Enriched Namespace to Support Content-aware Authorization Policies

Antonia Ghiselli†, Luca Magnoni†, Federico Stagni‡, Zappi Riccardo†
†INFN-CNAF, Viale Carlo Berti Pichat, 6/2, I40127, Bologna, IT
‡INFN Ferrara, Via Saragat,1 I44100, Ferrara, IT
E-mail: {antonia.ghiselli, luca.magnoni, riccardo.zappi}@cnaf.infn.it
E-mail: {federico.stagni}@fe.infn.it

Abstract. In the near future, data on the order of hundreds of Petabytes will be spread in multiple storage systems worldwide dispersed in, potentially, billions of replicated data items. Users, typically, are agnostic about the location of their data and they want to get access by either specifying logical names or using some lookup mechanism.

A global namespace is a logical layer that allows the view of data resources independently from the physical location. Usually, the naming scheme is designed to be easily interpreted by humans and it is organized into a purely user-defined directory hierarchy. Within this model, a data resource is uniquely addressed by file name and path. Nevertheless, this hierarchical structures of logical namespace lacks adequate flexibility to manage sophisticated organization of data. In particular the implicit classification of the data item derived from the path is not enough meaningful to classify data objects when different orthogonal dimensions are considered.

In this paper we expose an enriched namespace able to support a new type of data access authorization policy based on tags. The tags are organized in well-defined hierarchies providing a simple representation of the domain ontology. Only authorized users can label data resources with different tags taken from the domain tag hierarchies. In this way an overlay of classical hierarchical structure of logical namespace with faceted hierarchical tags provides a semantics classification of data entities. Authorization policies defined in respect of tags are content-aware.

1. Introduction
In the near future, data on the order of hundreds of petabytes will be spread in multiple storage systems worldwide dispersed in, potentially, billions of replicated data items. Users are typically agnostic about the location of their data and they want to get access by either specifying logical names or using some lookup mechanism. The objective of a global namespace is to provide the users the illusion of a single centralized file system, allowing the creation and access to distribute and replicated data items as they are local. It enables the independence of position for users and applications as well as transparency of data location.

Current namespaces are nearly identical to the first file systems: hierarchical, named directory trees. The main task of a namespace is to provide only a mapping between a name and a piece of associated data (i.e., a file), leaving users unassisted regarding the data logical organization and retrieval. Researchers, scientists, and users have largely relied on some simple methods to organize and record information about data items, like using descriptive directory and file names. These exotic solutions have demonstrated all their limits. These methods do not scale to terabyte and petabyte data sets consisting of millions of data items.
The data organization in directories allows classifying files in a trivial and effective fashion. Subdirectories contain files or further subdirectories and so on, all the way down. Often researchers want to classify data items using more than a single directory, because monodimensional classification is only a partial expressivity. Considering data content structured and complex, the strict hierarchical filling schemes fail in many cases. The use of symbolic links, aliases and shortcuts cross-tree linking augment the capabilities of a strict hierarchy, nevertheless these links are difficult to maintain in the face of file updates and removals, making them an imperfect solution.

One potential solution to this problem is to enrich the namespace with metadata, information that describe the contents of data items. Application-dependent information like the status, the meaning, and the quality of data items allow users to classify each file with multiple attributes, and to query for data items identified by descriptive attributes. Metadata are crucial for researchers in order to identify and locate those subsets of data objects that are of interest to a data analysis activity. Accurate identification of desired data items is essential for correct analysis of experimental and simulation results.

Another problem lies in the authorization architecture: in the scenario of a global namespace with billions of entries, the definition and the enforcement of the conventional access control model to this huge, dynamic, and dispersed set of data items, represents an issue in terms of management, scalability, and consistency. In current hierarchical file systems, access control is made specifying the authorizations on everyone of billions of files.

The traditional access control models, such as Identity Based Access Control (IBAC) [1] normally implemented using of Access Control List (ACL) [2], or the more flexible Role Based Access Control (RBAC) [3], have deficiencies in large open systems, since they do not solve scalability and governability problems. When an authorization policy changes for a specific user or role, the authorization manager must implement the adjustment in every entry involved, potentially all. Moreover, frequent authorization mutations and a big number of users or roles make worse the possibility of being managed in an effective way.

In this paper, we envision an enriched namespace able to support a new type of data access authorization policy based on tags. Tags are dedicated attribute added to metadata schema of data items to provide an expressive and schematic classification scheme. The tag values can be organized in well-defined hierarchies providing a simple representation of the domain ontology. Only authorized users should be able to label data resources with different tags taken from the domain tag hierarchies. In this way an overlay of classical hierarchical structure of logical namespace with faceted hierarchical tags provide semantics classification of data entities.

An enriched namespace allows the specification of a new variety of authorization policies based on defined tag values, and then it refers to the content of files. In order to make the access control mechanism more flexible, the attribute based access control (ABAC) [4][5][6] has been introduced. Moreover, to increase the flexibility and limit the complexity of specification of policies and the representation of attributes we suggest the use of tag values only within ABAC policies.

The rest of the paper is organized as follows: to set the background, Section 2 provides a brief overview of global namespace and metadata. Section 3 introduces the concepts of authorization and describes the attribute based access control model (ABAC). Section 4 explains our idea to implement ABAC applied to data management in grid using already in use technologies. Section 5 concludes the paper.

2. Global Namespace and Attributes
In a Grid environment, the applications and the users should be able to access data without knowing their location. The naming capability proposed in the current grid allows users to refer to specific data resources in a physical storage system using a high level logical identifier.
To implement location independence in the grid, data resources has to be recognized by name without any location information. Many efforts are made in the Global Grid Forum (GGF) and OASIS to document best practices, implementation guidelines, and standards for Grid technologies. In particular, the Open Grid Services Architecture (OGSA) work on data architecture [7] identifies a scheme with three level of naming. The data resources are identified through a hierarchically arranged scheme: human-oriented names, globally unique abstract names and low level addresses:

- **Human-oriented name**: based on a naming scheme that is designed to be easily interpreted by humans (e.g. human-readable and human-parsable).
- **Abstract name**: a persistent name suitable for machine processing that does not necessarily contain location information.
- **Address**: specifies the location of an data resource.

There are several terms in the context of the current grid to identify data, depending on the different resources involved. Typically, the users identify their files by **Logical File Name (LFN)**. The LFN represents the key by which the users locate the actual locations of their files. The LFNs are user friendly high level identifiers, which are human-oriented names. The users can organize LFN with a directory structure to simulate a global namespace. A same data resource can be addressed by various LFNs by different users, similar to the concept of alias.

As abstract name, the Global Unique Identifier (GUID) is given to each file when it is created on the Grid. In the Grid, we can also ponder replica of files, so for a GUID we can find various addresses corresponding to the different replicas.

Replicas are identified by the logical identifier named **Site URLs (SURLs)**. A SURL provides an abstraction of the data namespace living into a storage resource to allow different data access paths. Each replica has its own SURL and it specifies implicitly which Storage Resource needs to be contacted to extract the data. Usually, users are not directly exposed to SURLs, but only to the logical namespace defined by LFNs. To provide the user the illusion of a single file system, the Grid data management middleware has to keep track of LFN to GUID and SURL mappings in a scalable manner. Finally, the address is identified by the **Transfer URL (TURL)**. The TURL represents the physical identifier provided by the storage resource management.
Figure 2. An example of taxonomy

(SRM) [8] services for data access. A same SURL can hold different TURLs, one for each access protocol supported by the storage resource. The TURL used by the user to address the data resource results from the negotiation process between SRM service and user request; the outcome depends on the access protocol specified by the user to access the data resource and to the protocol supported by the storage resource. Figure 1 describes the relationship on terms.

2.1. Metadata as Semantic Layer
Metadata is information that describes the contents of data items. Binding metadata to data resources allow scientists to record information about the creation, transformation, meaning and quality of data items, and to query for data items based on these descriptive attributes, and to classify each file with multiple attributes. Accurate identification of desired data items is essential for correct analysis of experimental and simulation results. In the past, scientists have largely relied on ad-hoc methods (descriptive file and directory names, lab notebooks, etc.) to record information about data items. However, these methods do not scale to terabyte and petabyte data sets consisting of millions of data items.

Metadata can encompass a variety of information. We can distinguish two types of attributes: one is system-dependent metadata and application independent, which consists of file attributes such as creation time, author, size on disk, physical locations and file checksum, etc. The second type of metadata are application-dependent and user-defined attributes which consist of properties that depend on the contents of data resources and, in the case of data related to High-Energy Physics experiment the metadata could describe attributes such as experiment date, mode of production (simulation or experimental) and event type. In some case, scientists could tag data with free tag to express particular data flavor. System metadata are automatically generated by middleware, while user-defined attribute are decided by the scientists.

Metadata adds value to scientific data. Without metadata, the researcher is unable to evaluate the quality of the data. For example, it is impossible to conduct a correct analysis of a data set without knowing how the data was cleaned, calibrated, what parameters were used in the process, etc.

Normally, the generating and the binding process of accurate attribute values to categorize files content is a hard work for users. Moreover, users could tag the same file with different synonyms despite the fact that user is also the creator of the content. Metadata is not only generated by experts but also by consumers of the content. Lately, collaborative tagging (also knows as ”folksonomy”) is commonly used in contrast to traditional subject indexing.
2.2. Folksonomy, Taxonomy e Thesaurus
An overlay of classical hierarchical structure of logical namespace with faceted tags provides semantics classification of data entities. Resources annotated with machine-readable metadata may hold the key to developing the semantic grid, where resources can be discovered and joined up automatically by middleware. Collaborative tagging, also known as folksonomy, is an effective approach to enrich the metadata. In folksonomy many users can collaboratively create and manage tags to annotate and categorize data content.

In contrast to traditional subject indexing, metadata is not only generated by experts but also by creators and consumers of the content. Usually, these category names and tags are freely chosen by the user without any a-priori dictionary, taxonomy, or ontology to conform to. Without a shared taxonomy or ontology, social annotations suffer the usual problem of ambiguity of semantics. The same annotation may mean different things for different people and two seemingly different annotations may bear the same meaning. Without a clear semantics, these social annotations won’t be of much use for scientists and applications on data grids.

In HEP, as well as in every scientific domain, we can use a well specified thesaurus, classification scheme and taxonomy. Hopefully, the same file tagged by different users should hold the same attribute values, so more than one interpretation for the same content should not occur. Data creators and consumers can annotate data resources easily with tag taken from a controlled vocabulary and taxonomy with even knowing semantic. An example of taxonomy is shown in figure 2.

3. Attribute-Based Authorization
An authorization service deals with at least three kind of actors: consumers, objects, and rights [9]. We may say that authorization is concerned with the assignment of rights, permissions or privileges to determinate consumers, while access control, a closely related concept, evaluate access requests to determine whether they must be granted or not. Access control is applied to each resource in order to limit the activity of authenticated users. Resources include individual files or data items, programs, devices and applications. Examples of consumers are users, groups, or any kind of identified set of identifiable entity. Figure 3 shows the relations between the authorization concepts.

In Grid, authorization is usually defined in the limits of a Virtual Organization (VO) [10]. Since VOs are increasingly turning into distributed multi-institutional collaborations, authorization becomes a growing challenge. VO membership may change dynamically, rights may be granted to entities on a periodic basis, or a user’s role in an organization might dynamically evolve. Such factors make it more practical to express users’ rights based on their attributes, such as institutional affiliation or role in a collaboration, rather than identity alone.

3.1. From IBAC to RBAC
The oldest and simplest access control model is the IBAC model, which has the distinctive feature of being based on the subject entity, like an uid or pid in its UNIX implementation. This per person approach of establishing resource access becomes unmanageable as the number
of users requiring resources access grows. One of the most notable solutions to the problem is the application of Role Based Access Control model (RBAC) [11].

RBAC is inspired to the business world: its development coincides with the advent of corporate intranets. Corporations are usually hierarchically structured and access permissions depend on the position of the user in the hierarchy, i.e. the role played by the user. RBAC policies control access depending on the roles that users play within the system and on rules stating what accesses are allowed to users in given roles. Fig. 4 presents a simple RBAC example (on the left), and the core RBAC relations between the five sets of basic elements, namely users, roles, objects, actions (operations) and permissions.

RBAC is commonly considered a mature and flexible technology, it has been standardized, implemented, extended in many ways, and has been accepted for many years as the most appropriate paradigm for the implementation of access control in complex scenarios. However, very dynamic systems with high volume of heterogeneous data, like semi-structured data systems, require more flexible constructions for the expression of access control policies. RBAC provides a quite scalable solution classifying users with roles, but does not consider an object classification. Moreover, in an environment where not all the users that may need access to a document are known in advance, we need to have authorization models that can consider user attributes and credentials to determine access rights. Such models are normally identified as attribute based (ABAC) models, since access depends not only on the subject identity but also on the satisfaction of assertions including those attribute values describing subjects (consumers and providers) and objects.

3.2. Attribute Based Access Control - ABAC

The basic idea of ABAC is not to define rights directly between consumers and objects, but instead to use their attributes as the basis for authorizations. This way, subjects (the consumers) and objects attributes are security-relevant characteristics. Attributes of a consumer may include its identifier, name, organization, position or role in a company, as well as dynamic attributes like age, current location or an acquired subscription for a digital library. Objects can be characterized not only by their usual application-independent attributes, like file size or the date of the last access, but by their application-dependent peculiarities, like its physical scope.

The ABAC concept is illustrated in fig. 5: subjects and objects are both represented by a set of attributes and related attribute values, put together in a subject and an object descriptor. Permissions (rights) consist of the combination of a so-called object descriptor, which consists of a set of attributes and conditions, and an operation that is to be executed on the objects denoted by the descriptor. Authorizations are defined between a subject descriptor and permission. Using descriptors it is possible to assign dynamically permissions to subjects and objects, thus making

![Figure 4. A simple RBAC example, and the core RBAC relations](image-url)
We consider role and identity of a subject as attributes of a principal, so that ABAC fully encompass the functionalities of both IBAC and RBAC approaches. ABAC models are the natural convergence of existing access control models, surpassing their functionalities. With ABAC we can have semantic rich policies, with the possibility for an administrator to define the access control granularity that better suit her/his needs.

4. Current works and Proposed Architecture
We want to address the need to efficiently define the authorization access policy in a scenario with a huge data volume and with dynamic and vague set of data consumers.

The authorization policies are defined using ABAC model, where object properties used to define ABAC policies include the tags defined previously and other metadata, while subject attributes include not only on the subject identity but also those attribute values describing subjects, such as role within the Virtual Organization (VO). By using subject attributes and object properties for the definition of authorizations, administration is simplified and flexibility is improved. When changing attribute values, affected permissions will be automatically updated without the need to change explicitly role definitions.

Data creators and experts in related applications are able to annotate the data resources using tag taken from a defined application domain thesaurus and defined taxonomy.

The authorization policies can be defined only by authorized administrators, such as application managers at virtual organization level. In this way, data creator will take upon oneself to annotate the data with the more appropriate tags only, and they are released to determinate the proper access authorization.

The technologies currently available allow the definition and the planning of a prototype of a framework able to support content-aware authorization policies. In the next sections we propose a panoramic of the adoptable technologies and a possible arrangement.

Figure 6 shows a plausible integration of technologies to build a prototype of content-aware authorization policy framework.

4.1. Subject attributes
Examples of subject attributes can be the subjects identity, role, age, zip code, IP address, memberships, citizenships, etc. Currently, the Virtual Organization Membership Service
(VOMS) [12] is the most widely-used attribute authority service, and is a de-facto standard for AA in production Grids. VOMS is an Attribute Authority that exposes attributes and encodes the position of the holder inside the virtual organization. A Holder may be a member of several groups, and may hold a special role inside some of his groups. Groups are organized in a tree structure that comprises groups and subgroups, while roles are not hierarchical and are associated to the membership in a group.

4.2. Object attributes
Example of object attributes can be the resource identity, location, size, value, or may be environment attributes, such as time of day, date, system state, etc. Generally, the object attributes are stored in metadata catalog service. Currently there are several implementation available, for example MCAT [13] and AMGA Metadata Catalog [14].

MCAT is a meta information catalog system implemented at San Diego Supercomputer Center (SDSC) as part of the Data Intensive Computing Environment (DICE) with requirements mainly based on the Storage Resource Broker system (SRB) [15]. MCAT catalog is designed to serve both a core-level of meta information and domain-dependent meta information.

The ARDA Metadata Grid Application (AMGA) Metadata Catalog is part of the gLite [16] software stack of the EGEE [17] project. The AMGA implementation uses a file-system model for structuring metadata, supporting flexible schemas and a hierarchical organization.

Domain experts and scientist who are planning to define metadata schemas specific to their domains of interest (eg. High Energy Physics group) can use various facilities and languages. The representation semantics can be described by means of relevant domain terms based on ontologies.

The Resource Description Framework (RDF) [18][19] is a web standard for representing resources on the web. It restricts the description of resources to statements composed of subject, predicate and object triples. It uses XML ((extensible markup language) as the interchange syntax. An RDF Schema (RDFS) [20] defines the terms that will be used in the RDF statements and gives specific meanings to them. The Web Ontology Language (OWL) [21] may also be used to define those terms using representations that are more expressive. OWL builds on top of RDF. The use of RDF allows data to be linked to and/or merged with other RDF data by semantic metadata catalogues. In practice, this means that data sources can be distributed across the grid in a decentralized way enabling a distributed semantic metadata service.

A very interesting technology is the Simple Knowledge Organisation Systems (SKOS) [22]. SKOS is designed for representation of thesauri, classification schemes, taxonomies, subject-heading systems, or any other type of structured controlled vocabulary, so it is especially suitable for our use case. SKOS is built upon RDF and RDFS and it is developed, published, and maintained by the W3C Semantic Web Best Practices and Deployment Working Group [23]. It consists of a set of RDF properties and RDFS classes that can be used to express the content and structure of a concept scheme as an RDF graph.

4.3. ABAC Policy definition, distribution, and enforcement with XACML and G-PBox
A standard to define and evaluate attribute policies in distributed system such as grid computing system, is the Extensible Access Control Markup Language (XACML) standard [24]. The XML application XACML is an OASIS standardized markup language to describe access control policies as well as access control decision requests and responses. XACML is highly flexible and extensible; however, the flexibility and expressiveness of XACML results in verbosity and complexity. In this case, the Grid Policy Box (G-PBox) [25] helps to manage and distribute XACML policies. The G-PBox framework is an approach for the management of policy repositories hierarchically distributed to independent, administrative-based layers, where each layer contains only policies regarding itself. The G-PBox GUI offers facilities for policy and
distribution management, which means that it allows to create/remove/modify/move arbitrary XACML policies and policy sets and to send/receive policies to/from other PBoxes.

5. Conclusion
The focus of this paper was to investigate a possible content-aware authorization policies solution based on already in use technologies. We have showed the strong motivation. The lacuna on scalability and adequate flexibility to manage sophisticated organization of data. These limits of simple hierarchical namespace underline the importance of metadata. We have proposed to enrich the namespace with special metadata tags taken from a thesaurus and from domain taxonomy. The tags can be organized in well-defined hierarchies providing a simple representation of the domain ontology. Only authorized users can label data resources with different tags taken from the domain tag hierarchies. By enriched namespace, it is possible to support a new type of data access authorization policy based on tags. The authorization model is based on attribute based access control (ABAC) model. Finally, various technologies were selected to propose a prototype.

Future work will be targeted at the development and testing of the extension presented in this paper.

References
[1] Surridge M 2002 A rough guide to grid security Tech. rep.
[2] Grunbacher A Submitted for publication at the USENIX ATC, San Antonio, Texas, June 2003
[3] D Ferraio I R Sandhu S G D K R C 2001 ACM Transactions on Information and System Security (TISSEC) 224–274
[4] Yuan E and Tong J 2005 ICWS ’05: Proceedings of the IEEE International Conference on Web Services (ICWS’05) (Washington, DC, USA: IEEE Computer Society) pp 561–569 ISBN 0-7695-2409-5
[5] bo Shen H and Hong F 2006 PDCAT ’06: Proceedings of the Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies (Washington, DC, USA: IEEE Computer Society) pp 74–79 ISBN 0-7695-2736-1
[6] Swamynathan G, Close T, Banerjee S and McGeer R 2007 C5 ’07: Proceedings of the Fifth International Conference on Creating, Connecting and Collaborating through Computing (Washington, DC, USA: IEEE Computer Society) pp 93–98 ISBN 0-7695-2806-6
[7] Antonioletti M, Berry D, Chervenak A, Kunszt P, Lumiewski A, Laws S and Morgan M 2006 Ogsa data architecture v0.6.2 Tech. rep. Open Grid Forum URL https://forge.gridforum.org/sf/go/doc13635?nav=1
[8] Shoshani A, Sim A and Gu J 2004 Grid Resource Management (Norwell, MA, USA: Kluwer Academic Publishers) chap 20, pp 321–340
[9] Lampson B 1969 Dynamic protection structures
[10] Foster I, Kesselman C and Tuecke S 2001 Lecture Notes in Computer Science \textbf{2150}
[11] 2004 ANSI incits 359-2004 role based access control
[12] Alfieri R, Cecchini R, Ciaschini V, dell’Agnello L, Frohner Á, Lörentey K and Spataro F 2005 \textit{Future Generation Comp. Syst.} \textbf{21} 549–558
[13] 2007 MCAT (Version 1.1) \url{http://www.npaci.edu/DICE/SRB/mcat.html}
[14] Santos N and Koblitz B 2006 Distributed Metadata with the AMGA Metadata Catalog
[15] Baru C, Moore R, Rajasekar A and Wan M 1998 The sdsc storage resource broker URL \url{citeseer.ist.psu.edu/baru98sdsc.html}
[16] 2007 gLite - Lightweight Middleware for Grid Computing \url{http://glite.web.cern.ch/}
[17] 2007 EGEE - Enabling Grids for E-sciencE \url{http://public.eu-egee.org/}
[18] Resource Description Framework (RDF) Model and Syntax Specification, \url{http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/}
[19] Manola F and Miller E 2004 RDF primer
[20] RDF Vocabulary Description Language 1.0: RDF Schema. \url{http://www.w3.org/TR/rdf-schema/}
[21] McGuinness D L and van Harmelen F 2004 OWL Web ontology language overview
[22] SKOS Core Vocabulary Specification, W3C Working Draft 2 November 2005 \url{http://www.w3.org/TR/2005/WD-swbp-skos-core-spec-20051102/}
[23] Semantic Web Best Practices and Deployment Working Group. \url{http://www.w3.org/2001/sw/BestPractices/}
[24] OASIS 2005 Oasis extensible access control markup language (xacml) tc \url{http://www.oasis-open.org/committees/xacml}
[25] Caltroni A, Ciaschini V, Ferraro A, Ghiselli A, Rubini G and Zappi R 2004 G-PBox: A Policy Framework for Grid Environments