# An Adaptable Authentication Logic for Cloud Bursting in Cloud to Grid Infrastructure

Kisshor Kumar Sethuraman, Hemalatha Thangaraj

**Abstract**— The Cloud is an emerging technology derived from distributed computing of past, which holds grid as its infrastructure. Trust in service delivery through multi-level authentication over the cloud is a tedious task faced by both end users and cloud providers of today. Usage of Public Key Infrastructure PKI and Kerberos authentication are the existing authentication mechanisms in cloud. When the Service Level Agreement (SLA) could not be matched with the resources in the private cloud, it is necessary to do cloud bursting. Cloud bursting is a challenging issue, since when a job is being migrated from private cloud to public cloud; the heterogeneity with respect to authentication must be masked from the end user. We proposed a multi policy authentication scheme that acts as a delegator in migrating jobs from end user to hybrid cloud by bridging and managing SLA violations. The negotiations made between the delegator and third party authentication service and the complexity involved in multi-level authentication is hidden.

**Index Terms**— Cloud Computing, Cloud Bursting, Kerberos, Multi-level Authentication, PKI, Private Cloud, Public Cloud, SLA.

## 1 INTRODUCTION

THE cloud [1] is based on Service Oriented Architecture that acts as a service request-provider model that maps the services to a service registry. The SOA [2] is similar to the RMI [3] architecture where the former operates on large scale while later on the small scale. The cloud is a based on the distributed and utility computing. It provides the resources to the end user on pay per use model. The resources are provided by cloud in the form of services. There are three common services that are popular among the cloud providers. They are Software as a Service [4], Platform as a Service [5] and Infrastructure as a Service [6].

### 1.1 Software as a Service (SaaS)

The Software as a Service (SaaS) acts as a service delivery model that provides application usage over the Internet to its end users. The applications are of various types such as office, gaming, emails and file storage and sharing services. The SaaS is implemented through a web front end, interfaced by an application server connected to a back end database. There are various cloud providers involved in delivery of SaaS. They are Amazon (AWS-Amazon web services) [7], Abiquo [8], Salesforce [9], Akamai [10], Apprenda [11] and many others. The Application servers include JBOSS [12], Oracle Glassfish [13] and the database used mostly is MySQL [14]. In case of cloud database MongoDB [15] is used. The major support in the above tools is the open source type that helps to achieve the goals of developer, provider and the end user easily.

### 1.2 Platform as a Service (PaaS)

The Platform as a Service enable providers to service end users with set of tools and packages other than applications as in SaaS.The PaaS enable users to enjoy the services of SDKs, IDEs and other application infrastructure of specific platform as supported by the Cloud provider. The typical platform is Linux with supported tools such as gcc, tar, libre office and also the distributions such as fedora, Ubuntu etc. The Windows platform includes Visual studio 2012, SQL Server and Visual .NET runtime. The Apple platform includes tools such as Cocoa SDK and XCode IDE. The major PaaS providers are Amazon, AppScale [16] and Google (Google App Engine) [17] and many others.

### 1.3 Infrastructure as a Service (IaaS)

The Infrastructure as a Service enables cloud providers to provide hardware resources that enable to establish an infrastructure on end user’s domain. The IaaS allows cloud provide to provide resources such as memory in the form of hard disks,additional memory in the form of VMs [18] or Virtual machines,Servers,bandwidth allocation and other hardware services. The major IaaS providers are Amazon, AT&T, HP, GoGrid and many others.

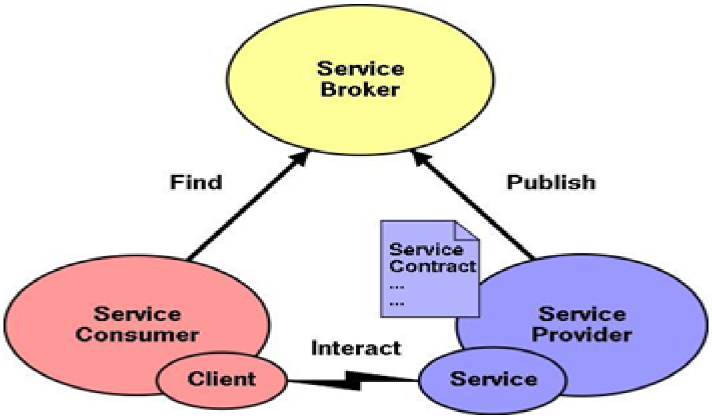


Fig. 1. Service Oriented Architecture (SOA)

- *Kisshor Kumar Sethuraman is currently pursuing masters degree program in Computer Science and Engineering in PSNA College of Engineering and Technology, Country, PH-9994154028. E-mail: kisshorgoody@gmail.com*
- *Hemalatha Thangaraj is working as an Associate Professor in the department of Computer Science and Engineering in PSNA College of Engineering and Technology, India, PH-9994980886. E-mail: hemashek@yahoo.com*

### 1.4 Multi-level Authentication

The cloud security on identifying the unique end user for better service requires authentication. The traditional authentication involves a single sign on mechanism. It involves a single user name and password passed as an input by the end user over the web UI. The authentication in the cloud involves multiple levels. There are two types of authentication to be performed in the cloud for identifying user identity.

- Broker level authentication
- Resource level authentication

#### 1.4.1 Broker-level Authentication

The cloud involved in providing the service can't be able to manage the requests sent by the end user. The cloud provider allocates the user request management, SLA negotiation and other allocation/verification tasks to a trusted third party called cloud broker. It can be either an application/web server hosted by a authority to manage a private cloud or a IT firm for managing public cloud. A commercial cloud provider can't handover large allocation/management tasks to a single cloud broker. It hands over the requests to multiple cloud brokers that are heterogonous and unknown to each other. Performing the authentication through series of cloud brokers is called broker level authentication.

#### 1.4.2 Resource-level Authentication

The end user is provided service through resources in the cloud. The resources range from data in a database, hardware validity from the manufacturer, digital certificates for data security and service tickets from the Kerberos. These resources are stored and deployed by the cloud in various geographical locations. Similar to the cloud brokers, resources are heterogynous to each other with respect to platform, environment, compatibility and capacity. Multiple resources are authenticated to provide better service and hence called resource level authentication. As both the broker and resource level authentication occurs via multiple levels through n brokers and resources, it can be called as multi level authentication.

### 1.5 Grid vs Cloud operability

The web services [19] along with virtualization [20] and SOA form the back bone of the cloud. It makes possible to enable simple resource management techniques through above two components in the clouds. The predecessor to the cloud is the grid computing [21]. The grid is a collection of resources in a server that is loosely coupled, heterogeneous and geographically distributed. The resources in grid are transferred to the cloud which can be instantiated into multiple copies by creating multiple instances of virtual machines.

### 1.6 User Request Management in Cloud

The server hosted in cloud can't handle all the resource management strategies and computations for better services to the end user. The negotiations between the cloud provider and the end user are managed by the SLA (Service Level Agreement) [22].It's a mutual agreement that defines the type of service, requirements, conditions and other factors that defines the agreement between the provider and the end user. The SLA is defined in xml as its light weight and compatible with interoperable web services. The cloud handles its partial work to the cloud broker or the trusted third party who handles the SLA negotiation and work management before transferring the jobs to the cloud. The single cloud broker can't handle million requests from end user. The real-time cloud employs multiple cloud brokers that negotiate each other to divide and handle the requests of users based on the requirements and other conditions. The set of negotiations performed by multiple cloud brokers before handling the job to the cloud is known as multi-level authentication. The job submission to the cloud via heterogeneous multiple cloud brokers faces multiple drawbacks and complexities which reduce user friendliness are overcome by our proposed work.

Our objective is to develop a user friendly portal to interconnect the private cloud with grid infrastructure, to host the computational intensive task by hiding the underlying heterogeneity with respect to authentication requirements.

The rest of this paper is organized as follows. In Section II we compare our work with the existing literature and highlight our novelties. In Section III, we present the proposed architecture. Experimentation Result is described in Section IV followed by Conclusion and Further Enhancement in Section IV.

## 2 RELATED WORK

There are various experts expressed their findings in the domain of cloud computing. Buyya et al. [23] in his work described and implemented Aneka cloud in .NET platform to provide better services to the end users. It consists of various modules to divide the computation and resource management tasks within itself. Although Aneka is best in service delivery, it can provide services only in the windows platform. Commercial cloud providers requires the platform independent solutions such as java virtual machines and Linux which cannot be provided by Aneka.The Aneka requires windows which requires purchase of valid license from Microsoft which incurs more cost. For better operations it also requires purchase of other tools such as SQL server which is not feasible.

Elisabetta Di Nitto et al. [24] in his work described various architectures for managing the dynamic brokerage of clouds in the environment. These include centralized, hierarchical, decentralized and combination of centralized and decentralized brokerage schemes. It uses multiple brokers in a predefined pattern to manage the resources in multiple clouds. The resource management strategies to make effective service delivery are not defined in the architecture. Rather the overall view of the operations or backbone of the environment is presented.

Jennifer G.Steiner et al [25] in her work describe the authentication via Kerberos server that authenticates the trusted user via ticket generation and service inputs. The steps involved in Kerberos Authentication are

- Request for TGS ticket
- Ticket for TGS
- Request for Server ticket
- Ticket for Server
- Request for service

Although Kerberos seems to be a secure protocol for authentication, it has its own drawbacks. The migration of user passwords in UNIX environment from a UNIX database such as etc/passwd or etc/shadow to a Kerberos database is a tedious task. Kerberos has only partial compatibility with the Pluggable Authentication Modules (PAM) system used by Red Hat Enterprise Linux platforms. Kerberos performs trusted authentication over an untrusted host or a network. If malicious user obtains tickets from Key Distribution Center (KDC) the Kerberos authentication is at maximum risk. When using Kerberos in a larger network all clients and servers are need to be deployed the Kerberos service. If a certain client/server doesn't possess the features of Kerberos service, it can't handle the unencrypted passwords, which makes no utilization of Kerberos. It leads to vulnerable threat under risk.

## 3 PROPOSED ARCHITECTURE

The user submits the job to the cloud portal via the web UI hosted over the Internet. The input job is a block of code or program stored in a file. It is either a single file written in High level Language or a set of files containing the libraries such as header files in C or STL (Standard Template Library) in C++ or classes packed in JAR file accessed by packages in Java.The job is in the form of executable or archived format as in the case of C, C++ or JAR file format in the case of Java code. The user submits the job over the cloud which is handled by the cloud broker as follows. The code signer upon checking if the certificate of the job is invalid or failed and has a trusted data in input, it manually signs the job with the certificate issued by CA.The job signer handles the job to the job handler that places the jobs in job queue. It also stores the job data in a xml repository. The job queue along with xml repository acts as a hash table with data in repository acts as a hash key that maps the jobs in job queue. The service when requested by user, cloud fetches the jobs from queue and pass it over to the SLA verification where the jobs are verified for service delivery that fulfills SLA as defined by the cloud provider. The job is then transferred to the job scheduler that schedules the jobs as per SLA and FCFS (First Come First Served) order. The proposed system hides the complexities of the multi-level authentication when the user's job is migrated from private cloud to public cloud. The modules involved in the proposed model are given below.

### 3.1 User

The user submits the job that consists of input files to be submitted to the cloud. The user submits the job in a web UI hosted over the internet.

### 3.2 Code Signer

The code signer acts as an interface between the user's environment and the cloud. It performs initial processing and verifications before passing the job over the cloud. The operations such as manual signing upon the trusted job data by the certificates supplied by the CA are performed.

### 3.3 Job Handler

The job handler upon verification and preprocessing forwards the submitted jobs to the job queue. The job handler manages the ports involved in the connections over the internet. It also requests additional storage resources from the cloud to be used for job management.

### 3.4 Xml Repository

The xml repository is a file system that acts as a database in storing the jobs submitted by the user. The queue is implemented in xml as its light weighted and compatible with existing web service standards by defining xml schema for storing the details of the job. Whenever the job is submitted the xml repository creates a new instance and stores it into the designated folder which holds all the jobs which are waiting to be scheduled for its execution.

### 3.5 SLA Verification

The SLA Verification modules verify the input job to be compatible with the existing SLA rules. It checks whether the job satisfies the type of application, service, duration and the subscriptions that are agreed upon both by the service provider and the consumer. If the job passes the verification successfully, it passes the job over the CTN.

### 3.6 Credential Translation Module (CTM)

Currently there are 3 different types of credentials in the internet, where the service providers align themselves with more than one provider. They are single user name and password, PKI and Kerberos. The third party authentication services used in the proposed architecture are PKI and Kerberos. When the user's SLA cannot be met with the private cloud, the proposed model migrates the job from the private cloud to the public cloud. While migration, the requirements of the public cloud has to be satisfied else the job migration will fail. To handle this requirement, CTM is implemented in the proposed system. The role of CTM is that it requests the official certificate authority to Kerberos server, to acquire a valid certificate and a service ticket.

The sample xml format that describes the job submitted to the cloud is presented as follows.

```
<Jobs>
<Job-Description>
<JobID>      </JobID>
<Name>       </Name>
<Affiliation>
<Organization>    </Organization>
<Address>      </Address>
<City>         </City>
<Pincode>    </Pincode>
</Affiliation>
<mailto>        </mailto>
<JobSpec>
<Source-Language> </Source-Language>
<No-of-files><No-of-files>
<No-of-input><No-of-input>
<Other>      </Other>
</JobSpec>
<SLA>
<JobType>   </JobType> {SIMD, MIMD}
<E-D-O-C>    </E-D-O-C>{Expected Date of Completion}
<Secrecy>       </Secrecy> {No, Moderate, High}
<Cost-Incurred>    </Cost-Incurred>
</SLA>
</Job-Description>
</Jobs>
```

The type of job submitted in the cloud comprised of two types. They are SIMD or Single instruction multiple data and MIMD or Multiple instruction multiple data.

### 3.7 Credential Database

The credential database is a xml repository of file system containing the user passwords and other essential information required by the CTM. The database is queried by CTM for authenticating the users upon with the decryption of job request is performed.

### 3.8 CA

The CA is the Certification authority involved in providing the certificate for the client requests on demand. The CA is a trusted third party operates in public key infrastructure.

### 3.9 Kerberos Server

The Kerberos is a server used to authenticate the users in the cloud. The Kerberos generates a ticket or unique value token which is assigned to the users in the cloud. The ticket is used throughout the session. It is stored in Kerberos database which is also used for the later sessions.

### 3.10 Job Scheduler

The job scheduler schedules the jobs submitted to the private cloud as per SLA and FCFS (First Come First Serve) manner. The job scheduler schedules in `such a way so that deadlocks and starvations are not observed in the cloud.

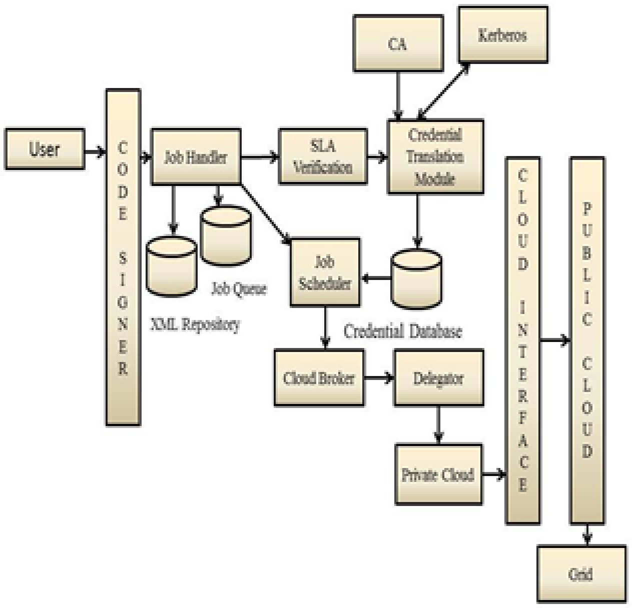


Fig. 2. User Negotiation/Job Migration in proposed Architecture

### 3.11 Private Cloud

The private cloud is a cloud environment hosted to service the end user   on behalf of the public cloud. If the private cloud cannot handle the job request it passes the job over to the public cloud for final computation. The private cloud is connected to the grid for providing the resources that are needed for the management and computation

### 3.12 Grid

The grid is a non-virtualized environment with abundant collection of resources and services. If private cloud cannot manage by itself regards the resource management and computation, it request the grid. The grid grants the resources to the private cloud in the form of grid services. The private cloud consumes the resources from grid through a web service.

### 3.13 Cloud Broker

The cloud broker upon completion of request validation and SLA negotiation performs the post processing of the submitted job request by the user. The tasks involved are the final verification and checking the public cloud availability status and packing the data in a format compatible for public cloud. If job is serviced to the end user from private cloud, the operation of cloud broker is not essential.

### 3.14 Delegator

The delegator involves in the delegation operations between the private and public cloud. The only task of the delegator is to pass the data to the public cloud through the cloud interface.

### 3.15 Cloud Interface

The cloud interface is the middleware. It performs the tasks as required by the public cloud so as to be compatible with it for better service delivery to the end user.

### 3.16 Public Cloud

The cloud environment hosted to the public customers built on the grid. The public cloud has abundant resources that can manage any kind of job requests that is compatible with the predefined SLA.

## 4 JOB DESCRIPTION IN THE CLOUD

The job is submitted as input through the web UI in the Internet. In the experimentation the job is tested against the local server built with the cloud. The sample job taken for the experimentation is a set of classes written in java.

The sample CRL format used to revoke the invalid certificates in certificate chain authentication is performed as follows.

Listing. 1. Sample CRL Format

```
-----BEGIN X509 CRL-----
MIIDQTCCAikwDQYJKoZIhvcNAQEF-
BQAwdTETMBEGCgmSJomT8ixkARkWA25ldDESMBA
GCgmSJomT
8ixkARkWAmVzMSAwHgYDVQQLExdDZXJ0aWZpY2
F0ZSBBdXRob3JpdGllczEZMBcGA1UECxMQRE9F
IFNjaWVuY2UgR3JpZDENMAsGA1UEAxMEcGtpMRc
NMDIwNTA5MjAwMjM2WhcNMDIwNjA4MjAwMjM2
WjCCAYEwEgIBXBcNMDI-
wMzE5MTcyNjI4WjASAgFbFw0wMjAzMTkwMDA0ND
JaMBICASUXDTAyMDIx
MjIwMTkzMVowEwICAK8XDTAyMDUwNzIzMzAxNF
owEgIBUBcNMDIwMzEyMjAzMjM4WjATAgIArhcN
MDIwN-
TA3MjMyMjM5WjASAgFPFw0wMjAzMjcxNDQxMTJa
MBICAR4XDTAyMDIwNDIxNTc1MVowEgIB
SRcNMDI-
wMzE0MjI0OTQzWjASAgF2Fw0wMjA0MDgxOTMwM
zNaMBMCAgChFw0wMjA0MzAyMDQwMjVa
MBICARMXDTAyMDEyOTI-
wMTQwOFowEwICAKAXDTAyMDQzMDI-
wNDAyNVowEgIBEhcNMDIwMTI5MTk1
NDIzWjATAgIAmhcNMDIwN-
TA5MjAwMjM2WjASAgENFw0wMjAxMjgyMzE0NDZa
MBICATwXDTAyMDMw
NTE5NDExM1owEgIBOBcNMDIwMzE5MjMxOTI5WjA
SAgE3Fw0wMjAzMDgyMDE4NDhaMA0GCSqGSIb3
DQEBBQUAA4IBAQBWt6fD7AsvcmuTsSx9GWPbFIR3C
CG7yIQUDiBSOOJi3guKh4tLqiCIQeIkGbMp
7XeEk+5oKRcuwZdMQpseKO6GYVVACEkqDczk2L62k
MiE/7cTbXryKJRg87fGF6MC+uXcU0bTCtpC
tByQ82yaKuPw/C+JYOurMzhyc8ZSxzJxz7WKYEiCzig5
ZiVBvqO7ksSJGUy08ABWSmPBIL3u3CG6
Lz7aV/GiME20eXQRW++9256NhkT2P2IYETa5c/UFWl
wyAFLq23C5u/R5e1sqpK5BcmAPqId957b9
+g7I9/ZsXj1ZRNlEPZ3wu6XHwVpC2TSLG95B+rl0TDN
zxEKho1Rc
-----END X509 CRL-----
```

## 5 EXPERIMENTATION RESULT

The UI for the job submission to the cloud is provided by an html web page that is authenticated by a web service from behind. The private cloud is hosted in a Linux environment running Fedora OS.The private cloud is a nimbus cloud client connected to a Globus tool kit. The Globus tool kit acts as a grid providing services to the private cloud. The CA is the Indian Certification Authority that provides certificates for the submitted jobs.

### 5.1 Hardware Requirements

The Globus tool kit installation is tested on the hardware platform comprised of Core i3 Intel Processor,4 GB RAM, CentOS 5 and Hard Disk space of 100 GB .

### 5.2 Software Requirements

The Software requirements for the Globus tool kit installation comprised of J2SE 1.5, Apache Ant 1.6.5. The php installation requires a wamp server version 2.2 with Apache 2.2.22 and php 5.4.3. It also requires Visual C++ 2010 Redistributable.

### 5.3 Sequence Diagrams

The sequence diagram that describes the proposed architecture operation is presented as follows. The following diagram shows the interactions involved between the user and xml repository.

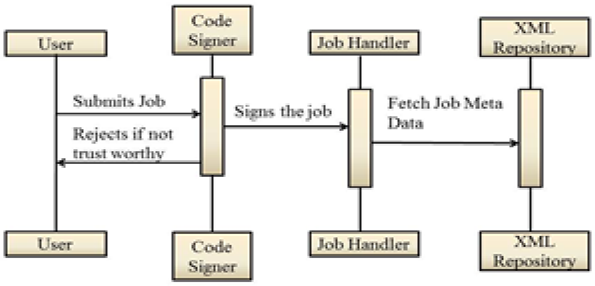


Fig. 3. Interactions between user and xml repository

The following diagram shows the interactions between the SLA verification and user database.

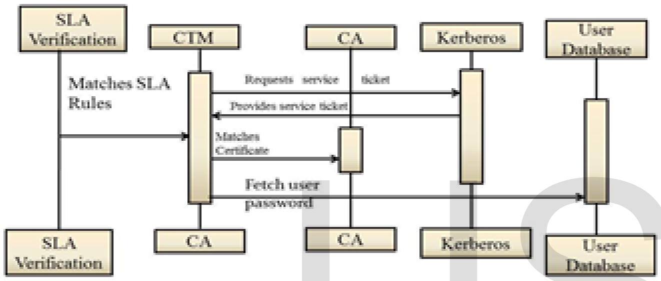


Fig. 4. Interactions between SLA Verification and User Database

The following diagram shows the interaction between cloud interface and the public cloud.

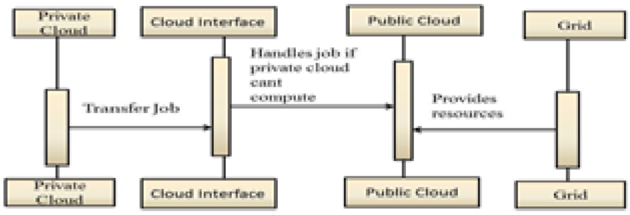
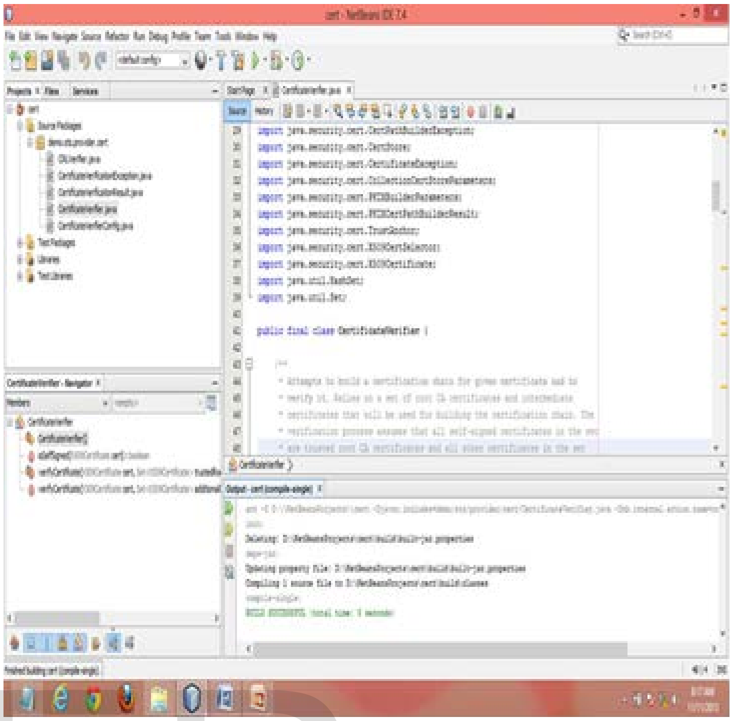


Fig. 5. Interactions between Private Cloud and Grid

## 5.4 EXPERIMENTATION SNAPSHOTS

The snapshots taken during the work experimentation are presented below. The following snapshot describes the compilation of CertificateVerifier class.

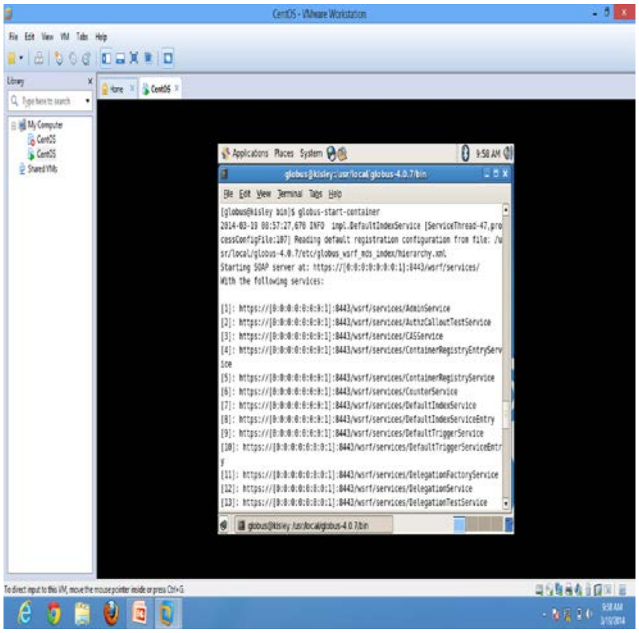


```
import java.security.cert.CertPathBuilderException;
import java.security.cert.CertStore;
import java.security.cert.CertificateException;
import java.security.cert.CollectionCertStoreParameters;
import java.security.cert.PKIXBuilderParameters;
import java.security.cert.PKIXCertPathBuilderResult;
import java.security.cert.TrustAnchor;
import java.security.cert.X509CertSelector;
import java.security.cert.X509Certificate;
import java.util.HashSet;
import java.util.Set;

public final class CertificateVerifier {

    /**
     * Attempts to build a certification chain for given certificate and to
     * verify it. Relies on a set of root CA certificates and intermediate
     * certificates that will be used for building the certification chain. The
     * verification process assumes that all self-signed certificates in the set
     * are trusted root CA certificates and all other certificates in the set
```

```
ant -f D:\\NetBeansProjects\\cert -Djavac.includes=demo/sts/provider/cert/CertificateVerifier.java -Dnb.internal.action.name=
init:
Deleting: D:\NetBeansProjects\cert\build\built-jar.properties
deps-jar:
Updating property file: D:\NetBeansProjects\cert\build\built-jar.properties
Compiling 1 source file to D:\NetBeansProjects\cert\build\classes
compile-single:
BUILD SUCCESSFUL (total time: 0 seconds)
```

Fig. 6. Compilatoin of CertificateVerifier class.

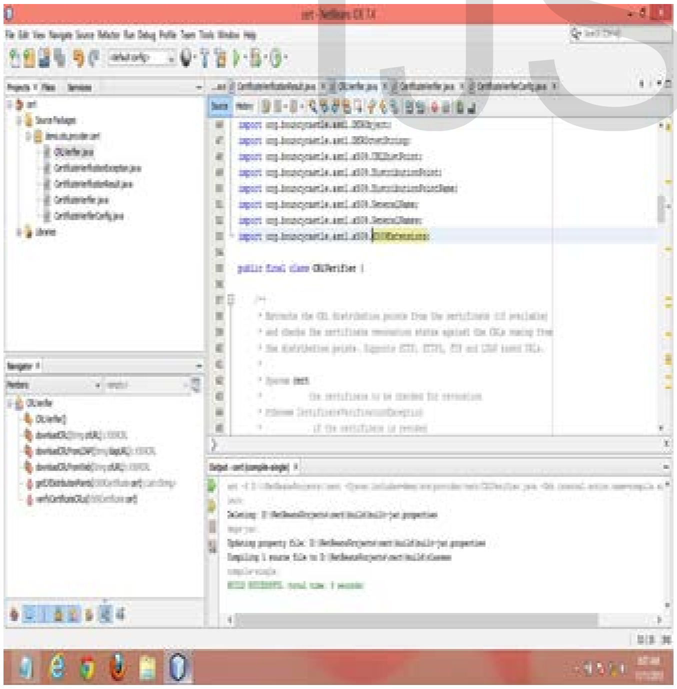
The following snapshot describes the starting of Globus container on CentOS 5.


CentOS - VMware Workstation
File Edit View VM Tabs Help
Applications Places System
9:58 AM
globus@kisley:/usr/local/globus-4.0.7/bin
File Edit View Terminal Tabs Help

```
[globus@kisley bin]$ globus-start-container
2014-03-19 08:57:27,670 INFO  impl.DefaultIndexService [ServiceThread-47,pro
cessConfigFile:107] Reading default registration configuration from file: /u
sr/local/globus-4.0.7/etc/globus_wsrf_mds_index/hierarchy.xml
Starting SOAP server at: https://[0:0:0:0:0:0:0:1]:8443/wsrf/services/
With the following services:

[1]: https://[0:0:0:0:0:0:0:1]:8443/wsrf/services/AdminService
[2]: https://[0:0:0:0:0:0:0:1]:8443/wsrf/services/AuthzCalloutTestService
[3]: https://[0:0:0:0:0:0:0:1]:8443/wsrf/services/CASService
[4]: https://[0:0:0:0:0:0:0:1]:8443/wsrf/services/ContainerRegistryEntryServ
ice
[5]: https://[0:0:0:0:0:0:0:1]:8443/wsrf/services/ContainerRegistryService
[6]: https://[0:0:0:0:0:0:0:1]:8443/wsrf/services/CounterService
[7]: https://[0:0:0:0:0:0:0:1]:8443/wsrf/services/DefaultIndexService
[8]: https://[0:0:0:0:0:0:0:1]:8443/wsrf/services/DefaultIndexServiceEntry
[9]: https://[0:0:0:0:0:0:0:1]:8443/wsrf/services/DefaultTriggerService
[10]: https://[0:0:0:0:0:0:0:1]:8443/wsrf/services/DefaultTriggerServiceEntr
y
[11]: https://[0:0:0:0:0:0:0:1]:8443/wsrf/services/DelegationFactoryService
[12]: https://[0:0:0:0:0:0:0:1]:8443/wsrf/services/DelegationService
[13]: https://[0:0:0:0:0:0:0:1]:8443/wsrf/services/DelegationTestService
```

Fig. 7. Starting of Globus Container in CenTOS 5

The following snapshot describes the compilation of job classes for certificate verification in Netbeans.

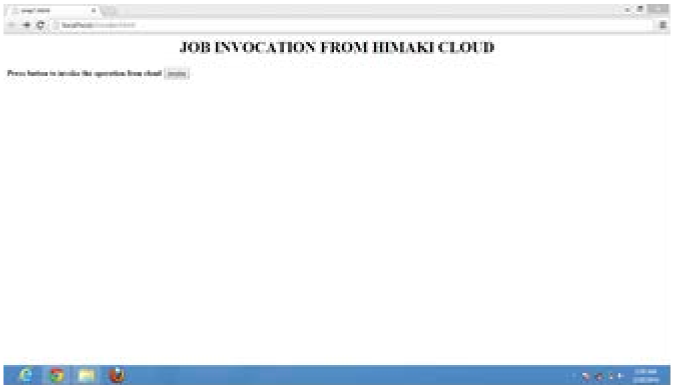


```
import org.bouncycastle.asn1.[illegible];
import org.bouncycastle.asn1.[illegible];
import org.bouncycastle.asn1.x509.[illegible];
import org.bouncycastle.asn1.x509.[illegible];
import org.bouncycastle.asn1.x509.[illegible];
import org.bouncycastle.asn1.x509.[illegible];
import org.bouncycastle.asn1.x509.[illegible];
import org.bouncycastle.asn1.x509.[illegible];

public final class CRLVerifier {

    /**
     * [illegible]
     * [illegible]
     * [illegible]
     *
     * [illegible]
     *         [illegible]
     * [illegible]
     *         [illegible]
```

```
[illegible]
BUILD SUCCESSFUL (total time: 1 second)
```

Fig. 8. Compilation of CRLVerifier class

The following snapshot describes the invocation of certificate verifier classes from PHP web client.



JOB INVOCATION FROM HIMAKI CLOUD

Press button to invoke the operation from cloud invoke

Fig. 9. Invocation of job from PHP Web Client

The following snapshot describes the result of invocation from PHP web client.

[illegible] BUILD SUCCESSFUL [illegible]

Fig. 10. Result of invoking job from PHP Web Client

## 6 CONCLUSION AND FURTHER ENHANCEMENT

The proposed work provides the architecture that hides the complexities of the multilevel authentication. It provides better user friendliness to the end user by providing better resource management and the credential security in the cloud. The architecture acts as a reusable framework that can be adopted in existing commercial cloud platforms by the providers. The future enhancement involves upgrading the architecture in more refined manner for performance. Scalability allows users to submit code and enhancements among the open source community for better performance.

## REFERENCES

[1] Peter Mell, Timothy Grance, "The NIST Definition of Cloud Computing- Recommendations of the National Institute of Standards and Technology", September 2011.

[2] Mahmoud Mohammed AbdAllah, Waseim Hashem Mahjoub, "A Quick Introduction to SOA", Software Engineering Competence Center 2013.

[3] Vijaykumar Krishnaswamy et al. "Efficient Implementation of Java Remote Method Invocation (RMI)" ,Proceedings of the 4th USENIX Conference on Object-Oriented Technologies and Systems (OOTS, Santa Fe, New Mexico, April 27-30, 1997-98.

[4] Michael Stadler et al. "Application of the Software as a Service Model to the Control of Complex Building Systems",ECEEE 2011 Summer Study 6-11 June 2011, Belambra Presqu'ile de Glens, France.

[5] Boniface, M et al. "Platform-as-a-Service Architecture for Real-Time Quality of Service Management in Clouds", Internet and Web Applications and Services (ICIW), May 2010, pp 155-160.

[6] Oberle, K, Stein, M, Voith, T, Gallizo, G, Kubert, R, "The network aspect of Infrastructure-as-a-Service",Intelligence in Next Generation Networks (ICIN), Oct 2010, pp 1-6.

[7] Amazon Web Services, aws.amazon.com, "Amazon Web Services (AWS) delivers a set of services that together form a reliable, scalable, and inexpensive computing platform "in the cloud".

[8] Amazon Web Services, aws.amazon.com, "Amazon Web Services (AWS) delivers a set of services that together form a reliable, scalable, and inexpensive computing platform "in the cloud".

[9] Salesforce, www.salesforce.com, "Customer Relationship Management (CRM)"

[10] Akamai, www.akamai.com, A Cloud platform for content delivery and Web security.

[11] Apprenda, www.apprenda.com, A Cloud provider for Platform as a Service.

[12] JBOSS Server, www.jboss.org, "Open Source Application Server acquired by RedHat".

[13] GlassFish Server, https://glassfish.java.net, "Open source Application Server developed by Oracle".

[14] MYSQL, www.mysql.com, "Open source Relational Database Management System".

[15] MongoDB, www.mongodb.org, "An open-source document database and leading NOSQL database".

[16] AppScale, www.appscale.com, "An open-source emulation of the award winning Google App Engine platforn".

[17] Google App Engine, https://appengine.google.com, "A Platform as a Service cloud computing platform for developing and hosting web applications in Google-managed data centers.

[18] James E Smith, Ravi Nair, "Virtual Machines", Elsevier, 2005.

[19] James A Chappell, "Java Web Services", O'Reilly Media 2002.

[20] Jun Huang, Yanbing Liu, Qiang Duan, "Service provisioning in virtualization-based Cloud computing: Modeling and optimization", Global Communications Conference (GLOBECOM),IEEE 2012,Dec-2012, pp 1710 – 1715.

[21] Rajan, A,Rawat, A, Verma, R.K, "Virtual Computing Grid Using Resource Pooling",Information Technology,2008,ICIT'08,Bhubaneshwar,Dec 2008,pp 59-64.

[22] Zhang Shu, Song Meina, "An architecture design of life cycle based SLA management",Advanced Communication Technology (ICACT),Feb 2010, pp 1351-1355.

[23] Rajkumar Buyyaa, Chee Shin Yeo, Srikumar Venugopal, James Broberg, Ivona Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," Manjrasoft Pty Ltd, Melbourne, Australia, ScienceDirectVolume 25,Issue 6, June 2009, pp 599–616.

[24] Elisabetta Di Nitto, Nicolo Maria Calcavecchia, Antonio Celesti, "Understanding Decentralized and Dynamic Brokerage in Federated Cloud Environments",Achieving Federated And Self-Manageable Cloud Infrastructures: Theory and Practice : May, 2012.

[25] Jennifer G. Steiner, Clifford Neuman, Jeffrey I.. Schiller, "Kerberos: An Authentication Service for Open Network Systems" USENIX Conference proceedings", Citations: 623 – 12 self.

[26] M. Noureddine, R. Bashroush, "An authentication model towards cloud federation in the enterprise" Microsoft Corporation, Seattle 2012, Journal of Systems and Software , Volume 86 Issue 9, September, 2013, Elsevier Science Inc. New York, NY,USA, pp 2269-2275.

[27] MoussaOuedraogo, Haralambo Mouratidis, " Selecting a Cloud Service Provider in the age of cybercrime" Service Science and Innovation Department, Public Research Center Henri Tudor, 29, Avenue John F. Kennedy, Luxembourg 1855, Luxembourg Luxembourg School of Architecture, Computing and Engineering, University of East London, United Kingdom,Elsevier-2013.

[28] S. Subashini, V. Kavitha, "A survey on security issues in service delivey models of cloud computing" Kavitha Anna University Tirunelveli,Tirunelveli,TN 627007,India,Elsevier-2010.

[29] BahmanJavadi, Rajkumar Buyya, Parimala Thulasiraman, "Enhancing Performance of Failure-prone Clusters by Adaptive Provisioning of Cloud Resources", The Journal of SuperComputing,February 2013, Volume 63, Issue 2.

[30] Attila CsabaMarosi, Gabor Kecskemeti, Attila Kertesz, Peter Kacsuk "FCM: an Architecture for Integrating IaaS Cloud Systems" , CLOUD COMPUTING 2011 : The Second International Conference on Cloud Computing, GRIDs, and Virtualization.

[31] Johan Tordsson, Ruben S. Montero, Rafael Moreno-Vozmediano, Ignacio M. Llorente, " Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers" ,Future Generation Computer Systems,Volume 28,Issue 2,February,2012,pp 358-367.

[32] Bouncycastle, www.bouncycastle.org, "A third party API to perform encryption and other cryptographic operations".