Granular Security for a Science Gateway in Structural Bioinformatics

Sandra Gesing[1*], Richard Grunzke[2*], Ákos Balaskó[3], Georg Birkenheuer[4], Dirk Blunk[5], Sebastian Breuers[5], André Brinkmann[4], Gregor Fels[6], Sonja Herres-Pawlis[7], Peter Kacsuk[3], Miklos Kozlovszky[3], Jens Krüger[6], Lars Packschies[8], Patrick Schäfer[9], Bernd Schuller[10], Johannes Schuster[4], Thomas Steinke[9], Anna Szikszay Fabri[3], Martin Wewior[8], Ralph Müller-Pfefferkorn[2], and Oliver Kohlbacher[1]

[1]Zentrum für Bioinformatik, Eberhard-Karls-Universität Tübingen, Germany.
[2]Zentrum für Informationsdienste und Hochleistungsrechnen, Technische Universität Dresden, Germany.
[3]MTA SZTAKI, Computer and Automation Research Institute, Hungarian Academy of Sciences, Budapest, Hungary.
[4]Paderborn Center for Parallel Computing, Universität Paderborn, Germany.
[5]Department für Chemie, Universität zu Köln, Germany.
[6]Department Chemie, Universität Paderborn, Germany.
[7]Fakultät Chemie, TU Dortmund, Germany.
[8]Regionales Rechenzentrum, Universität zu Köln, Germany.
[9]Konrad-Zuse-Institut für Informationstechnik Berlin, Germany.
[10]Forschungszentrum Jülich, Germany.

ABSTRACT

Structural Bioinformatics is concerned with computational methods for the analysis and modeling of three-dimensional molecular structures. There is a plethora of computational tools available to work with structural data on a large scale. Using these tools on distributed computing infrastructures (DCI), however, is often hampered by a lack of suitable interfaces. The MoSGrid (Molecular Simulation Grid) science gateway provides an intuitive user interface to several widely-used tools in structural bioinformatics. It ensures the confidentiality, integrity and availability of data via a granular security concept which covers all layers of the infrastructure. The concept applies SAML (Security Assertion Markup Language) and allows trust delegation from the user interface layer across the high-level middleware layer and the grid middleware layer down to the HPC facilities. SAML assertions had to be integrated into the MoSGrid infrastructure in several places: the workflow-enabled grid portal WS-PGRADE, the gUSE (grid User Support Environment) DCI services, and the cloud file system XtreemFS. The security infrastructure presented here allows single sign-on and thus lowers the hurdle for users to utilize large HPC infrastructures for structural bioinformatics.

Contact: sandra.gesing@uni-tuebingen.de,
richard.grunzke@tu-dresden.de

*to whom correspondence should be addressed
*these authors contributed equally

1 INTRODUCTION

Structural bioinformatics and computational chemistry have become indispensable tools in many fields of biomedical research. Molecular dynamics methods, quantum chemical methods, and protein-ligand docking provide deep insights into the structure of biomolecules and their interactions and are thus essential tools in such diverse areas as materials science and drug design. While very powerful, most of the tools and applications used for computational chemistry calculations reflect the complexity of the underlying scientific theories. Using these tools thus requires a lot of experience. Their usability is seriously lacking and thus frequently deters novice users.

The computational complexity of these theories make the according tools ideal candidates for high-performance computing infrastructures [1]. However, this has become one of the biggest challenges for quite a number of scientists, since powerful compute resources may not be easily usable for everyone. Here, DCIs come into play.

These issues, complexity of theory and tools as well as limited access to high performance infrastructures, have been in focus when the MoSGrid (Molecular Simulation Grid) project was conceived. It is part of the German Grid Initiative (D-Grid) and is designed to address the requirements of both commercial and academic users.

MoSGrid offers a science gateway for the computational chemistry community, providing easy access to tools from the field of quantum chemistry, molecular dynamics, and docking. Currently, the MoSGrid community consists of about 110 users or working groups, respectively. At this stage, the science gateway is opened

for about 15 users from academia and industry whose feedback and demands are invaluable for the further development. It is planned to offer the science gateway to the whole community in the near future. Novice and advanced users are enabled to run their sequences of work on grid resources. They are assisted by graphical user interfaces with different levels of sophistication to accommodate both user groups. Additionally, standard methods for specific problem classes are provided. MoSGrid provides a framework for developing, storing and providing simple and complex workflows. Furthermore, users are enabled to collect and process results of calculations and more generally are provided with molecular structures in databases.

Having left the first prototypic state, developments in MoSGrid continue to focus on the security requirements of the different communities. Distributed computing infrastructures are accessible by a number of users from different locations at the same time. The broad user community has to be provided with an infrastructure that protects their know how and molecular data by efficiently securing it.

The MoSGrid science gateway lowers the barrier of utilizing HPC infrastructures and allows access to UNICORE [2] infrastructures utilizing a single sign-on concept which applies SAML. This paper describes the recent developments in the MoSGrid security infrastructure. Especially considering both the demands of academic and commercial users, the paper focuses on the integration and interoperability of the employed components with respect to user authentication and authorization and data security.

The remainder of the paper is structured as follows. Section 2 introduces the background with the application domain and related work. The developments for the MoSGrid security infrastructure are presented in Section 3 and Section 4 demonstrates domain specific workflows utilizing the security infrastructure.

2 BACKGROUND

Some of the application cases of structural bioinformatics and computational chemistry, in particular applications in pharmaceutical industry, impose strict requirements on data security in order to protect potential intellectual property. We will discuss these issues briefly and then examine how a good level of security can be obtained while still providing a convenient single sign-on access.

2.1 Application Domain

Structural bioinformatics deals with the prediction and analysis of the structure, and the mechanisms of function of biological macromolecules [3], including proteins, nucleic acids [4], lipids and sugars. Some major issues handled by this field are e. g., the improvement of drug targeting [5], the derivation of enzymatic design principles, or the development of computational models that describe structure function relations. Knowledge is gained by both, experimentally derived structures as well as computational models. Regarding the computational methods, two fields have emerged among others: (i) quantum chemical calculations (QM) dealing with the electronic structure of molecules and (ii) molecular dynamics (MD) employing classical mechanics approaches. Since the target of the investigations are macromolecules and the processes of interest can consume a considerable time scale, a large amount of data will be generated in the course of these calculations [6].

Sensitive Data in Research and Science Both in an academic and in an industrial context the most valuable goods being produced by structural bioinformatics is data. This data has to be stored reliably in order to avoid data loss, but also securely in order to avoid unauthorized access to sensitive and valuable information. Keeping that in mind, it is essential that the scientist has full control over the access policies to all of his simulation data. With respect to a collaborative work strategy, the option to share selected data with co-workers is also an essential feature. One has to differentiate what kind of data should be shared. The pure simulation data, such as intermediate molecular structures, raw trajectories, and unanalyzed energies is usually only of interest for closest collaborators. In contrast, access has to be granted to a broader community if the knowledge is published.

Within an academic environment to publish is a prerequisite before analyzed and approved data is shared with third parties. In collaboration with industry partners the focus shifts to other priorities. Publications are out of question before a patent application is filed. In both areas a highly secure exchange of data including robust encryption and authentication techniques is immanent.

Another crucial requirement is a high degree of data persistence, i.e. protection from loss or inadvertent change of data. In regard to this goal several requirements have to be met.

The security demands in an industrial context comprise a multitude of details; (i) the data shall be transferred with robust encryption, (ii) the data shall not be visible or modifiable by third parties, and (iii) the jobs and even their existence shall not be transparent for third parties.

In academic environments, the demands are different due to the more open and collaborative approach to work. This distributed approach generates different challenges. (i) A great degree of transparency in terms of versions and changes for all contributors is desired because the project data is handled like a ”living” document. (ii) When a project is highly distributed, simultaneous access to data can cause problems with naming schemes and versions as well as concurrency issues. (iii) During a long-term project, a mass of preliminary data is produced which cannot be stored forever. Hence, criteria for secure long-time archiving of data and also reliable erasing of data have to be evaluated.

2.2 Related Work

Security is a key aspect for a science gateways [7] on top of DCIs. Currently, the established basis for authentication in grid middlewares (e.g., UNICORE, Globus Toolkit, gLite) are X.509 certificates. The basic security concept includes offering *single sign-on* to users. It is a principle for access control in connected systems. The user has to authenticate himself just once and gains access to all connected systems without the need for further authentication procedures. Another main advantage is that the user does not have to maintain several means of authentication, meaning no multiple passwords for multiple systems or several certificates are required.

Single sign-on relies on the principle of *trust delegation* with which systems can be allowed to act on behalf of the user. It is used, for instance, in workflow systems, where a whole workflow

consists of multiple jobs. Using trust delegation, a workflow engine acting in the name of the user, submits the individual jobs to suitable resources without further user interaction. This approach decouples job submission and user interaction.

To support single sign-on and thus trust delegation UNICORE 6 uses the approach of *explicit trust delegation* (ETD) [8] in its dynamic style [9]. It allows the dynamic creation of jobs in the name of the user, though the trust relationships are still static. ETD advanced to its dynamic style offers increased flexibility while maintaining robust security properties. The trust delegations assertions are encoded in *SAML 2.0*. It can contain several statements specifying the assertion in more detail. It also can be chained, meaning that an entity acting on the user's behalf can delegate trust to yet another entity, which is then also able to act on the user's behalf. SAML trust delegation assertions offer important security characteristics. They can be limited to one entity, to a specific validity time span, and to a trust chain of a maximum length. Furthermore, SAML is already supported by various single sign-on infrastructures (e.g., Shibboleth), which allow mapping of local accounts to federated identities.

Other grid middlewares like Globus Toolkit or gLite implement trust delegation via GSI (Grid Security Infrastructure) proxy certificates. GSI is a specification for secure communication in a grid environment and is based on public key cryptography using certification authorities (CAs) and X.509 certificates. These proxy certificates have several disadvantages compared to trust delegation based on SAML. The proxy certificate is always transfered along with its private key which is extremely sensitive since anyone, who possesses it, can impersonate the user. To mitigate this problem, the validity span is often severely limited which creates new problems. Furthermore, it is impossible to reconstruct each step of a trust chain build with proxy certificates. To lessen the problem of short validity time spans users can upload their certificate to MyProxy [10] servers and periodically generate proxy certificates valid for a certain duration of time. A MyProxy server also lessens certain security risks, because the private keys do not have to be stored on every machine used. However, it also creates news risks, because the central servers have to be very well secured. Also it does not improve the security of GSI proxy certificates by itself.

Both approaches for trust delegation introduced above are based on X.509 certificates, which demand that users go through a multistage application process to receive their user certificates. Additionally, they have to create essential files from their certificates for the trust delegation. These procedures are time-consuming and may discourage users to utilize DCIs. Therefore, several approaches are on the way to simplify the application process or to automatically generate the essential credential files.

The Java library GridCertLib [11] supports users of web-based science gateways by automatically obtaining X.509 certificates and using proxy certificates. The prerequisite is that the science gateway has access to a SAML assertion of a previous successful Shibboleth authentication. This library could be adapted for the use of SAML assertions and employed in the MoSGrid science gateway in case the D-Grid infrastructure will be extended for offering federated identities based on Shibboleth.

A similar concept has been implemented by the UK project SARoNGS [12]. However, the generation of a MyProxy certificate in the portal still needs the interaction of the users and a web service which demands Shibboleth authentications. This mechanism is analogously used in WS-PGRADE and therefore the MoSGrid science gateway. In both solutions the users are provided with an intuitive user interface to create their credentials without the need to use any command line invocations for generating credentials.

The GENIUS portal supports the concept of X.509-based robot certificates [13]. These are not associated with specific users but with communities, applications or science gateways. The certificates are handed over to the users on smart cards, which demands card readers connected to the users' computers. Users are authenticated via login and password in the GENIUS portal and are allowed afterward to use DCIs via the smart card. This solution has two major drawbacks. First, the need for additional hardware on the users' side. Second, the duplicated additional effort for already implemented processes in grid security infrastructures, like mapping user distinguished names (DN) to local accounts on HPC facilities.

The EU project EGI (European Grid Infrastructure) [14] presented in October 2010 the result of a questionnaire about requirements for authentication and authorization infrastructures for DCIs, which was answered by a number of projects from different domains, e.g., biomedicine. One result was that the key technologies include SAML and X.509 certificates and that the goal is to bridge security domains by using for example Shibboleth. Since the MoSGrid science gateway already uses SAML, its security infrastructure can be easily adapted to rely on Shibboleth for user authentication instead of certificates.

3 THE MOSGRID SECURITY INFRASTRUCTURE

The MoSGrid security infrastructure consists of four layers: the science gateway as intuitive user interface, the high-level middleware service layer including gUSE [15] (grid User Support Environment) and XtreemFS [16], the grid middleware layer with UNICORE and suitable HPC facilities in the D-Grid infrastructure (see Fig. 1).

In general, a science gateway can be defined as a single point of entry to a set of tools for a specific application domain operating across organizational boundaries. We characterize a grid portal as a web-based science gateway utilizing grid infrastructures and demanding solely a web browser on the user's side. The workflow-enabled grid portal WS-PGRADE [7, 17] (Web Services Parallel Grid Runtime and Developer Environment) is the basis for the MoSGrid science gateway. The chosen WS-PGRADE version employs the open source portal framework Liferay [18], which supports the JSR168 [19] standard and its successor JSR286 [20]. Additionally, WS-PGRADE is the highly flexible graphical user interface for gUSE. The latter provides a large set of services for the management of workflows in DCIs.

XtreemFS is an object-based file system which supports distribution of data up to a world-wide scale and allows simple access on local machines. Furthermore, the data availability is increased and the latency and network consumption reduced using its replica management.

As a fully developed grid middleware, UNICORE is deployed and used in a variety of settings. It consists of a full software stack including clients, a gateway, system services, and components for access to the actual computing or data resources. The latest version is UNICORE 6, which is based on Web Services and particularly the Web Service Resource Framework (WSRF) [21].

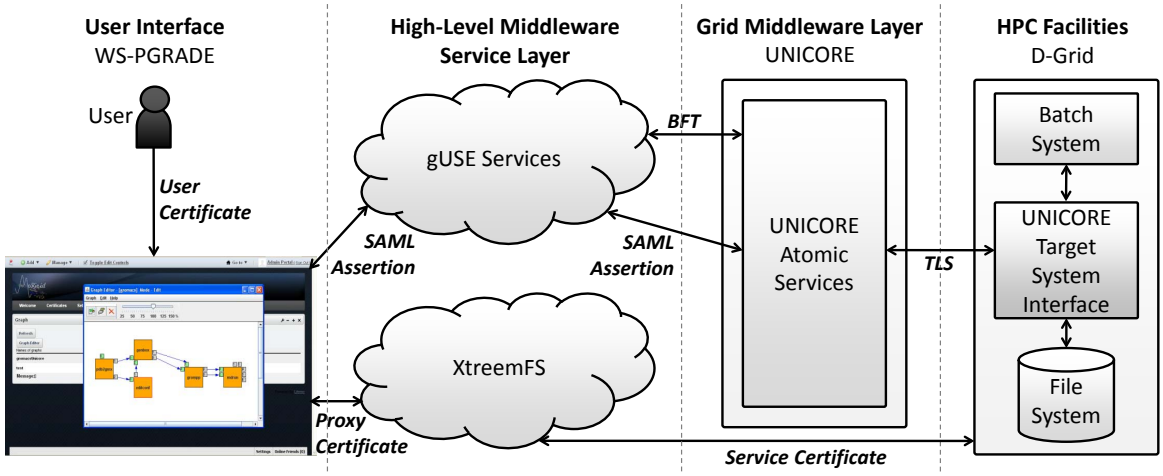

Fig. 1. The MoSGrid Security Infrastructure

MoSGrid employs security features of Liferay, extended WS-PGRADE and gUSE, and is extending XtreemFS for the use of UNICORE and SAML assertions. These extensions affect three major domains of security in DCIs: user and credential management, workflow and job management, and distributed data management. These are described in the following sections in detail.

3.1 User and Credential Management

Liferay consists of a portlet container with default applications and a portal interface which is deployed inside an application server. MoSGrid has chosen Apache Tomcat 6 [22] as the underlying application server.

Apache Tomcat handles the access control of users and programs to resources and the integrity of data during transfers via HTTP or HTTPS, respectively. Furthermore, the application server offers role-based authorization modules and supports the login with user name and password. Liferay facilitates these modules and extends the role-based authorization with more granular security mechanisms in the user management by providing organization, community, and group management. Organizations may present various divisions or various locations of a company and offer private and public accessible pages of the portal. In contrast, communities are designed for allowing access across organizational boundaries or to pages which are applicable for all users of a portal.

To meet the needs of the computational chemistry community, the organization and the community management is utilized in the MoSGrid science gateway. Hence, we implemented four main roles via Liferay: guests, novice users, advanced users, and administrators.

Guests are characterized by lacking an account for the science gateway. However, they can obtain information about the project and about essential steps for getting access to the MoSGrid science gateway and its features. Liferay offers the option that an account can be created by an unknown user. As soon as users have a login created for the MoSGrid science gateway, they can apply for the MoSGrid community membership via email and their accounts will be assigned to the MoSGrid user or MoSGrid advanced user role.

Novice users, in terms of being novice to structural bioinformatics tools, are classified as MoSGrid users. This role enables to choose pre-defined workflows to become acquainted to the tools and domain specific workflows. The latter are offered via intuitive graphical user interfaces which lowers the barrier for utilizing the tools as well as using them on high-performance computing facilities. The novice users are allowed to change input and parameters, to invoke and monitor workflows. The access rights are implemented as a community role for MoSGrid users.

The access to additional features for creating and changing workflows and for configuring settings for grid infrastructures is granted via the MoSGrid *advanced user* role.

Finally, the *administrators* are additionally enabled to manage all credentials, users, organizations, and communities.

The presented user management implements solely the access to the MoSGrid features in WS-PGRADE (e.g., creation of workflows) but not the access to the underlying grid infrastructures. The latter's essential file on the users' side is a security token generated via the certificate portlet of WS-PGRADE.

3.1.1 Managing Security Token with a Certificate Portlet Since UNICORE allows access to underlying HPC facilities based on X.509 certificates, every user who wants to utilize UNICORE infrastructures has to obtain an X.509 user certificate from an appropriate certificate authority (CA).

To protect the user's certificate it is fundamental to absolutely minimize its necessary transfers in the authentication process and the locations where it has to be stored.

WS-PGRADE achieves this goal by offering a certificate portlet for credential management without the need to upload personal certificates to the portal server. The original version of the certificate portlet solely supported proxy certificate based authentication via MyProxy servers. A new certificate portlet was created which provides features for diverse credential formats including SAML. In the near future, when XtreemFS fully supports SAML assertions, the use of the less secure proxy certificates will be discarded in the MoSGrid science gateway.

Applets ensure that processed data remains on the user's computer and signed applets additionally use policy files to ensure the integrity of the processed data. Therefore, a signed applet has been integrated into the certificate portlet for generating SAML assertion files locally on the user's computer. For the generation of a SAML assertion, the user only has to fill in the location of his certificate on his computer, the corresponding password and the location on his computer where the generated assertion file should be stored (see Fig. 2). The applet then automatically generates an assertion file with the same validity as the user's certificate.

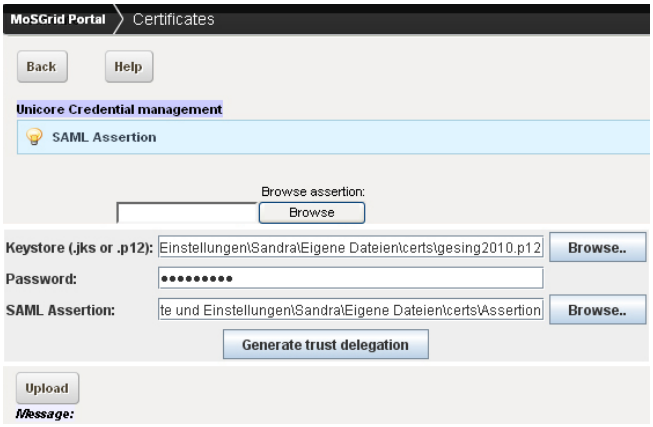

Fig. 2. Trust delegation generation with integrated certificate portlet.

Furthermore, the extended certificate portlet is adapted to simplify its use in MoSGrid. Users do not have to distinguish between diverse options but are still enabled to use all relevant options regarding a SAML assertion file and its management, e.g., generating, uploading, and deleting the assertion file. The uploaded SAML assertion file sets or will set the stage for the authentication processes in UNICORE, XtreemFS, and the domain specific portlets.

3.2 Workflow and Job Management

The collaborative, community-oriented application development environment of WS-PGRADE offers a graphical workflow editor and enables the users to create, change, invoke, and monitor workflows. The latter may contain jobs on local resources and distributed resources in grid and cloud infrastructures. Existing workflows, workflow graphs, workflow templates, and sophisticated workflow applications can be shared via a local repository.

WS-PGRADE allows to configure intuitively settings for various grid middlewares and corresponding resources. In the case of UNICORE 6, MoSGrid advanced users are enabled to add UNICORE registries which provide access to a number of infrastructures. MoSGrid users are enabled to choose the preferred UNICORE registry out of a list of configured registries. However, the whole integration process of UNICORE in WS-PGRADE additionally demanded the development of a so-called submitter plug-in in gUSE.

gUSE provides a set of services for the management of workflows in DCIs including the data-driven workflow engine and submitters. Jobs within the same workflow may be configured for diverse DCIs and the workflow engine invokes each with an appropriate submitter. In general, gUSE submitters are Java-based applications developed to provide authentication mechanisms and the management of single jobs for a specific DCI. They implement the interface GridService of the workflow engine with methods for the management of jobs including authentication, authorization, and data-staging.

gUSE offers various submitters for grid and cloud infrastructures, desktop grids, and web services. In MoSGrid we have additionally developed the submitter for UNICORE 6 [23]. The submitter utilizes the UCC (UNICORE commandline client) libraries, implements authentication with SAML assertions, and manages data-staging utilizing the secured BFT (Basic File Transfer) protocol of UNICORE.

To authenticate a user with SAML assertions against a UNICORE infrastructure, the submitter requires access to three files: the SAML assertion file created via the certificate portlet, the X.509 certificate to which the trust delegation is issued by the user, and a truststore which includes the public keys of the CAs used in the UNICORE infrastructure. The first file is unique for each user, the second and the third are the same for all users of the MoSGrid science gateway.

As soon as a user uploads his SAML assertion file via the certificate portlet to the portal server, the submitter is able to access the file. The public key of the MoSGrid science gateway is utilized by the certificate portlet to create the SAML assertion file. An administrator of the science gateway ensures that the X.509 certificate used for the trust delegation as well as the truststore is available for the submitter. Accordingly, the submitter uses these essential files to authenticate the user against the selected UNICORE middleware installation, which then checks whether the credentials are valid and authorizes the user or returns an error.

Once a user is authenticated, the submitter creates a job on the targeted UNICORE resource. As a result, UNICORE automatically provides a job working directory on a HPC facility (USpace) which is solely accessible for the user who invoked the job. Currently, the submitter utilizes the BFT protocol for uploading or downloading all files belonging to a job to or from the USpace. This mechanism will be extended in the near future to apply the cloud file system XtreemFS for specified input and output files.

3.2.1 Application Specific Module gUSE provides a sophisticated web-based way to create, configure, and execute grid applications on various types of DCIs. However, there is a demand to let the portal developers use features and functionalities of gUSE from portlets' codes. The developers can focus on creating domain specific portlets that are tailored especially for the applications and for the users' needs. The authentication on grid and cloud infrastructures and the submission and monitoring of workflows is handled by services of gUSE. Therefore, a new component is developed called ASM (Application Specific Module) that can be used as an API (Application Programming Interface).

Applications consist of workflows and corresponding parameters, input files and output files. Every application included in the local repository of gUSE can be reused via a portlet using the ASM libraries. ASM provides various interfaces for the management and contains functions to be able to manage the whole execution lifecycle.

The java functions can be called from portlets which themselves can use any technology and visualization methods suitable to the applications' needs, independently from the underlying solution. The security mechanisms rely on the implemented submitters

in gUSE. Hence, portlets developed for the MoSGrid science gateway can utilize the submitter for UNICORE via the configured applications and are unaffected in case modules in the security infrastructure are changed.

3.3 Distributed Data Management

XtreemFS was chosen as distributed file system for MoSGrid to safeguard data and provide each resource with secured access. It is an object-based file system which stores file data and metadata on different services. The object storage devices (OSDs) manage the physical files and the metadata and replica catalogs (MRCs) contain the directory tree and metadata such as the filename, DN of the owner and file size. Moreover, the MRC authenticates users based on GSI and authorizes access to files based on the X.509 user certificate's DN entry. The features for authorization and authentication based on certificates allow to easily integrate XtreemFS into existing services namely UNICORE, WS-PGRADE, and the D-Grid infrastructure. Currently, XtreemFS and its components support GSI proxy certificates for authentication while SAML support is being developed.

Users are enabled to access, upload, and download data to and from XtreemFS via a portlet deployed in WS-PGRADE. As soon as the portlet is initialized, XtreemFS is mounted using a proxy certificate issued by a MyProxy server.

3.3.1 Integration of XtreemFS in UNICORE To make an XtreemFS volume available in UNICORE, the latter manages the transfers of data between an XtreemFS volume and HPC facilities. UNICORE uses the FUSE [24] client of XtreemFS for this purpose. The client translates file system calls to requests to the corresponding MRC and OSD. The client as well as the UNICORE Target System Interface (TSI) shall be installed on every login node of participating HPC facilities. The TSI is the UNICORE component which forms the interface between the UNICORE grid middleware and the HPC facility, e.g., it manages the communication with the batch system of the HPC facility and handles data transfers via TLS (Transport Layer Security) [25] connections.

The XtreemFS client will mount the MoSGrid volume using the XtreemFS X.509 service certificate, which identifies a services instead of a person, and a file based on extended UNICORE User Database (XUUDB) information. It contains the mapping between the user DN and a login on a HPC facility. Using this information the local logins of the users are mapped to their corresponding DNs. Afterwards, the DNs are passed to XtreemFS for authorization and access to the users' files, which are identified by a DN. The DNs are thus the basis for the access rights on the HPC facilities.

The MRC regards XtreemFS clients using a service certificate as a trusted system component, meaning that the MRC will accept any DN sent by the client. Using the TSI, the mounted MoSGrid XtreemFS volume will be integrated into UNICORE and thus made available in the UNICORE middleware.

This way of integration offers important advantages. First of all, the integration is transparent in regard to XtreemFS. Independent of the available storage resources, XtreemFS provides one global namespace. Furthermore, XtreemFS as an efficient distributed data management system handles the transfer of data between the science gateway and the HPC facilities.

The UNICORE middleware will only take care of transferring the data from the mounted XtreemFS volume on the HPC facility to the USpace of the simulation job on the same machine. This way, UNICORE is avoided for extensive data transfers over long distances as it is less efficient in this regard. The simulation jobs are enabled to directly access the MoSGrid volume via UNICORE that provides a technically mature and proven way for this feature.

4 DOMAIN SPECIFIC WORKFLOWS

Beneath the supply of the WS-PGRADE based workflow oriented instruments to use grid resources, MoSGrid aims to provide novice users intuitive means to run chemical simulations. To serve this purpose, the chemical simulation codes, workflows, and IT infrastructures are hidden. The user accesses portlets that directly offer instruments to start and manage simulations for different subjects of structural bioinformatics.

Currently, MoSGrid offers specific portlets for molecular dynamics and quantum chemistry and conceives a portlet for docking. The connections of the portlets are established to the UNICORE grid middleware directly and the portlets use predefined certificates. In the near future, the portlets will be ported to utilize the newly introduced ASM library and with it the gUSE services like the UNICORE submitter. This enables the developers to focus on the domain related features to further improve the user experience. The design and functionality of the domain specific portlets are described in the following.

4.1 The Molecular Dynamics Portlet

The Molecular Dynamics (MD) portlet enables chemists to easily access molecular simulation codes in the area of molecular dynamics. Frequently used workflows are predefined and available for different *recipes*. On the one hand, the portal should ease the work of experienced users and lower the hurdle for novice users on the other hand. The scientists can submit molecular simulations without knowledge of the underling DCI. The MD portlet is organized in three main sections.

Connection In the connect widget users can connect to the underlying DCI and see how many HPC computing facilities can be accessed with their certificates.

Submission The MD submission widget is designed to provide a molecular dynamics service on multiple levels. It allows the user an easy use of standard chemical recipes. In the current state the user is enabled to submit a single simulation using a directly uploaded job description. Alternatively, the user can run a complex recipe that includes an energy minimization and a following equilibration. This recipe is an indispensable prerequisite for all kinds of production runs.

The user has, for both cases, to upload a file, containing either the job description (Gromacs TPR-Format) or the structural information (PDB-Format). In the background the portlet automatically checks the job description for correctness. Some unnecessary input information is automatically filtered. Other erroneous information, like missing residues, is detected and shown to the user in the portal. In the next development stage the MD portlet will detect topological features of the input structure ,e.g., if the protein is a monomer or a multimer and adapt the simulation to the different input files.

However, the portlet minimizes the necessary user input as far as possible but still needs some user input. First, it is hard to guess how long a chemical process should be simulated [26]. Therefore, the user has to define the simulation length in picoseconds. Secondly, the user has to define the resources for the simulation. This includes the number of parallel nodes and the maximum duration of the simulation (wall time). When all information is given and checked, the user can submit the job to the MoSGrid infrastructure.

Monitoring Finally, the user can monitor the job process. The jobs are named after the user login on the portlet, combined with submit time, and name of the workflow recipe. A traffic light for each simulation entry shows the status of the simulation. Further information, e.g., about the underlying HPC facility which the job utilizes, is hidden.

For each simulation the user can query the output files, even in an ongoing simulation. Files can be downloaded or displayed in the portal. The MD portlet shows either plain text, picture, or figures, and in case a molecule file is selected, a 3D view in Jmol (see Fig. 3).

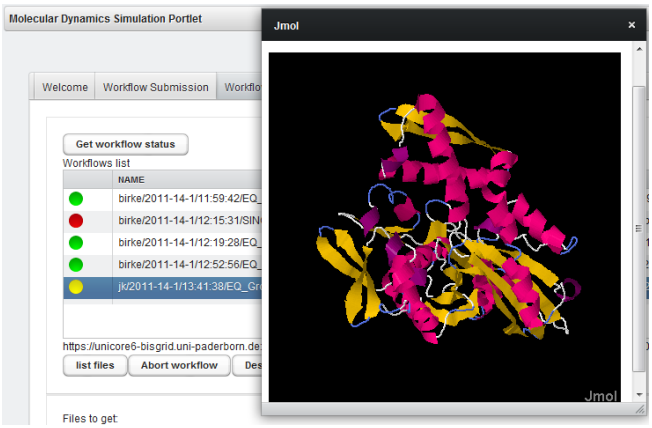

Fig. 3. MD Portlet - Monitoring and view of a molecule file in Jmol.

The next step of the development of the MD portlet is the incorporation of more simulation codes, additionally to the currently supported Gromacs [27].

4.2 The Quantum Chemistry Portlet

The Quantum Chemistry (MD) portlet is a fully functional prototype which implements a complete quantum chemical workflow. The platform enables both experienced and inexperienced researchers to submit their molecular simulations, monitor the progress, and retrieve the results. Moreover, pre- and post-processing routines are available. Among others, these can be used to extract the output of the simulation tools and format it in a standardized way.

On the start screen the user has three options to select from. The first two represent the two implemented workflows, the third provides access to a monitoring facility.

Graphical job creation is supported in the first workflow. The extensible interface provides the most common options to create molecular simulations. Using familiar user interface components, both less experienced and advanced users can configure and submit simulation jobs.

The interface is divided in different tabs, which group different functions and settings (see Fig. 4). This includes the job type specification (e. g. optimization, energy minimization), the selection of the simulation method, technical parameters of the resources to be used and additional options. After specifying the geometry, the job can be submitted for calculation.

Direct submission of an existing job file is provided as second option on the start screen. Users are enabled to directly upload and submit pre-generated job descriptions in Gaussian job file format [28]. They may parametrize the job with the specific requirements like maximum run time, number of processors to use, or memory requirements. This option is intended for advanced users. These users are accustomed to certain tools which generate the output or want to modify the job descriptions directly to achieve maximum control over the simulations and reuse existing job descriptions.

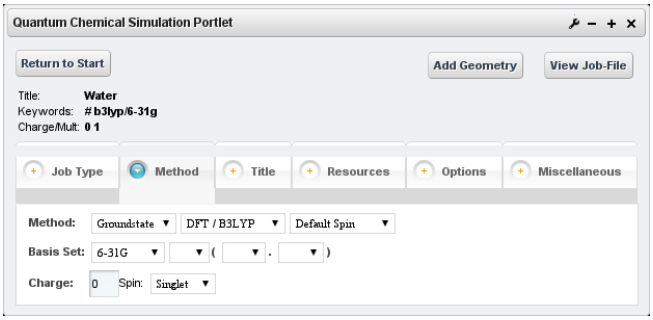

Fig. 4. QC Portlet - Graphical job creation.

Monitoring. No matter which method was used for creating the simulation job, the monitoring facilities can be used to acquire an overview of the currently active jobs. The status is represented by the well known items *queued*, *running*, *successful*, or *failed*. Furthermore, in case of a successful execution, the exit code of the tool is provided as well as the information that the data is available.

For successfully finished jobs, the workflow produces different results. Besides the native output format of the simulation tool, specific values and the development of these values are plotted to files, which can be viewed, downloaded, and processed in common spreadsheet applications.

5 SUMMARY AND OUTLOOK

We presented the security infrastructure of the MoSGrid science gateway offering single sign-on to HPC facilities via SAML assertions. Users are enabled to intuitively create SAML assertions and are provided with domain specific workflows and portlets. Furthermore, WS-PGRADE offers the ASM API which allows developers to focus on domain specific workflows and portlets without the need to become acquainted to the security infrastructure in detail. On the high-level middleware service layer, gUSE was and the cloud file system XtreemFS is being extended for the use of SAML assertions.

Our next steps regarding the security infrastructure will enhance the usability of the authentication mechanism. Therefore, we will utilize the user certificate embedded in the browser. This embedded user certificate will be employed for two purposes: first, for an automatic login of the user based on membership in the MoSGrid virtual organization, second, for an automatic creation of SAML

assertions. The latter step will eliminate the user interaction for creating a SAML assertion and for choosing a certificate from the local hard drive. Since personal certificates expire annually, information about how to renew the certificate will be presented when this case occurs. Together with the previous mentioned measures, this will further aid the user in smoothly using the MoSGrid science gateway.

ACKNOWLEDGEMENT

We would like to thank Valentina Huber for the basic version of the applet for generating SAML assertions.

Funding: This work is supported by the German Ministry of Education and Research under project grant #01IG09006 (MoSGrid) and by the European Commission's 7th Framework Programme under grant agreement #RI-261556 (EDGI), #RI-261323 (EGI-InSPIRE), #261585 (SHIWA), and #RI-283481 (SCI-BUS).

REFERENCES

[1]O. Niehörster, G. Birkenheuer, A. Brinkmann, B. Elsässer, D. Blunk, S. Herres-Pawlis, J. Krüger, J. Niehörster, L. Packschies, and G. Fels. Providing Scientific Software as a Service in Consideration of Service Level Agreements. In *Proceedings of the Cracow Grid Workshop (CGW)*. 2009.

[2]A. Streit, P. Bala, A. Beck-Ratzka, K. Benedyczak, S. Bergmann, R. Breu, J. M. Daivandy, B. Demuth, A. Eifer, A. Giesler, B. Hagemeier, V. Huber S. Holl, N. Lamla, D. Mallmann, A. S. Memon, M. S. Memon, M. Rambadt, M. Riedel, M. Romberg, B. Schuller, T. Schlauch, A. Schreiber, T. Soddemann, and W. Ziegler. Unicore 6 - Recent and Future Advancements. *JUEL-4319*, February 2010.

[3]N. Chandra, P. Anand, and K. Yeturu. Structural Bioinformatics: Deriving Biological Insights from Protein Structures. *Interdisciplinary Sciences: Computational Life Sciences*, 2(4):347–366, December 2010.

[4]M. A. Jonikas, A. Laederach, and R. B. Altman. *RNA STRUCTURAL BIOINFORMATICS*. Wiley-Liss Inc., 2003.

[5]E. B. Fauman, A. L. Hopkins, and C. R. Groom. *Structural Bioinformatics in Drug Discovery*.

[6]O. Niehörster, A. Brinkmann, G. Fels, J. Krüger, and J. Simon. Enforcing SLAs in Scientific Clouds. In *IEEE International Conference on Cluster Computing 2010 (Cluster)*, 2010.

[7]P. Kacsuk. P-GRADE portal family for grid infrastructures. *Concurrency and Computation: Practice and Experience*, 23(3):235–245, March 2011.

[8]D. Snelling, S. van den Berghe, and V. Li. Explicit Trust Delegation: Security for Dynamic Grids. In *Fujitsu Scientific and Technical Journal*, pages 282–294, 2004.

[9]K. Benedyczak, P. Bała, S. van den Berghe, R. Menday, and B. Schuller. Key Aspects of the UNICORE 6 Security Model. In *Future Generation Computer Systems*, number 27, pages 195–201, 2011.

[10]S. Tuecke, V. Welch, and J. Novotny. An Online Credential Repository for the Grid: MyProxy. In *Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10)*, pages 104–111. IEEE Press, August 2001.

[11]P. Kunszt, S. Maffioletti, R. Murri, and V. Tschopp. GridCertLib: Use Shibboleth to Access the Grid from Web Portals. `http://arxiv.org/PS_cache/arxiv/pdf/1101/1101.4116v1.pdf`, December 2010.

[12]X. D. Wang, M. Jones, J. Jensen, A. Richards, D. Wallom, T. Ma, R. Frank, D. Spence, S. Young, C. Devereux, and N.l Geddes. Shibboleth Access for Resources on the National Grid Service (SARoNGS). In *Fifth International Conference on Information Assurance and Security*, volume 2, pages 338–341, 2009.

[13]R. Barbera, G. Andronico, G. Donvito, A. Falzone, J.J. Keijser, G. La Rocca, L. Milanesi, G. P. Maggi, and S. Vicario. A Grid Portal with Robot Certificates for Bioinformatics Phylogenetic Analyses. *Concurrency and Computation: Practice and Experience*, 23(3):246–255, March 2011.

[14]EGI. European Grid Infrastructure. `http://www.egi.eu/`.

[15]MTA SZTAKI. gUSE. `http://www.guse.hu/`.

[16]F. Hupfeld, T.i Cortes, B. Kolbeck, J. Stender, E. Focht, M. Hess, J. Malo, J. Marti, and E. Cesario. The XtreemFS Architecture - A Case for Object-based File Systems in Grids. *Concurrency and Computation: Practice and Experience*, 20(17):2049–2060, 2008.

[17]Z. Farkas and P. Kacsuk. P-GRADE Portal: a generic workflow system to support user communities. *Future Generation Computer Systems journal*, 27(5):454–465, 2011.

[18]Inc. Liferay. Liferay. `http://www.liferay.com`.

[19]A. Abdelnur and S. Hepper. JSR 168: Portlet Specification. `http://www.jcp.org/en/jsr/detail?id=168`, Oct 2003.

[20]M.S. Nicklous and S. Hepper. JSR 286: Portlet Specification 2.0. `http://www.jcp.org/en/jsr/detail?id=286`, June 2008.

[21]OASIS Web Services Resource Framework (WSRF). `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf`, 2011.

[22]The Apache Software Foundation. Apache Tomcat. `http://tomcat.apache.org/tomcat-6.0-doc/`.

[23]S. Gesing, I. Marton, G. Birkenheuer, B. Schuller, R. Grunzke, J. Krüger, S. Breuers, D. Blunk, G. Fels, L. Packschies, A. Brinkmann, O. Kohlbacher, and M. Kozlovszky. Workflow Interoperability in a Grid Portal for Molecular Simulations. In Roberto Barbera, Giuseppe Andronico, and Giuseppe La Rocca, editors, *Proceedings of the International Workshop on Science Gateways (IWSG10)*, pages 44–48. Consorzio COMETA, 2010.

[24]FUSE. `http://fuse.sourceforge.net`.

[25]T. Dierks and E. Rescorla. TLS. `https://tools.ietf.org/html/rfc5246`, 2008.

[26]J. Krüger and G. Fels. Ion Permeation Simulations by Gromacs – an Example of High Performance Molecular Dynamics. *Concurrency and Computation: Practice and Experience*, 23(3):279–291, 2011.

[27]B. Hess, C. Kutzner, D. van der Spoel, and E. Lindahl. GROMACS 4: Algorithms for Highly Efficient, Load-Balanced, and Scalable Molecular Simulation. *Journal of Chemical Theory and Computation*, 4(3):435–447, 2008.

[28]M. J. Frisch, G.W. Trucks, E. Frisch, et al. *Gaussian 03, Revision E.01*. Gaussian, Inc., Wallingford CT, 2004.