Advanced Security Infrastructures for Grid Education

Prof R.O. Sinnott, A.J. Stell, Dr J.P. Watt,
National e-Science Centre,
University of Glasgow
ros@dcs.gla.ac.uk

Prof D.W. Chadwick,
Information Systems Security Group,
University of Kent

Abstract

This paper describes the research conducted into advanced authorization infrastructures at the National e-Science Centre (NeSC) at the University of Glasgow and their application to support a teaching environment as part of the Dynamic Virtual Organisations in e-Science Education (DyVOSE) project. We outline the lessons learnt in teaching Grid computing and rolling out the associated security authorisation infrastructures, and describe our plans for a future, extended security infrastructure for dynamic establishment of inter-institutional virtual organisations (VO) in the education domain.

Keywords: Grid, education, Security, PERMIS, Shibboleth.

1. Introduction

As Grid technology becomes ubiquitous across a wide range of application domains, there is an increasing demand for proven and effective security models and infrastructures. This can only be achieved if there is a generation of developers cognisant of the challenges and solutions that exists in the technologies underlying the Grid. Knowledge transfer and exposure to leading Grid solutions is thus essential for next generation middleware developers. In the current fluid middleware environment, it is especially difficult for educators to produce course materials that will have some kind of longevity and incorporate latest Grid developments.

To meet this challenge there is a need for courses that cover the fundamental principles of Grid computing in conjunction with exploration of today's solutions. Thus whilst there might be numerous technologies say for job scheduling (e.g. Condor [1], Sun Grid Engine [2], OpenPBS [3], Maui [4]), the basic principles of job scheduling and the specific demands of large scale, wide area job scheduling remain the same. The NeSC at the University of Glasgow has established a Grid Computing module as part of the advanced MSc in Computing Science addressing these challenges. This is one of the first full Grid computing courses available today.

Security is one area where education is critical to the future acceptance and take-up of the Grid, and has been a key aspect of the Grid Computing module at Glasgow. Understanding the technical and non-technical aspects associated with security is crucial, not least due to the degree of trust between resource providers and the potentially highly distributed, remote end users. For the most part, the Grid community has focused primarily upon *authentication* – verifying that users are who they say they are. This has largely been implemented using Public Key Infrastructures (PKIs) [5]. Through PKIs, it is possible to validate the identity of a given user requesting access to a given resource. For example, with the Globus toolkit solution [6], gatekeepers are used to ensure that signed requests are valid, i.e. from known collaborators. When this is so, i.e. the Distinguished Name (DN) of the requestor is in a locally stored and managed *gridmap* file, the user is typically given access to local account as defined in the *gridmap* file.

There are several key limitations with this approach with regard to security. For example, the level of granularity of security is limited. There is no mention of what the user is allowed to do once they have gained access to the resource. Further, this approach works on the assumption that user certificates are provided by an acknowledged certificate authority (CA). In the UK, a centrally managed CA at Rutherford Appleton Laboratories exists [7] which (necessarily) has strict procedures for how certificates are allocated. Users are expected to "prove" who they are in order to get a certificate, e.g. through presenting their passports to a trusted individual at their institution (contacted by the CA). This is a human intensive activity and one which has scalability issues once it is rolled out to the wider community, e.g. to industry and larger groups such as students taking Grid/e-Science courses. Having users personally take care of their private keys is another major limitation of this approach. The passwords associated with these private keys are necessarily strong, and as a consequence users are liable to write them down, thereby seriously compromising the overall Grid security.

In short, current experiences with PKIs as the mechanism for ensuring security on the Grid have not been too successful [8,9]. Whilst being a widely accepted foundation for security, authentication on its own is insufficient for fine grained control. *Authorisation* – defining and enforcing what end users are allowed to do on local resources – is essential. Authorisation infrastructures offer extended and finer grained security control when accessing and using Grid resources. Many authorisation solutions exist today, often using different paradigms of operation [10-13]. Examples of how these compare to one another are described in [14-16].

It is clear that defining and managing detailed policies on access to and usage of site resources will face scalability issues for large scale Grid infrastructures where many different users, services and resources exist. This is further compounded when new users join, leave, new resources are added and removed etc. Having a single centralised authority to manage a security infrastructure at a given site is not

realistic for large scale, evolving Grid infrastructures. Instead dynamic (rather than static) delegation of authority is required. Static delegation of authority implies that a central authority has to be contacted, and register local managers in its policy, before managers are entitled to assign privileges to subordinates. With dynamic delegation of authority, however, local managers do not need to be registered, but are given the privilege to delegate when they are first given privileges to use the system. Managers can then allocate privileges to staff and students as required, without having to contact the central authority first to get permission. Through this, a federated and scalable model of security authorisation can be realised. The DyVOSE project [17] has developed a dynamic delegation issuing service which supports such dynamic delegation of authority. Given the novelty of this security solution, large scale practical explorations of such extended authorisation infrastructures in realistic environments such as education are essential.

We note that these security models and solutions are broadly applicable across most Grids today, not just education, since they address the key challenge of dynamically linking collections of distributed individuals and resources together *in a secure manner* to form so called Virtual Organisations (VOs). Typically a VO will allow a collection of individuals and/or institutions to pool resources such as data sets, data archives, CPUs, or allow access to specialised equipment from astronomical radio-telescopes through to medical imaging scanners. With the open and collaborative nature of the Grid, ensuring that local security constraints are met and not weakened by Grid security solutions is paramount.

The rest of the paper is structured as follows. Section 2 provides an overview of the Grid Computing module contents and a justification for, and exploration of, the course structure. Section 3 provides an outline of existing authorisation infrastructures and describes in detail the PERMIS role based access control software used in teaching at Glasgow. Section 4 explores the experiences in applying these security infrastructures in a teaching environment and section 5 outlines the lessons learnt and plans for the future.

2. Grid Computing Module Composition

The Grid Computing module at the University of Glasgow was designed specifically to train future Grid engineers. One of the greatest challenges we faced in developing and delivering materials for educating future Grid engineers was (is!) the fluidity of the technological landscape. Grid technology and associated standards are continually evolving in a radical manner with new recommendations and software from standards bodies and solutions providers. This has been exemplified in recent times with the move from pre-web service based Grid infrastructures [18,19] to Open Grid Service Infrastructure (OGSI) based Grid services [20] and the current move towards Web Service Resource Framework (WSRF) web/Grid services [21]. The evolution of the Open Grid Service Architecture (OGSA) is also a key issue that makes the development and delivery of any form of education or materials difficult. Trainers and educators need to be sure that they are developing materials which has some expectancy of life time. Developing and delivering educational materials based upon explicit technology, e.g. Globus toolkit version 3, are fraught with dangers associated with a moving technology base. The nature of the Grid computing module at Glasgow was explicitly designed with these issues in mind. The overall structure of the Grid Computing module is given in Table 1.

Wk 1	Lecture 1	Introduction to Grid Computing
	Lecture 2	Scalability and Heterogeneity
Wk 2	Tutorial 1	Discussion of seminal Grid papers
	Lecture 3	Open standards and architectures
	Lecture 4	Implementations of Grid architecture
Wk 3	Lecture 5	Web services
	Lecture 6	Resource discovery and info. services
	Tutorial 2	Exploring web services with GT3
Wk 4	Lecture 7	Grid security concepts
	Lecture 8	Virtual organizations
	Lecture 9	Security in practise
Wk 5	Tutorial 3	Review of Grid security papers/Lab
	Lecture 10	Job scheduling and management
	Lecture 11	Job scheduling and management
Wk 6	Tutorial 4	Review of job scheduling papers
	Lecture 12	Workflow management
	Tutorial 5	Q&A on programming exercise
Wk 7	Lecture 13	Data access, integration and mgt
	Lecture 14	Data provenance and curation
	Tutorial 6	Review of data mgt/provenance
Wk 8	Lecture 15	Bulk Data Transfer
	Lecture 16	Peer-to-peer communication
	Tutorial 7	Discussion of networking papers
Wk 9	Lecture 17	Tools for Collaboration
	Tutorial 8	Discussion on future of Grid
	Lecture 18	The future of Grid Computing
Wk 10	Lecture 19	Sample applications
	Lecture 20	Review of major concepts

Table 1: Grid Computing module contents

This course structure was designed to give an overall impression of the key challenges and distinguishing characteristics of Grid computing. Linkage to previous work and architectures in distributed computing, and more recent activities such as peer to peer systems was deliberately undertaken to put Grid computing into perspective. It is a fact that many of the concepts associated with Grid computing are a refactoring of previous distributed systems ideas. Where Grid computing differs however is in scale, e.g. managing peta-bytes of data poses new computing science research challenges. Open challenges and unsolved issues such as long term data curation and data provenance were outlined in the course to give the students an awareness of research frontiers.

Establishing a course based solely upon principles and challenges associated with Grid technologies, is

unlikely to be suitable for a full time advanced course. Experiments and investigations using current state of the art in Grid technology are needed. At Glasgow this was through use of OGSI versions of the Globus toolkit [21] and Condor [1] (amongst other technologies), however, we emphasise that this technology did not provide the cornerstone of the educational material. Rather it provided a vehicle through which many of the basic principles could be demonstrated. It is this perspective we believe that underpins the difference between training and education more generally. Courses designed to train e-Scientists would have radically different characteristics and be more focused upon how to use existing technologies.

A key requirement on Grid education is a broad scope and balance. Grid technology touches on many areas from security, usability, job scheduling and data management etc and developing single courses attempting to provide a complete picture of Grid today needs to be targeted to the right audience. Whilst high level overviews of Grid can be provided say to undergraduate students, it is more likely the case that complete and detailed overview materials are best delivered to computer science students that have the necessary grounding in related materials. At Glasgow, various pre-requisites were in place for students wishing to take the Grid Computing module. Students were expected to either have taken various courses at Glasgow such as advanced networking systems, operating systems, distributed systems and algorithms etc, or have knowledge of the contents of these courses. This impacted upon the level of difficulty of the programming assignments which were developed to test advanced and knowledgeable computer scientists, as opposed to less experienced (novice) undergraduate students. That said the lecture material (as opposed to the implementation work) is more generic in nature and will we hope be more easily transferable to the wider community. Several sites have requested permission to re-use these teaching materials which we have granted.

It is also worth noting the strong emphasis on security in this course both in terms of lecture material and implementation/assignment work. The lectures on security provided an overview of the challenges of making Grids secure including concepts such as authentication, authorisation, accounting, auditing, confidentiality, privacy, data integrity, and trust. Exploration of current Grid security mechanisms, e.g. PKI based authentication and Globus GSI [6] based individual service/user based authorisation was presented, with focus on the many open challenges to be addressed to realise robust, scalable Grid security. Lectures addressing other aspects of Grid Computing were delivered in a similar manner, each with an emphasis on their own idiosyncratic issues. The structure of the Grid computing course itself and the lecture materials, associated background reading and tutorials on setting up secure Grid infrastructures for teaching purposes are available at [17].

We also felt that it was important to emphasise real working Grid solutions in a variety of application domains. Live demonstrations of significant Grids were presented to the students in later lectures – showing how real science is undertaken on large scale compute and data Grid infrastructures. We focused in particular on the life science domain [23] but outlined solutions from a wide variety of other domains such as nano-engineering and particle physics.

The module itself was assessed by a combination of a written examination (70%) and marked coursework (30%). The marked coursework consisted of three smaller problem sets and one large programming assignment. This course has been run one time thus far and it is planned that it will be repeated in early 2006. Sixteen students took the course the first time around – a significant amount for an elective module held for the first time. The infrastructure used in the course of the teaching consisted of a training laboratory at the NeSC at University of Glasgow comprising 20 PCs – each with Pentium III processors with 512MB RAM. Each PC had the associated technologies (Condor, Globus, etc) preinstalled and configured for students.

3. Background to Advanced Authorisation Infrastructures

In a Grid environment, authentication (being able to establish the identity of a user) should be augmented with authorisation capabilities, which can be considered as what Grid users are allowed to do on a given Grid end-system. Thus “what users are allowed to do” can be interpreted as the privileges that the users have been allocated on those end-systems. The X.509 standard [24] has standardised the certificates of a PMI. A PMI can be considered as being related to authorisation in much the same way as a PKI is related to authentication. Consequently, there are many similar concepts in PKIs and PMIs. An outline of these concepts and their relationship are discussed in detail in [25].

The Privilege and Role Management Infrastructure Standards Validation (PERMIS) project [26] was an EC project that built an authorisation infrastructure to realise a scalable X.509 attribute certificate (AC) based PMI. Through PERMIS, an alternative and more scalable approach to centrally allocated X.509 public key certificates can be achieved through the issuance of locally allocated X.509 ACs. The PERMIS software realises a Role Based Access Control (RBAC) authorisation infrastructure. It offers a standards-based Java API that allows developers of resource gateways (gatekeepers) to enquire if a particular access to a resource should be allowed. In addition, PERMIS realises the generic Security Assertion Markup Language (SAML) [27] AuthZ API [28] put forward by the Global Grid Forum [29]. This API provides a generic policy enforcement point (PEP) that can be associated with an arbitrary authorisation infrastructure. Thus rather than developers having to explicitly engineer a security policy checks on a per application basis, the information contained within the deployment

descriptor file (.wsdd) when the service is deployed within the container, is used. Authorisation checks on users attempting to invoke "methods" associated with a given service are then made using the information in the .wsdd file and the digitally signed (and tamper proof!) security policies defined and stored within the LDAP repository (Policy Decision Point (PDP) in X.509 parlance) together with the DN of the user. Note that this "method" authorisation basis extends current security mechanisms such as GSI which work on a per service/container basis. The Globus toolkit (version GT3.3+) and PERMIS both support this API.

The PERMIS RBAC system itself uses XML based policies defining rules, specifying which access control decisions are to be made for given VO resources. These rules include definitions of: subjects that can be assigned roles (students, staff etc); Source of Authority (SOA), e.g. local managers trusted to assign roles to subjects; roles and their hierarchical relationships; what roles can be assigned to which subjects by which SOAs; target resources, and the actions that can be applied to them; which roles are allowed to perform which actions on which targets, and the conditions under which access can be granted to roles. Roles are assigned to subjects by issuing them with X.509 Attribute Certificate(s). A graphical tool called the Privilege Allocator (PA) has been developed to support this process. Once roles are assigned, and policies developed, they are digitally signed by a manager and stored in one or more LDAP repositories.

To set up and administer PERMIS requires the use of a LDAP server to store the attribute certificates and reference the SOA root certificate. A local CA is required to be set up – at Glasgow we used OpenSSL [30] – this designates the SOA and all user certificates created from this CA must have a Distinguished Name that matches the structure of the LDAP server. The DN of the user certificate is what is used to identify the client making the call on the Grid service. From the user's perspective, once the administrator has set up the infrastructure, the PERMIS service is relatively easy to use. Unique identifiers are placed as parameters into the user's grid service deployment descriptor (.wsdd file). These are the Object Identification number of the policy in the repository, the URI of the LDAP server where the policies are held and the SOA associated with the policy being implemented. Once these parameters are input and the service is deployed, the user creates a proxy certificate with the user certificate created by the local CA to perform strong authentication. The client is run and the authorisation process allows or refuses the intended action in a generic and transparent manner.

4. Exploration of the Advanced Security Infrastructure

In exploring the advanced security infrastructure, the students were initially expected to develop their own security policies (in the second problem assignment set) for a basic GT3.3 based Grid service which was subsequently used in their main programming assignment.

Specifically the students were requested to create a policy for a GT3.3 service (*searchSortGridService*) which wrapped a Condor based application (this service offered two methods to search (*searchMethod*) and sort (*sortMethod*) a large (5MB) text file (the complete works of Shakespeare). The students themselves were split into groups (*studentteam1*, *studentteam2*) with the authorisation policy to ensure that method *sortMethod* could only be invoked by members of their student group and the lecturing staff, whilst method *searchMethod* could be invoked by everyone. This set-up was used to illustrate the use of RBAC, where users are allocated privileges based on what role they have been assigned rather than their local user credentials. The students were also requested to secure their service using Globus GSI (which provides service based security) and also with PERMIS (which uses finer grained based method level security). Performance aspects and benchmarks for the speed of the different systems were recorded by the students.

The intention of this assignment was multi-fold. We wanted to: undertake a detailed exploration of the PERMIS tool family (including the Policy Editor and the Privilege Allocator; explore in detail and document the usability of the GGF AuthZ SAML interface; take the students through a trivial Java programming exercise through to addressing the challenges of developing and deploying applications across a Grid infrastructure; gain an appreciation of the performance aspects when Grid middleware and associated security infrastructures are used.

Students could implement this system any way that they chose and a variety of search and sort methods were implemented – we deliberately told the students that we did not care how performant their implementations for search/sort were. Rather we were more interested in the performance impact of the Grid middleware on their implemented algorithms and their experiences of Grid technologies as a whole.

4.1 Observations and Feedback

Considerable feedback was generated on the general usability of the PERMIS policy editing tools which was subsequently sent to the PERMIS team (and has since been incorporated into their later releases). *All* students were able to create security policies using these tools however some students suggested that the HCI aspects of the tool (explicitly coded to be suited to non-computer literate folk) should be removed. This was counter to the HCI expert suggestions which had been incorporated into the tools' user interface on making them easier to use!

Most students were also able to develop the Condor based version of their search/sort system. A variety of solutions were implemented using Condor. Some students allowed the user to select how many nodes the job should be distributed over. Other students farmed out the data with the jobs whilst others came up with solutions whereby the data was pre-deployed.

However of the 16 students that took this module *only four* managed to successfully engineer the Globus

GT3.3 based version which wrapped the Condor version of their search/sort system. Of these four, two managed to get the PERMIS based solution working, whilst all four managed to get the GSI version working. It has to be said however that the students at Glasgow had significantly different levels of programming ability and experience of associated background technologies. The overall performance aspects of the different implementations are presented in table 2.

	Search (s)	**Sort (s)**
Single Processor	1.7 ± 0.4	5.7 ± 3.3
Condor Pool (16 nodes)	62.2 ± 4.4	60.7 ± 3.1
Condor Pool (4 nodes)	29.5 ± 6.9	35.2 ± 1.8
Grid Service (4 nodes)	31.8 ± 5.9	37.6 ± 11.2
GSI (4 nodes)	39.9 ± 8.6	48.3 ± 15.3
PERMIS (4 nodes)	34.5 ± 8.6	38.5 ± 9.8

Table 2: Job Completion Times

As may be seen it was far quicker to search and sort the file on a single PC. The overheads in distributing the sort/search algorithms were significant and typically resulted in taking over one minute to search and to sort the file using all of the nodes in the pool. The reasons for this are primarily due to the overheads involved in farming out the jobs across a network and collecting and merging the results. The time taken to split the text files, traverse the local network, prepare the Condor jobs, process them, come back to the original machine and concatenate the final results gave a significant time overhead.

A further key factor in the performance is due to the job being completed when *all* distributed Condor jobs have completed, i.e. one queued or delayed job delays the overall time. Other issues that contributed were the high network latency and non-deterministic nature of benchmarking on a multi-user system. The extent of the delays caused by these issues was nevertheless surprising.

The GSI-based authorisation of the application also resulted in a significant increase in the overall time required to complete the search/sort (approximately 8 seconds). The PERMIS based authorisation of the search/sort application took approximately 3 seconds more than the unsecured service. The reasons for these increases, compared to the unsecured service, are due to the time overhead in consulting the *gridmap* file and the LDAP repository, respectively, then proceeding through the necessary stages of credential validation. Once again the time overheads were surprising.

Of the students that managed to complete the full exercise, numerous observations on the state of the Grid middleware were made. Many of these were not especially positive. For example, in other courses at Glasgow students were asked to implement much more complicated distributed systems using Java RMI, and were quite scathing about how complicated Grid middleware is to use to implement such a seemingly basic distributed application.

The lack of programming environments and debuggers was also identified. Students often resorted to using web search engines for debugging purposes as opposed to middleware documentation. More often than not, students identified that the result sets returned from such searches contained other users who had faced similar problems with no answers being found. We note that leaving these students to resolve these issues largely by themselves was deliberate. This was an advanced computing course where we expected students to solve implementation issues themselves. That said it was often the case that direct help was necessary when students faced non-resolvable implementation errors.

Despite this we note that four students also went on to complete their advanced MSc dissertations in Grid related research and technologies.

5. Lessons Learnt and Future Work

One of the main challenges in teaching Grid computing we faced is striking a balance between what is achievable in terms of implementation and what can constitute ground-breaking research. For example, linking advanced security and Grid infrastructures is still non-trivial and there are numerous things that cannot be easily achieved right now, e.g. restricting access to subsets of data in evolving databases. Establishing the level of difficulty of implementation work is also non-trivial and much has been learnt in the first running of this course. Thus whilst searching and sorting a file is an almost trivial computing exercise for a student (never mind an advanced student), developing secure Grid services utilising Condor pools for searching and sorting proved a major challenge to students. For the upcoming running of the Grid Computing module we thus plan to hold more lab sessions where more hands on guidance and exploration of the technologies is undertaken. The knowledge base we have now established in running the course for the first time cannot be emphasised enough. The theory of Grid computing and the associated technologies is one thing and rolling-out a full advanced course exploring toolsets in detail is another. For example, one unconsidered issue that arose was in students using the same PCs for development. Typically short term (12 hour) proxy credentials are created by users using their own local certificates for Grid development and testing. However, when other students later used this PC (the PC was not closed down as it formed part of the Condor pool) conflicts arose with the existing credentials that existed. To resolve this issue, we decided that individual students would be allocated their own dedicated PCs. Disseminating such knowledge to the wider Grid and education community is essential for the overall success of Grid and e-Science technologies, and something we have been actively pursuing for example at e-Science education workshops [31].

Establishing a static privilege management infrastructure for teaching purposes where security policies are defined locally in advance and used to

restrict access to Grid services has been demonstrated, and we have seen that this can work. In the wider Grid world however, there will typically be many "local" security infrastructures each with their own security policies. Dynamically linking such infrastructures together – as essential in establishing VOs - is the focus of the last phase of DyVOSE. A delegation issuing service has now been implemented allowing local security administrators to delegate privileges to remote administrators to issue attribute certificates in a controlled manner for access to and usage of local resources. Through this, the issues in understanding heterogeneous roles, targets and associated actions in a distributed setting can be addressed. To explore this inter-institutional education scenario, use cases are being established with the University of Edinburgh where multiple security infrastructures are to be dynamically and securely linked.

5.1. Acknowledgements

The DyVOSE project is funded by a grant from the Joint Information System Committee. The authors would like to thank the collaborators in the project including Professor David Chadwick and Dr Sassa Otenko, University of Kent, and Dr Colin Perkins at the University of Glasgow.

6. References

[1] Condor, www.cs.wisc.edu/condor
[2] Sun Grid Engine, http://www.sun.com/software/gridware/index.xml
[3] Open Portable Batch System (OpenPBS), www.openpbs.org
[4] Maui Cluster Scheduler, www.clusterresources.com/products/maui/
[5] R. Housley, T. Polk, Planning for PKI: Best Practices Guide for Deploying Public Key Infrastructures, Wiley Computer Publishing, 2001.
[6] Globus Grid Security Infrastructure, http://www.globus.org/security/
[7] UK Certification Authority, www.grid-support.ac.uk/
[8] JISC Authentication, Authorisation and Accounting (AAA) Programme Technologies for Information Environment Security (TIES), http://www.edina.ac.uk/projects/ties/ties_23-9.pdf
[9] R.O. Sinnott, A.J. Stell, D.W. Chadwick, O.Otenko, Experiences of Applying Advanced Grid Authorisation Infrastructures, Proceedings of European Grid Conference (EGC), pages 265-275, Vol. editors: P.M.A. Sloot, et al June 2005, Amsterdam, Holland.
[10] Johnston, W., et al, M. Authorization and Attribute Certificates for Widely Distributed Access Control, IEEE 7th Int. Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, June, 1998.
[11] L Pearlman, et al., A Community Authorisation Service for Group Collaboration, Proceedings of IEEE 3rd International Workshop on Policies for Distributed Systems and Networks. 2002.
[12] Lepro, R., Cardea: Dynamic Access Control in Distributed Systems, NASA Technical Report NAS-03-020, November 2003.
[13] D.W.Chadwick, A. Otenko. The PERMIS X.509 Role Based Privilege Management Infrastructure, Proc 7th ACM Symposium On Access Control Models And Technologies (SACMAT 2002), pp 135-140, Monterey, USA, June 2002.
[14] R.O. Sinnott, A.J. Stell, J. Watt, Comparison of Advanced Authorisation Infrastructures for Grid Computing, Proceedings of International Conference on High Performance Computing Systems and Applications, May 2005, Guelph, Canada.
[15] A.J. Stell, Grid Security: An Evaluation of Authorisation Infrastructures for Grid Computing, MSc Dissertation, University of Glasgow, 2004.
[16] D. Chadwick and O. Otenko, A Comparison of the Akenti and PERMIS Authorization Infrastructures in Ensuring Security in IT Infrastructures, Proceedings of the ITI First International Conference on Information and Communications Technology (ICICT 2003) Cairo University, pages 5-26, 2003.
[17] Dynamic Virtual Organisations for e-Science Education (DyVOSE) project, www.nesc.ac.uk/hub/projects/dyvose
[18] UNICORE Forum, www.unicore.org
[19] Globus toolkit version 2, http://www.globus.org/toolkit/downloads/2.4.3/
[20] Open Grid Service Infrastructure (OGSI) version 1.0, http://www-unix.globus.org/toolkit/draft-ggf-ogsi-gridservice-33_2003-06-27.pdf
[21] Globus toolkit version 3, http://www.globus.org/toolkit/downloads/3.0.2/
[22] Web Service Resource Framework, http://www.globus.org/wsrf/
[23] Biomedical Research Informatics Delivered by Grid Enabled Services (BRIDGES) project, www.nesc.ac.uk/hub/projects/bridges
[24] ITU-T Recommendation X.509 (2001) | ISO/IEC 9594-8: 2001, Information technology – Open Systems Interconnection – Public-Key and Attribute Certificate Frameworks.
[25] D.W.Chadwick, A. Otenko, The PERMIS X.509 Role Based Privilege Management Infrastructure, Future Generation Computer Systems, 936 (2002) 1–13, December 2002. Elsevier Science BV.
[26] PERMIS software, http://www.openpermis.org
[27] OASIS, Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) v1.1, 2 September 2003, http://www.oasis-open.org/committees/security.
[28] Authorization Frameworks and Mechanisms WG https://forge.gridforum.org/projects/authz-wg
[29] Global Grid Forum, www.ggf.org
[30] OpenSSL: The Open Source toolkit for SSL/TLS, www.openssl.org
[31] R.O. Sinnott, Teaching Grid Computing, Workshop on Education and Training in UK e-Science, Edinburgh, November 2004, www.nesc.ac.uk/esi/events/487