GRID-BASED SECURE WEB SERVICE FRAMEWORK FOR BIOINFORMATICS

Dawei Sun, Xiaoyu Zhang

Department of Computer Science
California State University San Marcos
San Marcos, CA 92069
Email: xiaoyu@csusm.edu

Although the web-based bioinformatics is very popular after dozens of years' growth, biologists found out that it is inconvenient because they need to access many web sites manually in order to perform a single task. Web-service based bioinformatics was proposed to provide well-defined interfaces accessible to programs. However, security for web services is a very important issue that was not addressed in most web-service based bioinformatics systems. We developed a Grid-based Secure Web Service Framework for Bioinformatics (GSWSF). The GSWSF is designed based on the Open Grid Service Architecture (OGSA) [6] and Grid Security Infrastructure (GSI) [7], which provide two security mechanisms: transport level security and message level security. We can build secure and easy-to-use bioinformatics services using this framework. This paper covers the architecture and some design and implementation details of the framework. A preliminary implementation of the framework can be found at http://bioinfo.csusm.edu.

1. INTRODUCTION

Nowadays the web based bioinformatics is widely used. In order to perform a task, biologists may need to visit a number of web sites one by one, parse the temporary results, provide input to the next web sites, and check back results frequently because some computations probably last for a long time and people could not wait in front of computers. It is very inconvenient to repeat such manual procedures for large amount of data. While such procedures may be automated by programs or scripts, there are some intrinsic problems. First, the web interfaces are designed to be easily read by human beings but it is hard for programs to grab data directly from web pages. A parser must be written to retrieve useful data from every HTML formatted result returned for a web query. There is not a common and simple way to write a parser to get useful data from web pages. Second, the formats of results are usually not well documented and subject to change. Whenever a format changes, the parser is broken and has to be modified. To resolve the shortcomings of web based bioinformatics, the web service based method was proposed.

Web services are based on XML-formatted SOAP [8] over HTTP. So it is well-formatted to be communicated between computers and easily understood by programs. Furthermore, it can pass through firewalls or gateways easily. Hence, web services are widely used by many legacy and new systems throughout the internet nowadays. There are so many techniques for sharing data among entities in a network. Web service is the only technique which is endorsed by all the big software companies in the world. Web services will not ruin the current web-based database systems. It's just another way to share the information. Using web service, users would be able to aggregate biological data and applications simply and reliably. On the other hand, web service based bioinformatics could also provide the similar human friendly user interfaces.

Security is always an important issue in web applications. As time passes, the security issue becomes imperative in web-based and web service based bioinformatics systems. The purpose of applying web services to bioinformatics is to integrate multiple organizations' services to make them work together smoothly. Thus the web services must be exposed to the public on the Internet. Not all the services would like to let everyone access them without any restrictions especially for functionalities that may change the states of internal databases. So it's very important to construct secure web service based bioinformatics. The basic mechanism is that

administrators manage the web services and can grant some privileges to certain verified users on a set of web services. Only those who have the privileges can access the secure web services. Unauthorized user can only access public services but not the secure ones.

The bioinformatics project at California State University Sam Marcos (CSUSM) is intended to provide the research community the accesses to the latest data and tools on marine coccolithophroids. In order to improve the experience of biologists and enhance the system's security, we developed a grid-based secure web service framework (GSWSF) for bioinformatics and implemented it for CSUSM bioinformatics project. Sample secure web services and user-friendly web clients are developed in the framework to verify the framework works well. We discuss the basic design issues in the next section.

2. GSWSF

The Grid-based Secure Web Service Framework (GSWSF) for Bioinformatics is based on web service architecture. Figure 1 shows its relationship to the application and web service layers.

Figure 1 GSWSF layers.

The framework locates at both the Web Service Layer and the Client Layer. However, it does not cover the Bioinformatics Data and Logic Layer. That means the framework will not add any security functionality in the underlying data and logic, but only to web services. In other words, no changes are necessary to the existing bioinformatics applications.

The Web Service Layer and Bioinformatics Data and Logic Layer are always deployed on local servers. And the Web Service Layer wraps Bioinformatics Data and Logic Layer so that Web Service Layer could be exposed to the public rather than Logic Layer. So the security is handled in the Web Service Layer. The Client Layer is also included in this framework to be aware of security functionalities and to cooperate with secure web services.

The framework is based on the concepts of grid computation. The secure web services based on this framework is implemented based on a grid implementation, e.g. GLOBUS [1]. In the case of web services, "security" commonly means the communication between clients and web services is secure and non-attackable. This "security" feature is provided by the protocol stack technology. The implementation of protocol stack is done by grid implementation. In other words, web service developers and client developers don't need to care much about how to implement the details of protocol stack. What they need to do is just to fill out the arguments of Grid API. Then Grid will guarantee the communication is secure.

The security of Grid is based on PKI. In other words, essentially the secure web services based on this framework need to check a user's credentials to make the decision whether the user is authorized. But it is inconvenient for biologists to manage their credentials manually. So this framework is designed to provide a credential management mechanism such that the users do not need to know anything about credentials themselves. The credentials are issued by a Certificate Authorities (CA), which is thus a component in the framework.

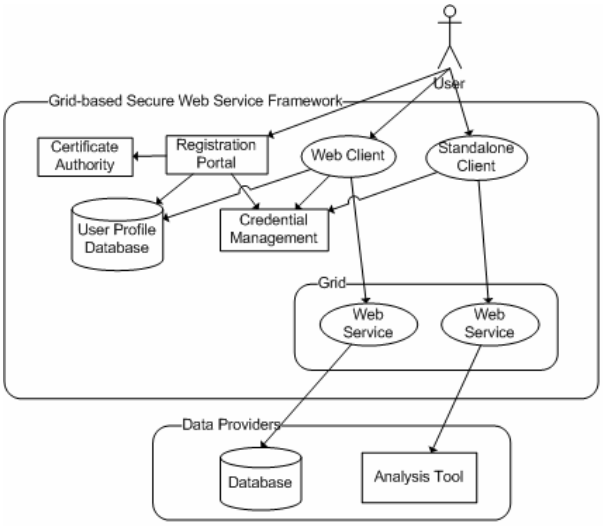

Figure 2. Component Diagram of GSWSF

There are five major modules in Grid-based Secure Web Service Framework for Bioinformatics: Certificate Authority Module, Credential Management Module, Registration Portal Module, Secure Web Service Module, and Client Module. Figure 2 shows an

overall view of these modules and their relationships. We discuss these modules and their implementations in more details in the next a few sections.

2.1. Certificate Authority Module

This module is responsible for credential generation. The responsibilities of the certificate authority module are to generate the credentials and sign them. Normally, the credential generation is done by users. But the framework is designed to hide the complex details of credentials from users. In other words, users do not even need to what are exactly in their credentials. SimpleCA [3] is a free implementation of certificate authority which can be installed anywhere for test or academic research. We choose SimpleCA as the certificate authority module in the framework for its simplicity. A more sophisticated certificate authority can certainly be used instead.

2.2. Credential Management Module

Every registered user must have one credential by which he might access the secure web services. So the safety of these credentials is critical to keep web services secure. Those credentials are usually stored as protected file on disk. However, risk increases when these credentials are distributed and stored physically on multiple disks if users want to access the services from multiple computers. Even if users understood the importance of credentials and kept them secret, it would be impossible for system administrators to manage them.

Instead of letting users store their own credentials, credentials are stored in a credential management module in GSWSF. Users can then access his credentials using a username and password pair that can be remembered in the user's mind. With the credential management module, users could never know that they have credentials. Instead, they only need to remember the username and password pair. The credentials are stored in this credential management module and will be retrieved by clients automatically when credentials are needed. MyProxy [2] is chosen as the implementation of the credential management module in the GSWSF framework.

2.3. Registration Portal Module

For any secure system, the security management is always based on user profiles. So all secure systems must offer the mechanisms of sign-up, login, and logout. In GSWSF, we designed a specific module called registration portal to deal with registration issues. This module depends on the certificate authority module and credential management modules. The registration portal will manage a registration procedure to verify users' identities. After a user is verified, the registration portal module will call certificate authority to generate the user's credential and to sign it, and then will store the signed credential into the credential management module. Also the user's profile is stored in a database called user profile database. In the design of GSWSF, the registration portal is responsible to the management of the registration procedure only, not for login or logout. PURSe [4] is adopted and modified as the implementation for CSUSM bioinformatics project.

2.4. Secure Web Service Module

Secure web service module is the core module of GSWSF. The responsibility of secure web service module is to publish a number of secure web services to the public. The interfaces of web services are some WSDL files. These web services could be thin as a simple wrapper or heavy enough to perform complicated computations. Currently for bioinformatics systems, the most important applications are to wrap the accesses to databases or analysis tools. The secure web service module is based on the GRID implementation. A secure web service module depends on multiple underlying data providers and/or other existing web services. So the web services in this module could be nested and interconnected. Therefore, the service module could be very flexible in practice. In the implementation of GSWSF, We select GLOBUS [1] as the GRID implementation because no one can defeat its dominative position in Grid area. We developed some example secure web services for the CSUSM bioinformatics project. One wraps secure insert, update, and delete functionalities of the EST database and the other supports secure BLAST [5] against databases in CSUSM bioinformatics project, which can be accessed at http://bioinfo.csusm.edu. Other services can be easily added into the framework when needed.

2.5. Client Module

Client module could be the most diverse and complicated module in GSWSF. This module should depend on web service module and credential management module. The responsibility of this module is to provide a user-friendly interface and to fulfill the requirements to invoke the web services. One client could call as many web services as needed. Theoretically, the clients could be developed by any language whose API or library supports networking. In practice, the client is developed in a language in which a stub library can be compiled from the service interface. The GSWSF supports two kinds of clients, web clients and standalone clients. There is a slight difference between them. The web client will verify the user's identity by user profile database; the standalone client will use credential management module to verify user's identity. In the implementation of CSUSM bioinformatics project we developed a web client for the secure web services.

3. CONCLUSION AND DISCUSSION

From a computer science point of view, the web service based of bioinformatics is an evolution of the web-based bioinformatics. Web service based bioinformatics can be convenient for both human and programs' access. Security issue is one of the important requirements of web-service based bioinformatics systems and was not usually considered by most web service based bioinformatics projects. We proposed a grid-based secure web service framework for bioinformatics and applied this framework to develop a implementation for CSUSM bioinformatics project.

GLOBUS Toolkit is selected to construct secure web services, which are then protected by PKI. Digital certificates play the role of keys to access these secure web services. Essentially, a few more protocols, SSL, XML Encryption, XML Signature and WSRF, are applied on network protocol stack comparing with the original non-secure web services. In order to make it convenient for end users to use the secure web services, the framework introduces three components, certificate authority module, credential management module, and registration portal module. Not only do The framework supports web clients but also standalone clients. Thus, the grid-based secure web service framework for bioinformatics fulfills the requirements of web service based bioinformatics. The preliminary implementation of the security framework on CSUSM bioinformatics project attests that the framework is fit for web service based bioinformatics and can provide secure web services very well.

We plan to develop more secure web services for CSUSM bioinformatics project according to GSWSF in the near future. Currently we have several secure web service and a few users in the preliminary implementation. That's fine for the administrator to manage them manually. As time passes, we'll have more web services and users. Each user could have more or less privileges for these web services. That will be a big matrix of users and services. So a good management module must be designed for administration in the future. Also we need to develop a good client for the upcoming secure web services to provide a human-friendly interface for biologists to link multiple services interactively. Probably this client needs to integrate more web services to construct a more powerful application dependent on the requirements of the biologists.

Acknowledgments

The authors thank Dr. B. Read, Dr. T. Wahlund, Dr. A. Hadaegh, and Mr. M. Ranji at CSUSM for useful discussions and assistances.

References:

1. The Globus Alliance and Toolkit, http://www.globus.org/
2. MyProxy Credential Management Service, http://grid.ncsa.uiuc.edu/myproxy/
3. SimpleCA, http://www.vpnc.org/SimpleCA/
4. Portal-Based User Registration Service (PURSe), http://www.grids-center.org/solutions/purse/
5. The Basic Local Alignment Search Tool (BLAST), http://www.ncbi.nlm.nih.gov/BLAST/
6. Open Grid Services Architecture, http://www.globus.org/ogsa/
7. Grid Security Infrastructure http://www.globus.org/security/overview.html
8. Simple Object Access Protocol http://www.w3.org/TR/soap/