Grid Computing

G00719

Mark Anderson

Security Report

Explore the developments in the implementation of security mechanisms for Grid computing. The particular emphasis for this study should be a critical analysis of the development of the security aspects of the Globus Toolkit from versions 2 to 4. You should pay particular attention to the alignment of the security mechanism with the move to a web-service oriented architecture. You should also consider any weaknesses in the current architecture and any possible future developments which you consider to be crucial to the successful adoption of Grid Computing.

Aisha Ijaz

10056967

Contents Page

1.0 Introduction

This report will explore the recent developments in the implementation of security mechanisms in Grid Computing. It will further, critically analyse the security developments of the Globus Toolkit (GT) version 2 to 4 with particular respects to the current move to a Web-Service (WS) orientated architecture. The report will further consider the weakness of GT4 security related issues and explore possible future developments that would be crucial for the successful adoption of grid.

2.0 Background

Initially, it is important to understand the concept of grid computing which "***refers to systems and applications that integrate and manage resources and services distributed across multiple control domains"*** [Welch et al; no date]. The rapid emergence of Grid has led to GT middleware developments that support the Grid Computing environment. GT is a software toolkit developed by The Globus Alliance (GA) which enables users to create grid systems via a low-level Application Programming Interface (API) so that grid application developers can build higher level clients [Anjomshoaa; 2002; 3]. It is the Grid Security Infrastructure (GSI) within GT that "***provides the fundamental security services needed to support Grids***" and provides mutually authenticated, integrity-checked encrypted channel of communication and further offers single sign-on support for users of Grid [TGST; 2005; 1]

Since Grid Computing entails crossing organisational boundaries, resources will be accessed by a many different multi-institutional "***Virtual Organizations***" (VOs) on a dynamic ad-hoc basis [Welch et al; no date; pg 1]. VOs encompass '***groups of individuals, associated resources and services which are united by a common purpose but not located within a single administrative domain***' [Welch et al; no date; pg 1]. The set of resources used by a single computation maybe large, dynamic and unpredictable and the resources itself can be valuable [The Globus Project; 2002]. Interactions for these resources may not only be client-server but also service-to-service. It is important to find an easy to use yet sensitive security environment for VOs which further incorporates Sotomayors [2006] three pillar of secure

communication (see section 3.0). The need to support integrated, interoperable heterogeneous systems and effectively 'manage resources' within VOs poses many challenges with regards to security issues especially with the recent move towards a more WS orientated architecture which bring along additional security threats. [Sotomayor et al; 2006; 271]

3.0 Secure communication and the Three Challenges

A secure communication is one that prevents "***unauthorised disclosure or modification of data***" by malicious users and/or enemies to ensure continued operation of the system [Foster et al; 1999; 395]. According to Sotomayor [2006; 257] all three common pillars to secure communication should be present, these are:

- Privacy – the sender and receiver are the only parties that understand the conversation via encryption/decryption techniques if eavesdropping occurs.
- Integrity – the receiver should know for sure that the original message sent was the initial message received and was not tampered or manipulated.
- Authentication – the need to verify the identity of a participant to an operation or request so as to prevent impersonation from malicious parties.

Authorisation is also an important concept in grid security and can be considered as an additional pillar of secure communication. Subsequent to authentication one would need to decide when the user is authorised to perform a certain task [Sotomayor; 2006; 259]. Thus, authentication establishes identity and authorisation establishes rights [The Globus Project; 2002]. With these concepts in mind, many technologies have been developed to ensure secure security mechanisms[1] within GT. It becomes a difficult task when different VOs require different combinations of these features for example, the implementation of privacy and integrity only.

According to Natgaratnum et al (2002) these security challenges within a Grid Computing environment can by grouped into three categories and should be considered by the Globus Alliance (GA) when exploring secure security mechanisms. (See Fig 1)

[1] Cryptography , Authentication and/or Certificates and Certificate Authorities (CA)

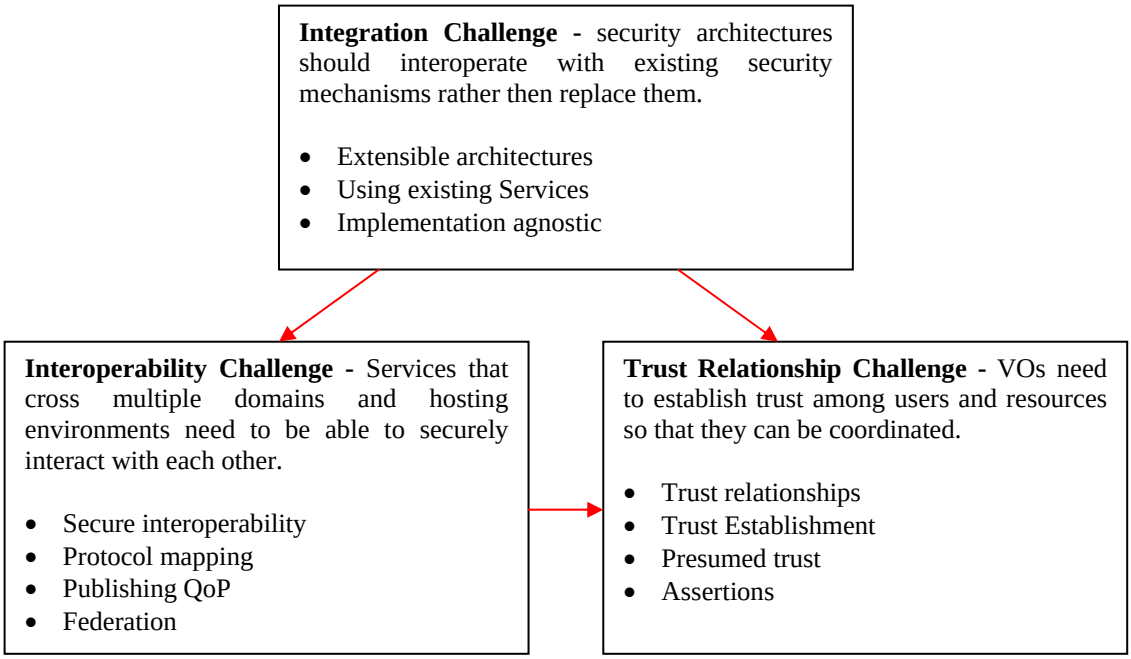


Fig. 1 Categories of Security Challenges in a Grid Computing Environment [Natgaratnum; 2002]

4.0 Security aspects of Globus Toolkit Version 2 to 4

Implementation of GSI[2] within the GT software tackles issues of local heterogeneity and this in conjunction with Generic Security Service API[3] (GSS-API) defines a standard procedure for obtaining credentials which enables vital security needs to be addressed in Grid Computing. GT software incorporates Sotomayors' [2006] common pillars for example, mutual authentication, message integrity, delegation and message confidentiality to form the essence of secure communication [Foster; 1999; 269].

4.1 Globus Tookit Version 2 – Pre-WS Components

For the scope of this project the Security Module of GT2 will only be critically analysed but an overall schematic diagram of GT2 can be viewed (see Appendix A) but the vital features of GT2 GSI are highlighted in Fig.2.

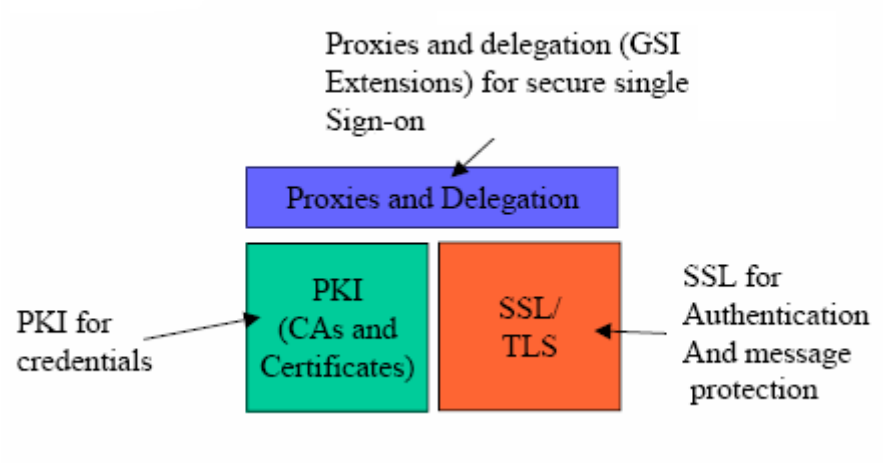


Fig. 2 Grid Security Infrastructure [Schopf; 2003]

The development of the Public Key Infrastructure (PKI) certificate and key aids in establishing authenticated and encrypted communication channels on the 'Secure Socket Layer' (SSL) and/or 'Transport Layer Security' (TLS)[4] [Agarwal; no date]. SSL/TLS is an Internet Engineering Task Force (IETF) standard that defines a cryptographic handshake protocol used for authentication via X.509 identity certificates in order to provide secured communication for all information exchanged between the client and the server. It is the implementation of these security layers that ensure authentication, message protection and message integrity during subsequent

[2] Global Grid Forum proposed GSI as part of a open source Globus project to deal with security issues
[3] Application Programming Interface
[4] The SSL/TLS have minor difference but are substantially the same [Wikipeida; 2006]

data streams via encryption algorithms and laid the necessary foundations for security mechanisms for Grid Computing [Dick; 510].

On the other hand, SSL implementation provides you with no real assurance that you are really talking to your intended company and is still vulnerable with regards to 'man in the middle attacks'[5]. [Swanberg; 2002] Furthermore, performance during the connection establishment phase may be an issue when considering SSL because of potential bottlenecks caused by public key operations [Hurley; 2002]. Alternatively Foster [1999] states that once a connection is established the performance of the conventional cryptosystems used is less of a factor.

Though, SSL/TLS ensures server side authentication it is the PKI developments that ensures mutual client-server authentication and establishes message integrity, user authentication and confidentiality before exchanging any vulnerable information. The sender may digitially sign messages using a private key, and the recepient can check the signature using the associated public key contained in the user's certificate issued by a Certificate Authority (CA) within the PKI [Schopf; 2003]. This aids in verifying to the recipient that the sender is who he claims to be since the digital signature has been verified by the CA and overcomes any sort of 'man in the middle attacks' [Wikipedia; 2006]. Conversely, PKI relies on trust and the users must protect the uniqueness of the private key [Hurley; 2002].

SSL was extended to further incorporate self-signed X.509 proxy certificates for single-sign on and delegation which encompasses similar functionality to that of Kerberos[6] [The Globus Project; 2003]. The developments of the X.509 proxy certificate allows the user to dynamically assign a new X.509 identity to an entity and then delegate some subset of their rights to that identity allowing new credentials and identities to be created quickly. Thus, proxy certificates can represent the user in all authentication and authorisation processes without the need for additional sign-on and delegate on behalf of the user [Welch et al; no date]. Despite, its strength proxy certificates also have their weaknesses for example, 'limited lifetime' of proxy

[5] For example, eavesdropping, tempering and/or message forgery
[6] Computer network authentication protocol, were the Kerberos tickets can be seen as equivalent to the GSI certificates to ensure authentication but the basis of both protocols differ for example, key based algorithms used and trust models.

certificates exists to secure communication but time is wasted in generating proxy certificates each time it expires. [Lock; 2002]

Simple trust relationships can be adapted using proxy certificates and GT2 using Community Authorisation Service (CAS). This allows VOs to express policy that has been outsourced to it via resource providers but was ineffective when it came to complex trust domains. [Welch et al; no date]

As a result, GT2 implemented fundamental security mechanisms for systems and applications to integrate and manage resources and services distributed across multiple control domains securely. However, it had been difficult to develop and extend GT2 since there were no common framework or procedures in place and it was dependable on the server. Furthermore, services that accept network connections are prone to attack because they are accessible to the attackers on the network and can result in severe consequences. There were still flaws in relation to Natgaratnum's three challenges[7] for example, CAS not being able to handle complicated trust domains and from this GT had not yet explored and implemented a truly secure Grid Computing environment.

[7] Integration, Interoperability and Trust

4.2 Globus Tookit version 3 – WS Orientated Architecture

The main development implemented in GT3 security mechanisms was bought about via joining grid protocols with Web Services (See Appendix A). "***WS are the technology of choice for Internet-based applications with loosely coupled clients and servers. It makes sense to use a similar structure for grid-based applications"*** [Sotomayor; 2004]**.** Thus, the result of applying a WS orientated architecture bought about the development of 'Grid Services'[8] (GS) which is basically 'WS but with improved characteristics and services' [Sotomayor; 2004]. GS general values improved from WS included, a sophisticated security infrastructure, a standard service invocation mechanism for service lifetime management and state management [Globus Tookit Alliance; 2006].

One of the initial motivations for this move to a WS orientated architecture was the WS standard invocation mechanism which is a foundation for interoperability. GT2 components combined with a WS orientated architecture provided an ease in extending services with standardisation to enable applications and users to operate in a 'seamless and automated manner' within Grid Computing environment [Welch et al; no date]. It is the Open Grid Service Infrastructure (OGSI) in GSI3 which specifies GS. OGSI basically consists of a set of WSDL specifications which help define mechanisms for creating, managing and exchanging information among GS in a stateful way. GT3 and its accompanying GSI3[9] also provide the first implementation in building the foundations for open GS via Open Grid Service Architecture (OGSA). [Tuecke; RamaKrishan]

OGSA is a standard service orientated architecture which consists of "***a set of core capabilities and behaviours that address key concerns in Grid systems***" and offers GS new challenges and opportunities [Foster et al; 2005]. OGSA standardised a service-orientated architecture that assures interoperability which integrates and manages distributed heterogeneous environments to deliver functionality when

[8] Concept bought about in GT3 as Grid Service is an extension to WS
[9] The Java GSI implementation is an implementation of the Java GSS-API. It supports the GSS-API extensions and the new proxy certificate format specifications as defined by the GGF.

interacting with services aligned with industry-accepted WS standards [Grid Forge; 2006].

Emerging WS-Security specifications deal with the idiom of WS-Security Policy[10], the standard formats for security token exchange[11] and standard processes for authentication and establishment of security context and trust relationships[12] [Welch et al; no date]. This specification had been exploited in GT2.4 but was later incorporated within GT3 and in turn incorporates the pillars of secure communication via a WS orientated architecture.

One of the vital developmental goals that GT3 wanted to achieve was in casting security functionality as OGSA services in order to allow them to be located and used by applications when needed. For example, a draft OGSA Security Roadmap [Siebenlist; 2002] presented in 2002 to the Global Grid Forum (GGF) itemised numerous security services some of which include Credential processing service, Authorisation service and Credential Conversation service. These services are well-defined protocols and interfaces in OGSA which permits an application to outsource traditional security functionality using a security service with a particular implementation to fit its current needs. [Welch et al; no date]

Another important goal of GT3 is the standards specified in exchanging security tokens to allow for interoperability. Similarly with GT2, GSI3 supported the formation of a security context that serves to authenticate two parties to each other and allow authorisation for the exchange of secured messages between the two parties. As a result of this, GT3 has made use of SSL and X.509 certificates. [Sandholm; 2003] It is also important to note that the assurance in interoperability offered in GT3 enables resources providers to specify course-grained access control policies in terms of communities[13] as a whole in addition to delegating fine-grained access control policy management to the community itself [The Global Alliance; 2006].

[10] WS-Policy (publishes services security which enables clients to discover dynamically what credentials and mechanisms are needed to establish trust with the service) and XACML (role-based access control enhancing authentication and authorisation)
[11] WS-Security and SAML
[12] WS SecureConversation, WS-Trust
[13] Reference to CAS

GT3 provides both transport- and message-level security. Both are based on GSI and PKI standards as covered in GT2. However, the use of transport-level security[14] is to be discouraged since its support is not guaranteed in future GT3 versions. The Globus Team instead recommended the use of message-level security. The message-level security is based on the WS-Security, XML-Signature, XML–Encryption standards and provides support for credential delegation [The Globus Alliance; 2006]. The GT3 core security infrastructure is based on the Java Authentication and Authorization service (JAAS) framework which allows Java GS to remain independent from the underlying authentication mechanisms [Sandholm; 2003].

As mentioned previously, GT2 implements TLS for both security context establishment and message protection as a stateful form of secured communication. In comparison GT3 implements SecureConversation and WS-Trust which uses WS SOAP message instead of Transmission Control Protocol (TCP) to transport "***context-establishment tokens***" used by GT2 [Welch et al; no date]. Once security context is established, GSI3 further implements message protection and confidentiality of SOAP messages by using WS standards of secured messages (XML-Signature[15] and XML-Encryption). This actual security context is established using GSS-API. Although GSS-API supports multiple security mechanisms only the GSI protocol is currently supported. [Foster; 2005]

Stateless form of secured communication is a vital development in GT3 which enables communication without the initial establishment. GT3 offers the ability to sign messages independent of any established security context via XML Signature specification. Thus, a message can be created and signed to allow the recipient to verify the message's origin and integrity, without the need to establish synchronous communication with the recipient. An important feature of this approach is that the identity of the recipient does not have to be known to the sender when the message is sent. [Welch et al; no date]

[14] via implementing a new "httpg" protocol to indicate a GSI-enables HTTP-based protocol
[15] A shared key is not required for the GSI XML Signature method[15] instead the client simply uses an X.509 certificate to sign the request.

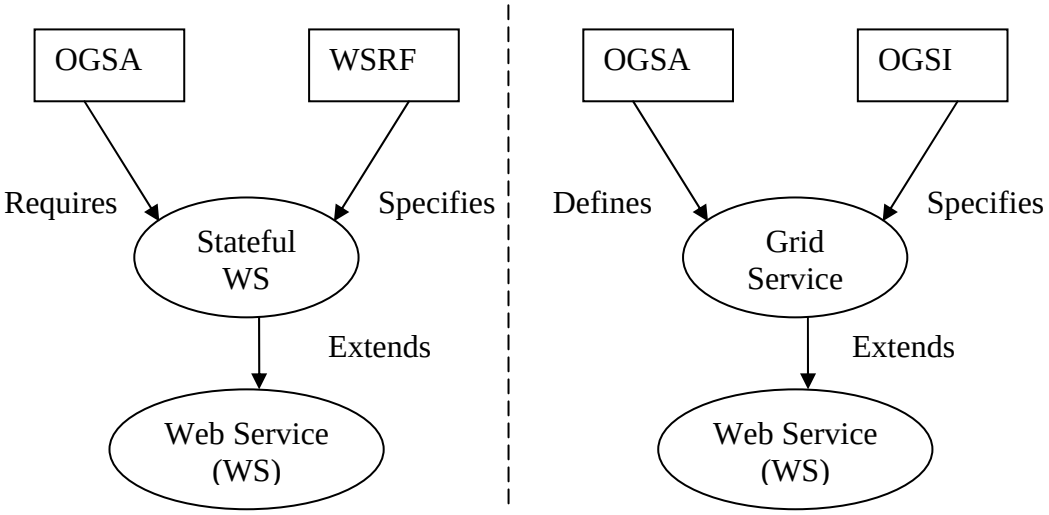
One of the key advantages of GT3 over its GT2 predecessor is the use of WS-Security protocol. The standards in which GT3 uses SOAP and WS-Security specifications for all its communications allows GT to leverage and use current and future WS tools and software [Welch et al; no date].

GT3 had established a framework in implementing security mechanisms via a WS orientated architecture for Grid Computing and has been explored in section 4.2. Much of the major features implemented primarily existed in GT2 but was actually standardised and widely deploy in GT3 (See Appendix). It was not long before GT3 was further extended resulting in GT4 entering the market in 2006 with the same framework as GT3 but improved functionality and performance. However, Harmer [2005] states the importance of having a 'period of stability… where new versions are not coming out since it is difficult to go back to partners and say we are using GT3 but we need to move to GT4'. Further improvements will offer further security when VOs integrate and manage resources and services but this will have an effect to organising trying to be apart of the Grid Computing Environment.

4.3 Globus Toolkit Version 4 – Extended Grid Services

GT4 is a key component for accelerating the adoption of enterprise-class grids. GT4 includes support for Web Services Interoperability Organization (WS-I) Web services standards, including Web Services Resource Framework (WS-RF) and Web Services Notification (WS-N) specifications; Security Markup Language (SAML); Extensible Access Control Markup Language (XACML) [The Global Alliance; 2006].

GT4 developments resulted in OGSI being obsolete and re-factored with the WS-Resource Framework (WSRF) (See Fig.3). Vital reasons for OGSI being made obsolete include too much material in one specification, it didn't work too well with exiting WS tools and it was excessively Object Orientated [Farber; 2006]. The WS group started to integrate their own approaches to capturing state into the WSRF which primarily has the same native interface as GT2.4 in addition to addressing the above OGSI concerns. [Wendler et al 2005] This report will not go into much detail with regards to WSRF but one thing that will be mentioned is that WSRF is a collection of different specifications which all relate to the management of WS resources in order to make WS-Resources stateful. [Sotomayor; 2006]



Much of the underlying WS orientated architecture of GT4 is very similar to GT3 but has the added advantages of user experience improvements, latest WS orientated upgrade, performance improvements plus some new features [Farber; 2006]. Further

security mechanisms were implmented and greatly improved GT4 development cycle. Changes made to WS-Security and HTTPs in turn reduced the message latency[16] of WS environment by 80% and had a major impact on all of the GT4 Web services tools. [Foster; 2005]

[Foster; 2005] GT4 provids secure support for usernames and password which are WS-I Base Security compliant and provides support for transport-level security with X.509 credentials which is the fastest and is set as default but is not well-defined enough to allow claim of compliance [Farber; 2006]. However, the security measures used with GSI have changed The Globus Teams focus to transport-level mechanisms rather than message-level mechanism[17]. This is with a view to improving performance levels when invoking secure services.

According to Harmer et al [2005] security requirements in GT4 are much more stringent and "***probably much more effective***" [Harmer et al; 2005]. However, the term 'probably' can represent a weakness of recent GT4-based grids since this version has not yet been widely deployed resulting in fewer security faults identified on exposed networks. Vital evidence illustrates recent security related issues with GT4 implementation via a recent article "***DOS Attack Bring Down Sun Grid Demo***" [Galli, 2006]. The 'text-to-speech' application was made available for the public without the need of registering in order to demonstrate current grid capabilities but was brought down by an attack. Furthermore, resource security model does not give the attacker any privileges on the local system and allows them to run setuid programs available to the services with constraints in place in addition to DOS acting as the only possible consequence. This development in turn reduces the amount of privileged code allows for easier code reviews and security audits.

Research suggested more stringent security mechanisms within GT4 for example any insecure clients are effectively barred from running services that demand security. Accessing resources without security explicitly specified results in various services that cannot be ran without these secure credentials. [Harmer et al; 2005]

[16] the time required to move a Web service message from the network interface to the service handler and return a response to the network interface
[17] GT4 WS authentication and authorisation component is comprised of two subcomponents Message/Transport-level Security and an Authorization Framework.

Another weakness of GT4 is that security mechanisms are very tedious to set up and not user friendly. Even though a certain level of security is demanded, the documentation and processes that allow a user to set up such security can be difficult for new users [Harmer et al; 2005]. Alternatively, the security demands are more relaxed within GT4 and enables users to gain an understanding of the basic features of the toolkit without struggling with obscure security issues.

Security policies (i.e. specification of grid-map files) are more flexible than was possible with previous toolkit versions with the implementation with new authorisation methods[18]. [The Global Alliance; 2006] In addition to this, grid-mapfiles found in earlier versions of GT which provides access to control based on a list of acceptable user identifiers. GT4 GSI uses the Security Assertion Markup Language (SAML) standard from the Organization for the Advancement of Structured Information Standards (OASIS) for OGSA Authorisation [Siebenlist; 2005a]. SAML purpose is to convey authentication and authorisation information in an XML based architecture. The intention here is to provide a message based architecture for "n" number of authorisations for an authenticated subject. Despites its benefits one of the vital disadvantages of SAML is its immaturity, limited number of vendors and potentially large message payloads. [TGST; 2005]

SAML defines formats for a number of types of security assertions and a protocol for retrieving those assertions. GSI uses SAML AuthorisationDecisions in two ways one being as a means of communicating the rights of CAS clients to services. Secondly, a SAML Callout parameter in which GSI is made available and allows easy and flexible integration with OGSA-authorisation compliant tools, such as PERMIS to allow the use of a third party authorisation decisions service for access control requests to GT4-based services (See Section 5.0). [Siebenlist; 2005b]

Deployed GT4 services have demonstrated good reliability and significant performance improvements on their GT3 equivalents. GT2.4 clients can inter-operate

[18] Such as *identity*, where the client identity subject must match that specified in the WSDD file, and *userName*, where a JAAS login module authorises the user based on the username and password that they supply. Resource level descriptors are also available that allow specification of authorisation of the service resource.

with servers and clients using the pre-WS components of a GT4 deployment. GT3 to GT4 migration has not proven to be a significant effort. The migration of code from GT2 to GT4 is still a largely unexplored area [Farber; 2006].

5.0 Possible future developments

Stability

Recent research indicated that there will be another GT4 standard that will have bug fixes and later version will have extended features [Schopf; 2005]. One of the vital importances noted for the successful adoption of grid is stability for users and VOs within the Grid Computing Environment. With new developments being made available it has made the adoption of grid quite difficult since organisations have to switch from the old version to the new version. Though there is GT2.4 and GT3 migration users would want the latest GT software which offers enhanced security, interoperability and integration on heterogeneous systems when managing resources and services.

Grid Security mechanisms for public use

From the research above there are a number of potential areas that are crucial to the successful adoption of grid computing. The most basic development being that of making security more stringent for public uses for example the article with regards to "***DOS Attack Bring Down Sun Grid Demo***". It is important to note that when making applications within a grid system externally available it is inevitably prone to Denial Of Service (DOS) attacks. This does not degrade the efficiency of the system which has been doing considerably well with its private members. According to Milani [2002] DOS is a common attack but no data is lost or exposed to malicious users instead, it causes an interruption of service hence the term Denial of Service.

PERMIS

It was GT3.3 that initially implemented PERMIS which was later migrated to GT4 functionality. However, according to Welch et al [2005] the implementation with PERMIS should help to provide Role-Based Access Control in VO built on Grids. It is very interesting to further note that The Globus Security Team [2005] stated that Role-Based Authorisation is clearly an emerging direction in grid computing. PERMIS is a security middleware solution that provides an interface to make use of

the standardised GGF authorization API in the Globus Toolkit to protect GS in an effective yet flexible way. "PERMIS and VOMS (Virtual Org. Membership Service) use assertions to bind attributes to users for the purpose of authorisation decision making as opposed the typical identity-based authorisation done today" [TGSL; 2005]. As a result, the use of PERMIS functionality within the GT4 environment is inevitable as the grid computing market moves towards a more role-based access control.

IPv6 and IPsec

GT software development would need to consider its compatibility with IPv6 which will replace existing IPv4. IPsec (IP security) is a standard for securing Internet Protocol (IP) communications by encrypting and/or authenticating all IP packets and will be made compulsory on IPv6 [Wikipedia; 2006]. Porting Globus Toolkit to be IPv6-enabled will bring considerable advantage to Grid Computing for example expanding the address space and enlarges Grid scaling potential and the mobility support could enable more Grid collaboration applications. (UCL; 2001)

6.0 Conclusion

In conclusion, the exploration and developments of security mechanism implemented from GT2 to GT4 have come along way. The security mechanisms defined in GT2 which had no standard means of invocation and no constituent component framework made it difficult to extend this initiative.

With the move towards a WS orientated architecture, GS set a precedent for future GT versions. This provided the standardisation and a standard means of invocation required in order to offer a secure Grid Computing environment. Limited research had made it difficult to examine fundamental flaws within GT3 and the establishment of newly deployed GT4 had been difficult to analyse. One thing that is made known is that GT4 brought along with it improved performance and extended features as mentioned above. As with all GT versions GT4 also suffers from weaknesses in which possible future developments have been stated in section 5.0 which would be crucial for the successful adoption of grid computing.

Bibliography

Books:

Coulouris et al (2005) Distributed Systems: Concepts and Design, Forth Edition, Addison Wesley, London

Dick, D (2002) The P.C. Support Handbook: The Configuration & Systems Guide, Dumbreck Publishing, Kirkintilloch

Foster, I and Kesselman, C (1999) The GRID: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, London

Sotomayor, B (2006) Globus Toolkit 4: Programming Java Services, Morgan Kaufmann, London

Online:

Agarwal et al (no date) Securing Collaborative Environments, dsd.lbl.gov/Collaboratories/Publications/final-security-2002.pdf, [Accessed: 14th April 2006]

Anderson, A et al (2005) Access Control for the Grid: XACML, http://www.globusworld.org/2005Slides/Session%201b(2).pdf, GlobusWORLD 2005, [Accessed: 14th April 2006]

Angelis et al (2004) Mechanisms for controlling access in the global grid environment, V14, N5 pp 347-352, [Accessed: 14th April 2006]

Anjomshoaa et al (2002) Analysis of Globus Toolkit V2.0: Sun Data and Comute Grids, EPCC, www.epcc.ed.ac.uk/sungrid/PUB/gtII-ooa.pdf, [Accessed: 14th April 2006]

Barbir, A (no date) Web Services Security: An Enabler of Semantic Web Services, cs.unb.ca/base**web**/base**web**03/papers/abbie-barbir-Base**Web**2003-paper1.pdf, [Accessed: 14th April 2006]

Farber, D (2006) Globus Toolkit 4.0 http://blogs.zdnet.com/BTL/?p=1326, [Accessed: 14th April 2006]

Cover Pages (2005) Security Assertion Markup Language (SAML) v2.0 Approved as OASIS standard, xml.coverpages.org/ni2005-03-14-a.html, [Accessed: 14th April 2006]

Foster, I (2005) Globus Toolkit Version 4: Software for Service Orientated Systems, www.**globus**.org/alliance/publications/papers/IFIP-2005.pdf, [Accessed: 14th April 2006]

Galli, P (2006) DOS Attack Brings Down Sun Grid Demo, www.eweek.com/article2/0,1759,1941574,00.asp, [Accessed: 14th April 2006]

Gawor et al (2003) GT3 Grid Security Infrastructure Overview, www-unix.globus.org/ogsa/docs/alpha/**GT3SecurityOverview**.pdf, [Accessed: 14th April 2006]

Grid Forge (2006) http://forge.gridforum.org/projects/ogsa-wg, [Accessed: 14th April 2006]

Harmer et al (2005) UK Engineering Task Force Globus Toolkit Version 4 Middleware Evaluation http://www.nesc.ac.uk/technical_papers/UKeS-2005-03.pdf, [Accessed: 14th April 2006]

Hurley, J.S. (2003) Overview of Grid Computing, www.educause.edu/ir/library/pdf/DEC0306.pdf, [Accessed: 14th April 2006]

IBM (2005) Security and privacy services

Kreger, H (2001) Web Services Conceptual Architecture (WSCA 1.0), www-306.ibm.com/software/solutions/**webservices**/pdf/**WSCA**.pdf, [Accessed: 14th April 2006]

Lock, R (2002) Grid Security Requirements, Interactions, Mechanisms and Models, comp.lancs.ac.uk/computing/**.../Security Requirements** for **Grid**s.pdf, [Accessed: 14th April 2006]

Milani, M & Brown, J.S (2002) Some Security Considerations for Service Grids, A preliminary white paper, www.johnhagel.com/**paper_security**grid.pdf, [Accessed: 14th April 2006]

Nagaratnam et al (2002) The Security Architecture for Open Grid Services, IBM, www.cs.virginia.edu/~humphrey/ogsa-sec-wg/OGSA-SecArch-v1-07192002.pdf, [Accessed: 14th April 2006]

Ramakrishan, L (2003) Writing secure grid services using Globus Toolkit 3.0: How to ensure message level security in a grid service, IBM, www-pnp.physics.ox.ac.uk/**.../Grid**/**globus**/dW-GT3-GSI-tutorial.pdf, [Accessed: 14th April 2006]

Sandholm, T & Gawor, J (2003) Globus Toolkit 3 Core – A Grid Service Container Framework, www-unix.**globus**.org/**toolkit**/3.0/ogsa/docs/gt3_**core**.pdf, [Accessed: 14th April 2006]

Sandholm, T (2005) Grid Security: General Considerations and Globus Specifics, Center for Parallel Computers, www.nsc.liu.se/ngssc-**grid**/**security**-s**and**holm.pdf, [Accessed: 14th April 2006]

Schopf, J (2002) Grid Computing and the Globus Toolkit, www.sztaki.hu/~vajda/**Grid** technika/**GridGlobus**(**Schopf**).ppt, [Accessed: 14th April 2006]

Schopf, J (2005) Grid Computing and the Globus Toolkit, www.sztaki.hu/~vajda/**Grid** technika/**GridGlobus**(**Schopf**).ppt, [Accessed: 14th April 2006]

Shirasuna et al (no date) Performance Comparison of Security Mechanisms for Grid Services, www.extreme.indiana.edu/xgws/papers/sec-perf-short.pdf, [Accessed: 14th April 2006]

Siebenlist et al (2002) OGSA Security Roadmap http://www.cs.virginia.edu/~humphrey/ogsa-sec-wg/ogsa-sec-roadmap-v13.pdf, [Accessed: 14th April 2006]

Siebenlist, F & Mori T (2005a) Globus Toolkit Authorization Processing, www.**globus**.org/**toolkit**/presentations/GW05-XACMLand**Globus**-Demo.ppt.pdf, [Accessed: 14th April 2006]

Siebenlist, F & Welch, V (2005b) Grid Security: The Globus Perspective, grid.ncsa.uiuc.edu/ggf12-sec-wkshp/panel5/firewalls-frank.ppt, GlobusWORLD 2005, [Accessed: 14th April 2006]

Sotomayor, B (2003) The Globus Toolkit 3 Programmer's Tutorial, www.cs.buffalo.edu/gridforce/fall2003/gt3**tutorial**.pdf, [Accessed: 14th April 2006]

Sun Microsystems (2003) Securing Web Services – Concepts, Standards, and requirements, White Paper, www.sun.com/software/whitepapers/**webservices**/**securing_webservices**.pdf, [Accessed: 14th April 2006]

Surridge, M (2002) A Rough Guide to Grid Security Issue 1.1a, IT Innovation Centre, esc.dl.ac.uk/ETF/public/Deployment/Level4/node16.html, [Accessed: 14th April 2006]

The Globus Alliance (2006) Components for Grid Security, www.globus.org/alliance/publications/papers.php, [Accessed: 14th April 2006]

The Globus Project (2003) GT3 Overview, www.nsfgrid.marist.edu/docs/introductionGT3.pdf, [Accessed: 14th April 2006]

The Globus Security Team (TGST) (2005) Globus Toolkit Version 4 Security Infrastructure: A Standard Perspective, www.**globus**.org/**toolkit**/docs/4.0/**security**/GT4-GSI-Overview.pdf, [Accessed: 14th April 2006]

UCL (2001) IPv6-enabled Globus Toolkit http://www.6net.org/publications/deliverables/D5.12.pdf#search='Globus%20Toolkit%204%20and%20IPV6', [Accessed: 14th April 2006]

Van, T (2005) Grid Stack; Security debrief – Top distributed computing security experts weigh in on new directions for grid security http://www-128.ibm.com/developerworks/grid/library/gr-gridstack1/?ca=drs-, [Accessed: 14th April 2006]

Welch et al (no date) Security for Grid Services www.globus.org/alliance/publications/papers/GT3-**Security**-HPDC.pdf, [Accessed: 14th April 2006]

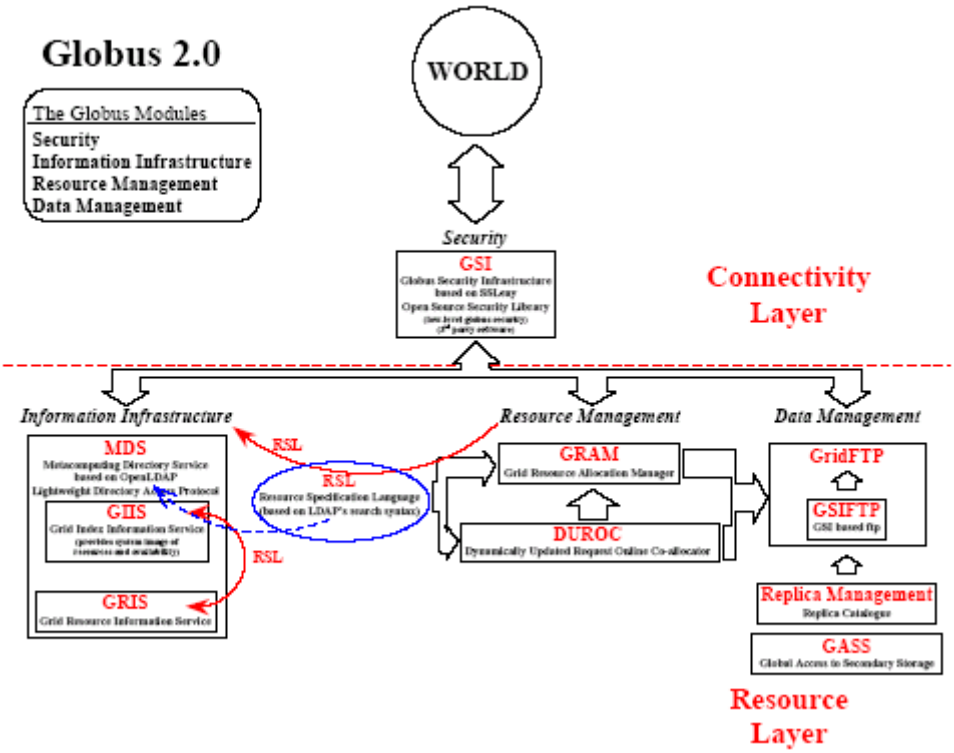
Wendler et al (2001) D3.8 Adoption of Flowserve to OGSA www.unizar.es/flowgrid/download/flowgrid-d38.pdf, [Accessed: 14th April 2006]

Wikipedia (2006) IPSec http://en.wikipedia.org/wiki/PKI, [Accessed: 14th April 2006]

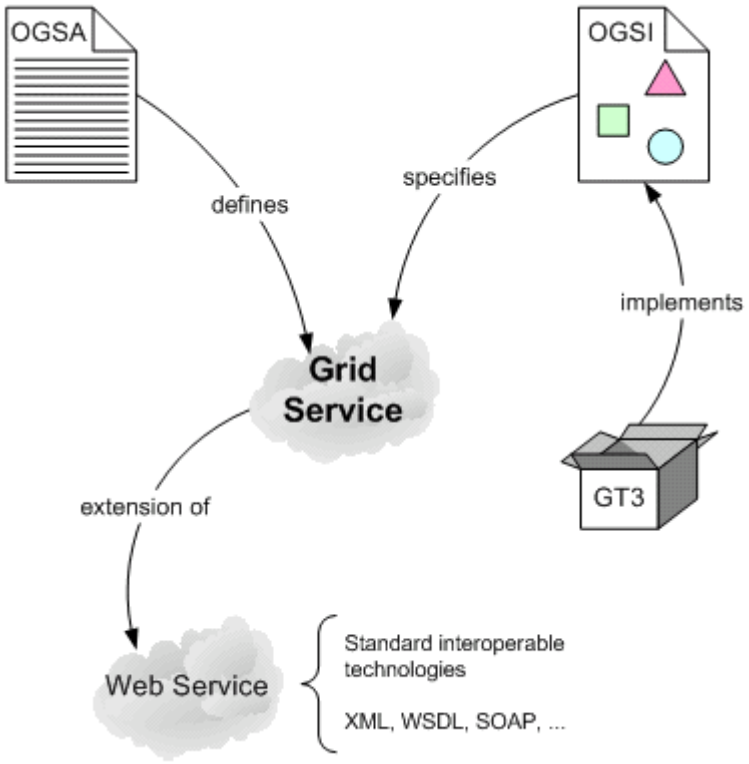
Wikipedia (2006) PKI http://en.wikipedia.org/wiki/Ipsec, [Accessed: 14th April 2006]

Wikipedia (2006) SSL http://en.wikipedia.org/wiki/Secure_Sockets_Layer, [Accessed: 14th April 2006]

Appendix A



GT2 schematic diagram [Anjomshoaa; 2002; 4]



GT3 Architecture [Sotomayor; 2004]

Globus Toolkit 3 improvement [The Globus Alliance; 2006]

The Grid Security Infrastructure (GSI) in the Globus Toolkit version 3 (GT3) represents the latest evolution of the Grid Security Infrastructure. GSI in GT3 builds off of the functionality present in early GT2 toolkit releases - X.509 certificates, TLS/SSL for authentication and message protection, X.509 Proxy Certificates for delegation and single sign-on.

Details of GSI secure can be found in the Security for Grid Services and the GT3 Security Overview papers. Highlighted improvements of GSI3 are:

- *GSI3-secured Web Services*: Access to GT3 services is secured using the GSI3 libraries. This includes GSI3 capabilities for authentication, authorization, delegation, message integrity and encryption.
- *No privileged services*: GT3 represents a redesign of the Globus Toolkit Grid Resource Acquisition and Management (GRAM) service with a strong eye towards the least privilege principle. No services in GT3 need any elevated privileges ("root" access). All privileged code is contained in two small setuid-root programs with tightly constrained functionality.
- *Use of Web Services Security Specifications*: GSI3 has protocols for authentication and message protection using Web Services specifications for securing messages using SOAP (XML-Signature and XML-Encryption) and the emerging WS-SecureConversation specification for context establishing.
- *Standards-based Approach*: GSI3 uses technologies that are defined in either existing or proposed standards in the IETF, GGF, W3C or Oasis. GSI3 will continue to be based on only public standards.
- *Proxy Certificates format.* The GT3 GSI libraries support Proxy Certificates as specified in the latest IETF/Global Grid Forum draft. This includes support for both impersonation and independent proxy certificates and a framework that allows for addition of supporting other delegation policies. The GT3 GSI libraries are also backwards compatible with GT2 proxies, in that they will accept GT2 proxies and treat them as GT3 impersonation proxies.
- *Enhanced client-side authorization*: Services in GT3 have credentials that not only indicate the resource name on which they are running, but the account in which they are running. This allows clients connecting to these services a greater level of assurance that they are interacting with an appropriate service.
- Some things have not changed from GT2 to GT3, for example:
- *GT2 Credential Compatibility*: GT3 uses the same long-term user and host/service credentials as GT2. Existing PKIs and certificates will continue to work in GT3.
- *Resource Authorization*. GT2 used a file known as the grid-mapfile to map Grid identities (the distinguished name from a user's X.509 identity certificate) to a local identity (a Unix account name). A GT3 installation uses the same grid-mapfile as used by a GT2 installation. This will allow GT2-based grids to continue to use their existing infrastructure to manage grid-mapfiles.
- *Application Interfaces.* The GT3 security library is still accessible through the Generic Security Service API (GSSAPI), as defined by RFC 2743 with extensions as defined by the Global Grid Forum GSS-extensions document.

GT3 Grid Security Infrastructure (GSI): Security Features [The Globus Alliance; 2006]

Area	Supported Feature	GT3.0 C Code	GT3.0 Java Code
Proxy Certificates	Authentication with Internet Draft compliant proxy certificates	Yes	Yes
	Authentication with legacy (GT2) proxy certificates	Yes, supported on in GridFTPd	Present, but unsupported
	Delegation of proxy certificates	Yes	Yes
CA Support	CA signing policy	Yes, documentation	No
	Configurable trust roots (CA certificates)	Yes	Yes
Revocation	CRLs	Yes	No
	OCSP	No	No
GSSAPI	GSSAPI	Yes, See RFC 2744	Yes
	GSSAPI extensions	Yes	Yes
	Integrity protection of user data	Yes	Yes
	Encryption of user data	Yes	Yes
Authorization	User authorization	grid-mapfile	grid-mapfile
	Client-side authorization of service using hostname	Yes	Yes
	Client-side authorization of service with GRIM	Yes	Yes

	credentials		
	Client-side authorization of service with wildcard matching of hostnames (e.g. foo matches foo-*: foo-1, foo-ethernet, etc.)	Yes	Yes
	CAS Support	In prototype	No
Kerberos	Relinking with Kerberos instead of PKI	Yes (but kludgy)	In theory as it is part of Java 1.4, but untested.
SOAP	SOAP independent message Signing	Yes	Yes
	SOAP independent message Encryption	Yes	Yes
	Context establishment over SOAP	Yes	Yes

OGSA supports [The Global Alliance; 2006]

The basic OGSA security model must address the following security disciplines:
• ***Authentication***. Provide plug points for multiple authentication mechanisms and the means for conveying the specific mechanism used in any given authentication operation. The authentication mechanism may be a custom authentication mechanism or an industry-standard technology. The authentication plug point must be agnostic to any specific authentication technology.
• ***Delegation***. Provide facilities to allow for delegation of access rights from requestors to services, as well as to allow for delegation policies to be specified. When dealing with delegation of authority from an entity to another, care should be taken so that the authority transferred through delegation is scoped only to the task(s) intended to be performed and within a limited lifetime to minimize the misuse of delegated authority.
• ***Single Logon***. Relieve an entity having successfully completed the act of authentication once from the need to participate in re-authentications upon subsequent accesses to OGSA-managed resources for some reasonable period of time. This must take into account that a request may span security domains and hence should factor in federation between authentication domains and mapping of identities. This requirement is important from two perspectives:
a) It places a secondary requirement on an OGSA-compliant implementation to be able to delegate an entity's rights, subject to policy (e.g., lifespan of credentials, restrictions placed by the entity)
b) If the credential material is delegated to intermediaries, it may be augmented to indicate the identity of the intermediaries, subject to policy.

• ***Credential Lifespan and Renewal***. In many scenarios, a job initiated by a user may take longer than the life span of the user's initially delegated credential. In those cases, the user needs the ability to be notified prior to expiration of the credentials, or the ability to refresh those credentials such that the job can be completed.
• ***Authorization***. Allow for controlling access to OGSA services based on authorization policies (i.e., who can access a service, under what conditions) attached to each service. Also allow for service requestors to specify invocation policies (i.e. who does the client trust to provide the requested service). Authorization should accommodate various access control models and implementation.
• ***Privacy***. Allow both a service requester and a service provider to define and enforce privacy policies, for instance taking into account things like personally identifiable information (PII), purpose of invocation, etc. (Privacy policies may be treated as an aspect of authorization policy addressing privacy semantics such as information usage rather than plain information access.)
• ***Confidentiality***. Protect the confidentiality of the underlying communication (transport) mechanism, and the confidentiality of the messages or documents that flow over the transport mechanism in a OGSA compliant infrastructure. The confidentiality requirement includes point–to–point transport as well as store-and-forward mechanisms.
• ***Message integrity***. Ensure that unauthorized changes made to messages or documents may be detected by the recipient. The use of message or document level integrity checking is determined by policy, which is tied to the offered quality of the service (QoS).
• ***Policy exchange***. Allow service requestors and providers to exchange dynamically security (among other) policy information to establish a negotiated security context between them. Such policy information can contain authentication requirements, supported functionality, constraints, privacy rules etc.
• ***Secure logging***. Provide all services, including security services themselves, with facilities for time-stamping and securely logging any kind of operational information or event in the course of time - securely meaning here reliably and accurately, i.e. so that such collection is neither interruptible nor alterable by adverse agents. Secure logging is the foundation for addressing requirements for notarization, non-repudiation, and auditing.
• ***Assurance***. Provide means to qualify the security assurance level that can be expected of a hosting environment. This can be used to express the protection characteristics of the environment such as virus protection, firewall usage for Internet access, internal VPN usage, etc. Such information can be taken into account when making a decision about which environment to deploy a service in.
• ***Manageability***. Explicitly recognize the need for manageability of security functionality within the OGSA security model. For example, identity management, policy management, key management, and so forth. The need for security management also includes higher-level requirements such as anti-virus protection, intrusion detection and protection, which are requirements in their own rights but are typically provided as part of security management.
• ***Firewall traversal***. A major barrier to dynamic, cross-domain Grid computing today is the existence of firewalls. As noted above, firewalls provide limited value within a dynamic Grid environment. However, it is also the case that firewalls are unlikely to disappear anytime soon. Thus, the OGSA security model must take them into account and provide mechanisms for cleanly traversing them—without compromising local control of firewall policy.
• ***Securing the OGSA infrastructure***. The core Grid service specification (OGSI) presumes a set of basic infrastructure services, such as handleMap, registry, and factory services. The OGSA security model must address the security of these components. In addition, securing lower level components (e.g., DNSSEC) that OGSI relies on would enhance the security of the OGSI environment.

Features that remained unchanged from GT2 [Welch et al; no date]

GT3 uses the same user and service credentials as GT2. PKI and CA credentials supporting GT2 based grids will effectively support GT3 based grids [Gawor; 2003]. While the format of GT3 proxy certificates have changed the user interface for creating and interacting with these proxies remained unchanged [Welch et al; no date].

Authorization in GT3 is based on a simple access control list placed in a flat file called a grid-mapfile which is used to map Grid identities and this has also remained unchanged. However, GT3 does extend this idea by applying access control policies which GT2 based grids can continue to use this existing infrastructure. [RamaKrishan; 2005]

Furthermore, the GT3 security library is still accessible via GSS-API. [Gawor; 2003] GT3 GSI libraries support proxy certificates for both independent and impersonation and a framework that allows for the support of other delegation policies. This is further made backward compatible in that GT2 proxies will be accepted and treated as GT3 impersonation proxies. [Gawor; 2003]