health care through information. Springer: 2004.

uate consent for the use of elec-
out strategy to reconcile individ-
it. J Royal Inst Pub Health 2005;

Z, Smith JP. Disease and Disad-
s and in England. JAMA 2006;

general practice. Oxford: Oxford

m CR, Carlson JA, Helgesson G.
o give broad consent to future
ol 2006;7(3):266–9.

ent approaches for biobank re-
gy; 2006. Letter responding to

Model Formulation ■

Sharing Data and Analytical Resources Securely in a Biomedical Research Grid Environment

Stephen Langella, Shannon Hastings, Scott Oster, Tony Pan, Ashish Sharma, Justin Permar, David Ervin, B. Barla Cambazoglu, Tahsin Kurc, Joel Saltz

Abstract **Objectives:** To develop a security infrastructure to support controlled and secure access to data and analytical resources in a biomedical research Grid environment, while facilitating resource sharing among collaborators.

Design: A Grid security infrastructure, called Grid Authentication and Authorization with Reliably Distributed Services (GAARDS), is developed as a key architecture component of the NCI-funded cancer Biomedical Informatics Grid (caBIG™). The GAARDS is designed to support in a distributed environment 1) efficient provisioning and federation of user identities and credentials; 2) group-based access control support with which resource providers can enforce policies based on community accepted groups and local groups; and 3) management of a trust fabric so that policies can be enforced based on required levels of assurance.

Measurements: GAARDS is implemented as a suite of Grid services and administrative tools. It provides three core services: Dorian for management and federation of user identities, Grid Trust Service for maintaining and provisioning a federated trust fabric within the Grid environment, and Grid Grouper for enforcing authorization policies based on both local and Grid-level groups.

Results: The GAARDS infrastructure is available as a stand-alone system and as a component of the caGrid infrastructure. More information about GAARDS can be accessed at http://www.cagrid.org.

Conclusions: GAARDS provides a comprehensive system to address the security challenges associated with environments in which resources may be located at different sites, requests to access the resources may cross institutional boundaries, and user credentials are created, managed, revoked dynamically in a de-centralized manner.

■ **J Am Med Inform Assoc.** 2008;15:363–373. DOI 10.1197/jamia.M2662.

Introduction

The informatics requirements of multi-institutional translational research projects are characterized by the need to securely share and access data and analytical resources hosted at different sites. In a multi-institutional project, sites participating in the collaborative effort can be viewed as being part of a *virtual organization*. One of the major obstacles to forming virtual organizations in biomedical research has been the lack of interoperability among disparate data and analytical resources. Another major problem has been the limited availability of infrastructure to provide secure and efficient access to these resources. Without mechanisms that can enable service providers to enforce access control policies to protect sensitive and proprietary information, data and analytical resources cannot be shared effectively. Traditionally, collaborative projects have created virtual organizations by employing a centralized system to host the databases and analysis tools at one of the institutions participating in the project. This approach, while alleviating some of the security and interoperability issues, is not scalable when the number of collaborating sites is large. It also is not efficient when it is desirable to rapidly and dynamically create, manage, and change virtual organizations.

A relatively recent effort, the cancer Biomedical Informatics Grid (caBIG™) program[a] of the National Cancer Institute (NCI), is targeting the informatics issues that arise in multi-institutional studies in biomedical research. This effort is developing informatics standards, a suite of common tools and applications, common data and analytical resources, and a Grid infrastructure, called caGrid,[1] to dynamically link applications, clients, and community provided resources. Security is of paramount importance in the caBIG™ program to ensure that any sensitive information such as patient demographics as well as the intellectual properties of

Affiliation of the authors: Department of Biomedical Informatics, The Ohio State University, Columbus, OH.

This work was supported in part by the NCI caGrid Developer grant 79077CBS10, the State of Ohio Board of Regents BRTT Program grants ODOD AGMT TECH 04-049 and BRTT02-0003, the NHLBI R24 HL085343 grant, the NIH U54 CA113001 and R01 LM009239 grants, and funds from the Ohio State University Comprehensive Cancer Center-Arthur G. James Cancer Hospital and Richard J. Solove Research Institute.

Correspondence: Tahsin Kurc, Biomedical Informatics Department, Ohio State University, 3184 Graves Hall, 333 West 10th Ave., Columbus, OH, 43210; e-mail: <kurc@bmi.osu.edu>.

Received for review: 11/15/07; accepted for publication: 02/13/08.

[a] https://cabig.nci.nih.gov/

researchers can be protected while promoting and facilitating collaborative projects.

Supporting authentication (i.e., determining whether or not a given user is who she/he claims to be) and authorization (i.e., controlling access to the functionality of a resource, once the user has been authenticated successfully) in the caBIG™ environment is difficult. User identities and credentials should be managed in a decentralized manner for scalability and manageability reasons, while allowing institutions to set up and enforce their access control policies locally. If there are many participants from different organizations, credentials should be managed in a federated environment. Tools are needed for system administrators to efficiently provision the credentials of users in their institutions in this federated environment. Another issue that becomes critically important in a dynamic and large-scale federated environment such as caBIG™ is the management of a *trust fabric*. Because institutions will have autonomous control over policies for granting, managing, changing, and revoking user credentials for their users, it can be expected that an institution will have different levels of trust for clients from different institutions when they want to access its resources. Moreover, there is a need to be able to efficiently propagate dynamic changes in policies and trust relationships and any security breaches (e.g., a user's credentials are revoked, because they have been compromised) to other entities in the environment. Tools are needed to create trust fabrics in the environment and manage them.

We have developed a software suite, called the Grid Authentication and Authorization with Reliably Distributed Services (GAARDS) infrastructure, to address these and other security requirements of caBIG™. In this paper, we report on the architecture of GAARDS and its main components. This paper significantly extends an earlier report on this project, which appeared in the proceedings of the AMIA 2007 Annual Symposium.[2] The current report presents a discussion on the requirements and challenges of supporting security in a large scale Grid environment and a more detailed description of the architecture of GAARDS and its components. It also illustrates the use of GAARDS in an application scenario involving review of images in a multi-institutional environment.

The salient features of the GAARDS infrastructure can be summarized as follows: 1) It provides services to support: a) integration of institutional identity provider and authentication systems with the Grid environment, b) efficient management and federation of user credentials, and c) easy deployment of a Grid-enabled identity provider system; 2) It implements support for group (role) based access control such that a service provider can use both community accepted roles and local roles to implement and enforce access control policies; and 3) It provides a service infrastructure for management of a trust fabric in the Grid environment, where institutions use different policies for provisioning of credentials for their local researchers and where credentials can be created, revoked, and reinstated dynamically. While the requirements for GAARDS have been motivated mainly by use cases from the caBIG™ program, the design and implementation of the infrastructure is generic and can be applied in other domains. The GAARDS infrastructure is available as both a stand-alone system and a component of the caGrid infrastructure,[1, 3] which is the Grid architecture of caBIG™. More information about GAARDS can be accessed at http://www.cagrid.org.

Security Challenges in a Large Biomedical Research Grid

The GAARDS infrastructure is designed to support three main components of security in a federated environment: authentication, authorization, and trust fabric. This section presents the issues that have motivated the design and implementation of support for these components in GAARDS. We describe the issues in the context of the caBIG™ environment, which is envisioned to span hundreds of institutions and thousands of researchers.

The objective of the caBIG™ program is to help accelerate research towards curing cancer by implementing the enabling informatics technologies for researchers to more efficiently find, share, retrieve, integrate, and process clinical and research data from disparate sources. The caBIG community consists of participants from cancer centers, research institutions, government organizations, and the informatics industry. Efforts underway in the caBIG™ program include the development and deployment of 1) informatics standards, 2) guidelines and tools to improve semantic and syntactic interoperability among data and analytical resources, 3) open-source, common applications for data management and analysis, 4) guidelines and processes for data and tool sharing, and 5) an open-source, standards based Grid infrastructure that is designed to federate distributed resources. While the spirit of caBIG™ is to promote and facilitate sharing of information and applications, not all information and tools can be made publicly available to everyone in the caBIG environment. Clinical information and the intellectual properties of researchers must be protected, and such information should be accessible only by those with appropriate privileges.

The caBIG™ project is implemented as a federated environment where individuals, groups, and institutions maintain their resources locally. Resources are exposed to the environment and shared among institutions and researchers using the caGrid infrastructure.[1, 3] Briefly, caGrid is the Grid architecture of caBIG™. It provides a core suite of tools and services and a runtime environment to enable secure federation of resources in the caBIG environment. Each data and analytical resource in caGrid is implemented as a Grid Service conforming to the Web Services Resource Framework standards.[4, 5] Interactions between caGrid services and clients are carried out using standard Grid Service protocols. The GAARDS infrastructure is designed to support security requirements in a caGrid like service oriented environment. We now discuss these requirements.

Users wishing to access caBIG resources are required to authenticate with caGrid services that expose those resources. To authenticate with a service, users must prove their identity to the service. For this purpose, users are issued *grid credentials*. In order for individuals to authenticate to services across organizational boundaries, it is necessary that a common type of credentials be adopted and those credentials be issued by a trusted set of credential issuers (also known as Certificate Authorities). The GAARDS infrastructure uses X509 Identity Certificates for researchers can be protected while promoting and facilitating collaborative projects.

Supporting authentication (i.e., determining whether or not a given user is who she/he claims to be) and authorization (i.e., controlling access to the functionality of a resource, once the user has been authenticated successfully) in the caBIG™ environment is difficult. User identities and credentials should be managed in a decentralized manner for scalability and manageability reasons, while allowing institutions to set up and enforce their access control policies locally. If there are many participants from different organizations, credentials should be managed in a federated environment. Tools are needed for system administrators to efficiently provision the credentials of users in their institutions in this federated environment. Another issue that becomes critically important in a dynamic and large-scale federated environment such as caBIG™ is the management of a *trust fabric*. Because institutions will have autonomous control over policies for granting, managing, changing, and revoking user credentials for their users, it can be expected that an institution will have different levels of trust for clients from different institutions when they want to access its resources. Moreover, there is a need to be able to efficiently propagate dynamic changes in policies and trust relationships and any security breaches (e.g., a user's credentials are revoked, because they have been compromised) to other entities in the environment. Tools are needed to create trust fabrics in the environment and manage them.

We have developed a software suite, called the Grid Authentication and Authorization with Reliably Distributed Services (GAARDS) infrastructure, to address these and other security requirements of caBIG™. In this paper, we report on the architecture of GAARDS and its main components. This paper significantly extends an earlier report on this project, which appeared in the proceedings of the AMIA 2007 Annual Symposium.[2] The current report presents a discussion on the requirements and challenges of supporting security in a large scale Grid environment and a more detailed description of the architecture of GAARDS and its components. It also illustrates the use of GAARDS in an application scenario involving review of images in a multi-institutional environment.

The salient features of the GAARDS infrastructure can be summarized as follows: 1) It provides services to support: a) integration of institutional identity provider and authentication systems with the Grid environment, b) efficient management and federation of user credentials, and c) easy deployment of a Grid-enabled identity provider system; 2) It implements support for group (role) based access control such that a service provider can use both community accepted roles and local roles to implement and enforce access control policies; and 3) It provides a service infrastructure for management of a trust fabric in the Grid environment, where institutions use different policies for provisioning of credentials for their local researchers and where credentials can be created, revoked, and reinstated dynamically. While the requirements for GAARDS have been motivated mainly by use cases from the caBIG™ program, the design and implementation of the infrastructure is generic and can be applied in other domains. The GAARDS infrastructure is available as both a stand-alone system and a component of the caGrid infrastructure,[1, 3] which is the Grid architecture of caBIG™. More information about GAARDS can be accessed at http://www.cagrid.org.

Security Challenges in a Large Biomedical Research Grid

The GAARDS infrastructure is designed to support three main components of security in a federated environment: authentication, authorization, and trust fabric. This section presents the issues that have motivated the design and implementation of support for these components in GAARDS. We describe the issues in the context of the caBIG™ environment, which is envisioned to span hundreds of institutions and thousands of researchers.

The objective of the caBIG™ program is to help accelerate research towards curing cancer by implementing the enabling informatics technologies for researchers to more efficiently find, share, retrieve, integrate, and process clinical and research data from disparate sources. The caBIG community consists of participants from cancer centers, research institutions, government organizations, and the informatics industry. Efforts underway in the caBIG™ program include the development and deployment of 1) informatics standards, 2) guidelines and tools to improve semantic and syntactic interoperability among data and analytical resources, 3) open-source, common applications for data management and analysis, 4) guidelines and processes for data and tool sharing, and 5) an open-source, standards based Grid infrastructure that is designed to federate distributed resources. While the spirit of caBIG™ is to promote and facilitate sharing of information and applications, not all information and tools can be made publicly available to everyone in the caBIG environment. Clinical information and the intellectual properties of researchers must be protected, and such information should be accessible only by those with appropriate privileges.

The caBIG™ project is implemented as a federated environment where individuals, groups, and institutions maintain their resources locally. Resources are exposed to the environment and shared among institutions and researchers using the caGrid infrastructure.[1, 3] Briefly, caGrid is the Grid architecture of caBIG™. It provides a core suite of tools and services and a runtime environment to enable secure federation of resources in the caBIG environment. Each data and analytical resource in caGrid is implemented as a Grid Service conforming to the Web Services Resource Framework standards.[4, 5] Interactions between caGrid services and clients are carried out using standard Grid Service protocols. The GAARDS infrastructure is designed to support security requirements in a caGrid like service oriented environment. We now discuss these requirements.

Users wishing to access caBIG resources are required to authenticate with caGrid services that expose those resources. To authenticate with a service, users must prove their identity to the service. For this purpose, users are issued *grid credentials*. In order for individuals to authenticate to services across organizational boundaries, it is necessary that a common type of credentials be adopted and those credentials be issued by a trusted set of credential issuers (also known as Certificate Authorities). The GAARDS infrastructure uses X509 Identity Certificates for identifying a user. An X509 Certificate with its corresponding private key forms a unique credential, or the so-called *grid credential* within the Grid. Although this approach is very effective and secure, it is difficult to manage in a multi-institutional environment. Using existing tools, the provisioning of grid credentials is a manual process, which is error-prone and very complex for most users and system administrators. The overall process is further complicated if a user wishes to authenticate from multiple locations, because a copy of his/her private key and certificate has to be present at every location. Not only is this process complex, securely distributing private keys is error prone and poses a security risk. Additionally, there are scalability and efficiency problems with vetting user identities. Organizations invest a significant amount of resources into their existing identity management systems and already have processes in place for issuing and managing user identities. In such settings, it would be more efficient to leverage existing identity management systems to provision Grid user accounts. Users would be able to use their existing credentials to "log on" to obtain Grid credentials and access Grid services. This scenario requires a mechanism to allow users to obtain Grid credentials using their existing organization-provided credentials. The mechanism should also remove the complications of using and managing Grid credentials. A discussion of how this issue is addressed in GAARDS follows.

Authorization is a challenging issue as well. It is desirable that access control policy be maintained and enforced locally, giving data providers the ability to determine who has access to their data. At the same time, it is important for scalability that access control policies be based on Grid-level information. Since most systems base their access control policies on membership to groups, a mechanism for organizing and managing groups spanning organizational boundaries is needed. The GAARDS approach to meeting these requirements is presented below.

Institutions participating in caBIG™ can have their own certificate authorities to issue credentials to their researchers. In such a setting, it is important to be able to verify and validate identities and privileges with a level of confidence. Because institutions will have different policies as to how they issue, control, audit, and revoke these credentials, it can be expected that a service provider will not have the same level of trust (in terms of authentication and authorization) for all users wishing to access the service. In addition, while institutions will want to collaborate, they will have services with different levels of security policy enforcement requirements. Services need to maintain a list of certificate authorities they trust. The main challenge is that there may be hundreds of certificate authorities, each issuing certificates for thousands of users. This problem is compounded by the fact that certificates will be issued and revoked continuously and certificate authorities may be added to or deleted from the environment dynamically. A Grid-wide mechanism is needed to create and manage a *trust fabric* so that services and users can make authentication and authorization decisions based on the most up-to-date security information. The support provided by GAARDS for management of trust fabric in a Grid environment is described below.

GAARDS Infrastructure

The GAARDS infrastructure has been developed as a suite of services and administrative tools on top of the Globus Toolkit[6,7] and its Grid Security Infrastructure (GSI) component.[b] The infrastructure consists of the following core services: *Dorian*[8] for management and federation of user identities, *Grid Trust Service*[9] for maintaining and provisioning a federated trust fabric within the Grid environment, and *Grid Grouper* for enforcing authorization policies based on both local and Grid-level groups. In the following sections, we present each of these components in greater detail.

Grid Account Management

Managing users and provisioning accounts in the Grid is complex and error-prone if done manually. A practical solution to this problem, from the point of view of both the users and their institutions, is to allow those users to authenticate with the Grid through the same mechanism by which they authenticate with their institution. Dorian[8] is a grid user management service that 1) hides the complexities of creating and managing grid credentials, and 2) provides a mechanism for users to authenticate using their institution's authentication mechanism, assuming a trust agreement is in place between Dorian and the institution—that is, Dorian is set up to trust the institution as an identity provider. Dorian's grid service interface provides mechanisms for adding, deleting, and managing trusted Identity Providers (IdPs). In a typical setup of Dorian, to obtain grid credentials (or a grid proxy), a user authenticates with his/her institution using the institution's conventional security mechanism. Upon successfully authenticating the user, the institution's security mechanism issues a digitally signed SAML assertion,[c] vouching that the user has authenticated locally. The user then sends this SAML assertion to Dorian in exchange for a grid proxy. If the user's SAML assertion is obtained from an Identity Provider trusted by Dorian, Dorian will issue the grid credentials to the user, which can be used to authenticate the user to services in the Grid.

Figure 1 illustrates an example usage scenario for Dorian. A Georgetown University user wishes to invoke a grid service that requires grid credentials. To do this, the user first supplies the application with his/her Georgetown username and password. The application client authenticates the user with the Georgetown Authentication Service, receives a signed SAML assertion, which it subsequently passes to Dorian in exchange for a grid proxy. These credentials can then be used to invoke the grid services. This illustrates how Dorian can leverage an institution's existing authentication mechanism and bring its users to the Grid.

To facilitate smaller groups or institutions without an existing IdP, Dorian has its own internal IdP. This allows users to authenticate to Dorian directly, thereby enabling them to access the Grid. It provides administrators with facilities for approving and managing users. All of the functionality provided by the Dorian IdP is made available through Dorian's grid service interface. Figure 1 illustrates a scenario of a client using the Dorian IdP to authenticate to the Grid. In this scenario, the unaffiliated user wishes to invoke a Grid

[b] http://www.globus.org

[c] http://www.oasis-open.org/committees/security/

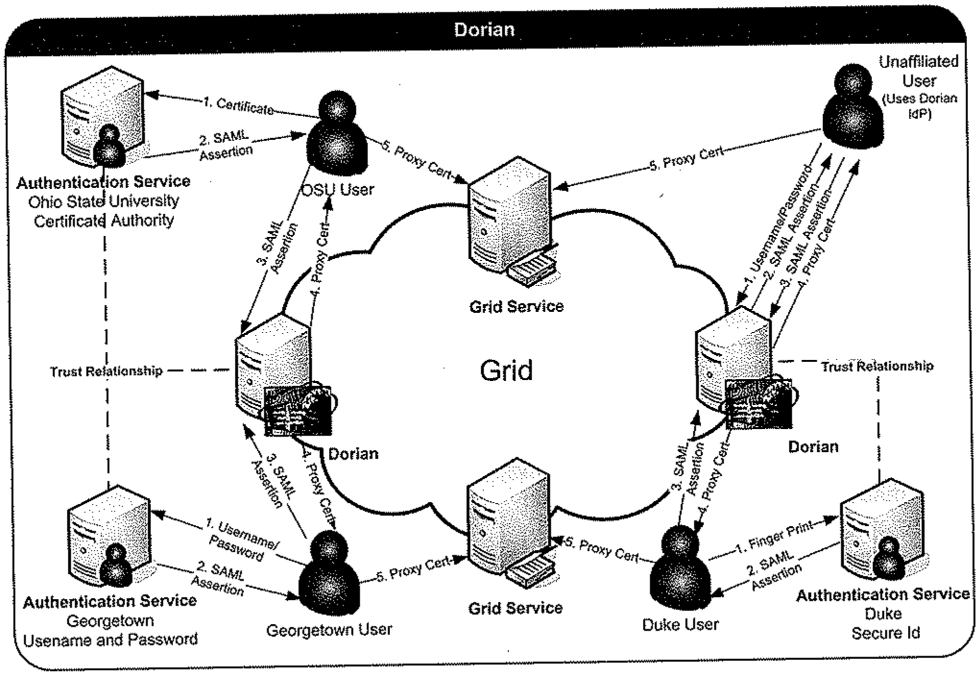

Figure 1. Example usage scenarios for Dorian. Users at Georgetown, OSU, and Duke use their institutional authentication services, while the unaffiliated user utilizes Dorian as an identity provider.

service. The user first needs to register and obtain an account, which is a one-time process. To obtain an account, the user may request an account on a Dorian instance maintained by an authority such as the NCI Center for Bioinformatics: the user can submit an account request through Dorian; her request is reviewed by the system administrators, and a decision is made on whether to grant an account to the user or not. Assume that the user has registered and been approved for an account, the user is able to authenticate with the Dorian IdP by supplying his/her username and password. Upon successfully authenticating the user, the Dorian IdP issues a SAML assertion just like institutional IdPs, which can be presented to Dorian in exchange for a grid proxy. The credentials can be used to invoke the Grid service.

In a production environment it is envisioned that multiple Dorian instances would be deployed and run in the Grid. One reason for this is that each Dorian operates a single certificate authority. Service providers in the Grid maintain a list of certificate authorities that they trust. Service providers often base their decision on whether or not to trust a given certificate authority based on the policies the certificate authority operates under. Since Dorian issues certificates to users based on assertions from organizations participating in the federation, the trust that a service provider puts in Dorian's certificate authority is closely tied to the account policies enforced at each organization. Grouping organizations with similar account policies together and associating each group with an instance of Dorian allows service providers to trust the instance(s) of Dorian whose organizations meet their required policies. Another reason for running multiple instances of Dorian in a Grid is for scalability reasons.

Authorization

The GAARDS infrastructure implements a group-based mechanism for authorization and provides a service called Grid Grouper in order to facilitate group management. Under the Grid Grouper approach, Grid services and applications can enforce authorization policy based on membership to *Grid-level* groups. Each service or application that delegates authorization decisions to Grid Grouper refers to one or more Grid Grouper service instances. In general, each service/application that uses grid grouper for authorization points to one Grid Grouper instance. Thus, the institution deploying the application to the grid determines which grid grouper instance they will use, typically also deployed by the same institution. There is no typical deployment scenario for Grid Grouper. In some Grids a single instance of Grid Grouper is employed in other Grids multiple Grid Grouper's are deployed. In determining a deployment plan for Grid Grouper, architects should consider scalability and organization factors.

Services can determine whether a caller is authorized by simply asking Grid Grouper, if the caller is in a given group. In addition, using the Grid Grouper service, an existing access control system can base its access control policies based on groups managed by the Grid Grouper as well as the local groups it manages. For example, the Common Security Module (CSM)[10] is a repository that can be deployed locally at a site to provide support for defining, managing, and enforcing access control policies—CSM has been adopted in the GAARDS infrastructure as its native access control system that can be employed by service providers that do not have an access control system in place. In such a setting, when a client invokes a Grid service, the Grid service can ask CSM whether the user can perform a given operation on a specified resource. Based on the access control policy maintained in CSM, CSM decides whether or not a user is authorized. Part of the CSM's access control policy can be based on Grid Grouper groups, allowing CSM to provide centralized access control based on these groups,

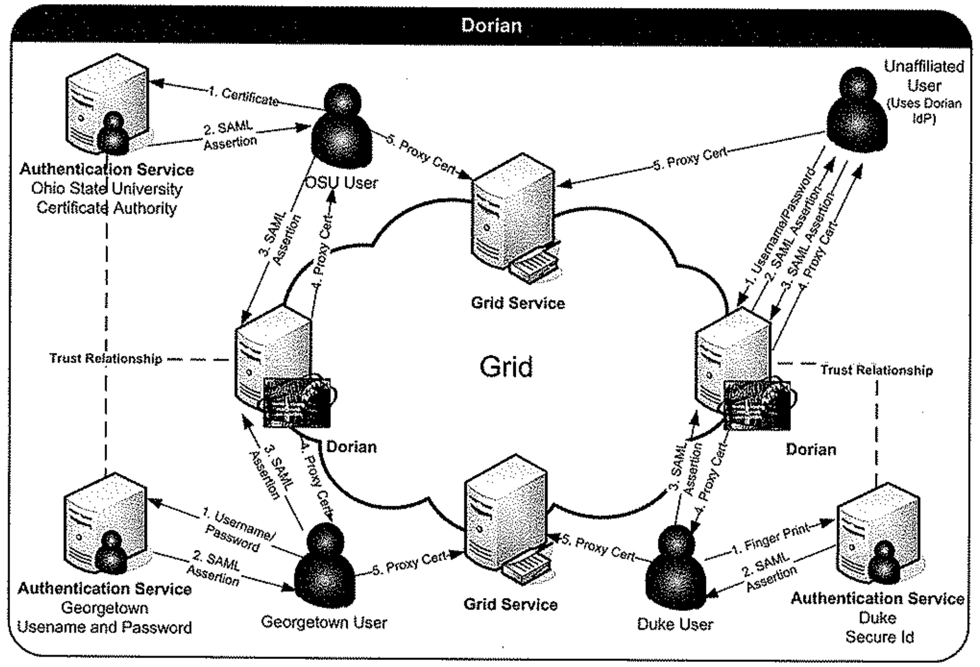

Figure 1. Example usage scenarios for Dorian. Users at Georgetown, OSU, and Duke use their institutional authentication services, while the unaffiliated user utilizes Dorian as an identity provider.

service. The user first needs to register and obtain an account, which is a one-time process. To obtain an account, the user may request an account on a Dorian instance maintained by an authority such as the NCI Center for Bioinformatics: the user can submit an account request through Dorian; her request is reviewed by the system administrators, and a decision is made on whether to grant an account to the user or not. Assume that the user has registered and been approved for an account, the user is able to authenticate with the Dorian IdP by supplying his/her username and password. Upon successfully authenticating the user, the Dorian IdP issues a SAML assertion just like institutional IdPs, which can be presented to Dorian in exchange for a grid proxy. The credentials can be used to invoke the Grid service.

In a production environment it is envisioned that multiple Dorian instances would be deployed and run in the Grid. One reason for this is that each Dorian operates a single certificate authority. Service providers in the Grid maintain a list of certificate authorities that they trust. Service providers often base their decision on whether or not to trust a given certificate authority based on the policies the certificate authority operates under. Since Dorian issues certificates to users based on assertions from organizations participating in the federation, the trust that a service provider puts in Dorian's certificate authority is closely tied to the account policies enforced at each organization. Grouping organizations with similar account policies together and associating each group with an instance of Dorian allows service providers to trust the instance(s) of Dorian whose organizations meet their required policies. Another reason for running multiple instances of Dorian in a Grid is for scalability reasons.

Authorization

The GAARDS infrastructure implements a group-based mechanism for authorization and provides a service called Grid Grouper in order to facilitate group management. Under the Grid Grouper approach, Grid services and applications can enforce authorization policy based on membership to *Grid-level* groups. Each service or application that delegates authorization decisions to Grid Grouper refers to one or more Grid Grouper service instances. In general, each service/application that uses grid grouper for authorization points to one Grid Grouper instance. Thus, the institution deploying the application to the grid determines which grid grouper instance they will use, typically also deployed by the same institution. There is no typical deployment scenario for Grid Grouper. In some Grids a single instance of Grid Grouper is employed in other Grids multiple Grid Grouper's are deployed. In determining a deployment plan for Grid Grouper, architects should consider scalability and organization factors.

Services can determine whether a caller is authorized by simply asking Grid Grouper, if the caller is in a given group. In addition, using the Grid Grouper service, an existing access control system can base its access control policies based on groups managed by the Grid Grouper as well as the local groups it manages. For example, the Common Security Module (CSM)[10] is a repository that can be deployed locally at a site to provide support for defining, managing, and enforcing access control policies—CSM has been adopted in the GAARDS infrastructure as its native access control system that can be employed by service providers that do not have an access control system in place. In such a setting, when a client invokes a Grid service, the Grid service can ask CSM whether the user can perform a given operation on a specified resource. Based on the access control policy maintained in CSM, CSM decides whether or not a user is authorized. Part of the CSM's access control policy can be based on Grid Grouper groups, allowing CSM to provide centralized access control based on these groups,

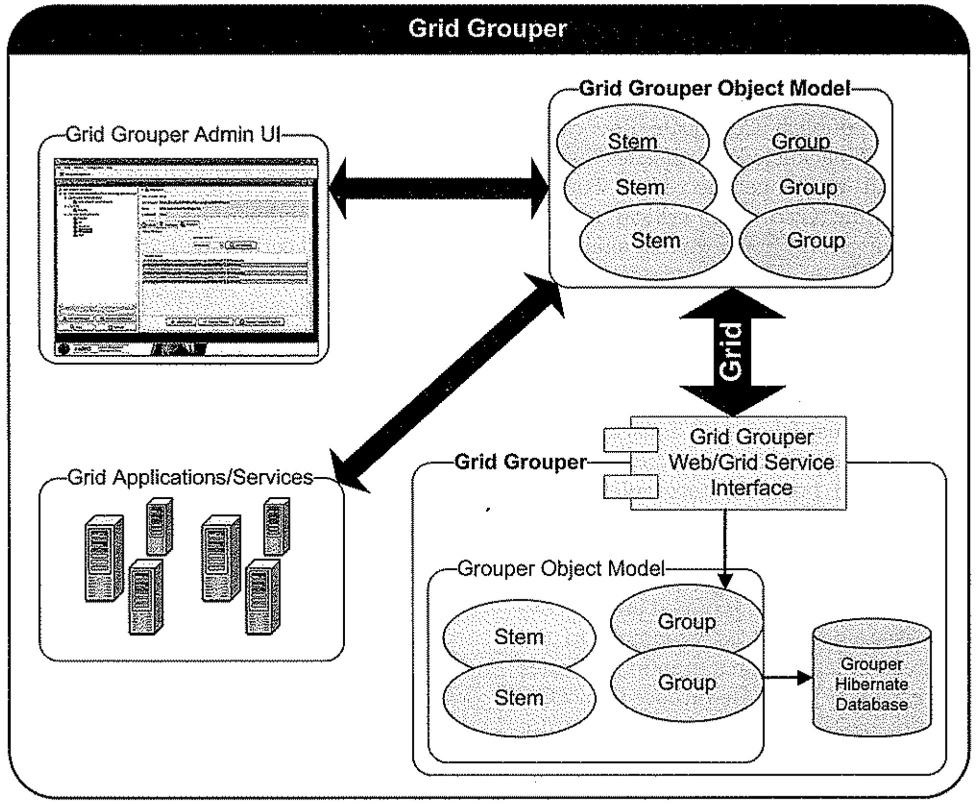
Grid Grouper
Grid Grouper Admin UI
Grid Grouper Object Model
Stem
Stem
Stem
Group
Group
Group
Grid
Grid Applications/Services
Grid Grouper
Grid Grouper Web/Grid Service Interface
Grouper Object Model
Stem
Stem
Group
Group
Grouper Hibernate Database

Figure 2. Grid Grouper Architecture. Group and stem creation and management can be done through a graphical user interface (Grid Grouper Admin UI) provided by the Grid Grouper infrastructure.

the memberships of which can cross organizational boundaries.

Grid Grouper is built on top of Grouper[d], which is an Internet2[e] initiative focused on providing tools for group management. It provides a service interface to the underlying Grouper object model—the object model can be used to enforce access control policies in applications; for example, the object model can be used for determining membership to a group in an application that allows access to a specific area of the application, if the user is a member of a specified group. With Grid Grouper, groups become available to applications and other services in the Grid. Applications and services can use the Grid Grouper object model much like they would use the Grouper object model to access and manage groups and enforce a group membership authorization policy. Grid Grouper provides support for basic group management by distributed authorities; subgroups; composite groups (whose membership is determined by the union, intersection, or relative complement of two other groups); custom group types and custom attributes; trace back of indirect membership; and delegation. The Grid Grouper object model provides an API for applications and services to access groups managed by Grid Grouper.

The architecture of Grid Grouper is illustrated in Figure 2. Grid Grouper groups are organized into namespaces or stems. Each stem can have a set of child stems and set of child groups with exception to the root stem which cannot have any child groups. Groups are compromised of a set of metadata describing the group, a set of members in the groups, and a set of privileges assigned to users for protecting access to the group. Grid Grouper provides three mechanisms for adding members to a group: 1) Directly adding a member 2) Adding a subgroup to a group 3) Making a group a composite of other groups. The Stem hierarchy in Grid Grouper is publicly visible to anyone accessing the Grid Grouper service; however, the ability to view a group within a stem publicly depends on the privileges for the group. To protect access to groups in Grid Grouper, users can be assigned the following privileges on a group: View, Read, Update, Admin, Opt-in, and Opt-out. Users with the View privilege can see that the group exists. Users with the Read privilege can read basic information about the group. Users with the Update Privilege can manage memberships to the group as well as administer View, Read, and Update privileges. Users with the Admin privilege can modify/administer anything on the group: metadata, privileges, and memberships. Users with the Opt-in privilege can add themselves as a member to a group, similarly users with the Opt-out privilege can remove themselves from a group. By default Grid Grouper grants Read and View privileges to all users on each group.

Grid Trust Fabric

In a Grid environment, there will be multiple certificate authorities each of which may be trusted by clients and services with different levels of assurance. Moreover, in a dynamic multi-institutional environment, the status of iden-

[d] http://middleware.internet2.edu/dir/groups/grouper/

[e] http://www.internet2.edu/

tities may be updated frequently. Identities and credentials can be revoked, suspended, and reinstated, or new identities can be created. In addition, the list of trusted certificate authorities (CAs) may change. In such settings, certificate authorities will frequently publish Certificate Revocation Lists (CRLs), which specify "blacklisted" certificates that the authority once issued but no longer accredits. For the security and integrity of the Grid, it is critical to both authenticate and validate a given grid credential against an accurate list of trusted certificate authorities and their corresponding CRLs.

The Grid Trust Service (GTS) is a federated infrastructure enabling the provisioning and management of a Grid trust fabric.[9] In implementing a trust fabric in a Grid environment, we envision that the trust fabric will consist of Grid users and administrators, Grid services, multiple CAs, and multiple GTS instances. GTS provides a complete Grid-enabled federated solution for registering and managing CRLs and the certificates of certificate authorities. It supports definition and management of levels of assurance, such that certificate authorities may be grouped and discovered by the level of assurance that is acceptable to the consumer. As a simple example, a CA that grants certificates automatically will have a lower level of assurance than a CA that reviews certificate requests. Due to its federated nature and its ability to create and manage arbitrary arrangements of authorities by level of assurance, GTS facilitates the curation of numerous independent trust overlays across the same physical Grid. It enables client validation, allowing a client to submit a certificate and trust requirements in exchange for a certificate verification and validation decision. The GTS infrastructure can also be used as a distribution mechanism of the CRLs from CAs.

In order to model different levels of trust in the trust fabric, the GTS provides a mechanism for its administrators to define and manage trust levels. When certificate authorities are registered into the trust fabric they are assigned one or more trust levels. Clients can specify the level of trust that they require when discovering trusted CAs or when requesting validation. Trust levels in the GTS each consist of a unique name (value) and description. The unique name is used to implicitly bind a certificate authority to a trust level. The description is used as a human readable method of understanding what a specific trust level represents. While GTS facilitates the management of certificate authority lists, the trust establishment with a CA and setting its trust level is a manual process. That is, the administrator of a GTS instance is expected to exchange correspondence with the owner of the CA to be added to the list of CAs managed by the GTS instance. Once a trust level, or set of trust levels, has been established, the CA can be added to the list of CAs so that it can be discovered by users and services.

The flexibility of GTS allows many possible deployment scenarios. For instance, an institution may set up a local CA and GTS instance. Alternately, a group of organizations may all share a common CA for certificates and a GTS to maintain the list of trusted external CAs. In any deployment, each Grid user will be given a certificate, signed by a CA that can be used by services to authenticate the user. Similarly, each Grid service will be given a certificate, also signed by a CA, so that a client application, user, or other service can check the integrity of the service. As deployments leveraging the GTS to maintain the trust fabric are effectively delegating this responsibility to the GTS, it is imperative the GTS instance(s) can be trusted. There are multiple possible deployment options for assigning certificates to GTS instances. A possible way is that each GTS instance has a self-signed certificate (i.e., serving as its own CA). In such a deployment, clients and services are manually configured to trust the self-signed certificates of the GTS instances they intent to interact with. Alternatively, there can be one (or a few) *trusted root-CA*, which will be used to assign the certificates to each GTS instance. Installations in the Grid are then bootstrapped to trust this authority or small set of authorities.

In a large Grid environment, it is desirable to have a federated trust fabric for redundancy and scalability, and for the integration of multiple trust overlays. A possible way of federating GTS instances is to create a hierarchical structure, in which there are *authority GTS instances* and *subordinate GTS instances*. The authority GTS instances maintain lists of trusted CAs and CRLs and synchronize with CAs for updates. The subordinate GTS instances can be designed to synchronize with one or more authority GTS instances. In this way, when the state of the trust fabric changes (e.g., because of publishing a new CRL), the updates need not be broadcast to all GTS instances individually.

Putting GAARDS to Work

We now describe how GAARDS can be employed in a Grid environment using an example application scenario. The application scenario draws from a multi-institutional clinical trial that involves collection and analysis of images obtained from patients. It is based on common use cases identified by the In-vivo Imaging Workspace of caBIG™. We should note that while the example scenario focuses on imaging studies, the security requirements can easily be generalized to other types of data and applications, in which one or more datasets are generated at multiple locations and accessed by remote clients.

Our description of how GAARDS is employed is a simplified version of the security mechanism implemented in the caGrid-enabled in-vivo imaging middleware (IVIM),[11–13] which is designed to provide secure, federated access to image databases, image analytical resources, and existing DICOM-based data repositories in the Grid and to facilitate development of Grid-enabled in vivo imaging applications. The IVIM is developed by our group and is motivated by biomedical image review and analysis use cases like the example scenario as well as other use cases from the caBIG In-vivo Imaging Workspace.

Example Application Scenario

The application scenario is a large clinical trial, in which Radiology images obtained from patients are analyzed along with other laboratory results to assess the effectiveness of a specific drug therapy. Large clinical trials often involve multiple-institutions and cooperative oncology groups that are supported by multiple data repositories. In these studies, patients may be recruited to the clinical trial at multiple institutions and reviews of images are performed remotely by experts from different institutions. Recruited patients are registered by the study coordinator into a cancer tities may be updated frequently. Identities and credentials can be revoked, suspended, and reinstated, or new identities can be created. In addition, the list of trusted certificate authorities (CAs) may change. In such settings, certificate authorities will frequently publish Certificate Revocation Lists (CRLs), which specify "blacklisted" certificates that the authority once issued but no longer accredits. For the security and integrity of the Grid, it is critical to both authenticate and validate a given grid credential against an accurate list of trusted certificate authorities and their corresponding CRLs.

The Grid Trust Service (GTS) is a federated infrastructure enabling the provisioning and management of a Grid trust fabric.[9] In implementing a trust fabric in a Grid environment, we envision that the trust fabric will consist of Grid users and administrators, Grid services, multiple CAs, and multiple GTS instances. GTS provides a complete Grid-enabled federated solution for registering and managing CRLs and the certificates of certificate authorities. It supports definition and management of levels of assurance, such that certificate authorities may be grouped and discovered by the level of assurance that is acceptable to the consumer. As a simple example, a CA that grants certificates automatically will have a lower level of assurance than a CA that reviews certificate requests. Due to its federated nature and its ability to create and manage arbitrary arrangements of authorities by level of assurance, GTS facilitates the curation of numerous independent trust overlays across the same physical Grid. It enables client validation, allowing a client to submit a certificate and trust requirements in exchange for a certificate verification and validation decision. The GTS infrastructure can also be used as a distribution mechanism of the CRLs from CAs.

In order to model different levels of trust in the trust fabric, the GTS provides a mechanism for its administrators to define and manage trust levels. When certificate authorities are registered into the trust fabric they are assigned one or more trust levels. Clients can specify the level of trust that they require when discovering trusted CAs or when requesting validation. Trust levels in the GTS each consist of a unique name (value) and description. The unique name is used to implicitly bind a certificate authority to a trust level. The description is used as a human readable method of understanding what a specific trust level represents. While GTS facilitates the management of certificate authority lists, the trust establishment with a CA and setting its trust level is a manual process. That is, the administrator of a GTS instance is expected to exchange correspondence with the owner of the CA to be added to the list of CAs managed by the GTS instance. Once a trust level, or set of trust levels, has been established, the CA can be added to the list of CAs so that it can be discovered by users and services.

The flexibility of GTS allows many possible deployment scenarios. For instance, an institution may set up a local CA and GTS instance. Alternately, a group of organizations may all share a common CA for certificates and a GTS to maintain the list of trusted external CAs. In any deployment, each Grid user will be given a certificate, signed by a CA that can be used by services to authenticate the user. Similarly, each Grid service will be given a certificate, also signed by a CA, so that a client application, user, or other service can check the integrity of the service. As deployments leveraging the GTS to maintain the trust fabric are effectively delegating this responsibility to the GTS, it is imperative the GTS instance(s) can be trusted. There are multiple possible deployment options for assigning certificates to GTS instances. A possible way is that each GTS instance has a self-signed certificate (i.e., serving as its own CA). In such a deployment, clients and services are manually configured to trust the self-signed certificates of the GTS instances they intent to interact with. Alternatively, there can be one (or a few) *trusted root-CA*, which will be used to assign the certificates to each GTS instance. Installations in the Grid are then bootstrapped to trust this authority or small set of authorities.

In a large Grid environment, it is desirable to have a federated trust fabric for redundancy and scalability, and for the integration of multiple trust overlays. A possible way of federating GTS instances is to create a hierarchical structure, in which there are *authority GTS instances* and *subordinate GTS instances*. The authority GTS instances maintain lists of trusted CAs and CRLs and synchronize with CAs for updates. The subordinate GTS instances can be designed to synchronize with one or more authority GTS instances. In this way, when the state of the trust fabric changes (e.g., because of publishing a new CRL), the updates need not be broadcast to all GTS instances individually.

Putting GAARDS to Work

We now describe how GAARDS can be employed in a Grid environment using an example application scenario. The application scenario draws from a multi-institutional clinical trial that involves collection and analysis of images obtained from patients. It is based on common use cases identified by the In-vivo Imaging Workspace of caBIG™. We should note that while the example scenario focuses on imaging studies, the security requirements can easily be generalized to other types of data and applications, in which one or more datasets are generated at multiple locations and accessed by remote clients.

Our description of how GAARDS is employed is a simplified version of the security mechanism implemented in the caGrid-enabled in-vivo imaging middleware (IVIM),[11–13] which is designed to provide secure, federated access to image databases, image analytical resources, and existing DICOM-based data repositories in the Grid and to facilitate development of Grid-enabled in vivo imaging applications. The IVIM is developed by our group and is motivated by biomedical image review and analysis use cases like the example scenario as well as other use cases from the caBIG In-vivo Imaging Workspace.

Example Application Scenario

The application scenario is a large clinical trial, in which Radiology images obtained from patients are analyzed along with other laboratory results to assess the effectiveness of a specific drug therapy. Large clinical trials often involve multiple-institutions and cooperative oncology groups that are supported by multiple data repositories. In these studies, patients may be recruited to the clinical trial at multiple institutions and reviews of images are performed remotely by experts from different institutions. Recruited patients are registered by the study coordinator into a cancer clinical participant database. This database contains patient name, demographic information, medical history number, initial set of laboratory values, and clinical observations. PET/CT images obtained at periodic intervals from each patient are stored in image databases at respective institutions. Since these images are linked to patient-related information, a mechanism is needed to protect patient privacy, and intellectual property of the researchers. Figure 3 illustrates the application scenario in a Grid environment.

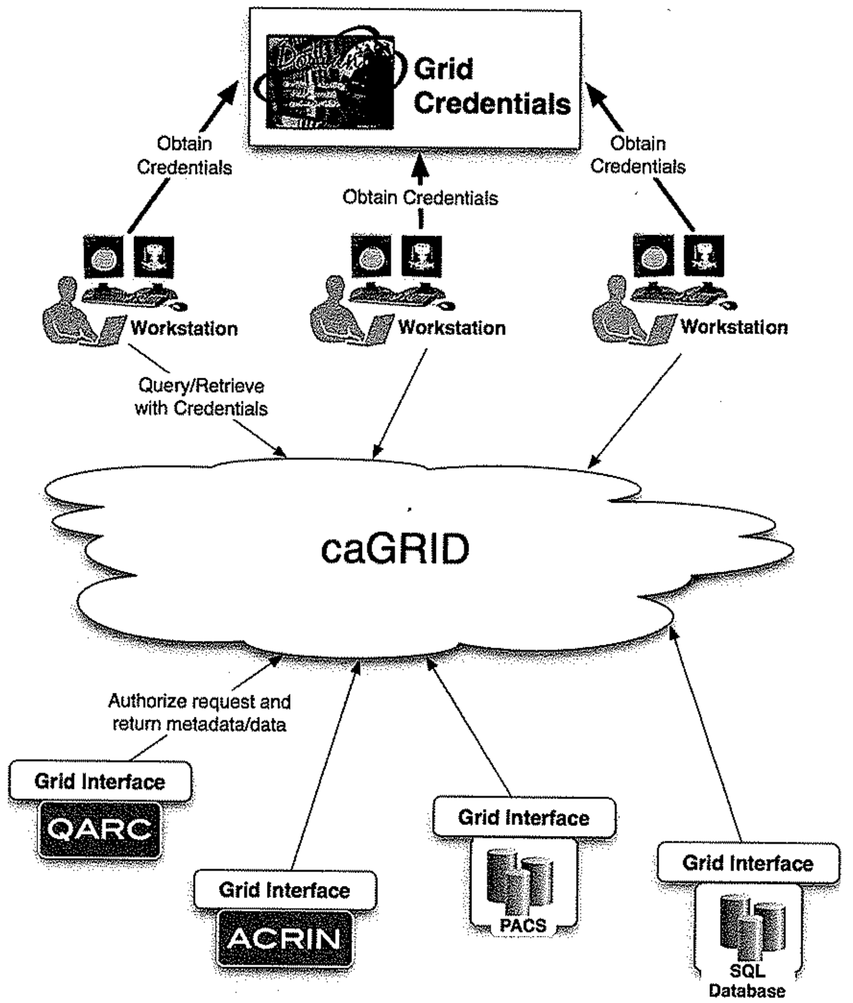

Figure 3. Multiple clients accessing Grid-enabled imaging services. In this setting, a client (e.g., a reviewer) needs to have Grid credentials to be able to interact with secure image services.

The images collected in the study are reviewed by expert Radiologists. When a Radiologist is ready to interpret an image, she uses a review workstation to access the image databases, implemented as secure data services. The Radiologist browses and annotates the images of interest based on her analysis and diagnosis. The review and analysis results (e.g., nodule location, shape, and texture data) in the form of annotations on the images are inserted to a results/annotation database, which also is implemented as a secure data service. In this application example, remote clients with the "Reviewer" role are authorized to access only the images in the clinical trial and are restricted from the other images in the databases. The study manager, who is assigned a "Trial Manager" role, is authorized to accrue subjects and review existing patient records, but not to access the image data.

Authentication

In order for the reviewers and the study coordinator in the application scenario to communicate with secure data and analysis services, each of them needs a Grid credential, i.e., a Grid-wide identity that can be authenticated. Dorian provides two methods to register for a Grid user account. The user can register 1) directly with Dorian, or 2) indirectly via his/her existing user account provided by his/her institution. To illustrate these two methods, we assume that the study coordinator wants to use his/her existing institutional account and that one of the reviewers is affiliated with an institution without a local IdP system.

The identity provider system of the study coordinator's institution is registered with Dorian as a trusted identity provider. When the study coordinator logs on to his/her client workstation with his/her username and password, the client program authenticates the user with his/her institution's local identity provider system. After successfully authenticating the user, the system issues a digitally signed SAML assertion, vouching that the user has authenticated. The review workstation program then sends this SAML assertion to Dorian in exchange for Grid credentials using the Dorian client APIs. These credentials can then be used to invoke the services in the environment. In the case of the reviewer, the reviewer should first request an account with Dorian. After the reviewer has obtained an account, she can log on to the environment using an imaging client application which supports Dorian-based authentication. Dorian will issue Grid credentials in the form of a proxy certificate for the reviewer that can be used for authentication.

After a user (the reviewer or the study coordinator) has obtained Grid credentials from Dorian, she may submit a request to an image data service to retrieve a set of images or to an analytical service to process images. The request will carry her Grid credentials and present them to the service. The service then authenticates the reviewer by validating the credentials. Part of the verification process is checking that the supplied Grid credentials were issued by a trusted Grid credential provider (i.e., Dorian or other certificate authorities). To perform this verification, the service checks with the Grid Trust Service (GTS) to ensure that the credentials provided were issued by a trusted credential provider that meets the service's level of assurance requirements.

Access Control and Authorization

In our example application, when a reviewer wants to examine a specific patient's images stored at a remote service, the primary decision points for data authorization reside on the service side. The authorization mechanism must enforce a policy that will deny all users from accessing the data by default, only authorizing specific users for each unit of data, which may correspond to a single image. In our implementation, this is achieved with a two-level authorization scheme. The first level is the service level authorization. At this level, an authorization decision is made on whether a user can access the functionality provided by the service or not. The decision is made based on the authorization group the user belongs to. The second level of authorization is the data level authorization. Once a user is authorized to access the service, he/she can submit requests to retrieve data from the service. At this stage, the data-level authorization is used to control access to individual data objects (e.g., images and metadata on images) and/or sets of data objects. For example, a reviewer may have access rights only to a patient's images obtained in studies that were performed at the hospital for which the reviewer works. The data-level authorization facilitates a finer grain access control on data.

Implementation of service- and data-level authorization requires several components: data group and membership definition based on authorization policy, authorization decision implementation, and data and user group management via a standard interface. These components are described next.

Group Definitions Using Grid Grouper

The service-level authorization requires that a group administrator first defines user groups in Grid Grouper and adds users to the groups. Group definitions may be based on virtual and real organizations (e.g., "Institution A," "Institution B"), or may be based on functional roles (e.g., "Reviewer" or "Study Coordinator"). A service administrator can then configure the service and operation authorization by adding one or more users. For instance, if a service allows access to only the "Study Coordinator" group and a reviewer belongs to the User Group "Reviewer," the reviewer will not be able to invoke any of the service's methods.

Data-level authorization policies can be formulated as rules such as "User A in User Group B has access to Data Instance C and D." The rules can be represented as group membership statements: "*User Group* B is a member of *Data Group* C (or D) which correspond to Data Instance C (or D)," and authorization policy statements: "if User A in *User Group* B is in *Data Group* C (or D) then grant User A access to Data Instance C (or D)." Members of a child group are by definition members of the parent group as well. This approach transforms the original authorization policy from a user centric view to a data centric view by representing each data instance with a data group and authorization policy as group memberships. Management of the authorization policy is done through the management of the group membership. This approach also simplifies the evaluation of the authorization policy by transforming it into a group membership determination. The data-level authorization policies can be defined as data groups in Grid Grouper. Authorization exceptions, such as "User A can access Patient 1 and all of its studies and series, except for Study 1A", are handled via Boolean operations on the groups including AND, OR, and Complement operations.

Example data group definitions in Grid Grouper are shown in Figure 4. Each visit generates a "Study," which may contain one or more exams whose output is represented by "Series." Each series contains one or more images. Each study also maintains patient information that is typically used as an implicit hierarchy level in review workstations for the purpose of organization.

Authorization Decision Implementation

The typical use case for data authorization is that a user may be granted access to a particular data set or a subset of it (see Figure 5). A user may be given access to a *Patient*, in which case she would also gain access to all its children studies and series, unless an explicit exception is made for the children data instance. Since images within a series are parts of the same dataset, authorization is only managed at the series level at its deepest traversal.

User Group Management

To ensure correct, consistent and efficient creation of the groups and their members based on the described membership rules, a configuration step creates the groups and sets up the memberships according to the rules described in the previous section. Once the data groups have been created in Grid Grouper, user groups can be added to these data groups to allow authorization. The user groups are managed using the client interface provided by GAARDS.

Related Work

A number of toolkits and service architectures have been developed to address issues in Grid security. Butler et al.[14] discuss the creation and deployment of an authentication and authorization infrastructure for the Grid. Humphrey et al.[15] discuss the challenges that must be faced in securing grid environments by grouping required activities under four categories as naming and authentication; secure communication; trust, policy, and authorization; and enforcement of access control. Sinnott et al.[16] describe a federated security model for virtual organizations in the

the reviewer, the reviewer should first request an account with Dorian. After the reviewer has obtained an account, she can log on to the environment using an imaging client application which supports Dorian-based authentication. Dorian will issue Grid credentials in the form of a proxy certificate for the reviewer that can be used for authentication.

After a user (the reviewer or the study coordinator) has obtained Grid credentials from Dorian, she may submit a request to an image data service to retrieve a set of images or to an analytical service to process images. The request will carry her Grid credentials and present them to the service. The service then authenticates the reviewer by validating the credentials. Part of the verification process is checking that the supplied Grid credentials were issued by a trusted Grid credential provider (i.e., Dorian or other certificate authorities). To perform this verification, the service checks with the Grid Trust Service (GTS) to ensure that the credentials provided were issued by a trusted credential provider that meets the service's level of assurance requirements.

Access Control and Authorization

In our example application, when a reviewer wants to examine a specific patient's images stored at a remote service, the primary decision points for data authorization reside on the service side. The authorization mechanism must enforce a policy that will deny all users from accessing the data by default, only authorizing specific users for each unit of data, which may correspond to a single image. In our implementation, this is achieved with a two-level authorization scheme. The first level is the service level authorization. At this level, an authorization decision is made on whether a user can access the functionality provided by the service or not. The decision is made based on the authorization group the user belongs to. The second level of authorization is the data level authorization. Once a user is authorized to access the service, he/she can submit requests to retrieve data from the service. At this stage, the data-level authorization is used to control access to individual data objects (e.g., images and metadata on images) and/or sets of data objects. For example, a reviewer may have access rights only to a patient's images obtained in studies that were performed at the hospital for which the reviewer works. The data-level authorization facilitates a finer grain access control on data.

Implementation of service- and data-level authorization requires several components: data group and membership definition based on authorization policy, authorization decision implementation, and data and user group management via a standard interface. These components are described next.

Group Definitions Using Grid Grouper

The service-level authorization requires that a group administrator first defines user groups in Grid Grouper and adds users to the groups. Group definitions may be based on virtual and real organizations (e.g., "Institution A," "Institution B"), or may be based on functional roles (e.g., "Reviewer" or "Study Coordinator"). A service administrator can then configure the service and operation authorization by adding one or more users. For instance, if a service allows access to only the "Study Coordinator" group and a reviewer belongs to the User Group "Reviewer," the reviewer will not be able to invoke any of the service's methods.

Data-level authorization policies can be formulated as rules such as "User A in User Group B has access to Data Instance C and D." The rules can be represented as group membership statements: "*User Group* B is a member of *Data Group* C (or D) which correspond to Data Instance C (or D)," and authorization policy statements: "if User A in *User Group* B is in *Data Group* C (or D) then grant User A access to Data Instance C (or D)." Members of a child group are by definition members of the parent group as well. This approach transforms the original authorization policy from a user centric view to a data centric view by representing each data instance with a data group and authorization policy as group memberships. Management of the authorization policy is done through the management of the group membership. This approach also simplifies the evaluation of the authorization policy by transforming it into a group membership determination. The data-level authorization policies can be defined as data groups in Grid Grouper. Authorization exceptions, such as "User A can access Patient 1 and all of its studies and series, except for Study 1A", are handled via Boolean operations on the groups including AND, OR, and Complement operations.

Example data group definitions in Grid Grouper are shown in Figure 4. Each visit generates a "Study," which may contain one or more exams whose output is represented by "Series." Each series contains one or more images. Each study also maintains patient information that is typically used as an implicit hierarchy level in review workstations for the purpose of organization.

Authorization Decision Implementation

The typical use case for data authorization is that a user may be granted access to a particular data set or a subset of it (see Figure 5). A user may be given access to a *Patient*, in which case she would also gain access to all its children studies and series, unless an explicit exception is made for the children data instance. Since images within a series are parts of the same dataset, authorization is only managed at the series level at its deepest traversal.

User Group Management

To ensure correct, consistent and efficient creation of the groups and their members based on the described membership rules, a configuration step creates the groups and sets up the memberships according to the rules described in the previous section. Once the data groups have been created in Grid Grouper, user groups can be added to these data groups to allow authorization. The user groups are managed using the client interface provided by GAARDS.

Related Work

A number of toolkits and service architectures have been developed to address issues in Grid security. Butler et al.[14] discuss the creation and deployment of an authentication and authorization infrastructure for the Grid. Humphrey et al.[15] discuss the challenges that must be faced in securing grid environments by grouping required activities under four categories as naming and authentication; secure communication; trust, policy, and authorization; and enforcement of access control. Sinnott et al.[16] describe a federated security model for virtual organizations in the grid. In this model, each organization manages its security, delegating to trusted local or remote entities as necessary. Dwoskin et al.[17] focus specifically on the security issues in interactive applications running on the grid. They show how the grid security infrastructure can be extended to set up secure, interactive sessions at the remote host. The main differences of the GAARDS infrastructure from the previous work is that it provides mechanisms for federation of existing institutional security infrastructure and local user accounts, for access

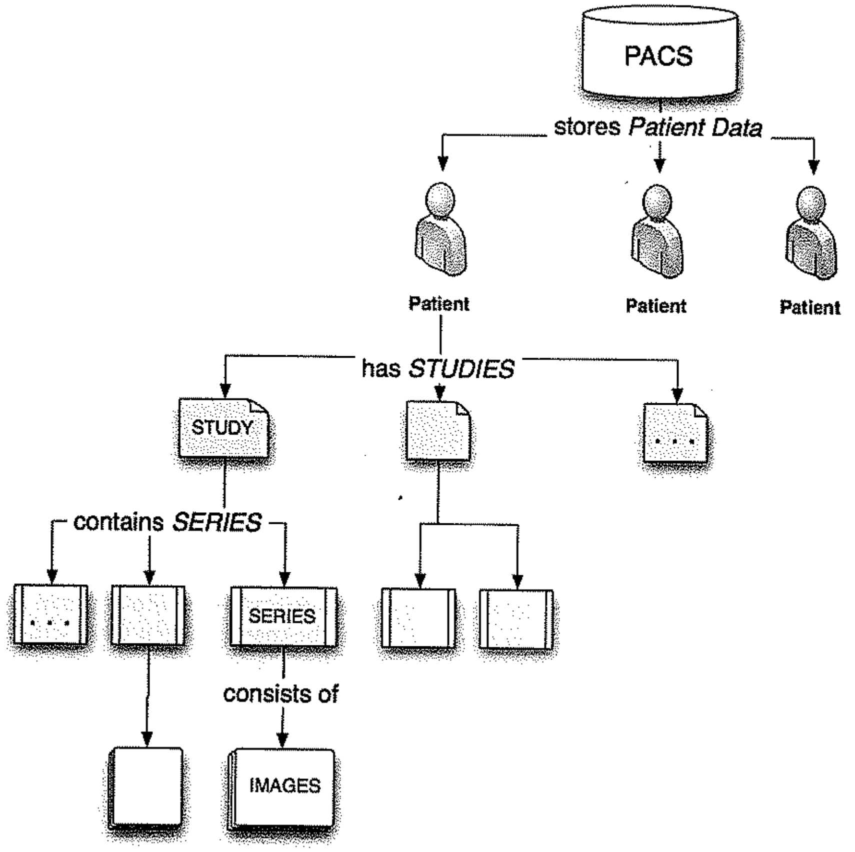

Figure 4. Data groups for images are organized hierarchically by patient, study, series, and image.

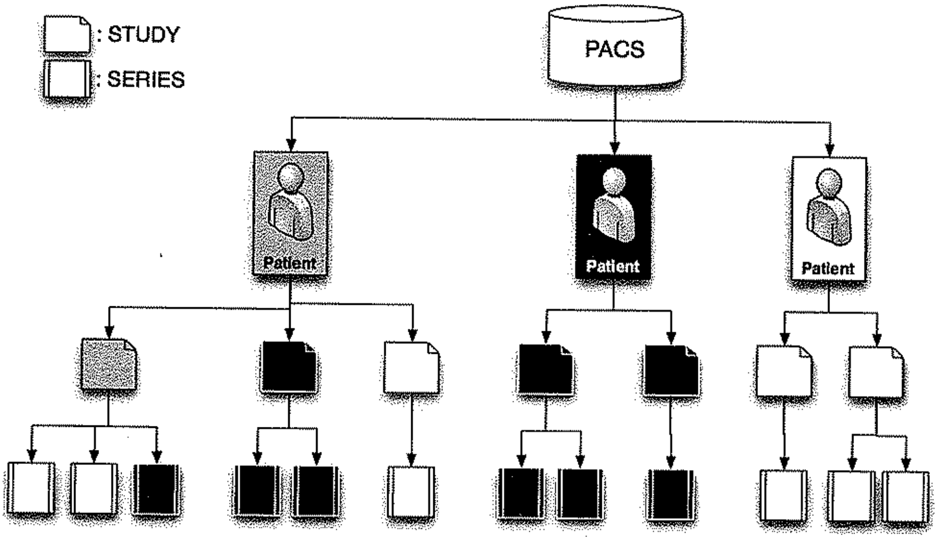

Figure 5. A user group's access to data can be controlled for individual patient, study, or series. Authorization is inherited by children nodes in the information hierarchy as shown here for the two patients on the right, where the user is allowed to access one patient (white documents) while disallowed from accessing the other (black documents). The inherited authorization can be overridden at a child node, as shown in the leftmost patient, where the user is allowed to access the patient, except for one study, and one series.

control based on Grid-wide and local groups, and for the management of trust fabric in a Grid environment.

Few middleware systems have been developed to facilitate the creation, management, and federation of user credentials in the Grid. One such system is the MyProxy Credential Management Service.[18] It enables users to supply a password to securely create grid proxies based on their private key and certificate stored in the MyProxy repository. When the predecessor of Dorian was developed, MyProxy did not have a built in certificate authority and it required its users to upload their private keys and certificates. This was one of the factors in the decision to develop Dorian, because a built-in certificate authority greatly reduces the complexity of creating grid user credentials. Since that time, MyProxy (MyProxy 3.0) has added the ability to act as a Certificate Authority. The main difference between MyProxy and the GAARDS Dorian is the support for Web Service interfaces in Dorian and the ability of Dorian to federate existing users in institutions to the Grid. The Portal-based User Registration System (PURSe)[f] provides a friendly interface for users of web applications to register for and obtain access to their grid credentials. PURSe uses SimpleCA[g] and MyProxy for the creation and management of grid credentials. PURSe differs from GAARDS as it is purely web-based and is a toolkit used for simplifying the development of web applications, whereas the Dorian service of GAARDS is a free-standing grid service focused on solving the identity management and federation problem. The GAMA infrastructure[19] is based on Globus GSI.[h], [6, 7] It consists of a backend server for creating and managing X.509 credentials for users and portal interfaces for users and administrators to access its functions. The Dorian component of GAARDS provides a Grid service infrastructure, based on the use of public key certificates and SAML assertions, for managing and federating user identities in the Grid. It makes use of SAML and Grid certificates to authenticate users to the Grid environment through their institution's authentication mechanism.

The GAARDS infrastructure provides a suite of core services and tools to support security requirements in a Grid environment. These tools and services leverage community accepted mechanisms, e.g., the Grouper model from the Internet2 initiative, X.509 certificates, SAML for local authentication assertions. There are also efforts in the healthcare information technology developing standards for security. Recently the Healthcare Information Technology Standards Panel (HITSP; http://www.hitsp.org) has identified and defined such security components as entity identity assertion, access control transaction package. These components make use of standards such as SAML, SOAP, and the IHE (Integrating the Healthcare Enterprise; http://www.ihe.net) Cross-Enterprise User Assertion Profile[i]. Some of these standards (e.g., SOAP, SAML) are already leveraged in GAARDS. As new standards develop, we plan to review them to determine how they may be employed by GAARDS.

Group- and role-based access control is a common methodology employed in security infrastructures.[20–22] Earlier work in this area focused on mechanisms to enforce access control at a single institution. More recent works have developed systems to support authorization in distributed environments.[23, 24] The Grid Grouper component of GAARDS is similar to these systems in that it provides group-based authorization in a distributed environment. It implements a WSRF compliant solution using the Grouper[j] system from the Internet2 initiative[k].

Management of trust is recognized as an important component of security in distributed environments. Manchala[25] describes trust models and metrics in e-commerce applications and discusses how risk can be analyzed under different models. Azzedin and Maheswaran[26] present a trust model for a Grid environment. Their approach models trust based on behavior and reputation of entities that interact with others. They describe techniques for computing this type of behavior trust, how it evolves in an environment, and how it can be managed in a Grid setting. GridAdmin, proposed by Quillinan et al.[27] is a system that provides support for automatic handling of requests for administrative actions and resource allocations. The system incorporates trust metrics in responding to and ranking such requests. Weaver et al.[28] discuss trust-sharing agreements and an IT infrastructure for federated security in distributed healthcare applications. Grandison and Sloman[29] present a toolkit that provides support for specifying and monitoring trust relationships for Internet applications. Ahsant et al.[30] discuss how business trust relationships can be propagated to the Grid environment and how these relationships can be federated dynamically. Basney et al. describe extensions to the basic Grid security architecture in order to support negotiation and dynamic establishment trust relationships between entities in the Grid. Our work complements the previous work on trust management in that earlier work focused on specification of trust and establishment and management of trust between entities. The GAARDS GTS, on the other hand, enables Grid-wide management of trusted Certificate Authorities with different trust levels.

Conclusions

Comprehensive security infrastructures are critical to the success of large-scale, multi-institutional biomedical research efforts. GAARDS is designed to address challenging issues such as federation and provisioning of user credentials, group-based access control, and management of trust fabric in federated environments. The GAARDS infrastructure is in use in the production Grid in the caBIG program, which is a national-scale informatics effort to facilitate sharing of data and tools among biomedical researchers. We believe that security systems like GAARDS will be increasingly important as research in biomedicine becomes more collaborative.

[f] http://www.grids-center.org/solutions/purse/

[g] http://www.globus.org/toolkit/docs/4.0/security/simpleca/index.html

[h] http://www.globus.org/security/overview.html

[i] http://wiki.ihe.net/index.php?title=Cross-Enterprise_User_Assertion_Profile

[j] http://middleware.internet2.edu/dir/groups/grouper/

[k] http://www.internet2.edu/

control based on Grid-wide and local groups, and for the management of trust fabric in a Grid environment.

Few middleware systems have been developed to facilitate the creation, management, and federation of user credentials in the Grid. One such system is the MyProxy Credential Management Service.[18] It enables users to supply a password to securely create grid proxies based on their private key and certificate stored in the MyProxy repository. When the predecessor of Dorian was developed, MyProxy did not have a built in certificate authority and it required its users to upload their private keys and certificates. This was one of the factors in the decision to develop Dorian, because a built-in certificate authority greatly reduces the complexity of creating grid user credentials. Since that time, MyProxy (MyProxy 3.0) has added the ability to act as a Certificate Authority. The main difference between MyProxy and the GAARDS Dorian is the support for Web Service interfaces in Dorian and the ability of Dorian to federate existing users in institutions to the Grid. The Portal-based User Registration System (PURSe)[f] provides a friendly interface for users of web applications to register for and obtain access to their grid credentials. PURSe uses SimpleCA[g] and MyProxy for the creation and management of grid credentials. PURSe differs from GAARDS as it is purely web-based and is a toolkit used for simplifying the development of web applications, whereas the Dorian service of GAARDS is a free-standing grid service focused on solving the identity management and federation problem. The GAMA infrastructure[19] is based on Globus GSI.[h], [6, 7] It consists of a backend server for creating and managing X.509 credentials for users and portal interfaces for users and administrators to access its functions. The Dorian component of GAARDS provides a Grid service infrastructure, based on the use of public key certificates and SAML assertions, for managing and federating user identities in the Grid. It makes use of SAML and Grid certificates to authenticate users to the Grid environment through their institution's authentication mechanism.

The GAARDS infrastructure provides a suite of core services and tools to support security requirements in a Grid environment. These tools and services leverage community accepted mechanisms, e.g., the Grouper model from the Internet2 initiative, X.509 certificates, SAML for local authentication assertions. There are also efforts in the healthcare information technology developing standards for security. Recently the Healthcare Information Technology Standards Panel (HITSP; http://www.hitsp.org) has identified and defined such security components as entity identity assertion, access control transaction package. These components make use of standards such as SAML, SOAP, and the IHE (Integrating the Healthcare Enterprise; http://www.ihe.net) Cross-Enterprise User Assertion Profile[i]. Some of these standards (e.g., SOAP, SAML) are already leveraged in GAARDS. As new standards develop, we plan to review them to determine how they may be employed by GAARDS.

Group- and role-based access control is a common methodology employed in security infrastructures.[20–22] Earlier work in this area focused on mechanisms to enforce access control at a single institution. More recent works have developed systems to support authorization in distributed environments.[23, 24] The Grid Grouper component of GAARDS is similar to these systems in that it provides group-based authorization in a distributed environment. It implements a WSRF compliant solution using the Grouper[j] system from the Internet2 initiative[k].

Management of trust is recognized as an important component of security in distributed environments. Manchala[25] describes trust models and metrics in e-commerce applications and discusses how risk can be analyzed under different models. Azzedin and Maheswaran[26] present a trust model for a Grid environment. Their approach models trust based on behavior and reputation of entities that interact with others. They describe techniques for computing this type of behavior trust, how it evolves in an environment, and how it can be managed in a Grid setting. GridAdmin, proposed by Quillinan et al.[27] is a system that provides support for automatic handling of requests for administrative actions and resource allocations. The system incorporates trust metrics in responding to and ranking such requests. Weaver et al.[28] discuss trust-sharing agreements and an IT infrastructure for federated security in distributed healthcare applications. Grandison and Sloman[29] present a toolkit that provides support for specifying and monitoring trust relationships for Internet applications. Ahsant et al.[30] discuss how business trust relationships can be propagated to the Grid environment and how these relationships can be federated dynamically. Basney et al. describe extensions to the basic Grid security architecture in order to support negotiation and dynamic establishment trust relationships between entities in the Grid. Our work complements the previous work on trust management in that earlier work focused on specification of trust and establishment and management of trust between entities. The GAARDS GTS, on the other hand, enables Grid-wide management of trusted Certificate Authorities with different trust levels.

Conclusions

Comprehensive security infrastructures are critical to the success of large-scale, multi-institutional biomedical research efforts. GAARDS is designed to address challenging issues such as federation and provisioning of user credentials, group-based access control, and management of trust fabric in federated environments. The GAARDS infrastructure is in use in the production Grid in the caBIG program, which is a national-scale informatics effort to facilitate sharing of data and tools among biomedical researchers. We believe that security systems like GAARDS will be increasingly important as research in biomedicine becomes more collaborative.

[f] http://www.grids-center.org/solutions/purse/

[g] http://www.globus.org/toolkit/docs/4.0/security/simpleca/index.html

[h] http://www.globus.org/security/overview.html

[i] http://wiki.ihe.net/index.php?title=Cross-Enterprise_User_Assertion_Profile

[j] http://middleware.internet2.edu/dir/groups/grouper/

[k] http://www.internet2.edu/

References ▪

1. Saltz J, Oster S, Hastings S, Langella S, Kurc T, Sanchez W, et al. caGrid: Design and Implementation of the Core Architecture of the Cancer Biomedical Informatics Grid. Bioinform 2006;22(15): 1910–6.
2. Langella S, Oster, S, Hastings, S, Siebenlist, F, Phillips, J, Ervin, et al. The Cancer Biomedical Informatics Grid (caBIG™) Security Infrastructure. Proceedings of the 2007 American Medical Informatics Association (AMIA) Annual Symposium. Chicago, IL; 2007.
3. Oster S, Hastings, S, Langella, S, Ervin, D, Madduri, R, Kurc, T, et al. caGrid 1.0: A Grid Enterprise Architecture for Cancer Research. Proceedings of the 2007 American Medical Informatics Association (AMIA) Annual Symposium. Chicago, IL; 2007.
4. Czajkowski K, Ferguson DF, Foster I, Frey J, Graham S, Sedukhin I, et al. The WS-Resource Framework version 1.0. 2004 [cited 2004; Available at http://www.globus.org/wsrf/specs/ws-wsrf.pdf. Accessed March 2008.
5. Foster I, Czajkowski K, Ferguson DF, Frey J, Graham S, Maguire . T. Modeling and Managing State in Distributed Systems: The Role of OGSI and WSRF. Proceedings of IEEE. 2005;93(3):604–12.
6. Foster I. Globus Toolkit Version 4: Software for Service-Oriented Systems. J Comp Sci Technol 2006;21(4):523–30.
7. Foster I, Kesselman C. Globus: A Metacomputing Infrastructure Toolkit. Int J High Perform Comput Appl. 1997;11(2):115–28.
8. Langella S, Oster S, Hastings S, Siebenlist F, Kurc T, Saltz J. Dorian: Grid Service Infrastructure for Identity Management and Federation. The 19th IEEE Symposium on Computer-Based Medical Systems, Special Track: Grids for Biomedical Informatics; 2006 June; Salt Lake City, Utah; 2006.
9. Langella S, Oster S, Hastings S, Siebenlist F, Kurc T, Saltz J. Enabling the Provisioning and Management of a Federated Grid Trust Fabric. 6th Annual PKI R&D Workshop, Gaithersburg, MD. 2007 April.
10. Covitz PA, Hartel, F., Schaefer, C., Coronado, S., Fragoso, G., Sahni, H., Gustafson, S., Buetow, K.H. caCORE: A Common Infrastructure for Cancer Informatics. Bioinform 2003;19(18): 2404–12.
11. Gurcan MN, Pan T, Sharma A, Kurc T, Oster S, Langella S, et al. GridImage: A Novel Use of Grid Computing to Support Interactive Human and Computer-Assisted Detection Decision Support. J Dig Imag 2007;20:160–171.
12. Pan TC, Gurcan MN, Langella SA, Oster SW, Hastings SL, Sharma A, et al. GridCAD: Grid-based Computer-aided Detection System. RadioGraphics. 2007;27(3):889–97.
13. Siegel E, Siddiqui K, Pan T, Oster S, Langella S, Saltz J. A Novel Use of GRID Computing to Provide a Real-Time, High Performance Interactive CAD Decision Support Tool. infoRAD presentation and demonstration at Radiology Society of North America (RSNA) 2005, Chicago, IL, Nov 27–Dec 2; 2005.
14. Butler R, Welch V, Engert D, Foster I, Tuecke S, Volmer J, et al. A National-scale Authentication Infrastructure. Computer 2000; 33:60–6.
15. Humphrey M, Thompson MR, Jackson KR. Security for Grids. Proceedings of the IEEE. 2005;93:644–52.
16. Sinnott RO, Chadwick DW, Koetsier J, Otenko O, Watt J, Nguyen TA. Supporting Decentralized, Security Focused Dynamic Virtual Organizations across the Grid. e-Science and Grid Computing; the Second IEEE International Conference on e-Science '06; 2006.
17. Dwoskin J, Basu S, Talwar V, Kumar R, Kitson F, Lee R. Scoping security issues for interactive grids. Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers; 2003.
18. Basney J, Humphrey M, Welch V. The MyProxy Online Credential Repository. Software: Practice and Experience. 2005;35(9): 801–16.
19. Bhatia K, Chandra S, Mueller K. GAMA: Grid Account Management Architecture. The First International Conference on e-Science and Grid Computing (e-Science '05). Melbourne, Australia; 2005.
20. Thomas RK. Team-based access control (TMAC): a primitive for applying role-based access controls in collaborative environments. Proceedings of the Second ACM workshop on Role-based Access Control; 1997. p. 13–9.
21. Bacon J, Moody KN, Yao W. A Model of OASIS Role-Based Access Control and Its Support for Active Security. ACM Transactions on Information and System Security. 2002;5:492–540.
22. Sandhu RS, Coyne, E.J., Feinstein, H.L.,Youman, C.E.,. Role-based Access Control Models. Computer. 1996;29:38–47.
23. Ramakrishnan L, Rehn, H., Alameda, J., Ananthakrishnan, R., Govindaraju, M., Slominski, A., et al. An Authorization Framework for a Grid Based Component Architecture. Proceedings of the 3rd International Workshop on Grid Computing; 2002.
24. Pearlman L, Welch, V, Foster, I, Kesselman, C., Tuecke, S. A Community Authorization Service for Group Collaboration. Proceedings of Third International Workshop on Policies for Distributed Systems and Networks; 2002. p. 50–9.
25. Manchala DW. E-Commerce Trust Metrics and Models. IEEE Internet Computing. 2000;4(2):36–44.
26. Azzedin F, Maheswaran, M.,. Evolving and Managing Trust in Grid Computing Systems. Proceedings of Canadian Conference on Electrical and Computer Engineering; 2002. p. 1424–9.
27. Quillinan TB, Clayton, B.C., Foley, S.N. GridAdmin: Decentralising Grid Administration using Trust Management. Third International Symposium on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Networks; 2004. p. 184–92.
28. Weaver AC, Dwyer, S.J., Snyder, A.M., Van Dyke, J., Hu, J., Chen, X., Mulholland, T., Marshall, A. Federated, Secure Trust Networks for Distributed Healthcare IT Services. Proceedings of IEEE International Conference on Industrial Informatics (INDIN 2003); 2003. p. 162–9.
29. Grandison T, Sloman, M. Trust Management Tools for Internet Applications. The First International Conference on Trust Management; 2003.
30. Ahsant M, Surridge, M., Leonard, T.A., Krishna, A., Mulmo, O. Dynamic Trust Federation in Grids. Proceedings of the 4th International Conference on Trust Management. Pisa, Tuscany, Italy; 2006.