Journal of Theoretical and Applied Information Technology
10th December 2015. Vol.82. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN ANALYSIS OF TECHNOLOGIES FOR BUILDING INFORMATION SECURITY INFRASTRUCTURE OF GLOBAL DISTRIBUTED COMPUTING SYSTEMS

[1]PAVEL SERGEYEVICH PTITSYN, [2]DMITRY VLADIMIROVICH RADKO

[1]Research Institute of Semiconductor Engineering, JSC
Leninsky Prospekt, 160a, Voronezh, 394033, Russian Federation
[2]Voronezh innovation and technology center, LLC
Leninsky Prospekt, 160a, Voronezh, 394033, Russian Federation

ABSTRACT

The implementation of global distributed information systems based on cloud and grid approaches. There are many problems of ensuring a high level of information security of these systems because they operate critical or confidential data, and the elements of these systems found in different physical locations to communicate with that uses open standards and protocols of the Internet. The existing distributed information systems implemented using a variety of architectural and technology platforms, which usually do not meet the current challenges in the field of ensuring a high level of information security. In addition, the actual question of the integration of these systems with corporate information systems, and providing a high level of security used integration solutions. The aim of this work was the systematization and analysis of proven technologies for building high reliable information security infrastructure of global distributed computing systems. As part of the work identified approaches to implementation of information security infrastructure based on technical standards Globus Toolkit, OGSA, UNICORE, gLite. The features of the implementation of security infrastructure based on these standards, as well as the possibility to interact with external systems based on industry standards such as SOA, Web Services.

Keywords: *Distributed Computing Systems, Security Framework, Security Infrastructure*

1. INTRODUCTION

Distributed information systems allow creating a geographically distributed computing infrastructure that brings together diverse resources and the ability to implement multiple access to these resources.

Currently the most prevalent approach to the implementation of distributed information systems based on Grid and Cloud technologies, which represent a cluster solution, connected by a network of loosely coupled heterogeneous computers [1, 2, 3].

The principal novelty of the grid and cloud technologies is the pooling of resources through the creation of a computer infrastructure of a new type, providing the global integration of information and computing resources. Given approach based on network technologies and special middleware software, and a set of standardized services for secure sharing geographically distributed information and computing resources including separate computer clusters, data storages and networks [4].

The basic elements for providing the functions of information security distributed systems are the following [5]:

Authentication - process of identity participant interaction. In traditional systems, the authentication process provide the protection of the server. In the global distributed systems, however, in order to protect themselves from intruders, it is equally important to check the authenticity of the server.

Authorization - determine the acceptability of the system requested operation. The global distributed system for the adoption of similar solutions, such mechanisms must act based on the rules established for each resource.

Confidentiality and data integrity - transmitted or stored data should be protected by adequate mechanisms to prevent illegitimate access. In some cases, it is necessary to ensure complete isolation of a specific data set from illegitimate users.

Billing and auditing. For creation large-scale distributed structures, organizations require

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

mechanisms that monitor and calculate the amount of resources used. Accounting mechanisms also ensure that all parties comply with the agreement on the use of resources. The audit of the transaction information makes it possible to identify the source of danger or security breach.

At each level of the architecture global distributed systems, the information security have their own specifics. It should take into account the fact that a complete solution of information security of distributed systems must interact successfully with existing local solutions for information security. However, the diversity of local solutions significantly complicates creation of integrated security systems. It should be noted that one of the possible ways to address the difficulties of this kind is the use of common standardized approaches to the implementation of local solutions to ensure information security of distributed systems [6].

2. METHODS

In terms of standardization (architecture, protocols, interfaces, and services) distributed computing technologies described by the following set of criteria [7]:

Coordination of resources in the absence of centralized management of these resources.

The usage of standard, open, universal protocols and interfaces.

Providing high-quality user service.

For building, a fully functional distributed computing system requires middleware software, built based on existing development tools.

Particular relevance acquire information security issues of these systems related to the prevention of unauthorized access to resources and information , as well as ensuring a high level of availability of distributed resources and services.

For solving of standardization issue are used specialized software frameworks that are able to provide a high level of security and reliability of the developed distributed systems. The basis of these systems is the concept of services, or components that interact over a global network. One of the most important indicators of their successful operation is the level of accessibility, i.e. opportunity in a reasonable time to process the entire stream of incoming requests from users. The level of availability at a given time defined as the current workload resource distributed system or the network environment, and the fact that the implementation of DDoS-attacks aimed at generating conditions for denial of service. The following is an analysis of the existing systems on the market, including the Globus Toolkit, OGSA, UNICORE, gLite. Presented their main features and distinctive characteristics.

3. RESULTS AND DISCUSSION

3.1 Globus Toolkit security infrastructure

Globus Toolkit security Infrastructure is set of programming modules (framework) for building global distributed computing systems. Each module defines group of interfaces, which used high-level components and had related implementation for different run-time environments. Globus Toolkit had the following types of modules [8]:

Security services.

Access services.

Information services.

Monitoring services.

Job management services.

Data services.

The conceptual architecture of Globus Toolkit framework represented in Figure 1.

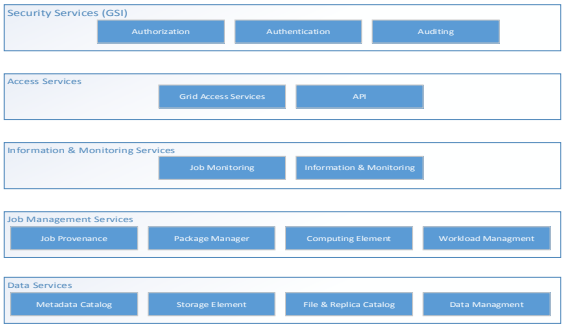

Figure 1. The Conceptual Architecture Of Globus Toolkit Framework

Globus Toolkit framework implements Globus Grid Security Infrastructure (GSI) model for solving the security information tasks [8]. Key points of GSI security policy are the following [9]:

- The computing environment consist of many trust domains. Given trust domain responsible for association with heterogeneous local resources and end-

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

users. The security of distributed computing environment in given case is management of cross-domain interactions and timely mapping cross-domain policy on local policy.

- The restrictions of one trust domain is subject of local security policy. Hence, there are no additional actions for creation security services on computing environment side except the local security infrastructure of trust domain. The implementation of local security policy will take place generally using many methods including firewalls, Kerberos, SSH.
- There are two types of subject: local and global. The global subject will have related rights for each trust domain depending on the rights of local subject. Each user of the resource will have two names, global name and local name. For example, the web site could define global user name depending on local user name, and its related user rights. Thereby the global subject will be able to pass through single sign on authorization in global computing environment.
- The transactions between the objects located in different trust domains require mutual authentication.
- Trusted global subject, which mapped to the local subject, adopted as a local domain entity. In other words, within the domain trust the combination of authentication policy and local display meets the safety requirements of the host domain.
- All access control performed locally, at the level of local entities. That is, the access control managed by local system administrators.
- A program or process that acts on behalf of the user has a subset of the rights of the user. This element of security policy is needed to ensure that the long existing programs that can dynamically request resources without interference and interaction with the user. It is also necessary to create processes in other processes.
- The processes running on behalf of the same subject within the same domain trust can share a single set of mandates of confidence (called credentials) global computing environment can include hundreds of processes running on a single resource. This component allows for the scalability of the security policy of the security architecture for its use in large-scale parallel applications and eliminates the needs for a unique mandate for each process.

GSI infrastructure implements the above security policy and has the following important features [10]:

- Single sign-on registration allows the user to only one time to pass the procedure of authentication and, thus, create a proxy certificate, which may be brought against any of the remote service to authenticate the user's behalf. Delegation enables the creation and transmission of remote service of delegated proxy certificate that can be used by the service to perform actions on behalf of the user (perhaps with some limitations). This feature is important when performing an operation with nested structure.
- As a basis for identifying, the user GSI uses certificates X.509, the widespread standard for PKI certificates. In order to accommodate the X.509 to support single sign-on and delegation of authority, GSI defines proxy certificate X.509 [9, 10]. Typically, an authentication GSI uses Transport Layer Security (Transport Layer Security - TLS), which is a modification of the Secure Sockets Layer (Secure Socket Level - SSL). Although other authentication protocols are based on public key technologies can be used to work with a proxy certificate X .509. Remote delegation protocol proxy X.509 certificates overbuilt above TLS.
- The rights of using X.509 certificates as a private keys are "personality», or the approach of identification of each object - the user, resource or program, indicating the name of the object and additional information, for example, the public key. Authorized to issue certificates (certification authority - CA), some credible independent organization signing certificate binds a means of identifying an object with a public key.
- Authentication algorithm defined by Secure Socket Layer (SSL) protocol, performs the identification of the object. The reliability of the results of such testing is determined by the degree of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

trust in the CA, so the local administrator receives the certificate, and then using them to verify the certificate chain.

- The object can delegate a subset of their rights (for example, the process of generating a different process) to a third party, creating a temporary means of identification, called an intermediary (proxy). Certificates form a chain of intermediaries, which begins with the CA, and then built up when the user first, and then signed by the mediators certificates. By checking the certificate chain processes are initiated by the same user on different nodes can authenticate to each other by holding back the chain of certificates until it finds the original certificate user.
- Each resource can set its own rules to determine how to respond to incoming requests. GSI initially uses a simple access control list, but it can be enhanced more advanced techniques.
- Authentication protocol checks the global names of the parties involved, but GSI have to convert the name to the local (e.g., login name or the Kerberos name), before the local security system can use it. GSI performs this procedure on a local node, referring to a simple text file of correspondences, which establishes the relationship between global and local names.
- Standard GSS-API interface (Figure 2) provides access to the functions of protection. GSI uses OpenSSL or SSLeay (freely available implementation of SSL) to its authentication protocols and certificate support intermediaries. SSL is widely used to secure web-based environment. The implementations of security algorithms in terms of standard GSS-API allows to take into account the heterogeneity of the local domain in global distributed environment [11, 12], which allows one object to convey a specific subset of the "pool" their privileges to another entity. Such a restriction is important in terms of harm reduction in the deliberate or accidental misuse of the delegated certificate.

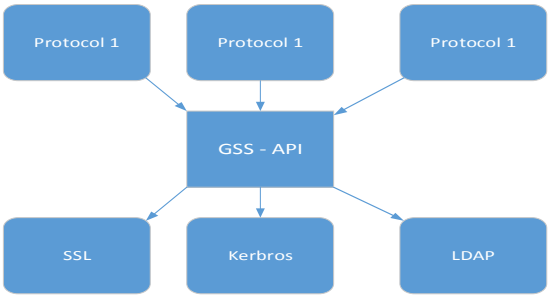

Figure 2. Using Standard GSS-API Interface In Globus Toolkit Architecture

Despite its relative simplicity, the architecture corresponds to all the critical requirements of users and systems [13]. From a user perspective, the global name and rights of intermediaries means that the user to gain access to all the resources, it needs only once to pass the authentication procedure, and the rights of the mediators and the procedure for delegation of authority allows programs running on behalf of the user to access the resources . Using the standard X.509, SSL, and GSS-API encourages the development of common tools, based on the GSI and the more complex applications. From nodes of the system perspective, the architecture does not require a revision of the local security infrastructure. Instead, the nodes set a relatively simple server supporting GSI, who use the well-known standards. The nodes rules govern with the help of an access control list and map file, so convenient to work with administrators to GSI, and they are ready to deploy it in parallel with SSH and other remote access mechanisms.

3.2 OGSA (Open Grid Services Architecture) security infrastructure

Globus Toolkit was widespread because it was the first complete set of tools for development infrastructure in the field of Global computing distributed systems. However, even the most common the second version of Globus Toolkit was not without drawbacks, the main of which was the lack of unified application development tools that can interact with each other and provide each other with a variety of services [14].

To solve given problem it was developed Open Grid Services Architecture (OGSA) standard. OGSA defines a core set of services that provide Global distributed system and describes their architecture. In the terminology of these services are called OGSA capabilities. Examples of these capabilities are launching applications, data access, etc. In OGSA, the Global distributed system is

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

viewed as a set of independent services that can be used independently or together to build the required infrastructure [15, 16].

OGSA offers to design Global distributed system on the principle of service-oriented architecture (Service-Oriented Architecture, SOA), which determines the method of constructing software systems in the form of a set of independent or loosely coupled services. It is assumed that each service perform their function well defined and has a rigid semantics. Services allow multiple implementations, but have a standard, strictly specified interface through which can interact with each other and with third party applications. Thus, in OGSA Global distributed system is represented as a set of services that implement various functional facilities.

OGSA architecture has the following layers (Figure 3):

- Data layer is represented by resources which may be part of the Global distributed system.
- Service layer is set of functional services. At this layer, there is a generalization (virtualization) resource. Given high-level services provided well-defined interfaces. Strict specification of this level is the main task of OGSA.
- Application layer is set of related business applications that use the services to perform certain tasks. This layer is not specified in the OGSA.

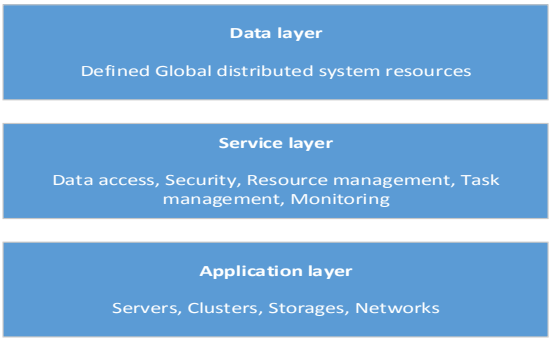

Figure 3. Three Layers Of OGSA Architecture

OGSA offers to design Global distributed system on the principle of service-oriented architecture (Service-Oriented Architecture, SOA), which determines the method of constructing software systems in the form of a set of independent or loosely coupled services. It is assumed that each service perform their function well defined and has a rigid semantics. Services allow multiple implementations, but have a standard, strictly specified interface through which can interact with each other and with third party applications. Thus, in OGSA Global distributed system is represented as a set of services that implement various functional facilities.

OGSA architecture as part of the security and reliability infrastructure implements the following functions [16]:

- Shared access - allow to organize shared access to resources owned by different organizations. It would provide a very flexible control over shared resources.
- Authentication and authorization - will enable authentication and authorization of users Global distributed system based on the policy domain for the provision of administrative resources and the policy community (virtual organizations) using these resources.
- Integration of security systems - will enable the integration of various models of cooperation and security systems.
- Delegation of rights - provides mechanisms for delegating user rights for avoiding multiple authentication.
- Quality assurance service. Global distributed system have to ensure quality of service for applications with such conditions as the minimum allowable network bandwidth, guaranteed performance of calculations, and guaranteed level of security.
- Reliability. Global distributed system have to ensure high reliability and fault tolerance. This may require the use of spare resources, data backup, resource monitoring, and automatic recovery from failures. For jobs, working for a long time may be required repair mechanisms, e.g., using control points.
- The ease of use and extensibility. Working of users with Global distributed system should be as simple as possible. Requires the support of various levels of tasks - from simple (with a minimum flexibility) to complex, require special knowledge and skills. Global distributed system architecture has to take into account possible extension of the range of applications and the emergence of new requirements.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

- Scalability. The architecture of the Global distributed system should not be bottlenecks to scaling given system.

Implementation of the OGSA architecture based on GSI infrastructure is show in the figure 4. It is greatly simplifies the construction of the infrastructure of information security for heterogeneous distributed subsystems on the relevant service-oriented services OGSA, and related Web standards [17, 18].

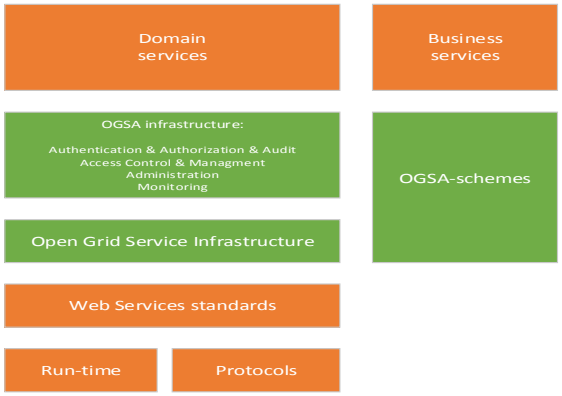

Figure 4. Implementation Of The OGSA Architecture Based On GSI Infrastructure

Implementation of the OGSA architecture in component model Globus Toolkit 4 is show in the figure 5.

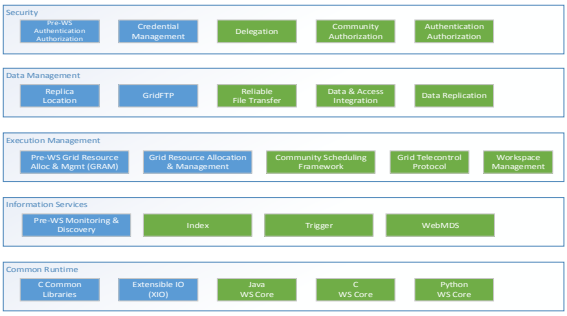

Figure 5. Implementation Of The OGSA Architecture In Component Model Globus Toolkit 4

3.3 UNICORE infrastructure

UNICORE (UNiform Interface to COmputing REsources) - a uniform interface for distributed computing. UNICORE provides online access to the nodes of the system, where the difference in hardware platforms, security mechanisms and exchange of information hidden from end users [18].

UNICORE has three-layer architecture (represented in the figure 6). It includes the user layer, UNICORE service layer and system layer, where it will run the target system applications. The components interact with each other in an open network via SSL (Secure Socket Layer) protocol [19]. To establish a client- server connection SSL uses public key cryptography, so that each component must have public and private keys. Keys will receive a certificate that the component could be identified as safe. The user sent data, which signed with his private key. Other users can use the public key to decrypt and verify the authenticity of the data.

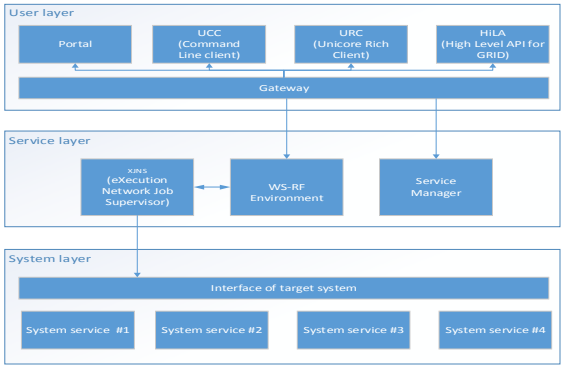

Figure 6. UNICORE Architecture

The key of UNICORE model is the concept of an abstract job (AJO - Abstract Job Object). This object based on data received from the user interface and the runtime. AJO goes through the gateway and UNICORE protocol, based on SSL. AJO sets standards for the exchange of data between the user and the network management software, and communicates directly with the UNICORE server [20, 21].

The client part of UNICORE is a graphical user interface. It provides an opportunity to prepare UNICORE tasks and monitor their implementation, and manage security settings. UNICORE user interface indicates the parameters related to the problem - for example, resources, command-line arguments, files, etc. Dispatching service determines whether the job to be performed on the selected system, and if the answer is positive, makes the task of the next stage. Translation of the abstract definition of the problem into an executable package on the real machine also takes place via a network controller (NJS).

Separate systems or clusters with a single file and the user environment in the general UNICORE model presented as virtual sites. Each site operates

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

a network job supervisor (NJS). NJS performs the following functions:

- Analyze an abstract job and creates of a package locally executed tasks.
- Produces authorization, i.e. mapping UNICORE user ID on the target system.
- If the job includes a group of tasks, forwards them on at the appropriate gateway.
- Provides information about local resources, the status of assignments, as well as the output of the executable tasks.

When the user enters an abstract task, NJS unpack it using the user's public key. Then, when the job should be distributed among other resources, according to the dependency graph, it is sent through the gateway to the next site or another NJS on the local network. Each NJS has its own certificate X.509, which used when transferring jobs via the gateway. This gateway knows forwarded abstract task is part of another job and that should use a custom key for unpacking.

For those tasks to be executed locally, NJS mapped the user certificate to the local authentication system. The information indicating a user's account belongs to a certificate stored in a user database and maintained at each UNICORE node. Through given mechanism, the user only has to log in only once, and the involvement of other virtual sites they pass without re-authorization.

NJS has a correspondence table according to which the abstract commands are converted to executable instructions of the target system. Therefore, the user only has to formulate the task in general terms and does not go into the details of the implementation of the platform. After unpacking, the task managed by the interface of target system (TSI - Target System Interface).

User authentication, secure data transmission and information about available resources provides the gateway. This is one of the elements of the UNICORE server-side. It is also responsible for relaying tasks, data, commands and authentication attributes. UNICORE Gateway supports additional local security systems.

3.4 gLite infrastructure

gLite infrastructure is a software middleware construction of Global distributed systems . gLite is the main middleware software of projects EGEE, EGEE-II, came to replace the complex LCG-2 [22].

gLite has service-oriented architecture (represented in the figure 7). Services can be run in parallel, performing the task of the end user, or being applied in the context of an independent task [23]. gLite services divided into the following groups:

- Security services.
- Information and monitoring services.
- Services of control tasks.
- Data management services.

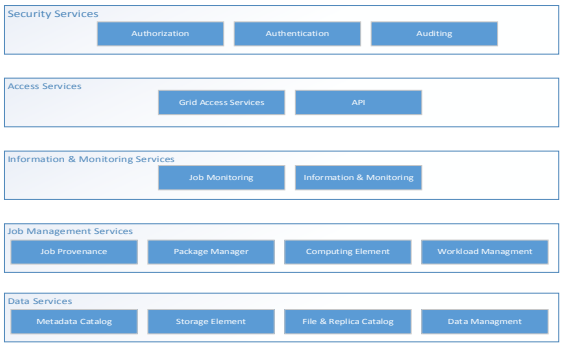

Figure 7. Glite Architecture

Security in gLite based on the proven technology of public and secret access keys and certificates X.509. Each transaction in the Global distributed system mutually verified the client and the server sides. The global access and registration of virtual organizations monitored by service belonging to the virtual organization VOMS (Virtual Organization Membership Service) [21].

The integrity, authenticity and confidentiality based on digital signature technology. gLite uses a widely used specification GSI, protected data transfer protocols TLS and WS-Security (Web Services Security) [23].

4. CONCLUSIONS

The result of this work is research of the technologies of information security of global distributed information systems based on specialized frameworks, including technical implementation of Globus Toolkit, OGSA, UNICORE, gLite.

The main distinguishing characteristics of these approaches to build information security infrastructure of distributed information systems are the following:

- The security infrastructure built on a modular principle, provides flexible

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

configuration and expansion of the functional and technical capabilities.

- The security infrastructure uses a single centralized distributed database that contains all user data as well as a description of the business logic to process the given data.
- The security infrastructure supports distributed access to data and distributed business transaction that provides operating modes in the active cluster, grid and cloud systems.
- The security infrastructure implemented based on n-tier architecture that includes a data layer, business logic, user interface level, at all levels of implementation and will provide the context of information security.
- The developed modules used a technique of designing distributed and scalable applications (SOA, MDA, ESB, MVC), as well as open standards (XML, SOAP, WSDL, REST, Web Services).
- The security infrastructure has flexible settings (user interface, business logic, data forms) and is open, which allows for integration cooperation with a wide range of information systems.

5. ACKNOWLEDGMENTS

The Ministry of Education and Science of the Russian Federation supported the work (Agreement #14.576.21.0078, unique identifier agreement RFMEFI57614X0078).

REFERENCES:

[1] Foster, I., & Kesselman, C. (2004). *The Grid 2: Blueprint for a New Computing Infrastructure* (2nd ed.). San Francisco, CA: Morgan Kaufmann Publishers Inc.

[2] Foster, I. (2001). The Anatomy of the Grid. Enabling Scalable Virtual Organizations. *Supercomputer Applications*, 15 (3), 200-222.

[3] Foster, I., Kesselman, C., Tsudik, G. & Tuecke, S. (1998). A security architecture for computational grids, *Proceedings of ACM conference on Computer and communications security* (pp. 83-91). Washington, DC.

[4] Ian, D. A. (2010). *A security framework for distributed batch computing*. Madison, WI: University of Wisconsin–Madison.

[5] Hee-Khiang, N., Quoc-Thuan, H., Bu-Sung, L., Dudy, L. & Yew-Soon O. (2005). Nanyang Campus Inter-organisational Grid Monitoring System, *Proceedings of Grid Asia Workshop on Grid Computing and Applications*, (pp. 118-127). Singapore: Nanyang Technological University.

[6] Purd, J. (2007). *Data Grids and Service-Oriented Architecture*. Redwood Shores, CA: Oracle Corporation.

[7] Raj, K., Stuart, M., Mihalo, B. (2010). *Setting up and using a Globus Toolkit 5 based Grid*. Chicago, IL: Argonne National Laboratory.

[8] GGF OGSA Security Workgroup. (2002). *OGSA Security Roadmap*, Chicago, IL: Argonne National Laboratory.

[9] Neha, M. (2014). Security issues in grid computing. *International Journal on Computational Sciences & Applications*, 4 (1).

[10] Benedyczak, K. (2005). UNICORE as uniform grid environment for life sciences, *EGC'05 Proceedings of the 2005 European conference on Advances in Grid Computing*. Amsterdam, Netherlands.

[11] Montagnat, J. (2008). A Secure Grid Medical Data Manager Interfaced to the gLite Middleware. *Journal of Grid Computing*, 4 (6).

[12] Seung-Hyun, K., Kyong H. K., Jong, K., & Sung-Je, H. (2004). Workflow-Based Authorization Service in the Grid. *Journal of Grid Computing*, 2 (1).

[13] Shingo, T., Susumu, D., & Shinji S. (2003). A user-oriented secure filesystem on the Grid, *The 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid*. Shenzhen, China.

[14] Jianmin, Z. (2006). *Secure Grid Computing*. Richardson, TX: The University of Texas at Dallas.

[15] Webber, J., Parastatidis, S., & Robinson, I. (2010). *REST in Practice*. Sebastopol, CA: O'Reilly Media Inc.

[16] Kalmady, R. (2009). *GridView: a Grid monitoring and vizualization tool*. CERN, Geneva, Switzerland.

[17] Sonvane, D. (2010). *Computation of Service Availability Metrics in Gridview*. CERN, Geneva, Switzerland.

[18] Tulloch, M. (2003). *Microsoft Encyclopedia of Security*. Redmond, WA: Microsoft Press.

[19] Tryfonas, T. (2009). *An Alternative Model for Information Availability*. Bristol, United Kingdom: University of Bristol.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[20] Cornwall, L. A. (2004). Authentication and authorization mechanisms for multi-domain grid environments. *Journal of Grid Computing*, 9(1).

[21]Aiftimiei, C. (2006). Requirements, Architecture and Experience of a Monitoring Tool for Grid Systems. *In Proceedings of the International Conference on Computing in High Energy and Nuclear Physics*, Mumbai, India.

[22]Handley, M. (2006). *Denial-of-Service Considerations. RFC4732*. Ottawa, Ontario, Canada: Network Working Group.

[23] Butt, A. (2003). *Grid-computingportals and security issues*, New York: Academic Press.