Open access Journal International Journal of Emerging Trends in Science and Technology

User Compatibility Checking Model For Cloud

Authors
V. Krithika, I. Soundarya, Mr. A. Sandanakaruppan
SSN Engineering College, Chennai, TamilNadu
Email: *venkatkrithi@gmail.com, ss.sowmya2011@gmail.com, sandanakaruppan@ssn.edu.in*

ABSTRACT
Cloud computing is the fastest growing technology. It is used as a metaphor for the "Internet". It is a type of "Internet-based computing" providing different services such as servers, storage and applicants. This paper deals with checking the compatibility of the Web Application with the Cloud Service composition and to build a Virtual Machine (Instance) for Deployment and various Preferences of Users. This paper primary focus is on creating virtual instance using the OpenStack cloud provider. Setting up such a complex combination of appliances in traditional hosting environments is costly and error prone. Virtual appliances provide an best solution for this problem. Previous work have been done to create virtual instance but to our we are the one among the first to address real time process of virtual Instance in Cloud environment.

Keywords: *Cloud computing, Virtual Instance, OpenStack Cloud provider, Virtual appliances, cloud service composition*

1. INTRODUCTION

Cloud Computing is becoming one of the next emerging IT industry technologies. With cloud, you no need to maintain the hardware and software, it is utility where you have to pay for what you need. To deliver their required solution, the application service provider can either make use platform as a service(PaaS) such as GOOLE ENGINE to build their hosting environment but PaaS as some restrictions on the programming language, development Platform. Such restrictions can be overcome by using IaaS .It encourage service providers to build their own platforms using Amazon EC2 or GoGrid.

Service deployment remain biggest challenges in cloud, If we consider the deployment requirements of a web application service provider, it will include security devices (e.g. firewall), web servers, application servers, database servers, and storage devices. Setting up such a complex combination of appliances in traditional hosting environments is costly and error prone. Virtual appliances provide an best solution for this problem. Our system uses OpenStack Cloud provider which is the best to build. Virtual Instance. They are built and configured with a necessary operating system and software packages to meet software requirements of a user.

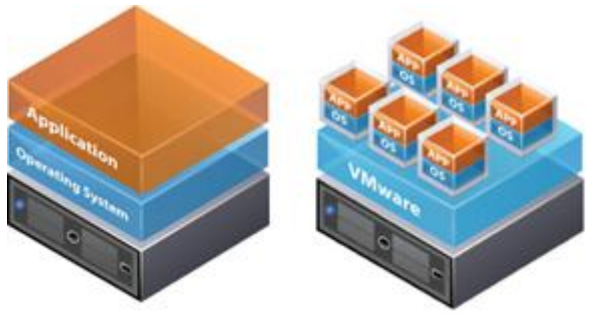


Figure 1: Traditional and Virtual Architecture

Virtual appliances, a set of virtual machines which includes operating systems, pre-built, preconfigu-red, ready-to-run applications are to solve the complexities of service deployment .Virtual appliances are proved to provide a better service deployment mechanism. It also reduce the cost of deploying and maintaining the software.

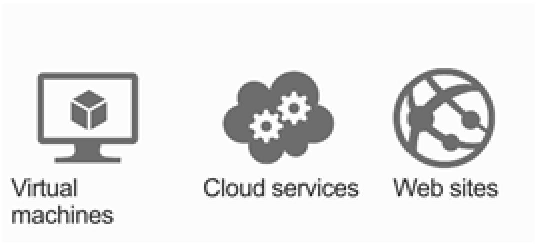


Figure 2: Cloud Service Composition

It is difficult to choose the best composition because none of the service provider provides ranking system to user For example, if an appliance format is Open Virtualization Format (OVF) it cannot currently be deployed on Amazon EC2.All the choices of individual cannot simply put in the traditional hosting environment . Unskilled users cannot handles all these complexity and cost.

In this paper, to minimize the process of selecting the best preferences an approach is introduced in our system. The non-expert user can make use of Image format compatibility issue to deploy their service. For this purpose we have build two default Instance in the cloud service repository and advertisement is also provided for those Instance.

A Cloud service repository is used to provide all the services automatically, it allows the user to select their user preferences according to their requirements. Ranking system is provided to minimize the complexity in choosing the user's preferences.

2. EXISTING SYSTEM

In this section, we are going to see the board overview on existing work. The main difficulty lies in setting up complex combination of appliances in the traditional hosting environment , it is also difficult to choose the best instance type from the cloud pool.To solve this problem virtual appliance is used.But this system does not provide effective use of virtual appliances. The non-expert user are not aware of cloud service composition. They were not able choose the web dependence for their task. Without the help of expert knowledge they were not enable to select the Instance .Because of this Deployed application may be misconfigured. user preference is difficult .There is no ranking system provided. Combination of appliance, chooses by the individual cannot simply put in the hosting environment, because it may not be capable in the hosting environment.

3. PROPOSED WORK

Proposed system uses OpenStack Cloud provider which is the best to build Virtual Instance. To build virtual instance, 4GB of storage is enough. Our system helps non-expert users with limited or no knowledge on legal and image format compatibility issues to deploy their services. User can build their own instance based on his/her requirements. Web applications dependencies are available, the only thing is user has to select their web application dependencies based on his/her preferences. User can deploy application. Real time process can be viewed in OpenStack Dashboard.

4. PROPOSED SYSTEM ARCHITECTURE

Proposed system provides whole service deployment life cycle for users. The main aim of this system is to provide service for non-expert user, ranking system for selecting user preferences. The major components of proposed system are listed below.

Cloud Service Repository: It is an cloud storage where the default instances are stored. Advertisement for the default instances is also provided. For example, an advertisement of a computing instance can contain descriptions of its features, costs, and the validity time of the advertisements, Then the user have to choose their instance depend upon their requirements.

Create new Instance: The user can create their own instance by giving their user preference, which includes OS, Server, Database, RAM and Budget. In this system we are providing the ranking system to select the preferences. The configuration detail is also provided to the users.

Compatibility Checking: It checks the service provided user with the cloud service composition, if the checking is done successfully, it gets the detail about the user application, it compares the complexity of the application with the RAM size provided by the user, if it is enough to run the application, then the user can buy the instance.

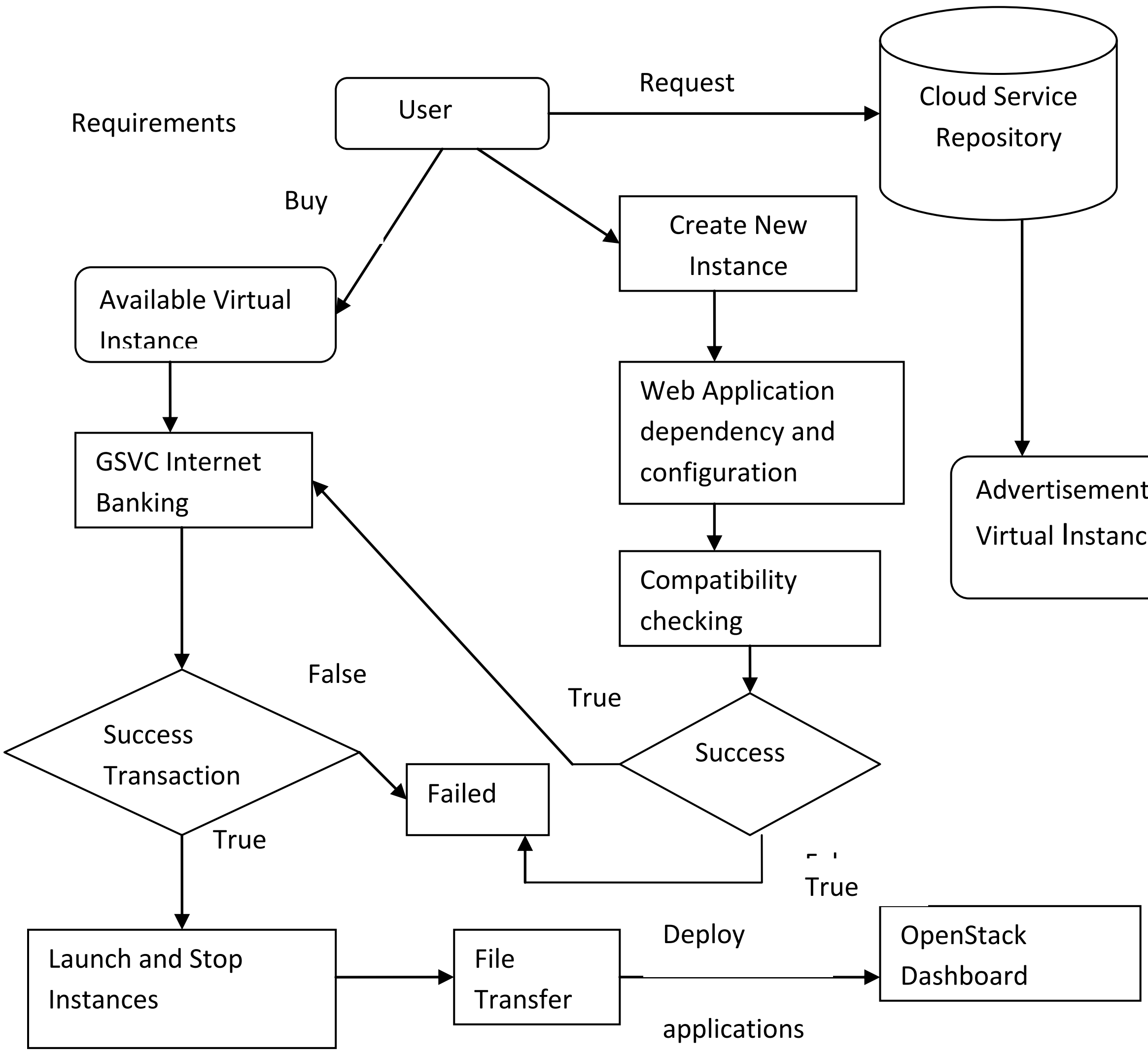


Figure 3: Architecture for User Compactibibity Checking Model For Cloud

5. PROPOSED SYSTEM METHODS

5.1. CLOUD SERVICE COMPOSITION

The user has to register his/her details. The server in turn stores the user information in its database. User can modify their information and updated information stored in database. Advertisement of two default instances of service provider is available in this module. User buys instances based on their requirements. The user checks their bought instances in launching an Instance module. User owned instances details are stored in database.

5.2. BUILD AND CONFIGUREINSTANCE

User can create new instance based on his/her requirements. In this module, user can select their Web application dependency which includes Operating Systems, RAM, and Database etc. Compatibility checking of cloud services is done in this module. If the cloud services are compatible, user redirects into GSVC Internet Banking. After successful transaction, the amount will be debited from his/her account. The user checks their newly created instances in launching an Instance module.

5.3. LAUNCH AND DEPLOY INSTANCES

In this module, instances owned by the user and running instances are displayed. Now user can launch any instances. Instance launched using jclouds, based on specified RAM and instance name. Start and end execution time of the instance will be noticed. Instance will be created in OpenStack Dashboard. A separate IP will be created for each instance. Now user can view their virtual instance in OpenStack. Using putty, user connects to IP assigned for the virtual instance. Now user connects into Virtual Instance. Giving file transfer command, user transfer their web application to Virtual instance. User can also transfer data base files to Virtual Instance. Now user can deploy their web applications in Virtual Instance. Real time process is shown in Open Stack Dashboard.

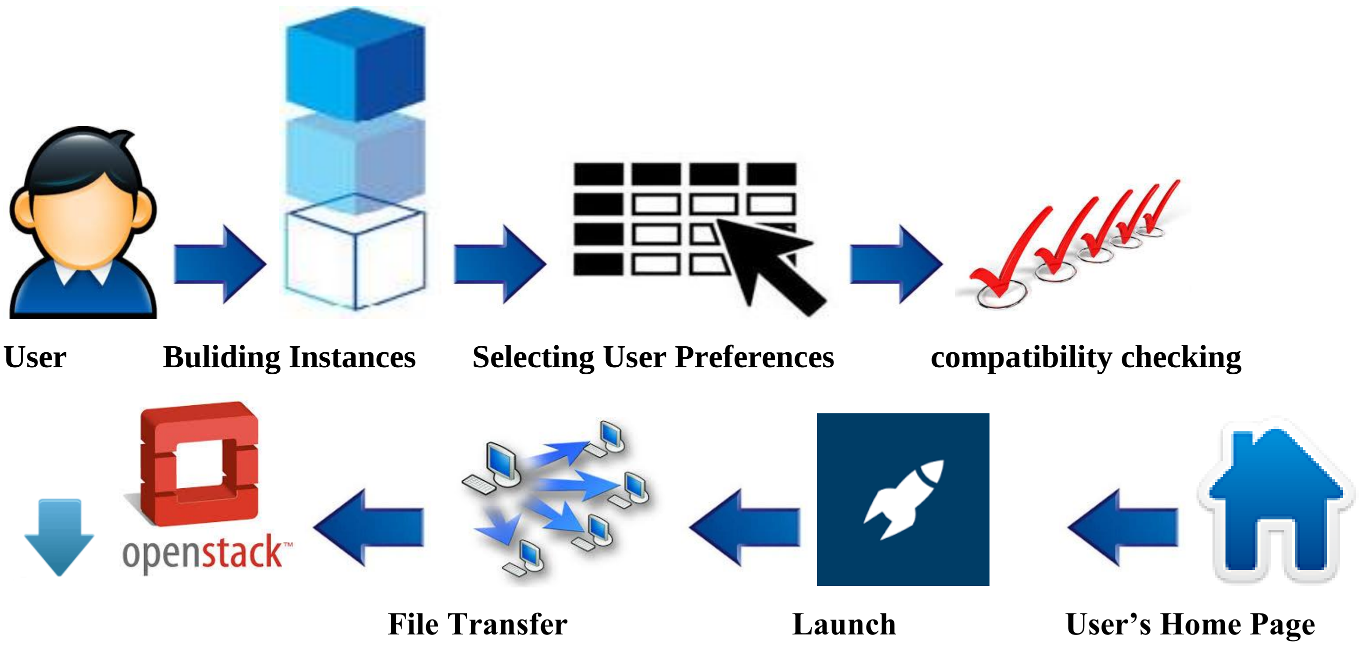
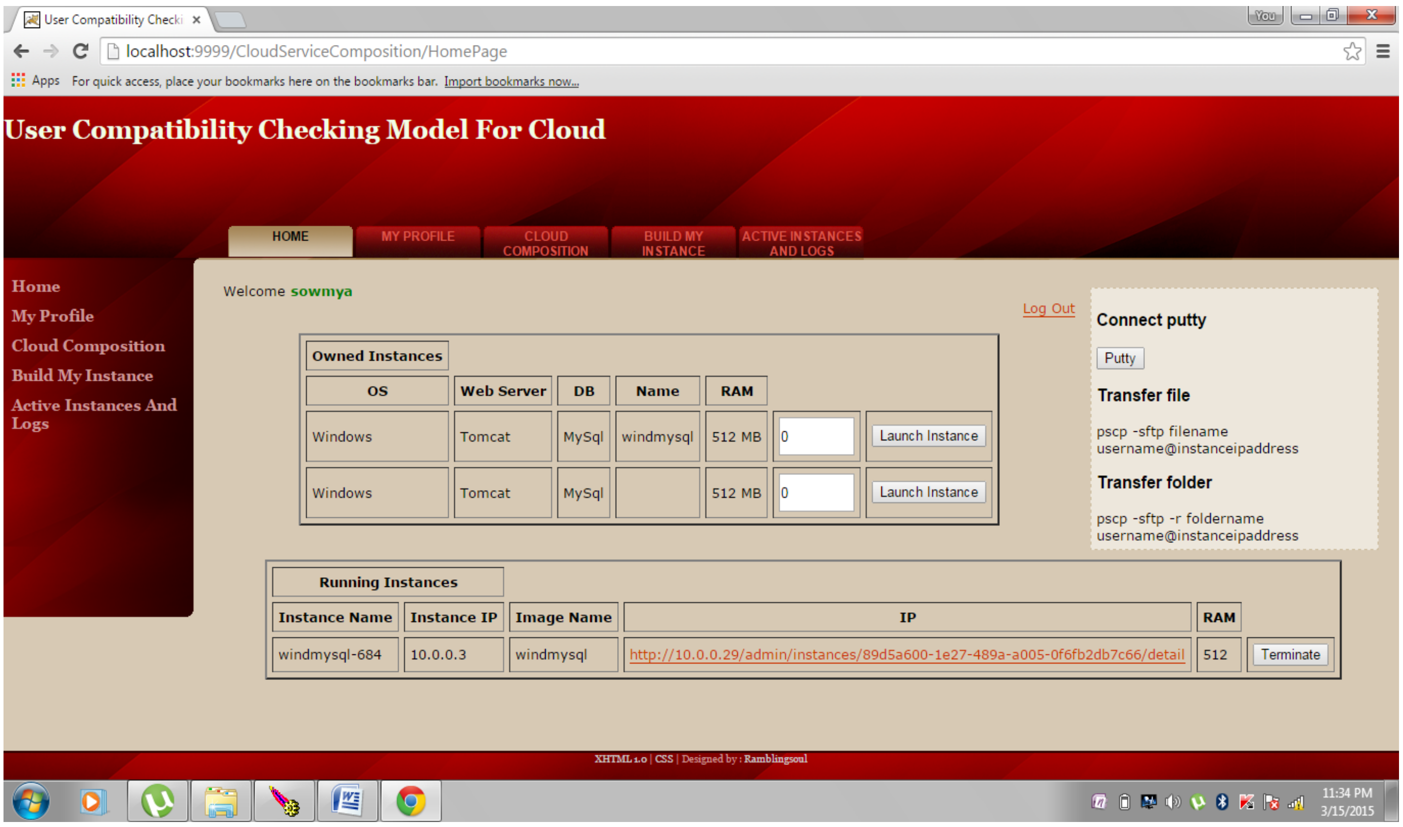


Figure 4: Configure And Deploy Model In Cloud

6. SAMPLE SCREENSHOTS


User Compatibility Checki
localhost:9999/CloudServiceComposition/HomePage
Apps For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
HOME
MY PROFILE
CLOUD COMPOSITION
BUILD MY INSTANCE
ACTIVE INSTANCES AND LOGS
Home
My Profile
Cloud Composition
Build My Instance
Active Instances And Logs
Log Out
Putty
Launch Instance
Terminate
XHTML 1.0 | CSS | Designed by : Ramblingsoul
11:34 PM
3/15/2015

User Compatibility Checking Model For Cloud

Welcome sowmya

Owned Instances

OS	Web Server	DB	Name	RAM		
Windows	Tomcat	MySql	windmysql	512 MB	0	Launch Instance
Windows	Tomcat	MySql		512 MB	0	Launch Instance

Connect putty

Transfer file

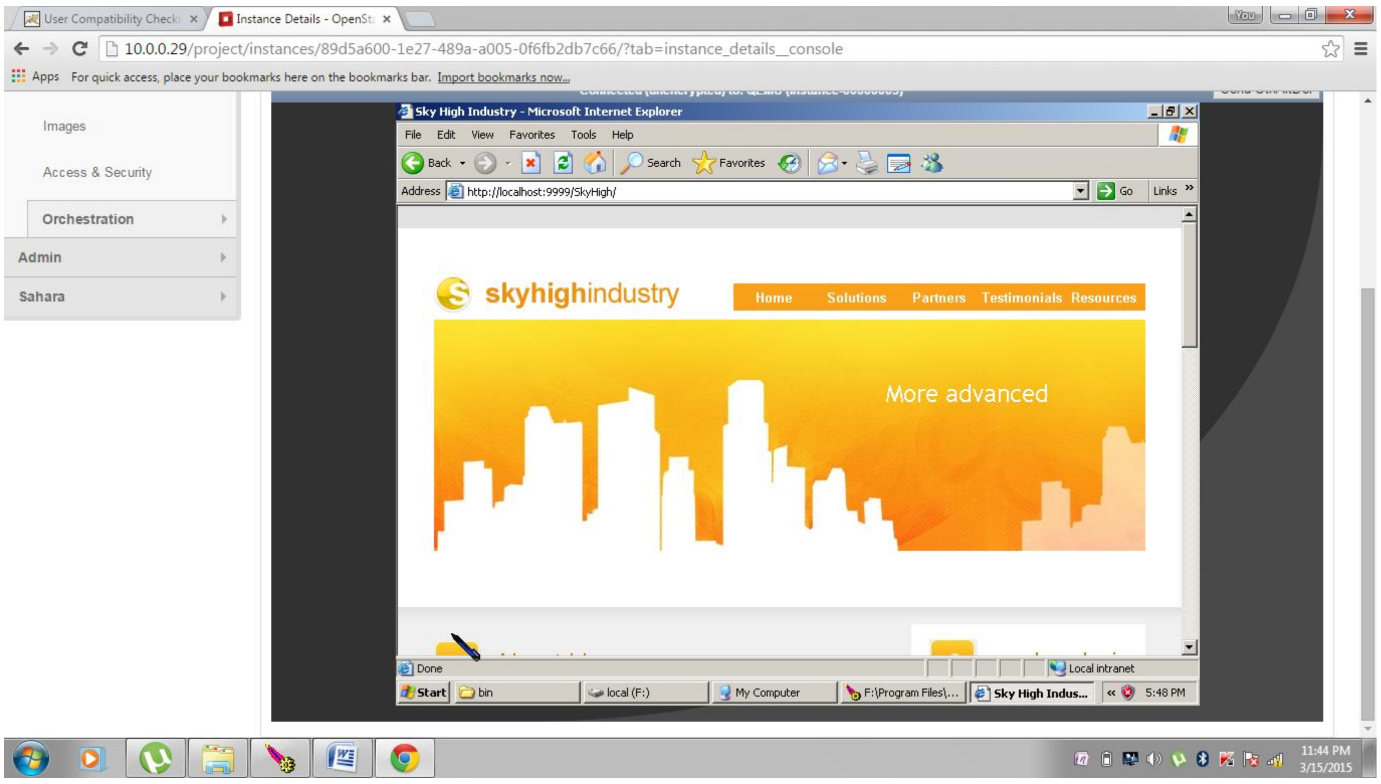
pscp -sftp filename username@instanceipaddress

Transfer folder

pscp -sftp -r foldername username@instanceipaddress

Running Instances

Instance Name	Instance IP	Image Name	IP	RAM	
windmysql-684	10.0.0.3	windmysql	http://10.0.0.29/admin/instances/89d5a600-1e27-489a-a005-0f6fb2db7c66/detail	512	Terminate


User Compatibility Check
Instance Details - OpenSt
10.0.0.29/project/instances/89d5a600-1e27-489a-a005-0f6fb2db7c66/?tab=instance_details__console
Apps For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
Images
Access & Security
Orchestration
Admin
Sahara
Sky High Industry - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:9999/SkyHigh/
Go Links
skyhighindustry
Home Solutions Partners Testimonials Resources
More advanced
Done
Local intranet
Start bin local (F:) My Computer F:\Program Files\... Sky High Indus...
5:48 PM
11:44 PM
3/15/2015

7. CONCLUSION

In this paper, we proposed a method of compatibility checking algorithm to check the compatibility of web service with the cloud service composition. This algorithm allow the user to select their preferences depends upon the requirements. In addition, it will display the real time process of the virtual instance. In future, the deployment time of the process can be reduced. The number of real time servers can also be increased.

REFERENCES

1. C. Sun, L. He, Q. Wang, and R. Willenborg, “Simplifying service deployment with virtual appliances,” in Proceedings of the IEEE International Conference on Services Computing (SCC),2008.
2. J. De Bruijn, H. Lausen, A. Polleres, and D. Fensel, “The web service modeling language wsml: An overview,” in Proceedings of the 3rd European conference on The Semantic Web: research and applications, 2006.
3. A. Dastjerdi, S. Tabatabaei, and R. Buyya, “An effective architecture for automated appliance management system applying ontology-based cloud discovery,” in Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, 2010.
4. A. Dastjerdi and R. Buyya, “An autonomous reliability-aware negotiation strategy for cloud computing environments,” in Proceedings of 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. IEEE, 2012.
5. DMTF, “Open virtualization format,” http://www.dmtf.org/standards/ovf.
6. A. Dastjerdi, S. Tabatabaei, and R. Buyya, “A dependencyaware ontology-based approach for deploying service level agreement monitoring services in cloud,” Software: Practice and Experience, vol. 42, no. 4, pp. 501–518, 2011.
7. I. Foster, Y. Zhao, I. Raicu, and S. Lu, “Cloud computing and grid computing 360-degree compared,” in Proceedings of the Grid Computing Environments Workshop (GCE), 2008.
8. J. Kopeck` y, D. Roman, T. Vitvar, M. Moran, and A. Mocan, “Wsmo grounding. wsmo working draft v0. 1, 2007.”
9. D. Lambert, N. Benn, and J. Domingue, “Integrating heterogeneous web service styles with flexible semantic web services groundings,” in Proceedings of the 1st International Future Enterprise Systems Workshop, 2010.
10. R. Barth and C. Smith, “International regulation of encryption: technology will drive policy,” Borders in Cyberspace: Information Policy and Global Information Infrastructure, pp. 283–299, 1999.