Global Information Assurance Certification Paper

About Grid and Security

GIAC Certification

GIAC Security Essentials Certification Practical

Christian Mathieu

Version 1.4b (Option 1)

December 2003

Abstract:

In the early 80's I read a sci-fi book titled "Web of Angels" [1] by John M Ford. It was the story of a hacker in a future world in which there was a world wide web of computers and services. The hero could log into there at different "levels" according to special rights he might have gained access to. Nowadays the reality is close to the story of this book.

Twenty years ago we were in the beginning of the PC era, we used to have host computers the size of a room, and networks were emerging. In this paper I will shortly retrace the evolution of computing and explain how this happened to be. From a user point of view I will show many problems that have been solved.

New user, Mr Hacker was just hired into the "world company" and his mission is to launch bunches of jobs to calculate large prime numbers. So he would like to submit his jobs on the very best machine of the company, check how the jobs are doing, and get access to some intermediate results. I will use Mr Hacker's experience to show all the main problems the company has solved, mention the Grid vocabulary and associated protocols. I will start by a short, high level view, history of computing which lead to the Grid. Then I will describe what is a Grid and what are the main concepts used by Globus technology, describing architecture, and security features pertaining to this architecture.

Using Mr Hacker's simple task, I will define the different steps to go to the Grid, mentioning the different Grid organisations. I will focus on the security problems that have been solved, and on how to increase global security keeping in mind the variety of this company's organisation.

I will end up by describing work to be done for the new area of Web Services which are the natural mixing of Grid technology and web technology.

History from cloud point of view to the Grid

At the beginning was the first computer, as big as a house, and then came the PC era (first Apple computer in 1977, first IBM PC 1981), where we used "floppy" diskettes to share information. At the same time the first internet experiments began, and we could connect a machine to a local area network (LAN). Then we connected these LANs together, and we started to use internet mail, IRC chat….. , and the first worms that appeared in 1988!

During the same time client-server programs appeared, first on the same machine then on different machines (one such service was Xserver). The client and the server were on different workstations, connected through a LAN. We submitted batch files to compute servers using "remote queuing system" first on a single machine, then popped up software that could make decision on which machine to submit the job on, such as Condor or Loadleveler, the latest being able to choose a machine based on its CPU/load or disk space availability. These programs need a single user community to work properly (AFS, NFS)

And eventually the web was created, allowing easy sharing of information with HTTP. First there were static pages, then dynamic pages, allowing some interaction with the client. The Web matures, protocols evolve to specify more services, more client server applications with SOAP (Simple Object Access Protocol), and lately the WSDL (web Service Design Language) [2].

From Moore Law, every two years CPU speed doubles. Within 20 years we went from the first network bandwidth of 50 Kbps to 155 Mbps. The number of connected machines rose from 4 machines to more then 10 million.

Today, in a company, workstations have more capacity then ever, but this capacity is used around 10 hours a day, this results in a huge amount of CPU cycles, storage capacity, and network bandwidth being wasted. The time has arrived where we can add more and more abstraction to modelize complex processes, resulting in more and more processes running (more CPU used) on a single machine to achieve complex tasks.

What is GRID and what it is not?

The term Grid term was first coined in the mid '90s to describe a vision for a distributed computing infrastructure for advanced science projects, the Grid was first properly explained by Ian Foster and Carl Kesselman in their book The Grid: Blueprint for a New Computing Infrastructure(Morgan Kaufmann, 1999; ISBN 1-55860-475-8)[3].

So the Grid is the usage of a large amount of heterogeneous devices all connected together with a reliable and high performance LAN or WAN network. The purpose of this technology is to use as much CPU (CPU Grid) as it can get, all needed disk space available for the job, disk space being located either on a local machine or on a remote machine (data Grid). Organisation could be a single or many companies wanting to share “resources” or information. One must have CPU power, the second one the important data, the third one the application programs. Grid must then assure some kind of services such as resource balancing for better performance, scheduling to allow better utilisation of all idle CPU cycles of machines, discovery to “find” all machines pertaining to the Grid (with machine state as maintenance schedule, breakdown …), resource management and user management. In addition it should provide a collection of tools to hide the complexity behind the scene, and of course, be secure.

The Grid is the illusion of having a single large and powerful virtual computer made of a large collection of connected heterogeneous systems sharing various resources.

I will not consider peer to peer networks as being part of the Grid, as they are used more to share information, even when they are used to build some portion of a larger program such as seti@home[4], or genome project in France[5]. I will next describe the new emerging technology intermixing Grid technology with web applications, the Web Services. This new technology is being defined by the GGF (Global Grid Forum) [6].

Grid and security

To resolve the problem of collaboration between different organisations, the virtual organisation (VO) concept has been defined and described in the documents “physiology of the Grid” and “anatomy of the Grid” [7][8] .

Security into this VO is hard to reach, as users may not belong to the same organisation, data might be sensitive by nature (confidential), the resources needed by a single job could be large enough to fit onto many physical devices, the system must be available all the time.

All these problems could only be solved by using a secure common set of tools, and protocols to allow this heterogeneous world to collaborate, open source is mandatory to show how implementation is done and how security is handled, protocol must be standardized, accepted and deployed.

The defacto security protocol is GSI (Grid Security Infrastructure) [9] coming with the Globus Tool kit (Gtk). [10]

The Globus toolkit

Globus alliance[11] has defined a layered model to solve the VO problem; they are supplying documentations, specifications and tools. The toolkit consists of utilities and of API services, all relying on standard protocols. This can be described in the following schema:

Resource management GRAM	**Information Services MDS**	**Data Management GASS**
HTTP	**LDAP**	**FTP**
GSI		
TSL/ SSL		
TCPIP		

The toolkit uses four key protocols (HTTP, LDAP, FTP, TSL/SSL) and provides tools suite (GRAM, MDS,GASS,GSI) to support this architecture [12]. The tools can be organized as a set of resource services using a common connectivity layer

- Resources layer
  - Resource management GRAM
  - Information services MDS
  - Data management GASS
- Connectivity layer:
  - Security using GSI

The three resources layer services rely on the underneath GSI layer for security features:

MDS (Metacacomputation Directory Services) is responsible for reporting all the resources available for a specific host. All of this information will be used by GRIS, a collection of tools needing the knowledge of this particular node (GRIS: Globus Resource Information Services). All data representation is done using LDAP protocol, hiding physical differences, and locations. It answers two questions: what is the state of the Grid? What resources are available?

GRAM (Grid Resource Allocation Management) is the main function for remote computation, enabling secure and controlled remote access. This service is in charge of the authentication and authorization, resources discovery, computation monitoring and control. The underlying protocol is HTTP based RPC.

The “gatekeeper” tool, entry point of the Grid, is to the Grid what “inetd” is for Unix, it controls the execution of the job managers.(condor-g, or other).

The Globus Resource Specification Language (GRSL) provides a common language to describe jobs and the resources required to run them. GRAM components use RSL (Resource Specification Language) to specify job parameters.

GASS (Globus Access Secondary Storage) The Global Access to the Secondary Storage service is the Globus tool that simplifies the porting and running of applications that use file I/O in the Grid environment. This is the piece of code in charge of carrying the requested files to the machine needing it. The Globus tool which performs this task is named “GridFTP”, it is an FTP extension allowing secure and reliable data transfer.

The connectivity layer:

GSI (Global Security Infrastructure) is in charge of the security. Based on industry standards, available on many platforms including Linux and Windows, it exploits PKI and X.509 certificates [13] for mutual user-to-server authentications, it issues user and server credentials, it uses session level encryption to protect data.
GSSI is the major security stone used in the Grid.

Certificate and mutual authentication

A (digital) certificate is a file used in cryptography, to bind some pieces of information pertaining to an identity,

All users or services are identified by certificates, holding vital information, such as: user’s name, his public key, the identity of a Certificate Authority (CA), the digital signature of the CA…

Certificates are used for authorisation, digital signature, delegation, non repudiation, secure communication. Accessing a service is granted based on mutual authentication, which implies a Certificate Authority that both parties trust.

Mutual authentication : If two parties have certificates, and if both parties trust the CA that signed each other's certificates, then the two parties can prove to each other that they are who they say they are. This is known as mutual authentication.

The GSI uses the Secure Sockets Layer (SSL) [14] for its mutual authentication protocol.

Delegation

To avoid the user identifying himself each time he accesses a resource, GSI has defined a user proxy mechanism which consists of a new certificate (with a new public key in it) and a new private key. This certificate is signed by the user, and contains a flag stating it came from a proxy. For mutual authentication, the remote party receives not only the proxy's certificate (signed by the owner), but also the owner's certificate. This trusted chain can be repeated as long as needed; the process is known as delegation. Policies on different organizations can set a different life time on the proxied certificate, minimizing the risk in case an intermediate machine has been compromised [15] [16].

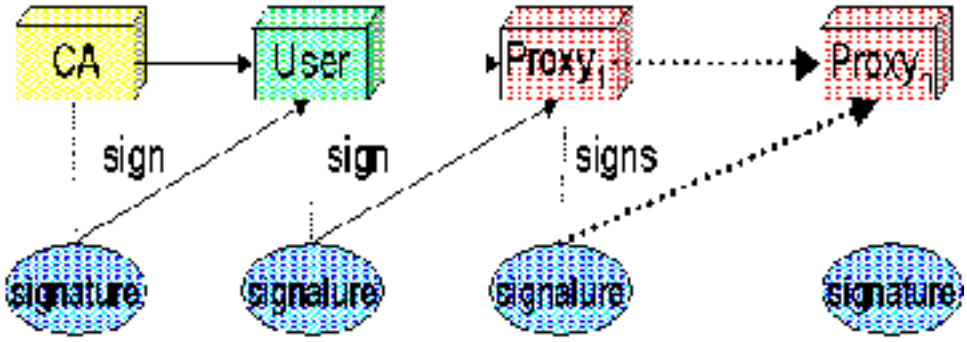


Figure 1 : Proxy and certificate

The user has received a private key, which is stored in a file on his local machine. He then must enter a pass phrase (a password) to decrypt his private key before using it. This is to prevent someone else using (stealing) his identity.

The security of the communication between the two parties is assured by TSL. Globus Toolkit software such as GridFTP and Grid SSH, support this delegation extension of TSL.

User related tasks

Mr Hacker has just received a user ID and a password: welcome to the Grid! The accompanying notice said his userid belongs to the Grid username space. This Grid is using Single Sign On.

.

At logging time, the user sends a request to the authentication server; which responds with a challenge string. The user responds by coding the challenge string with his private key (his password). The challenge string is sent back to authentication server that will decrypt the challenged string with a copy of the user's password. We can notice the password is never sent in clear text.

If the server can decode the string, the server will send back a token (a ticket) holding user's credential. This token will be presented each time the user wants to access a controlled service.

When the user logs into his local machine he identifies himself, the logging process creates a proxy which will carry user's certificate and sends this certificate to "remote" services. The transmission is secured using TSL (SSL). Of course he is now using a wireless connection to connect to his intranet as in the sci-fi book.

Single Sign On (SSO)

When Mr Hacker logs into the machine, he looks around his local account, then he runs a program that will submit a file transfer from a remote file located on a remote machine to another remote file located into another remote host. This FTP program should be given the right to perform this transfer securely, without asking again for password. This is done by delegation (proxy certificate). This process is known as Single Sign On.

Computation Grid

Mr Hacker just logs into the Grid, he would like to launch his program to calculate prime numbers.

First he will make some queries to know the status of the Grid: which nodes are up, which nodes are busy, which nodes are down (failure or maintenance). To accomplish his job, he will describe all resources he needs using RSL (Resources Specifications Languages). He will submit his job to a job dispatcher/scheduler specifying such specific features as main storage capacity, local file system size and so on. The scheduler will choose a machine satisfying the request, fires the program on behalf of that user; sends the result back to the originator. To elect a machine the job queuing system (Condor-g, or the like) is mainly talking to the "gatekeeper" of the remote host. The gatekeeper program is the main Grid entry point process running on the remote host.

The user in the meantime can submit queries to get intermediate status.

Data Grid

After the CPU cycles, the second most used resource in a Grid, is the files space (data storage). The Grid gives an integrated view of data storage, which can be located into physical memory or into (secondary storage) disk space.

Secondary storage can be used to increase capacity performance (by data replication), increase data sharing (setting correct Access Control List) and reliability of data by using mountable files systems as Network Files Systems (NFS) or more secured files systems as the Andrew Files System [17], (AFS), Distributed Files System DFS, General Parallel Files Systems GPFS[18], and San Files System SANFS.

Capacity and reliability can be obtained using the replication feature of the distributed files systems. AFS/DFS can also provide a discrete files or repository

access based on comparing user's token to files/repository Access Control List (ACL). Tokens/Tickets are needed to enter the different AFS cells or DFS realm.

Globus delivers a toolkit and an API services to help access to remote files systems. This is the most "customized" part of the Grid, and different Grid applications design team have designed their tools.

About globus Grid security

Globus Grid is more than cluster computing using a distributed files system such as AFS or GPFS. In a cluster you have a single community of administration, simplifying user registration, security policy definition and enforcement, clear monitoring, and system management. Also in a cluster you have a single type of operating system running on a certain type of hardware (mono cultural). In a Grid it is not that simple, as there are many operating systems, many policies and many organisations.

For local administration there are two types of administrators:

- for the basic operation system supporting the distributed files systems,
- for distributed files systems administration

.

These two kinds of administrators have different roles, hence different privileges. The Root administrator can "customize" operating system parameters, including those controlling DFS behaviour, but usually they can not access files in the DFS files systems without having a proper ticket granting access to specifics files or directory.

A contrario the DFS administrators have no right to manage the underlying operating systems.

Usually systems are located on a campus and are connected by a local area network which is easier to control, to administer and to secure then a Wide Area Network managed by more then a single organisation. Single management controls lead to clear and relatively simple rules and "single" policy management. The underlying files systems organisations named "AFS Cells", "DFS cells", can be deployed over a LAN or WAN network. AFS does not scale well when it comes to manage different organisation and security policy across cells boundaries.

We can imagine a Grid as an aggregation of different organisations, using different types of:

Operating and files systems,
User policy,
Security standards,
Management…

The communication into this "Babel tower" can only be achieved by a common set of standards and procedures. Resources sharing, systems and management collaboration do not imply that every collaborating organisation loses his identity, but rather have to publish their own policy standards. Grid needs an effective mechanism to map Grid policies to local policies and enforce locally the

different resulting policies. This is the first security concern, as this result in some manually customized procedures.

Grid foundations are based on security standards and protocols as HTTPS, SSL/TSL and LDAP. These protocols are secure by construction (from a protocol point of view). The tools using those standards have their own security features, and are as good as the correct security setting for these tools are correctly set. What is the point of having HTTPS, if the server running HTTP is not up-to-date with the latest fix applied! SSH is really good, as long as there is no telnet daemon running on that server, and the server administrator installs every patch when vulnerabilities are published.

“Grid login” gives the credential to a Grid user. The “userid” is then “mapped” to a local system userid using a single file, which could be easily modified by a simple file edit by the machine’s administrator. This is a security issue that could be fixed by “developing” tools to integrate Grid credentials to distributed users management such as Kerberos, this login procedure can be named kGrid_login()!

New way of computing

We went from distributed computing to Grid computing. Why can’t we use Web technology with Grid infrastructure? The merging of these two concepts has led industry to Web Services applications described by new standards: “Open Grid Services Architecture (OGSA)”.

Web Services (WS)

Web services are the integration of application programs with Web technology. The main idea is to put some glue around application programs (query weather forecast, query bank account position, stock quote service …) to make these applications accessible to everyone through a Web interface. These Web Services should be independent of the platform they are running on. There should be no dependence of the programming language (should not need specific libraries), and should be independent of the communication protocol. Web Services are a set of standards and techniques for distributed applications. The needs of the Web applications could be fulfilled by Grid technology (heterogeneous, availability, security …). Web Services should advertise the services they are performing, as should, computational Grid node or Data Grid elements. Web Services applications could be run on remote machines could be integrated with other services…

As for the Grid technology, focus is put on standards definition. WSDL (Web Services Description Language (an XML extrapolation) as a language to describe interfaces to access the applications. WSDL is an abstract languages defining application interaction, defining how they can be “bound” to different protocols regardless of the underlying technology. Web Services are built on existing technology (SOAP, HTTP, and XML).

These features are basic Grid requirement so Web Services could be deployed using Grid infrastructure.

OGSA

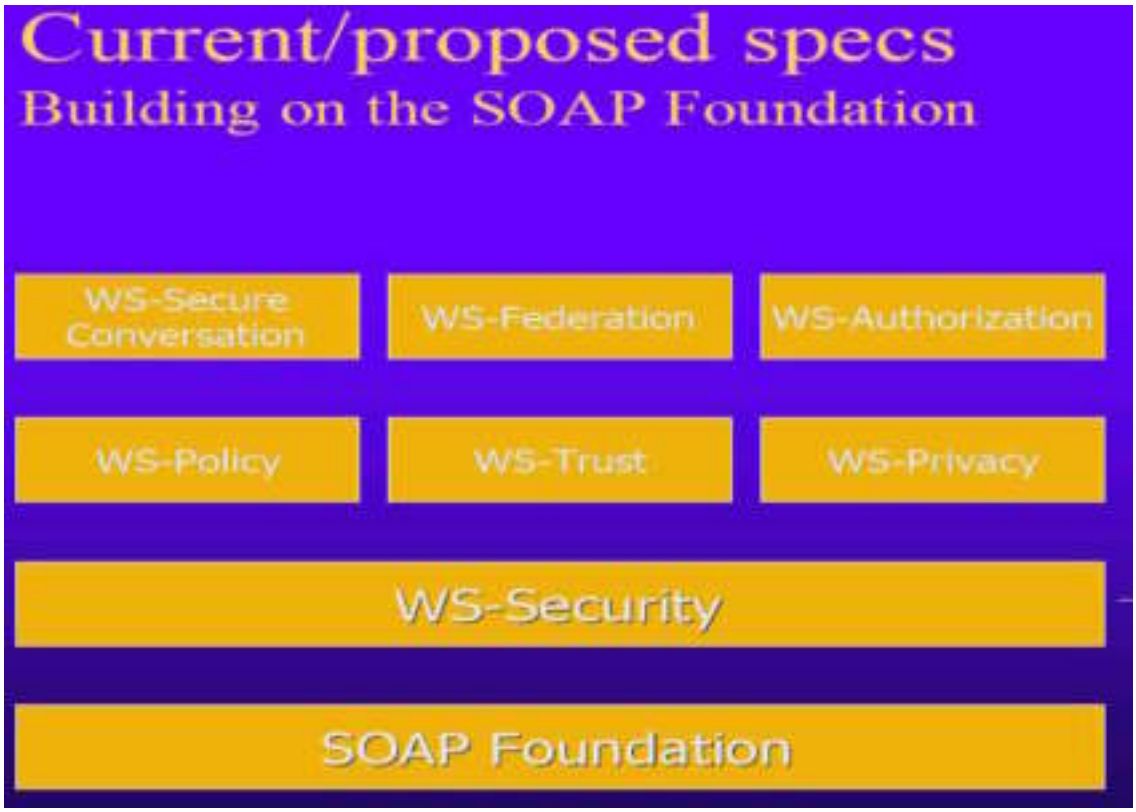
Web was crafted for scientific and engineering people to share data. These services have evolved towards e-business, using Web Services. WS did not take into account any resources provisioning, nor system integration and thus could not guarantee any quality of services (QoS).

OGSA is a framework description architecture using enhanced WSDL languages to describe Web services.

OGSA should handle standard services specification such as resources management, databases services, workflow support, security, diagnostics, alarm and event forwarding.

GGF(Global Grid Forum) has for objectives to define and standardize the Web Services(WS), and to address security issues within the Web Services environment (WS-Security [19]).

The Web Service security model looks like the Globus toolkit model, in which all applications will end up talking to the WS-Security layer, which in turn communicate with the SOAP layer as a transmission layer (see fig 2). SOAP protocol does not take any assumption about the underlying protocol necessary to carry information (as does TSL or SSL).



.
Fig 2: WS-Security architecture

WS-Security specifies SOAP messaging enhancements to assure message integrity, message confidentiality, has a proposition to manage a security token [20], and of course (as all Grid technology) inter-operate with existing security standards (SSL, TLS, IPSEC, XML signatures, X509 certificate).

The same problems as with computational Grid arise with WS-Security:

- What are the tools to distribute WS-Policy ?
- How to handle WS-Trust?
- What about WS-Authorization,
- What about privacy?
- How to handle dynamic Grid (WS) services deployment?

OGSA is being supported by:

- IBM with OGSA enabled WebSphere, OGSA enabled DB2. IBM also supports Globus project, OGSA research and development activities.
- Microsoft within .net technology
- Entropia, United Devices, Avaki OGSA compliant products

Autonomic computing

Grid computing and Web services have brought us computer facilities at work and at home. The development of wireless remote access is creating a new need for direct access to computer services, either for home banking, movie schedules, or checking partial results of our long lasting computer run. As the needs for “instant access” increase, our frustration increases also when the “system” does not work anymore, or worse, when we get “poor” response time. From a business perspective, the system should be up and running year round as customers are spread all over the world.

As the overall IT system is more and more difficult to understand in its totality, have more and more power, more and more network bandwidth, is more and more Grid oriented, why can not we use all this Grid intercommunications to provide self managing, self repairing, self controlling, self healing and self securing IT resources. This is the definition of Autonomic Computing, and this is what we all want to use.

Grid technology has prepared the road to Autonomic Computing (AC).

.

New security threat

With Web Services come a lot of middleware applications written in Java. Is Java secure? It depends on what we are considering.” Official” java modules (java applet) are secure as they run into a dedicated sandbox, in which you scrutinize carefully the inter-action between the program and the host the program is running on. Java program does not need to install itself, so it is virus free, as it can not replicate itself, nor store data in the operating files system (90% of virus are targeted at Microsoft). Another mean to decrease the risk is to control how much interaction is allowed to the java code. It is a good habit to filter java code execution within your browser.

WS Security 1.0 standard is published, and we can now find WS-Security Gateway appliance filtering according to this standard (layer 7 firewall), but like IP fire-walling, it is as good, as it can be securely configured. Is this appliance a stateful firewall? We do not have tools to check the WS-Trust, nor WS-Authorization …

All the messages used in WS are coded with XML. SSL is used to secure the transmission between client and server, but how can you assure the messages have not been modified on the server (or in the intermediate nodes) itself, and/or on the database?

This is why W3C consortium and IETF are working to define XML signatures, and XML encryption. The XML signature and XML encryption could be applied only on a part of the message. XML signatures as digital signatures (PKI based) can be used for authentication, data integrity, non –repudiation of the signed data.

Conclusion

I have presented two main flavours of Grid, Grid computing and Web Services. Both technologies will use benefits of autonomic computing even if autonomic computing is not mature yet. Securing autonomic computing is a huge task, but security has been taken into account from the beginning, as we are all aware of computer hacking, and not afterwards as it used to be.

Computational Grid is a reality, and a technology success. It relies on collaboration, open source community, and standardization which lead to reliability and security. The best way to secure the Computational Grid is by defining, using and publishing clear standard, and showing source code, and using public Grid policy.

Web Services are also a success, more and more applications are being made aware of new inter-communication based on Web Services technology. The different protocols involved in web communication are being made secure either by construction or by using already secured technology.

Computer are becoming more and more smarter than ever, through more and more concept abstractions, computers are running faster, having larger primary storage, more network bandwidth, are being made aware about other computer collaboration, and intelligent enough to win chess against the more than average chess player.

As computers will have almost human behaviour, the big question is what could be, and how to define, a security model?

Annexe 1: Some samples of scientific GRID realisations
From the web: http://www.ifae.es/pic/pic_communities.htm

The GRID philosophy can be applied at small and large scale, from projects involving a small number of centers up to projects on European scale and beyond. The scope of PIC's activities has been set to address support for scientific communities that need technological innovation in order to benefit from GRID technologies in subjects which need treatment of massive amounts of data under extremely difficult conditions.

Some examples of such communities are found in the application examples within GRID technology development projects with European Union financing, such as the DataGrid and the CrossGrid projects in which IFAE and UAB are already participating:

1. High Energy Physics, which is building at CERN, the European Organization for Nuclear Research, the Large Hadron Collider (LHC). When this accelerator starts its operation, between 5 and 8 million GigaBytes per year have to be stored and analyzed from different points of the European, American and Asian geography.

2. The studies of the human genome, which is only one of the many genomes that have been sequenced. An instrument to understand how the human DNA works is to compare it with that of other species, which implies traversing many times over thousands of GigaBytes information, searching for non-predictable patterns.

3. The European Space Agency (ESA), with its project of Earth intensive monitoring. ENVISAT generates about 500 GigaByties per day in Earth images for complex studies such as climate change.

In all three cases, we are dealing with research that needs to analyze a large amount of data and that, moreover, have to be shared between scientists in different locations.

Apart from these communities, which have already adopted GRID computation for their studies, we can think about others which are just around the corner. An example would be the collaboration among different hospitals to share, in a secure and anonymous way, their data -to go forward in the prevention, detection and treatment of breast cancer for example. If you give freedom to your imagination, there are many other examples of emerging data-intensive collaborative science.

References

Fig 1: http://www-fp.globus.org/security/overview.html

Fig 2: Web Security description page 15
URL: http://www.cs.virginia.edu/~humphrey/ogsa-sec-wg/OGSASec-ggf5.pdf

[1] John M. Ford. *Web of Angels*. Tor. 1980

[2] Web Services Description Language (WSDL) 1.1W3C Note 15 March 2001
URL: http://www.w3.org/TR/wsdl

[3] David Shotton (revised rahtz). What is the Grid?
August 2001 (revised Fri, 21 Sep 2001)
URL: http://e-science.ox.ac.uk/what/index.xml?style=printable

[4] Seti@Home The search for extraterrestrial Intelligence
URL: http://setiathome.ssl.berkeley.edu/

[5] Genome@Home:
URL: http://www.stanford.edu/group/pandegroup/genome/

[6] Global Grid Forum overview
URL: http://www.Gridforum.org/L_About/about.htm

[7] The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration
Author: Ian Foster, Carl Kesselman, Jeffrey M. Nick, Steven Tuecke
URL: http://www.globus.org/research/papers/ogsa.pdf

[8] The Anatomy of the Grid
Enabling Scalable Virtual Organizations
Author: Ian Foster, Carl Kesselman , Steven Tuecke
URL: http://www.globus.org/research/papers/anatomy.pdf

[9] Grid Security Infrastructure
URL: http://www.globus.org/security/

[10] Globus Toolkit Download page
URL: http://www.globus.org/gt2.4/download.html

[11] Globus Allicance
URL: http://www.globus.org

[12] Introduction to Grid computing and GTK
URL: http://www.globus.org

[13] X.509 digital certificates
URL: http://www.horseplay.demon.co.uk/x509.html#x509

[14] Transport Layser Security extension
Network working group RFC 3546
Author: S.Black-Wilson,M.Nystrom, D.Hopwood, J.Milkensen, T.Wright
URL: http://www.ietf.org/rfc/rfc3546.txt

[15] An Online Credential Repository for the Grid: MyProxy
Author J.Novotny, S.Tuecke, V.Welch
URL: http://www.globus.org/research/papers/myproxy.pdf

[16] Internet X.509 Public Key Infrastructure, Proxy Certificat Profile
Internet Draft, Document: draft-ietf-pkix-proxy-09
author: S. Tuecke, V. Welch, D. Engert, L. Pearlman, M. Thompson, Nov 2003
URL: http://www.ietf.org/internet-drafts/draft-ietf-pkix-proxy-09.txt

[17] *AFS* Distributed File System
URL : http://www.bo.infn.it/alice/alice-doc/mll-doc/impgde/node49.html

[18] A GPFS primer
URL: http://webdocs.caspur.it/ibm/power4_doc/gpfs/gpfs_primer.pdf

[19] Security in a Web Services World: A Proposed Architecture and Roadmap
URL: http://www-106.ibm.com/developerworks/security/library/ws-secmap/

[20] IBM Adds security support to its Web Services
URL: http://www.webservices.org/index.php/article/view/308/1/2/

Glossary

GIIS Grid Index Information Service
GRAM: Grid resource allocation management
GridFTP Grid File Transfer Protocol
GRIP: Grid Resource Information Protocol
GRIS: Grid Resource Information Service
GSI: Grid Security Infrastructure
GSS : Generic Security Services
LDAP Light Directory Access Protocol
MDS : Metacomputing Directory Service
MPI : Message Passing Interface
RSL: Resource Specification Language
SRB Storage Request Brokers