RPs along with their authorization information.

### 2.3. Credential format

As seen above, every user in a VO is characterized by the set of his attributes, i.e. 3-tuples of the form (group, role, capability). The combined values of all of these 3-tuples form unique attributes, the so-called “Fully Qualified Attribute Names” (FQANs) [14].

We can represent an FQAN as a sequence of group names; each of them may be qualified with one or several roles that the user can hold in that group, and one or several capabilities that this group/role combination may grant to that user (since roles and capabilities are not hierarchically structured but specific to groups).

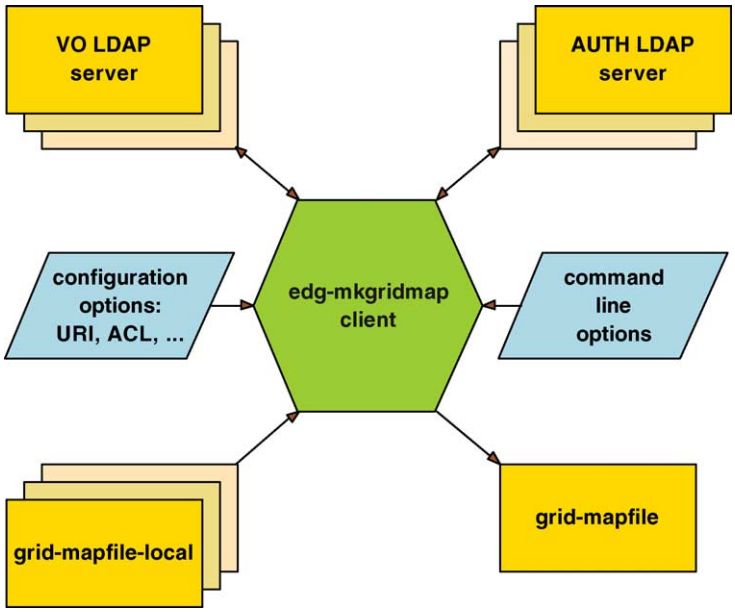


Fig. 1. VO LDAP authorization mechanism.

In general an FQAN has the following form:

/**VO**[/**group**[/**subgroup**(**s**)]][/**Role** = **role**]

[/**Capability** = **cap**]

For example, the FQAN corresponding to the role **Administrator** in group **Nerds** of VO **campus.example.org** is:

/**campus**.**example**.**org**/**Nerds**/**Role**=**Administrator**

where **/VO[/group[/subgroup(s)]]** is the full group name.

Based on RFC 3281-style Attribute Certificates [15], we defined a qualifier for *vo-roles* (not the "role" that the RFC 3281 defines), *groups* and *capabilities* in a new attribute, which follows the IetfAttrSyntax:

**name: voms-attribute**
**OID:** {**voms 4**}
**syntax: IetfAttrSyntax**
**values: one attribute value only; multiple values within the IetfAttrSyntax**

where { voms} is 1.3.6.1.4.1.8005.100.100.1[2].

[2] The 1.3.6.1.4.1.8005 enterprise subtree is registered for EDG.

## 3. "Gridmap-file" and VOMS

A first interim solution adopted by the collaborations of EDG and DataTAG consisted in using Lightweight Directory Access Protocol (LDAP) servers to manage the authorization information at the VO level, and developing the *mkgridmap* utility to allow resources to automatically download this information and generate the "gridmap-file" (see Fig. 1).

With this scheme, authorization is boolean, since there is no way to implement a fine grained authorization granting users permissions other than simple membership in the VO. Hence, a differentiation among users is only manageable at the local sites and can only reflect local policies. This clearly fails to fulfill the basic requirements described in [13].

Moreover, the use of a periodically updated RP-based database (i.e. the "gridmap-file") hardly scales in a production environment with a large number of users, each potentially with a different set of permissions; whereas in a testbed the users situation is almost static, and user policy is very simple. , with this solution, the rising of the number of users or the complexity of organization would entail a large and frequent number of changes to all the gridmap-files at all the local facilities.

### 3.1. The VOMS system

The Virtual Organization Membership Service (VOMS) has been developed in the framework of EDG and DataTAG collaborations to solve the above-mentioned LDAP VO server limitations. In fact, the purpose of VOMS is to grant users authorization to access the resources at the VO level, providing support for group membership, roles (e.g. administrator or student) and capabilities (free-form string).

Our preliminary effort was the evaluation of the first release of the CAS System [16], which was not satisfactory as it did not allow – at least in a simple way – the local site to know from which "real" user the request came (more problems will be mentioned in Section 4.2). This issue has been then addressed in successive releases of CAS.

We also evaluated PERMIS [17], but the software was not sufficiently mature when we performed the study, and the development times foreseen were incompatible with our deadlines (for a comparison with VOMS see Section 4.1).

For efficient management and access, the data are stored in an Relational DataBase Management System (RDBMS) — MySQL [18] in the present implementation (see Fig. 3).

The VOMS system consists of the following parts (see Fig. 2):

- *User Server*: receives requests from a client and returns information about the user.

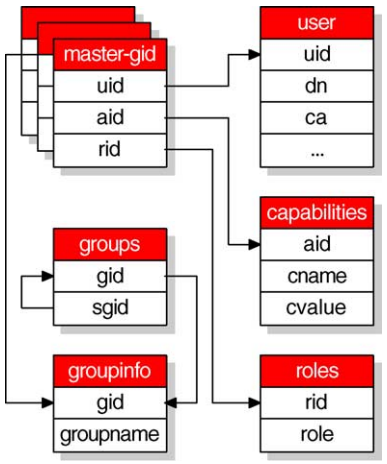


Fig. 3. The VOMS database structure.

- *User Client*: contacts the server presenting a user's certificate and obtains a list of groups, roles and capabilities of the user. All client-server communications are secured and authenticated.
- *Administration Client*: used by the VO administrators (adding users, creating new groups, changing roles, etc.)
- *Administration Server*: accepts the requests from the clients and updates the database.

### 3.2. VOMS operations

In this section, we describe how VOMS works both for the users and administrator.

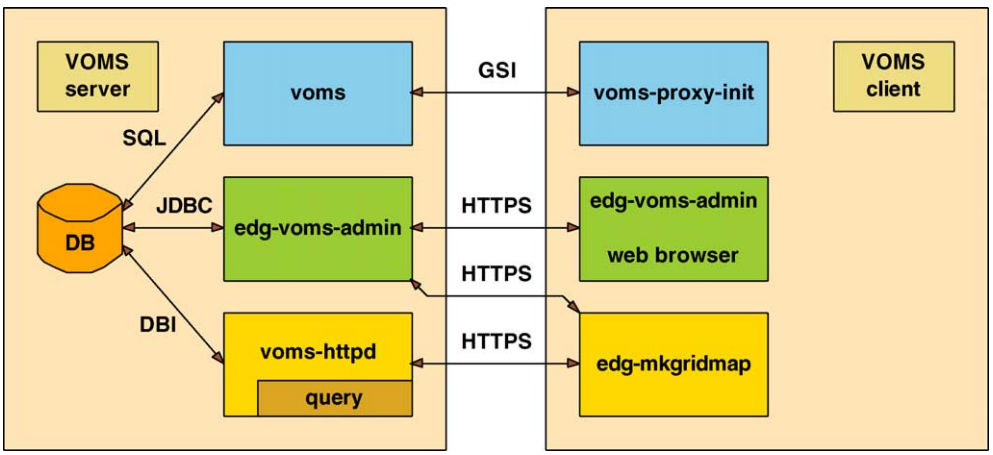


Fig. 2. The VOMS system.

*3.2.1. User part*

The first operation that a user must perform in order to access the Grid in EDG and DataTAG is generating a proxy certificate [19] that will be used to access the resources.

For backward compatibility reasons, we added the command (voms-proxy-init), analogous to the old grid-proxy-init, which generates the user proxy certificate necessary to access the Grid. The difference is that the proxy certificates produced by the new command contain the user authorization information from the VOMS server(s). Information is included in an RFC 3281-style Attribute Certificate [7,15] signed by the VOMS server itself. The user can request certificates from more than one server.

The procedure is the following (see Fig. 4):

1. The user and the VOMS Server authenticate each other using their certificates (via the standard Globus API);
2. The user sends a signed request to the VOMS Server;
3. The VOMS Server verifies the user's identity and checks the syntactic correctness of the request;
4. The VOMS Server sends back to the user the required information (signed by itself);
5. The user checks the validity of the information received;
6. The user optionally repeats this process for other VOMSes;
7. The user creates the proxy certificate containing all the information received from the VOMS Server into a (non-critical) extension;
8. The user may add user-supplied authentication information (e.g. Kerberos tickets).

At the RP level, the authorization information provided by VOMS needs to be extracted from the user's proxy certificate and combined with the local policies in order to make the authorization decision. The *Gatekeeper*, that is, the Grid interface to fabric in Globus-based Grids, first checks the validity of the proxy certificate, then uses an external service in order to process the authorization information.

In EDG and DataTAG, the Local Credential Authorization Service (LCAS) [20] is used to make the authorization decision; it can use both gridmap-files and Attribute Certificates signed by VOMS. However, as the

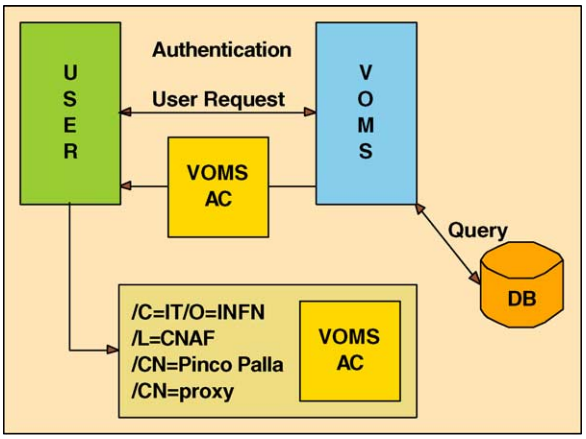


Fig. 4. VOMS operations.

VOMS information is included in a non-critical extension of the certificate, this can be used even by "VOMS-unaware" *Gatekeepers*, thus maintaining compatibility with previous releases. The support for VOMS has been enforced in all other components (e.g. the component in charge of accessing a database, via the *Authorization Manager* [21]).

*3.2.2. Administration*

The Administration Server supports the Simple Object Access Protocol (SOAP) [22] for connections, so that it can be easily converted into an Open Grid Services Architecture (OGSA) [23] service. It consists of five sets of routines grouped into services: the *Core*, which provides the basic functionality for the clients; the *Admin*, which provides the methods to administrate the VOMS database; the *History*, which provides the logging and auditing functionality (the database scheme provides full audit records for every changes); the *Request*, which provides an integrated request handling mechanism for new users and for other changes; and the *Compatibility*, which provides a simple access to the user list for the *mkgridmap* utility.

Two administrative interfaces (web and command line) are available.

*3.2.3. Security considerations*

The authentication to the VOMS server makes use of the standard Grid Security Infrastructure (GSI) [6] security controls on the user's certificate before granting rights; it must be signed by a "trusted" Certification

Authority, be valid and not revoked. Hence, the access control to the VOMS service leverages the robustness of GSI and OpenSSL [24] implementations.

Compromising the VOMS server database could allow a malicious user to grant credentials with access rights for any service, but this would not be enough to grant illegal, indiscriminate access because:

- it is not possible to impersonate a user without also possessing his credentials (including the private key), since the authorization data must be inserted in the user's proxy certificate (i.e. countersigned by the user himself);
- access control ultimately depends on the decisions made by local administrators, who can ban a user regardless of what the credentials say.

Possible large scale vulnerabilities of the VOMS server are denial of service attacks (e.g. to prevent VO users from getting their authorization credentials), but this is not specific to VOMS. The implementation of a replica mechanism will decrease this risk.

The main security issue is a global GSI problem related to proxy certificates: the lack of a revocation mechanism. On the other hand, these certificates have short lifetimes (12 h typically), which limits the problem to a certain extent.

In summary, we think that the VOMS technology does not add significant security problems to the existing GSI-based security infrastructure of the Grid.

## 4. Related work

In this section, we compare the VOMS system with three similar systems: Permis, CAS and Akenti.

### *4.1. PERMIS*

The Privilege and Role Management Infrastructure Standards Validation (PERMIS) [17] system, developed at Salford University, also implements an access control mechanism based on roles. While it may seem extremely similar to VOMS at first glance, there are several important differences:

1. VOMS considers an attribute to be composed of three different elements: group, role and capability. PERMIS only takes into account the role. While this, in principle, is not an important difference since any specific 3-tuple of the VOMS attributes may be simulated by using a properly named PERMIS role, the loss of flexibility entailed by this is, in our opinion, bound to cause problems with large and complex organizations.
2. VOMS distributes the Attribute Certificates it creates to the users themselves, who are then responsible for presenting their certificates whenever they want access to a particular system, or even to request a specific subset of it. Conversely, PERMIS pre-generates the ACs and holds them in a database from which they are retrieved when they are needed. This makes almost impossible for a user to request a subset of his permissions.
3. PERMIS also includes a policy engine that can make decisions based on a policy file and the attribute certificates received. On the contrary, VOMS does not focus on this problem at all and leaves the interpretation of the Attribute Certificates to other components.

In the authors' opinion, PERMIS is clearly superior to VOMS as a policy engine, but because of its architecture, it lacks flexibility to manage large organizations with many members.

### *4.2. CAS*

The Community Authorization Service (CAS) [16] was developed by the Globus team to solve the same problem as the one tackled by VOMS in EDG. In our opinion, there are two major differences between CAS and VOMS.

1. CAS does not issue Attribute Certificates, but a whole new proxy certificate with the CAS server's Distinguished Name as the subject; the authorization information of the user (i.e. the Distinguished Name and attributes) is included in an extension. As a consequence, services that are not specifically CAS-enabled cannot determine the identity of the user to whom the CAS server has granted the certificate. On the contrary, VOMS uses a completely standard proxy certificate, with the addition of a non-critical extension for the authorization information. For this reason, its certificates are also com-

patible with non- VOMS-enabled services, which gracefully ignore the extra data.
2. CAS records user permissions (e.g. down to the control of access to a specific file), as opposed to user attributes. This means that the ultimate decision about the user access to a PC farm is removed from the farm administrator and taken under the control of the CAS administrator. This clearly breaks one of the fundamental rules of the Grid; the farm administrator has total control about what happens on his machines.

### *4.3. Akenti*

Akenti [25] is another authorization system, developed at the Lawrence Berkeley National Laboratory, to facilitate setting access policies by independent organizations, and to provide a virtual, organization-wide user identity. In our opinion, there are three major differences between Akenti and VOMS.

1. Akenti uses a pure pull model. There is no need for the user to request a specific credential from an external authority. The authority is contacted independently by the service once the user has successfully authenticated. This poses flexibility problems because it makes it impossible for users to receive only a subset of their full capabilities without possessing a second independent identity.
2. Like PERMIS, Akenti also contains a policy engine. This feature is outside the scope of VOMS.
3. The data format used by Akenti for its Attribute Certificate is not standard ASN.1 [15], but a non-standard XML format.

## 5. Conclusion

In this paper, we described the authorization technology adopted by the EDG and DataTAG projects for Grid environments. The VOMS architecture provides a centralized – at Virtual Organization level – Authorization Repository for users' attributes.

VOMS, in its current implementation, requires the Globus environment. However, this is not an architectural constraint, with the implementation changes due to the different libraries and basic credential formats, it could also be used as an Attribute Authority with LEGION [26] or PRIMA [27].

Moreover, the use of standard Attribute Certificates makes it possible to inter-operate with different administrative domains and security infrastructures.

We think that in future Grid applications, VOs, now relatively stable and large organizations, will evolve towards much more dynamic entities, composed of few people and short-lived, which can not only coexist but also interact with each other in a secure way, forming federations.

In this respect, we think that the main shortcoming of VOMS is its relatively high weight; the setup (and management perhaps) of a VOMS server is not as easy, as it should be to allow for dynamic VOs.

For this reason, future developments will be in the direction of more lightweight setup and administration schemes, and the availability as an OGSA service. Moreover, we are working to provide support for other RDBMSes, e.g. Oracle, more sophisticate time validity constraints for the VOMS certificates, and support for different operating systems.

## Acknowledgments

This work was partially funded by the European Commission programs IST-2000-25182 and IST-2001-32459 through the European DataGrid and DataTAG projects. One of the authors (K.L.) was supported by by National Office of Research and Technology, Budapest (Hungary).

## References

[1] M. Draoli, G. Mascari, R. Piccinelli, Project Presentation, DataGrid-11-NOT-0103-_1.
[2] http://www. datatag. org/.
[3] R. Alfieri, et al., Managing Dynamic User Communities in a Grid of Autonomous Resources, TUBT005, in: Proceedings of CHEP 2003, 2003.
[4] I. Foster, C. Kesselman. The globus project: a status report, in: Proceedings of the IPPS/SPDP'98 Heterogeneous Computing Workshop, 4–18, 1998.
[5] I. Foster, C. Kesselman, S. Tuecke, The anatomy of the Grid, Int. J. High Perform. Comput. Appl. 15 (2001) 3.
[6] I. Foster, C. Kesselman, G. Tsudik, S. Tuecke, A security architecture for computational grids, in: Proceedings of the Fifth

ACM Conference on Computer and Communications Security Conference, 1998, pp. 83–92.
[7] R. Housley, T. Polk, W. Ford, D. Solo, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, RFC 3280, 2002.
[8] L. Cornwall, J. Jensen, D. Kelsey, A. McNab, EU DataGrid and GridPP authorization and access control, in: Proceedings of the UK e-Science All Hands Meeting 2003, pp. 382–384, EPSRC 2003 (ISBN 1–904425-11–9).
[9] The Grid2003 Project. The Grid2003 Production Grid: Principles and Practice, GriPhy N-2004–23, Grid3 2004–19.
[10] http://www. uscms. org/s&c/VO/doc/doc. html.
[11] http://lcg. web. cern.ch/lcg/.
[12] http://public.eu-egee. org/.
[13] D. Kelsey, L. Cornwall, Security Requirements and Testbed-1 Security Implementation, DataGrid-07-D7.5–0111-4–0, 2002.
[14] Á. Frohner, V. Ciaschini, VOMS Credential Format, EDG Draft, 2004.
[15] S. Farrel, R. Housley, An Internet Attribute Certificate Profile for Authorization, RFC 3281, 2002.
[16] L. Pearlman, V. Welch, I. Foster, K. Kesselman, S. Tuecke, A community authorization service for group collaboration, in: IEEE Workshop on Policies for Distributed Systems and Networks, 2002.
[17] D.W. Chadwick, O. Otenko, The PERMIS X.509 role based privilege management infrastructure, Fut. Gen. Comput. Syst. 19(2), Elsevier Science Publishers B.V., 2003, pp. 277–289.
[18] M. Widenius, D. Axmark, MySQL Reference Manual, 2002 (ISBN 0–596-00265–3).
[19] S. Tuecke, D. Engert, I. Foster, V. Welch, M. Thompson, L. Pearlman, C. Kesselman, Internet X.509 Public Key Infrastructure Proxy Certificate Profile, draft-ggf-gsi-proxy-04, 2002.
[20] Architectural design and evaluation criteria: WP4 Fabric Management, DataGrid-04-D4.2–0119-2–1, 2001.
[21] European DataGrid, Security Coordination Group: Security Design, DataGrid-07-D7.6–0112, 43–45, 2003.
[22] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. Nielsen, S. Thatte, D. Winer, Simple Object Access Protocol (SOAP) 1.1, Note 20000508, 2000.
[23] I. Foster, C. Kesselman, J. Nick, S. Tuecke, The physiology of the grid: an open grid services architecture for distributed systems integration, Open Grid Service Infrastructure WG, Global Grid Forum, 2002.
[24] J. Viega, M. Messier, P. Chandra, Network Security with OpenSSL Cryptography for Secure Communications, 2002 (ISBN: 0–596-00270-X).
[25] M. Thompson, A. Essiari, S. Mudumbai, Certificate-based authorization policy in a PKI environment, ACM Trans. Inform. Syst. Secur. 6 (4) (November 2003) 566–588.
[26] A.S. Grinshaw, Wm. A. Wulf and the whole legion team, Legion – A view From 50,000 Feet., in: Proceedings of the Fifth IEEE International Symposium on High Performance Distributed Computing, 1996.
[27] M. Lorch, et al., The PRIMA system for privilege management, Authorization and enforcement in grid environments, in: Proceedings of Fourth International Workshop on Grid Computing: Grid 2003, Phoenix, AR, USA, 2003.

**Roberto Alfieri** received his MSc degree in Physics (1985) from University of Parma (Italy). He has been Technical Coordinator of the Advanced Computing Laboratory (LCA) at the Physics Department of University of Parma (1986–2001). Since March 2001, he is Assistant Professor in Computer Science at the same University. His research interests are on Grid computing and high-performance computing. As member of the INFN and DataGrid Authorisation groups, he has been involved with the design and the implementation of new authorisation models in a Grid environment. His teaching activities include various graduate courses in Computer Science at the University of Parma.

**Roberto Cecchini** received his MSc degree in Physics in 1978. He has been working in the Computer Security field since 1995. He implemented the INFN Certification Authority in 1997 and is currently managing it. He is also the coordinator of the INFN Security Group since 1998. He founded in 1999 and manages since then the CSIRT of the GARR Network (GARR–CERT). He is the coordinator of the group which develops and mantains the VOMS server.

**Vincenzo Ciaschini** received his MSc degree in Computer Sciences in 2002. He has been since working in the computer security field. Specifically, he has been working in the INFN Authorisation group in the framework of the European DataTAG project, contributing to the development of VOMS. He is now working at INFN in the framework of the European EGEE project, continuing his work on VOMS and on a new Grid-aware policy architecture.

**Luca dell'Agnello** received his MSc degree in Physics (1992) from University of Firenze (Italy). As IT consultant, he has been working in the computer security field since 1994. Since 1996, he has been working for INFN and the GARR network for which he has also been technical representative (1997–2001). He has been one of the founders of the Computer Security Incident Response Team for the GARR network (1999). He has been working in the INFN Authorisation group in the framework of European DataGrid and DataTAG projects, contributing to the development of VOMS. Presently, he is involved in the management of the INFN Tier1.

**Ákos Frohner** received his MSc degree as a software architect and mathematician (1996), and he is about to receive his PhD degree. During his studies, he was a regular lecturer at his university and various commercial companies in computer security, distributed systems and object-oriented design. He is also co-author and editor of books on these fields. Later, he had been working as a research fellow at CERN on grid security. He is currently working on grid data management solutions at CERN in the EGEE project.

**Károly Lõrentey** is an undergraduate student at Eötvös Loránd University, in Hungary. Besides his studies, he has participated in several courses assuming the duties of a demonstrator. He is also a lecturer in operating systems for undergraduate computer scientists. He is currently a CERN Associate working on VOMS integration in the LHC Computing Grid project. He is co-maintaining the VOMS administration interface for the gLite project.

**Fabio Spataro** received his BSc degree in Physical Methodoly (1998) from the University of Roma "La Sapienza" (Italy) and his BSc degree in Physics (2003) from the University of Parma (Italy). From 2000, he is employed by the Parma University as System Administrator at the Advanced Computing Laboratory (LCA). In the Europead DataTAG project, he has been working in the Authorisation group (2001–2003). Teacher of the "C++ Programming" course for undergraduate students in Physics and Material Science at the University of Parma (2004).