A Secure and Scalable Grid-based Content Management System

Benjamin Aziz, Alvaro Arenas
STFC Rutherford Appleton Laboratory
Oxfordshire OX11 0QX, United Kingdom
{*benjamin.aziz, alvaro.arenas*}*@stfc.ac.uk*

Giovanni Cortese
Interplay Software S.r.l
Trento, Italy
g.cortese@ipsoft.it

Bruno Crispo
University of Trento
Trento, Italy
bruno.crispo@unitn.it

Silvio Causetti
Istituto Geografico De Agostini S.p.A.
Novara, Italy
silvio.causetti@deagostini.it

***Abstract*—We present in this paper a secure and scalable Grid-based content management system for the management of high-volume multimedia data in the domain of the publishing industry. This is achieved by leveraging on existing individual solutions, such as the Alfresco content management system, the SRM standard for building scalable solutions based on the Grid and the GridTrust services for building trustworthy and secure Grid systems. Our solution brings closer the use of the Grid to the enterprise community within the context of a real world use case scenario. The solution facilitates the fine-grained usage control of the storage resources and a reputation-based matching between resource policies and users' past behaviour.**

***Keywords*-Security; Grids; Scalability; Knowledge Management**

I. Introduction

A content management system allows organisations to collaboratively create, edit, manage and publish any type of digital information such as text, images, video, sound, documents, etc. This kind of systems are characterised by the need for efficient storage and retrieval of high volume of content under strict rules controlling the sharing of information among users and organisations. This paper presents the implementation of a content management system using Grids as the underlying technology.

Our approach has consisted in adapting an open-source content management system (Alfresco [1]) to use the standard Storage Resource Manager (SRM) [2] to access Grid storage, taking into account security needs required by content management systems, such as usage control of resources at service-level and obligation policies. There are advantages in using the Grid for content management systems. Firstly, it allows the management of digital information across different administrative domains; secondly, the use of standardised access to Grid resources allows designers to implement the content management system on the available implementations; and thirdly, it is fairly straightforward to integrate other solutions, for example to enhance security and usage control, to achieve better characteristics such as robustness and dependability.

In order to develop a Grid-based content management system, we have borrowed some concepts and ideas applied by the Virtual Enterprises community. To support the rapid formation of dynamic and short-lived Virtual Organisations (VOs), we use the concept of Virtual Breeding Environment (VBE) [3]. A VBE can be defined as an association of organisations adhering to common operating principles and infrastructure with the main objective of participating in potential VOs. We have adopted the view that organisations participating in a VO are selected from a VBE; such organisations may provide resources/services, and include users that utilise VO resources.

The main contributions of this paper are as follows:

- It brings the Grid closer to the enterprise community through the development of a prototype solution for achieving secure and scalable Grid-based content management within the domain of publishing industry.
- It facilitates better fine-grained usage control of Grid storage resources through incorporating a policy decision point within the proposed solution, which is capable of enforcing usage control policies [4].
- It provides a history-based reputation measure for matching resource usage policies with users, based on users' past behaviour.

The structure of the paper is the following. In Section II, we give an overview of a real world use case from the domain of multimedia publishing industry, which motivated the work presented here. We also review technologies from literature for achieving user-friendly content management and scalability. Then in Section III, we give an overview of some Grid security solutions relevant to the case study and we present our approach in integrating these solutions with the user-friendly content management and scalability solutions. In Section IV, we define the architecture and prototype description and present the implementation of our prototype solution. In Section V, we discuss some performance results for our solution from the perspective of one of its components. Finally, Section VI concludes the

paper and gives directions for future work.

II. Case Study: Collaborations in the Publishing Industry

In the publishing industry, companies may collaborate under the umbrella of VOs in order to produce and use data and documents related to all aspects of knowledge, such as, for example, books, magazines, cartographic documents and touristic material. Hence, in a *Content Management Virtual Organization* (CMVO), multimedia content is stored and shared through services offered by the participating publishers. The owner of the CMVO can use it to carry out the different phases of the publishing workflow, starting from content gathering and ending with the final media product.

A CMVO, as shown in Figure 1, has three main actors:

- *Publishers*, who are responsible for creating the product for their clients based on well-established Service-Level Agreements (SLAs) stating the deadline, cost and desired quality of the products.
- *Content Owners*, who own the multimedia content. These may not be the same as the Publishers, but may have established a contract with the Publishers allowing them to utilize their databases.
- *Server Farms*, that provide extra computational and storage resources to the Publishers in order for the latter to be able to meet their client SLA commitments.

Additionally, there could be other actors such as the *Reviewers* of the digital content and the products. Some possible interactions among these actors are shown in Figure 1.

Such a CMVO highlights a number of issues:

1 Scalable provisioning of resources: The processing of high-quality digital multimedia content, such as cartographic maps and images, can be a resource-intensive process in which CPU power and disk storage pose a challenge to the task of producing the end products. Therefore, a solution for scalable provisioning of resources in such a collaborative environment is needed.

2 Trust and Security: As in any collaborative environment, permitting the sharing of resources raises issues of trust and security both at the local resource level as well as the global VO level. This is particularly of relevance assuming that such sharing is carried out across administrative and trust domains.

3 User-friendly content management: The publishers are mainly concerned with achieving their tasks leading to the the final goal, i.e. publishing and selling the end product. Therefore, any content management system they use must be friendly to the domain of their expertise and usage. Additional technical functionality must be hidden from these users of the system.

Within the scope of this case study, some of the main tools that are used for addressing the user-friendly content management and the scalability requirements are Alfresco and the Grid SRM systems, respectively.

A. Alfresco

Alfresco Enterprise Content Management [1] is an open source application platform for enterprise-level content management, which provides a user-friendly interface for most business domains dealing with content management. The platform is comprised of a repository specialized for content storage, based on the standards JCR-170[1] and CMIS [5], a set of content services (locking, versioning, metadata, search, workflow etc.), a set of APIs (e.g. SOAP [6] and REST Web Services [7]) for external applications to access the content repository and services, and a graphical user interface components to access the content services.

The Alfresco architecture is J2EE server-based. Files are stored by default on a content store based on a file system, and metadata are stored on a Relational Database Management System (RDBMS). The simplest configuration uses a single content store and a single server. More complex installations can have multiple, replicated content stores (on different storage media) and multiple servers in clusters. With respect to security, Alfresco provides authentication and authorization features, based essentially on role-based access control [8]. Finally, the platform is highly extensible via Java programming for integration with other solutions.

B. Grid Storage Resource Management

One solution to the problem of scalability in the implementation of content management systems is to use Grid computing. The challenge is to provision storage in a way that is both flexible and affordable and to store digital content with considerable size. With respect to storage, several architectures exist to implement scalable content management repositories. In a business environment, solutions to achieve content storage scalability based on clustered architectures have been developed. For example, Content Addressable Storage (CAStor) [9] and Centera[2] allow easy plug-in of new storage units on demand.

More recently, solutions based on Grid and Cloud computing allow an organization to exploit storage services on a pay-per-use metered basis. Examples in this class include Amazon's Simple Storage Service (S3) and Nirvanix. In the scientific applications domain, the Open Grid Forum's Storage Resource Manager (SRM) [2] standard has evolved from several Grid projects developing storage management infrastructures and providing scientific applications with a uniform software interface to reserve and store files. SRM, defines a common service-oriented interface for accessing storage resources in order to allow clients to use storage resources from several providers.

III. GridTrust Security Services

While several scalable solutions for content management have been proposed, almost none of them provides flexible

[1] http://jcp.org/en/jsr/detail?id=170

[2] http://www.emc.com/products/family/emc-centera-family.htm

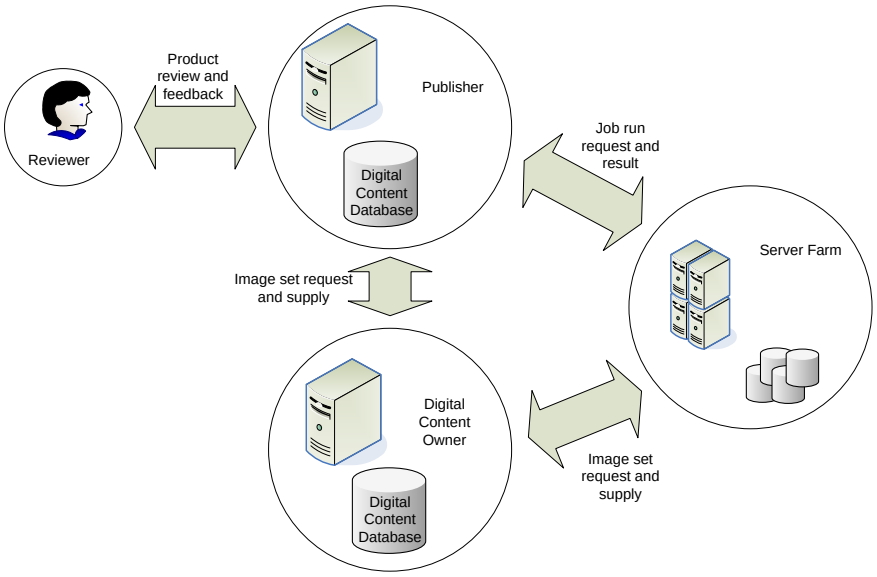

Figure 1. The Content Management Virtual Organization.

and rich enough support for managing trust and security requirements of users and service providers. Thus far, we discussed how a scalable content management infrastructure can be built using Grid architectures thanks to standards such as OGF's SRM. However, one of the main requirements of the CMVO scenario is to provide a secure and trustworthy collaborative environment, in which the sharing of resources requires certain quality of protection. Such requirements could include the definition and deployment of security and usage policies for distributed Grid systems.

Project GridTrust [3] provides a technical framework consisting of theoretical models, design tools and Globus-based software services for the enhancement of trust and security in Grid computing [10]. One of the main research novelties in GridTrust is the adoption of a vertical view, which starts at the level of requirements engineering and system design, and ends at the applications, middleware and foundation levels. For example, access and usage control VO policies generated from domain-specific security requirements can be enforced at the VO level or be refined and enforced at the computational level (middleware and foundational).

Here, we review three services that were developed in GridTrust and that were utilised in our solution. These are the *VBE Certificate Authority*, which is used to establish Public-Key Infrastructure (PKI) trust relations within the system, the *Enforcer*, which provides a policy decision point for evaluating and enforcing usage control policies, and finally, the *Reputation Management* service used for the matching of resource policies with users' past behaviour.

The VBE Certificate Authority: The project developed a VBE Manager service, which aims at coordinating the registration of users and service providers in the VBE as well as the creation of VOs and populating them with users and service providers. As part of this service, a Certificate Authority (CA) is provided, which issues users and service providers with identity certificates. The CA uses the Bouncy Castle library[4] to generate X-509 certificates [11]. It also uses the certificate and private key of a Globus SimpleCA for signing certificates.

The Enforcer Component: The Enforcer receives requests to access and use content, it then evaluates these requests against the security policies that apply to the required content and enforces the final decision by allowing/rejecting the request. Policies are written in XACML [12] and the Enforcer extends the XACML runtime support in order to support not only access control but also usage control policies [4]. In particular, it supports object and subject attributes, including mutable attributes, such that it can enforce all types of history-based policies. Furthermore, the Enforcer implements *obligations*, which are actions that must happen before, during or after the access to the resource (e.g. the content must be deleted after 20 days from when it is uploaded for the first time).

The Reputation Management Service: The responsibility of the reputation management service is to keep track of the past behaviour of users of a VO and transform such usage history information in reputation credentials that can be considered by service providers when taking decisions. The service builds reputation by collecting feedback from the usage control services, such as the Enforcer, which monitors whether users of the VO make appropriate usage of the VO resources granted to them [13]. Security and usage control policies establish what is meant by *appropriate usage* of VO resources and therefore define the expected behaviour that users should follow when using those resources.

[3]http://www.gridtrust.eu

[4]http://www.bouncycastle.org/java.html

A. Our Approach

The solution adopted in this paper is based on building a prototype that glues all the above technologies into one system, which we call the *Distributed Content Management Demonstrator* (DCMD). The purpose behind DCMD is to provide a platform for secure scalable content management in the business world based on the Grid paradigm. More specifically, the DCMD extends Alfresco by providing the following: a) an integration with SRM as a back-end content storage mechanism for Alfresco, b) an integration with the VBE CA to provide a trust infrastructure and c) an extension of the role-based authorization mechanism that includes usage control policies and that uses the Enforcer component. In the upcoming sections, we present the architecture, implementation and evaluation of the DCMD. For simplicity, we shall refer to the CMVO as the "VO".

IV. Architecture and Prototype Description

In this section, we describe the architecture and prototype of our solution, which integrates the individual technologies presented in the previous section.

A. Architecture Description

The architecture of the DCMD is shown in Figure 2 both at the VO set-up and the VO usage phases. The architecture utilises a subsystem (termed the *CMVO Manager*), which coordinates access and usage of storage, computing and application services needed to implement an effective content management system to be used by several users in the VO. Users can be Content Providers, Consumers or both. They rely upon the VO Manager to store, transform, index, search and download documents and multimedia content.

Service providers can be providers of storage, or application services such as specialized indexing functionalities, geolocation functionalities etc. In the implementation of the DCMD, the focus has been on providers of Grid storage services. The VO Manager should be considered both an organizational entity and a software system - it takes care of user requests dispatching them to appropriate service providers, enforcing resource usage policies and collecting evidence of fair behavior (i.e. reputation) of VO users.

Before users can access and use the VO resources, however, a VO setup phase is required. In this phase, the VO Manager collects information about Grid service providers, including identity information, as well as a description of the service and security policies they provide. The VO Manager matches service provider profiles and policies with the VO policies to ensure they are compatible and extends invitations to service providers to join the VO. At some point, the service providers join the VO and are then eligible to advertise their services. Similarly, users must register with the VO manager providing their identity and other information about themselves before they can join a VO.

B. Prototype Description

In this section we provide a few highlights and technical details on the prototype design.

User Interface: The interface for the VO users provides access to search, upload, download as well as categorization and metadata-related functionality. Users are presented with a sort of "file explorer", allowing the organization of content according to the usual "file and folder" visual metaphor. This allows users to access all available operations on the content. The interface is implemented as a customization and extension of an Alfresco community component. On the other hand, the VO Manager has a few administrative tools available, which it can use to manage users, service providers and policies. For example, the interface shows the reputation of users and all other attributes relevant to usage control policies, which may determine the constraints users accesses to resources. Finally, The interface also shows the list of service providers that join a VO, the type of services they provide and the endpoint to access the service.

Extending Alfresco to Support Usage-Control Policies: The GridTrust project developed two mechanisms for usage control: the first being computation-level usage control (i.e. based on the Grid Resource Allocation Management (GRAM) [14] protocol), where usage control is performed at the individual Grid node executing the users requests. The second being service-level, where enforcement is done at the VO level and can be used to enforce a wider variety of user requests beyond computational tasks. In the DCMD, we focused on enforcing usage of resources at the VO level.

Policy enforcement and decision is implemented in an application component, which enforces policies related to usage of resources for the whole VO, and uses as a library the Enforcer's policy decision point. Technically, the Policy enforcement and decision points have been integrated as part of the Alfresco platform. The Enforcer knows about users policies, which are valid for the whole VO, and the service providers' policies, which are relevant and applied to each service. Any service providers that join the VO register first their own policies with the Enforcer. User policies are centrally managed and apply to all users of the VO.

We identified a set of *protected operations* within the set of Alfresco public services that need to be protected with respect to usage control. Specifically, we selected as protected operations content reading and writing. Turning an Alfresco public service into a protected operation is implemented using the Alfresco extensible security architecture, which allows to plug-in security interceptors to process each call to the service. Content reading and writing is then subject to policies defined as XACML in the Enforcer.

The following policies were identified and implemented.

- Consumable resource usage policies (for storage resources and subscription-type "pay-per-use" services)
- History-based usage control policies

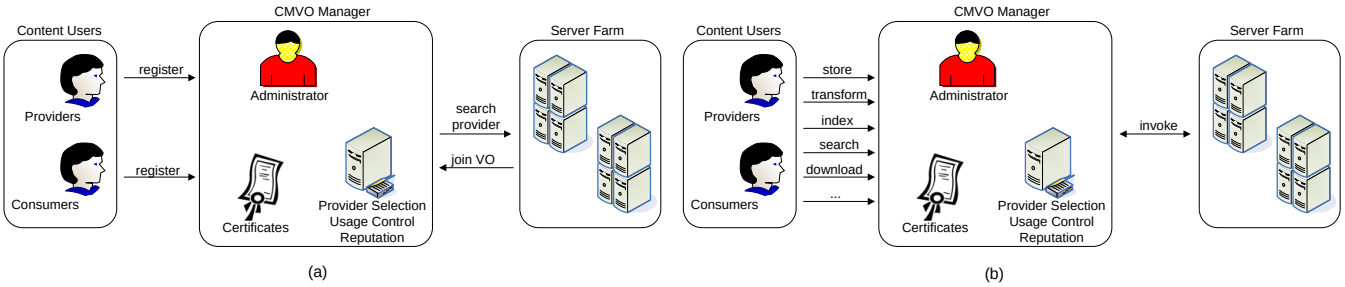

Figure 2. The High Level Architecture for (a) VO setup and (b) VO usage phases.

- Policies expressing obligations

Examples of the above policies include:

1. A content provider (e.g. touristic content, map data) obtains space/application services from the VO to create and manage content libraries. An example policy would be that the "content provider should not use more than its storage space as agreed with the VO (e.g. subscription to storage service)".
2. Content consumers (e.g. publishing companies) access/query/download content provided by the VO in libraries. An example policy would be "content consumers can use content only according to their rights (e.g. subscription to a library/content services)"
3. VO users can obtain credits to get more services from the VO (e.g. download of premium content) through their contributions to the VO.
4. Trial subscription should expire in 30 days, thus related accounts should be canceled after 30 days.

Extending Alfresco to Interoperate with SRM: The implementation maps Alfresco methods to corresponding SRM 2.2 methods invoked via a BeStMan-based web interface. The resulting implementation is pluggable to an Alfresco installation to achieve more resilient and scalable configurations. For example, one can use the local file system for primary content storage and use a secondary content storage, based on SRM, for replication and backup purposes. Alternatively, one can use Alfresco's clustering architecture, which shows how an Alfresco installation may use multiple SRMs to store digital content, therefore addressing scalability concerns. Each Alfresco instance stores content on a different SRM server and the Replicating Content Store (a component in Alfresco's clustering system) forwards requests to the other SRM servers when looking-up of content fails in the primary store. Finally, if a single-primary/multiple-SRM configuration is used, one would require implementing a scheduler/balancer at the low level, in order to assign load to the different SRMs.

V. Performance Results

We carried out a performance evaluation under heavy load of the main GridTrust service used, i.e. the Enforcer component. The performance overhead is shown in Figrue 3, which depicts the time to evaluate from 10 up to 100 parallel requests against a database of 10 XACML policies each composed of 5 rules. These numbers show expected delay for enforcing access and usage control policies.

VI. Conclusion and Future Work

We presented in this paper a prototype called the Distributed Content Management Demonstrator, which provides a user-friendly secure and scalable solution for the management of high-volume digital multimedia content based on Grid computing. Our solution enhances the Alfresco content management system with Grid capabilities by integrating it with BeStMan, an implementation of OGF's SRM standard for Grid-based storage management. The Demonstrator also provides better control over access and usage of resources since it utilises the Enforcer PDP to enforce usage control policies allowing it to express consumable resource usage, history-based usage and obligation policies.

The main advantage of this work is that it provides a solution for a real use case in which Grid computing is utilised commercially in the domain of distributed content management for the publishing industry. During the test phase of the Demonstrator, the cartography institute of De Agostini, the main users of the system, evaluated the use of usage control security policies, which have been so far very infrequently used inside the company. These policies have allowed De Agostini to open up new business possibilities. For example, De Agostini can now offer its customers contracts which allow them to read a fixed number of eBooks or images per month. The requirement is that customers content consumption must be limited to the amount specified in their contracts (enforced by the new policies). Further the end user can download premium content only if it uploads its own contents, for example personal reviews, to the books just downloaded. The Demonstrator has provided De Agostini with the possibility of re-evaluating the use of Grids for the management and sharing of digital content.

There very few solutions in literature that have been proposed for providing Grid-based scalable and secure back-up for content management technologies [15], [16]. In [15], the authors propose a distributed P2P-based Grid content management architecture, which combines Grid and Client-

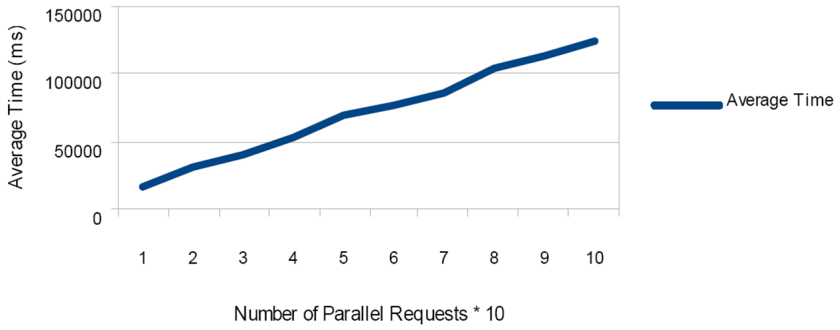

Figure 3. Average Enforcement Time for Parallel Requests (100 iteration for each request).

Server paradigms. Security is managed through classical access control and audit trail mechanisms, although these are left underspecified by the authors. In [16], the authors present the Scientific Annotations Middleware (SAM), which a set of services facilitating the management of metadata describing data objects and their semantic relationships. The middleware provides a content management front-end to clients backed-up by data Grid storage. In SAM, authentication and authorisation is applied using PKIs. Finally, the Alfresco Management and Security Toolkit[5] is a project that had the goal of produce a set of Web service workflows allowing the use of Alfresco by the educational community in repository projects and learning design.

Future work will be to exploit the proposed Grid-based solution in other data-intensive domains that require security and scalability. One such domain could be the health care domain, where data sharing bring in new security issues such as anonymity and integrity.

REFERENCES

[1] J. Newton, "Alfresco a fresh approach to content management," white paper, 2005.

[2] A. Sim, A. Shoshani, P. Badino, O. Barring, J.-P. Baud, E. Corso, S. D. Witt, F. Donno, J. Gu, M. Haddox-Schatz, B. Hess, J. Jensen, A. Kowalski, M. Litmaath, L. Magnoni, T. Perelmutov, D. Petravick, and C. Watson, "The Storage Resource Manager Interface Specification Version 2.2," OGF, Tech. Rep., 2008.

[3] L. M. Camarihna-Matos and H. Afsarmanesh, "Elements of a Base VE Infrastructure," *Journal of Computers in Industry*, vol. 51, no. 2, pp. 139–163, 2003.

[4] J. Park and R. Sandhu, "The UCON_{abc} Usage Control Model," *ACM Transactions on Information and System Security*, vol. 7, no. 1, pp. 128–174, February 2004.

[5] "Content Management Interoperability Services: Defining Web Services for Sharing Information among Disparate Repositories: Technology Concepts and Business Considerations," EMC[2] White Paper.

[5]http://amset.leeds.ac.uk:8080/amsetwiki/

[6] N. Mitra and Y. Lafon, "SOAP Version 1.2 Part 0: Primer (Second Edition)," W3C, Tech. Rep., 2007.

[7] R. T. Fielding and R. N. Taylor, "Principled Design of the Modern Web Architecture," *ACM Trans. Internet Technol.*, vol. 2, no. 2, pp. 115–150, 2002.

[8] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based Access Control Models," *Computer*, vol. 29, no. 2, pp. 38–47, Feb. 1996.

[9] A. Arkin and K. Visco, "CASTOR," in *O'Reilly Java Conference*, 2000.

[10] P. Massonet, A. Arenas, F. Martinelli, P. Mori, and B. Crispo, "GridTrust – A Usage Control Based Trust and Security Framework for Service-Based Grids," in *At your Service: Service Engineering in the Information Society Technologies Program*, E. di Nitto, A.-M. Sassen, P. Traverso, and A. Zwegers, Eds. MIT Press, 2008.

[11] C. Adams and S. Farrell, "Internet X.509 Public Key Infrastructure Certificate Management Protocols," RFC 2510 (Proposed Standard), Mar. 1999.

[12] OASIS, "Oasis extensible access control markup language (xacml) tc," http://www.oasis-open.org/committees/xacml, 2005.

[13] A. Arenas, B. Aziz, and G. C. Silaghi, "Reputation Management in Grid-Based Virtual Organisations," in *SECRYPT 2008, International Conference on Security and Cryptography*. INSTICC, 2008.

[14] M. F. I. Foster and S. Martin, "GT4 GRAM: A Functionality and Performance Study," in *Proceedings of the Teragrid 2007 Conference*, 2007.

[15] Q. Zhang, Y. Sun, Z. Liu, X. Zhang, and X. Wen, "Design of a distributed p2p-based grid content management architecture," in *CNSR '05: Proceedings of the 3rd Annual Communication Networks and Services Research Conference*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 339–344.

[16] J. D. Myers, A. Chappell, M. Elder, A. Geist, and J. Schwidder, "Re-integrating the research record," *Computing in Science and Engg.*, vol. 5, no. 3, pp. 44–50, 2003.