Data Access and Virtualization within ViroLab

Matthias Assel[1], Bettina Krammer[1], and Aenne Loehden[1]

HLRS - High Performance Computing Center of University Stuttgart
Nobelstr. 19, 70569 Stuttgart, Germany
email: {assel,krammer,loehden}@hlrs.de
phone: (+49 711) 685 62515, *fax:* (+49 711) 685 65832

Abstract

This paper describes the general approach developed and already deployed within the EU-funded research project *ViroLab* to manage distributed data resources via a single point of access, the Data Access Services (DAS). We explain how existing concepts such as OGSA-DAI and Shibboleth can be basically applied to designing appropriate services but also how they need to be further enhanced in order to guarantee a certain level of flexibility, reliability, sustainability, and, last but not least, security and trustworthiness. We furthermore present the main functionalities of the different service components and indicate their level of integration with the mentioned technologies. Finally, we list further activities and developments needed to make those operations more robust and scalable for the next generation services of the future Internet.

1 Introduction

One of the major difficulties to overcome in order to improve medical diagnoses and to find the best available treatment for a patient consists in making any relevant information accessible for medical experts in a fast, secure, and easy way, and, especially, in allowing to share updated information as soon as new data become available.

From an eBusiness perspective, the concept of a Virtual Organization (VO) is widely used to approach similar issues, namely to make (data) resources available dynamically, securely, and on-demand. The main purpose of such a concept consists in enabling dynamic, secure collaborations with easy access to different resources, respectively sharing of relevant data/knowledge, tools/services, and workflows [3]. To achieve this, a set of virtualization services that guarantees access to resources in a consistent, resource-independent, and efficient way shall be provided to facilitate a smooth integration of distributed and heterogeneous resources, thus enabling collaborative research and workflow execution.

Within the *ViroLab* [1] project, a virtual laboratory for HIV[1] research and medication support is being developed that allows different experts in this field to share their expertise and results interactively while working together on the same

[1] Human Immunodeficiency Virus

data and information sets, which are widely dispersed over Europe and currently without cross-national or even cross-institutional collaboration [2].

In order to meet the specific requirements for exchanging confidential biomedical information within such a virtual environment [5], the solution introduced in *ViroLab* is built on existing Grid technologies such as Globus Toolkit [6], OGSA-DAI [7], and Shibboleth [13] providing the basis for our own designed services, the Data Access Services (DAS).

In the following sections, we firstly describe the overall approach of data sharing within *ViroLab* and afterwards, the different service capabilities together with their technological challenges and functionalities are presented in more detail. We conclude the paper with an outlook on future developments and activities planned within *ViroLab*.

2 Data Sharing within ViroLab

A challenging task in *ViroLab* deals with the integration of distributed and heterogeneous data resources belonging to different organizations into the overall laboratory environment. This heterogeneity has multiple origins and requires quite some effort to finally come up with an efficient and elegant procedure for biomedical data resource integration. The main problems concerning data heterogeneity are based on the database management system (DBMS) technology and the specification of the data itself. In general, the data available from one particular research field is very similar, but the schemes that are implemented in different institutional databases usually show significant differences. These inconsistencies together with the fact that the data shared within *ViroLab* is sensitive and confidential and thus can be exchanged only in a highly secured way, make this task an important but also difficult and critical endeavor.

The approach chosen to realize this challenge is to develop a middleware system that hides the different data resources and their internals behind a layer of so-called virtualization services. Those services shall offer several functionalities to users and applications in order to interact with widely dispersed data resources as if accessing only one large single database including specific transformation methods in order to achieve a quasi-unification of inhomogeneous data sets. To relieve complex transformation processes, which might be time-consuming and error-prone, to disburden the information exchange among the partners, and also to ease the storage of medical data sets especially in the field of HIV analysis and treatments, the idea of installing the same specific HIV database management system at each data provider site - the RegaDB HIV Data and Analysis Management Environment [8] developed by the Rega Institute of the Katholieke Universiteit Leuven - came up and was suggested to all data providers involved in the project. Following this approach alleviates integrational difficulties and ensures beneficial properties for both data owners and developers. The remainder of this section briefly highlights the benefits and summarizes the overall concept and required steps.

Benefits of the proposed solution:

- External access into the hospitals' security regions is not required.
- Data conversion to the RegaDB schema (also anonymization, extraction) occurs within each hospital.
- Data update within the virtual lab (onto the collaborative RegaDB) can occur regularly.
- Central access for data queries is provided via a single access point.

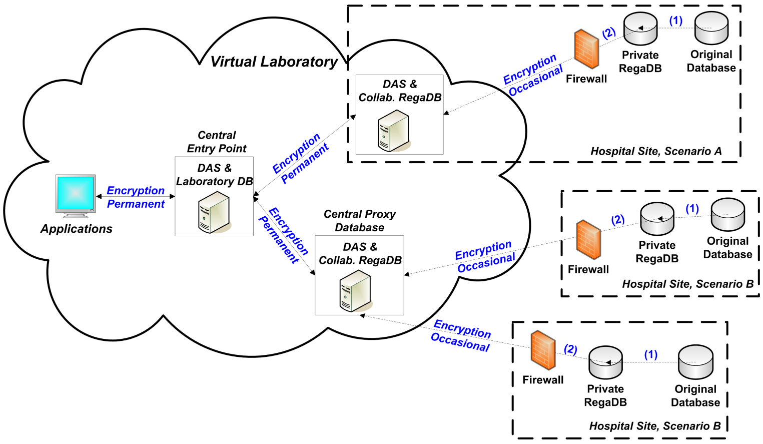

Fig. 1: Data Integration Scenarios

General Approach (refer to figure 1):

At every data provider site, within the hospitals' security regions (behind their firewalls), data from the original database(s) are transferred into a private RegaDB installation. The transfer is done by exporting data from the original database and converting that data extract through a custom script into the latest RegaDB schema. The transfer can be conducted repeatedly over time at the discretion of the database administrator(s).

Data anonymization can occur either while transferring data from an original database into a private RegaDB or alternatively when transferring data from a private RegaDB onto a collaborative one.

To contribute data to the *ViroLab* virtual laboratory, there are two alternative scenarios, both including the upload of data from a private RegaDB into a collaborative RegaDB. The main difference between both solutions is the physical location of the collaborative RegaDBs. Data providers can either host their own collaborative RegaDB installation within a trusted region outside their institutional firewall (the so-called Demilitarized Zone or DMZ), or they utilize one of the "centrally managed" collaborative RegaDBs hosted by some trusted third

parties via a secure connection or simply by sending data extracts to the corresponding site administrator.
Currently, both of the described scenarios are in the scope of *ViroLab*. Since some hospital policies basically prohibit an installation of additional server machines and/or software components within their administered networks, the only way to contribute data to the project's workspace is limited to the second possibility as mentioned above.

3 Data Access Services (DAS)

The Data Access Services are principally designed and implemented as a set of virtualization services ensuring unified data access in a transparent and resource-independent way. In figure 2, the two basic types of access into the virtual laboratory infrastructure - via logging into the working environment through a web portal or via the Data Access Client (DAC) being part of the Experimental Planning Environment (EPE) [9] - together with the four main service modules providing specific functionalities for dealing with several heterogeneous databases concurrently, are shown. These four stand-alone units, each serving a different purpose within the overall services' infrastructure, have been developed independently to guarantee a certain level of flexibility, scalability, and sustainability by following the general approaches of the so-called Service Oriented Architecture (SOA) standards [10]. The data handling components, for example, are forming a separately usable container with many functions for accessing single and/or federated databases by relying on basic security features such as SSL encryption and simple user authentication. Due to working within a collaborative working environment where sharing of resources takes place among multiple institutions, security plays one of the most important and at the same time challenging roles, and needs to be considered from a different perspective than inside a local company network [4]. Therefore, the security handling constituents have been equipped with sophisticated mechanisms allowing a secure sharing of sensitive data and/or information across organizational boundaries but also to become shareable in an easy and efficient way with other DAS parts or third party software applications.
The three following sections will look into the main technological aspects for each of the main service modules including implemented functionalities and their relationships with existing systems, like Globus Toolkit, OGSA-DAI, and Shibboleth [12].

3.1 Data Handling

The data handling subsystem is mainly built up on an existing framework - the OGSA-DAI toolkit - which provides a web service interface structure and can be used within the Globus Toolkit container. This framework offers a nice set of activities in order to deal with several databases via one central access point in a very common way and is basically the engine for all data access related

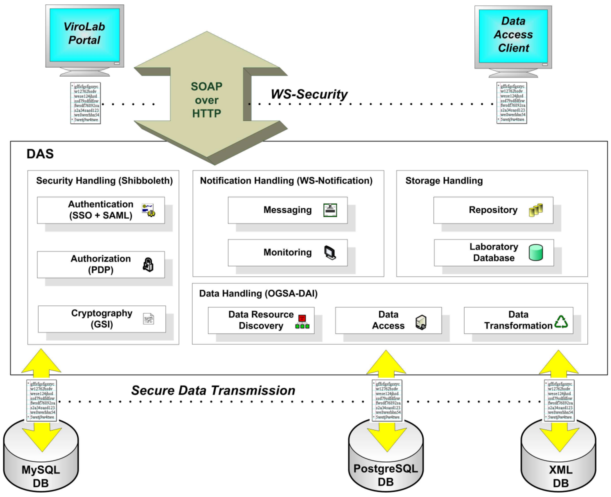

Fig. 2: Higher-level Architecture of Data Access Services

tasks. Unfortunately, some specific functionalities required within a biomedical environment such as *ViroLab* are currently missing in OGSA-DAI but due to its design and implementation OGSA-DAI can be easily extended with own application-specific services. Our data handling unit principally consists of three individually implemented subsystems having the following purpose:

- **Data Resource Discovery:** The discovery service virtualizes the location of data resources and forms one of the basic parts. Applications shall specify data resources in terms of logical names qualified by predicates using a so-called Meta Query Language (MQL) rather than naming exact endpoints. The service maps such terms onto concrete data resource-dependent statements, typically SQL queries for identifying available resources and querying corresponding schema information. Since OGSA-DAI does not take a mapping of a natural language into plain SQL statements into account, the discovery module is more or less developed from scratch but in cooperation with other components of the virtual laboratory like the Provenance System (PROTos) [11], which also requires a higher language to track back the way of certain data sets.
- **Data Access:** The data access infrastructure is the most important part of the virtualization layer. It provides interfaces to access different types of resources including relational and XML databases and allows users to perform different activities on these resources, like querying, updating, and delivering data. Therefor, OGSA-DAI provides various data resource-dependent data handling activities, so-called Data Resource Accessors, to interact with corresponding types of resources. These accessors allow accessing data resources in a well-known way by using standardized SQL commands.

- **Data Transformation:** Dealing with heterogeneous data resources, transformations of queried data sets with respect to the requirements of client applications is one of the challenging tasks. The transformation service utilizes the very basic functions of OGSA-DAI's transformation activities and implements enhanced methods for dynamically applying new transformation schemes with respect to the RegaDB schema in order to change the output format according to current needs of user applications during runtime.

3.2 Security Handling

Security handling particularly in an eHealth scenario has to be considered as an essential and very important part, since distributed data management implies crossing organizational boundaries by sending data over an untrusted network. Therefore, the security module will focus on established security standards like Shibboleth [13] and the Grid Security Infrastructure (GSI) provided by the Globus Toolkit to protect sensible sources of data with sophisticated mechanisms and to keep the privacy of single data sets (patients). Many of the existing principles and techniques can be simply reused or adapted, but they also need to be further extended to ensure the highest-available level of secureness for confidential information. The security handling system is divided into the following three subcomponents, each of them responsible for a certain task within the entire framework.

- **Authentication:** The authentication module is responsible for the identification of a user based on his/her credentials. To provide a very flexible and easily manageable solution, the basic user authentication is performed by Shibboleth's federated Single-SignOn (SSO) and attribute exchange framework, which provides extended functionality allowing the users and their home sites to control the attribute information being released to each service provider. Using this approach simplifies the management of user identities without harming the users' privacy. The DAS authentication module integrates with Shibboleth's Service Provider features, thus enabling a verification of user identities by proofing the authenticity at the corresponding home organization.
- **Authorization:** The authorization interface decides whether a user is allowed to perform a certain action by mapping user attributes onto OGSA-DAI's data handling activities and/or connected resources. It requests the public set of attributes by taking the current user's identity token, which is based on the Security Assertion Markup Language (SAML) specification and contains the current user handle (unique id) and the corresponding home site address. Having received the set of attributes, the service contacts a so-called Policy Decision Point (PDP) in order to evaluate against pre-defined policies whether the user can execute activities or queries.
- **Cryptography:** The service provides capabilities for decrypting incoming and encrypting outgoing messages to ensure secure transmission between different endpoints. Further associated mechanisms for ensuring privacy

and integrity of delivered messages shall also be available via the cryptographic interface. Usually, the underlying grid middleware already contains such mechanisms innately so that there will be a close cooperation between both, the cryptographic module and the Globus Toolkit.

3.3 Storage and Notification Handling

Both storage as well as notification services are additional components and are also developed completely independently. Currently, all actions are monitored using standard logging concepts as provided by Globus, OGSA-DAI, and Shibboleth by default. There is no support for any eventing or messaging between involved components at the moment.

However, the storage system is closely associated with setting up and accessing some databases. The main purpose having such a kind of a central storage system is that intermediate as well as experiment results and general data sets need to be recorded separately from local resources. *ViroLab* stores these kinds of data within a central laboratory database (long-time storage of results) and a temporary repository (storage of intermediate data from running applications/jobs) using the capabilities of the DAS.

4 Conclusions

Developing sustainable, partially autonomous, and in some way intelligent software services, lots of today's middleware systems need to be enhanced and also completely separately developed technologies need to be combined into more powerful solutions in order to roughly fulfill application/user-specific requirements on a certain level of robustness, scalability, and, especially, trustworthiness.

Having presented services enabling secure access to sensitive, distributed, and heterogeneous data resources, the services' main functionality is basically considered stable and already in usage. Nevertheless, DAS has got some limitations and yet unimplemented features.

Performing currently a more or less static user authorization by granting either full access or no access to all service methods and resources respectively, one major feature of the next release will be the integration of a dynamic authorization model based on a PDP and user-defined policies (usually defined and managed by data providers themselves) that controls the access to different service activities but also data resources. To increase the performance and reliability of complex and long-time running user queries, the next generation services will submit the single data queries in parallel instead of processing them sequentially. A higher-level query language, a so-called Meta Query Language (MQL), shall be defined and introduced that allows application users as well as developers to query data from different resources without typing concrete SQL statements but rather well-known terms. Finally, to support the advantages of automatic notification, the services shall be connected with an additional subsystem managing

the automatic delivery of events of interest to particular components and/or users. This may lead to first self-manageable mechanisms helping the service providers and the users in becoming more convinced with future software services.
With all these extensions, DAS can be a seen as a kind of prototype integrating different established technologies into one sophisticated system to provide a robust and flexible solution applicable also outside the project's scope and certainly interesting for other communities as well.

Acknowledgements. The results presented in this paper are partially funded by the European Commission under contract IST-027446 through the project *ViroLab*. The authors want to thank all members of the consortium who contributed to this paper.

References

1. ViroLab. EU IST Project (IST-027446). *http://www.virolab.org*.
2. Sloot P., Boucher C., Bubak M., Hoekstra A., Plaszczak P., Posthumus A., van de Vijver D., Wesner S. and Tirado-Ramos A. *VIROLAB - A Virtual Laboratory for Decision Support in Viral Diseases Treatment*. Proceedings of the Cracow Grid Workshop 2005, Cracow, Poland, November 2005.
3. Schubert L., Wesner S., Dimitrakos T. *Secure and Dynamic Virtual Organizations for Business*. Paul Cunningham & Miriam Cunningham, ed., Innovation and the Knowledge Economy: Issues, Applications, Case Studies, IOS Press Amsterdam, 2005, pp. 1201 - 1208.
4. Assel M., Kipp A. *A Secure Infrastructure for Dynamic Collaborative Working Environments*. Proceedings of the 2007 International Conference on Grid Computing and Applications (GCA'07), Las Vegas, USA, June 2007.
5. Assel M., Krammer B., Loehden A. *Management and Access of Biomedical Data in a Grid Environment*. Proceedings of the 6th Cracow Grid Workshop 2006, Cracow, Poland, October 2006.
6. The Globus Toolkit. *http://www.globus.org/toolkit/*.
7. The OGSA-DAI Project. *http://www.ogsadai.org.uk*.
8. The RegaDB Framework. *http://www.rega.kuleuven.be/cev/regadb/*.
9. Wlodzimierz F., Pegiel P. *GScript Editor as a Part of the ViroLab Presentation Layer*. Proceedings of the 6th Cracow Grid Workshop 2006, Cracow, Poland, October 2006.
10. M. C. Brown. *Build grid applications based on SOA*. 2005, Available on http://www-128.ibm.com/developerworks/grid/library/gr-soa/.
11. Balis B., Bubak M. and Wach J. *Provenance Tracking in the Virolab Virtual Laboratory*. In Proc. PPAM2007 Conference, Lecture Notes in Computer Science. Gdansk, Poland, September 2007.
12. Foster, I., Kesselman, C., Nick, J. and Tuecke, S. *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration*. Globus Project, 2002, Available at http://www.globus.org/research/papers/ogsa.pdf.
13. The Shibboleth Project. *http://shibboleth.internet2.edu*.