A policy-based cloud broker for the VCL platform

Karuna P. Joshi*, Yelena Yesha,
Tim Finin and Yaacov Yesha

Computer Science and Electrical Engineering Department,
University of Maryland, Baltimore County,
Baltimore, MD 21250, USA
E-mail: kjoshi1@umbc.edu
E-mail: yeyesha@umbc.edu
E-mail: finin@umbc.edu
E-mail: yayesha@umbc.edu
*Corresponding author

Abstract: Managing and delivering virtualised cloud-based services in systems like the virtual computing lab (VCL) is an open challenge. Current research is focused on enhancing specific components like service discovery, composition and monitoring and there is no holistic view of what would constitute a lifecycle of virtualised services delivered on a cloud environment. We have developed a policy-based integrated framework for automating acquisition and consumption of cloud services. This framework divides the cloud service lifecycle into five phases: requirements, discovery, negotiation, composition, and consumption. We have also developed a cloud broker tool to automatically discover, negotiate and reserve images from the VCL platform. It was developed using web technologies and is built upon our service lifecycle framework. This paper describes our methodology and the tool as implemented on the VCL platform.

Keywords: cloud broker; services lifecycle; virtual computing lab; VCL.

Reference to this paper should be made as follows: Joshi, K.P., Yesha, Y., Finin, T. and Yesha, Y. (xxxx) ‘A policy-based cloud broker for the VCL platform’, *Int. J. Cloud Computing*, Vol. X, No. Y, pp.000–000.

Biographical notes: Karuna P. Joshi received her PhD in Computer Science in 2012 from the University of Maryland, Baltimore County (UMBC). Her research interests include cloud computing services, databases, web technologies and data mining. She has been awarded the prestigious IBM PhD Fellowship. She completed her MS in Computer Science from UMBC and her Bachelors in Computer Engineering from University of Mumbai. She has over 15 years of industrial experience primarily as an IT Project Manager. She worked at the International Monetary Fund for over nine years. Her managerial experience includes portfolio/programme/project management across various domains.

Yelena Yesha is a Professor of Computer Science and Electrical Engineering at UMBC. She received her BSc in Computer Science from York University, Toronto, Canada, and MSc and PhD degrees in Computer and Information Science from The Ohio State University. She is also the Associate Director for the Multicore Computational Center. In addition, she served as the Director of the Center of Excellence in Space Data and Information Sciences at NASA. Her research interests are in the areas of distributed databases, distributed

systems, digital libraries, electronic commerce, and trusted information systems. She co-authored 14 books and authored over 180 refereed articles in these areas.

Tim Finin is a Professor of Computer Science and Electrical Engineering at UMBC. He has over 30 years of experience in applications of artificial intelligence to problems in information systems and language understanding. His current research is focused on the Semantic Web, mobile computing, analysing and extracting information from text and online social media, and on enhancing security and privacy in information systems. He holds degrees from MIT and the University of Illinois and has also held positions at Unisys, the University of Pennsylvania, and the MIT AI Laboratory.

Yaacov Yesha is a Professor of Computer Science and Electrical Engineering at UMBC. He received his BSc in Chemistry from Tel-Aviv University, Israel, and MSc and PhD degrees in Computer Science from Weizmann Institute of Science, Israel. His research interests are in the areas of parallel computing, wireless communication, distributed systems, computational complexity, algorithms, source coding, speech and image compression.

This paper is a revised and expanded version of a paper entitled ‘Policy based cloud services on a VCL platform’ presented at the 1st International IBM Cloud Academy Conference (ICA CON 2012), Research Triangle Park, North Carolina, USA, 19–20 April 2012.

1 Introduction

With broader adoption of cloud computing, organisations are increasingly procuring information technology (IT) components like software, hardware or network bandwidth as services from providers based all around the world. These services are hosted on the cloud and are delivered to the organisation via the internet or mobile devices. The service is acquired and consumed on an as needed basis.

Cloud services are increasingly based on the composition of multiple component services and assets (technological, human, or process) that may be supplied by one or more providers distributed across the network – in the cloud. Moreover, a single service component can be a part of many composite services as needed. The service, in effect, is virtualised on the cloud. This is becoming the preferred method to deliver services ranging from helpdesk and back-office functions to infrastructure as a service (IaaS). The virtualised model of service delivery also extends to IT enabled services (ITeS), which typically include a large human element.

A key barrier hindering organisations from successfully using services on the cloud is that they have complex internal policies, as well as legal and statutory constraints that require compliance. Such policies are today enforced on internal resources controlled by the organisation. When acquiring remote services, it requires significant human intervention and negotiation – people have to check whether a provider’s service attributes ensure compliance with their organisation’s constraints. This can become very complex if the provider is composing services, some of which it gets from other providers. A related issue is the lack of an integrated methodology for service creation and deployment that provides a holistic view of the service lifecycle on a cloud.

We have developed a methodology (Joshi et al., 2009) to address the lifecycle issue for virtualised services delivered from the cloud and describe it briefly in Section 3. This lifecycle provides ontologies to describe services and their attributes. In particular, we used semantically rich descriptions of the requirements, constraints, and capabilities that are needed at each phase of the lifecycle. Policies can be described using the same ontology terms so that compliance checks can be automated. This methodology is complementary to previous work on ontologies, e.g., OWL-S (Martin et al., 2005), for service descriptions in that it is focused on automating processes needed to procure services on the cloud. The methodology will enable practitioners to plan, create and deploy virtualised services successfully.

We have developed and implemented a cloud broker application to demonstrate and evaluate our methodology. We used web technologies like PHP and Perl to develop this application. The prototype allows users of the virtual computing lab (VCL) (Moothoor and Bhatt, 2012; Schaffer et al., 2009) to discover the available images and reserve the one that best meets their needs by specifying the service constraints, security policies and compliance policies via a simple user interface. In this application, we have incorporated some of the cloud data and security policies identified by National Institute of Standards and Technology (NIST). We have integrated this tool with the VCL cloud platform and describe our preliminary results for the same in Section 4.

2 Related work

Since cloud computing is a nascent field, there is lack of standardisation and a need to more clearly define some of its key elements. NIST has released a special publication 800-145 (NIST Special Publication 800-145, 2011) defining cloud computing as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. One of the key characteristics identified by NIST is that a cloud service should have the capability of on-demand self-service whereby a consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider. This capability of automatically acquiring a service is currently either missing, or very limited, in most cloud-based services. Our methodology aims to make it possible to automatically discover, negotiate/acquire and consume cloud-based services.

In addition to the standard definition of cloud computing, NIST has also released the Cloud Computing Reference Architecture document (NIST Special Publication 500-292, 2011) that describes a reference architecture for cloud computing and also the key roles and responsibilities of stakeholders. The authors of this paper were part of the NIST cloud computing reference architecture and taxonomy working group that participated in developing the standard. Our framework, described in the next section, makes use of the NIST cloud computing standards and definitions of the key stakeholders, viz., cloud provider, cloud consumer, cloud broker and cloud auditor.

Current research on cloud or web services so far has been limited to exploring a single aspect of the lifecycle such as service discovery, service composition, or service quality. There is no integrated methodology for the entire service lifecycle – covering service planning, development and deployment in the cloud. In addition, most of the work is limited to the software component of the service and does not cover the service processes or human agents which are a critical component of IT services.

Papazoglou and Van Den Heuvel (2006) have proposed a methodology for developing and deploying web services using service-oriented architecture (SOA). Their approach, however, is limited to the creation and deployment of web services and does not account for virtualised environment where services are composed on demand. Providers may need to combine their services with other resources or providers' services to meet consumer needs. Other methodologies, like that proposed by Bianchini et al. (2006), do not provide this flexibility and are limited to cases where a single service provider provides one service. Zeng et al. (2003) address the quality-based selection of composite services via a global planning approach but do not cover the human factors in quality metrics used for selecting the components. Maximilien and Singh (2004) propose ontology to capture quality of a web service so that quality attributes can be used while selecting a service. While their ontology can serve as a key building block in our system, it is limited by the fact that it considers single web services, rather than service compositions.

Black et al. (2007) have proposed an integrated model for IT service management. Their model is limited to managing the service from the service provider's perspective. Paurobally et al. (2007) describe a framework for negotiation of web services, and mention that the Ontogrid project (ONTOGRID Project, 2005) enables an iterative contract net protocol (CNP) (Smith, 1980). In Paurobally et al. (2007), they implement an iterative CNP, but their implementation is limited to pre-existing web services and does not extend to virtualised services that are composed on demand. Our negotiation protocol accounts for the fact that the service will be composed only after the contract/SLA listing the constraints is finalised. GoodRelations (Hepp, 2008) is an ontology developed for e-commerce to describe products. While this ontology is useful for describing service components that already exist on the cloud, it is difficult to describe composite virtualised services being provided by multiple vendors using this ontology. Cardoso et al. (2010) have described a Unified Service Description Language (USDL) a specification language to describe services from a business, operational and technical perspective.

The Information Technology Infrastructure Library (ITIL) is a set of concepts and policies for managing IT infrastructure, development and operations that has wide acceptance in the industry. The latest version of ITIL lists policies for managing IT services (Van Bon et al., 2008) that cover aspects of service strategy, service design, service transition, service operation and continual service improvement. However, it is limited to interpreting 'IT services' as products and applications that are offered by in-house IT department or IT consulting companies to an organisation. This framework in its present form does not extend to the service cloud or a virtualised environment that consists of one or more composite services generated on demand.

We use Semantic Web techniques for our services lifecycle development. The Semantic Web deals primarily with data instead of documents. It enables data to be annotated with machine understandable meta-data, allowing the automation of their retrieval and their usage in correct contexts. Semantic Web technologies include languages such as resource description framework (RDF) (Lassila et al., 1999) and Web Ontology Language (OWL) (McGuinness et al., 2004) for defining and using ontologies as well as tools for reasoning over these descriptions. These technologies can be used to provide common semantics of service information and policies enabling all agents who understand basic Semantic Web technologies to communicate and use each other's data and services effectively. The Ontology Web Language for Services (OWL-S) (Martin et al., 2005) was developed to provide a vocabulary for describing the properties and capabilities of web services in unambiguous, computer-interpretable form. OWL-S allows service providers or brokers to define their services based on agreed upon ontologies that describe the functions they provide. We have integrated the OWL-S ontology into our ontology.

SPARQL Protocol and RDF Query Language (SPARQL) is the query language for RDF that has been standardised by W3C (Prud'hommeaux and Seaborne, 2008). SPARQL can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware. It has capabilities for querying required and optional graph patterns and their conjunctions and/or disjunctions. SPARQL also supports value testing and altering of results. The results of queries can be results sets or RDF graphs.

The VCL (Moothoor and Bhatt, 2012; Schaffer et al., 2009) is a cloud computing platform developed at North Carolina State University (NCSU) that can deliver dedicated compute environment to a user for a limited time. Remote users can connect to the VCL scheduling application (the web VCL portal) and request access to a desired application environment. The application environment consists of an operating system and a suite of applications. The currently supported operating systems are Windows, Redhat Linux, and Solaris. The computer types are machine room blade servers, VMware virtual machines (VMs), and standalone machines, including those that may be in a traditional computing lab. The VCL infrastructure consists of three tiers: a web server, a database server, and one or more management nodes. In VCL terminology, a computer is called a compute node.

1. Web server is the VCL portal that provides tools to request, manage and govern all VCL resources. It is built using Linux, Apache and PHP. All its transactions occur with the database.
2. Database server holds all data related to VCL reservations, access controls, machine and environment inventory, log history, etc. It is built using Linux and MySQL.
3. Management node(s) is the processing engine that controls a subset of the VCL resources or can provision a computer that it has control over. There can be more than one management node in a VCL installation. It has been built using Linux, VCLD (Perl) and VCL Imagelibrary. VCLD is the VCL middleware that processes reservations/jobs assigned by the VCL web portal and loads the VCL images.

In VCL, the term *image* (or VCL image) is a software stack that incorporates the following utilities (Moothoor and Bhatt, 2012):

- base-line operating system, and if virtualisation is needed for scalability, this will allow on hypervisor layer
- desired middleware or application that runs on the selected operating system
- end-user access solution that is appropriate for the selected operating system,

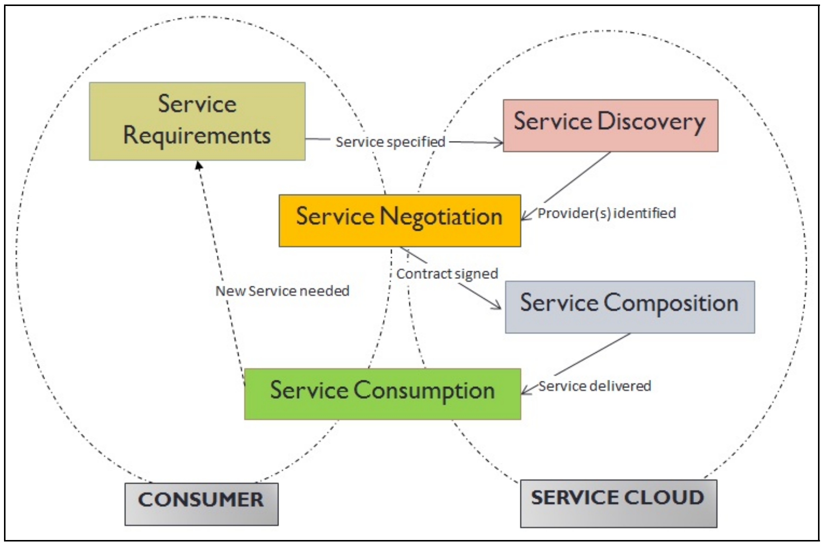
Images can be loaded on 'bare-metal' or to an operating system/application virtual environment of choice. If the user's desired combination of images is not available, the user has the privilege to construct the images in their own choice from the VCL component library.

We have added the VCL broker functionality by making additions to the existing web server and the database server tiers. We detail these changes in Section 4.

3 Lifecycle of cloud services

We have developed a methodology which integrates all of the processes and data flows that are needed to automatically acquire, consume, and manage services on the cloud. We divide this IT service lifecycle on a cloud into five phases. In sequential order of execution, they are requirements, discovery, negotiation, composition, and consumption; and are illustrated in Figure 1. We have described these phases in detail along with the associated metrics in Joshi et al. (2009). We have developed the ontology for the entire lifecycle in OWL 2 DL profile (Joshi, 2010).

Figure 1 The IT service lifecycle on a virtualised cloud comprises five main phases: requirements, discovery, negotiation, composition and consumption (see online version for colours)



3.1 *Service requirements phase*

In this phase, the consumer details the technical and functional specifications that a service needs to fulfil along with the organisational policies for the providing agent, data quality policy and security policies for the service. Service compliance policies such as required certifications standards to be adhered to, etc. are also identified. Depending on the service cost and availability, a consumer may be amenable to compromise on the service quality. Functional specification describe in detail what functions/tasks should a service help automate. The technical specifications lay down the hardware, software, application standards, and language support policies to which a service should adhere. Once the consumers have identified and classified their service needs, they issue a request for service (RFS). This request could be made by directly contacting a few service providers for their quotes. Alternatively, consumers can use a service discovery engine or service broker on the cloud to procure the service.

3.2 Service discovery phase

Service providers are discovered by comparing the specifications listed in the RFS. The discovery is constrained by functional and technical attributes defined, and also by the budgetary, security, compliance, data quality, and agent policies of the consumer. While searching the cloud, service brokers can be employed. The broker engine queries the service providers to match the service data and security policies, compliance needs, functional, and technical specifications; and returns the result with the service providers in priority order. Sbodio et al. (2010) have presented semantic approaches for service discovery that can be incorporated in our methodology.

One critical part of this phase is service certification, in which the consumers will contact a central registry, such as UDDI (Ran, 2003), to get references for providers that they narrow down to. The NIST reference architecture (NIST Special Publication 500-292, 2011) identified a 'cloud auditor' role that will be primarily responsible for security, performance and privacy impact audits of the cloud. We use this role in our framework to be the 'provider certifying agent' that will be referenced in the service discovery phase.

If a consumer finds the exact service within their budgets, s/he can begin consuming the service immediately upon payment. However, often the consumer will get a list of providers who will need to compose a service to meet the consumer's specifications. The consumer will then have to begin negotiations with the service providers. Each search result will return the primary provider who will be negotiating with the consumer.

3.3 Service negotiation phase

The service negotiation phase covers the discussion and agreement that the service provider and consumer have regarding the service delivered and its acceptance criteria. The service delivered is determined by the specifications laid down in the RFS. Service acceptance is usually guided by the service level agreements (SLAs) (SLA, 2009) that the service provider and consumer agree upon. SLAs define the service data, delivery mode, agent details, quality metrics, and cost of the service. While negotiating the service levels with potential service providers, consumers can explicitly specify service quality constraints (data quality, cost, security policies, response time, etc.) that they require.

At times, the service provider will need to combine a set of services or compose a service from various components delivered by distinct service providers in order to meet the consumer's requirements. The negotiation phase also includes the discussions that the main service provider has with the other component providers. When the services are provided by multiple providers (composite service), the primary provider interfacing with the consumer is responsible for composition of the service. The primary provider will also have to negotiate the quality of service (QoS) with the component providers to ensure that SLA metrics are met.

The key deliverable of this phase is the service contract between the service consumer and service provider. The SLA is a key part of this service contract and will be used in the subsequent phases to compose and monitor the service. Another deliverable of this phase are the service sub contracts between the service provider and component (or dependent services) providers. The QoS are the essential part of the service sub-contracts and are used in the consumption phase to monitor service performance.

3.4 Service composition phase

In this phase one or more services provided by one or more providers are combined and delivered as a single service. Service orchestration determines the sequence of the service components.

3.5 Service consumption/monitoring phase

The service is delivered to the consumer based on the delivery mode (synchronous/asynchronous, real-time, batch mode, etc.) agreed upon in the negotiation phase. After the service is delivered to the consumer, payment is made and the consumer then begins consuming the service. In a cloud environment, the service usually resides on remote machines managed by the service providers. The provider is responsible for managing and monitoring the service. In this phase, consumer will require tools that enable service quality monitoring and service termination if needed. This will involve alerts to humans or automatic termination based on policies defined using the quality-related ontologies. The service monitor measures the service quality and compares it with the quality levels defined in the SLA. This phase spans both the consumer and cloud areas as performance monitoring is a joint responsibility. If the consumer is not satisfied with the service quality, s/he should have the option to terminate the service and stop service payment.

The composite service is made up of human agents providing the service, the service software, and dependent service components. All the three elements, agents, software, and dependent services, must be monitored to manage the overall service quality. For the service, software providers have to track its performance, reliability, assurance, and presentation as they will influence customer's satisfaction rating (CSATs). Since the dependent services/components will be at the backend and, will not interface directly with the consumers, the service provider only needs to monitor their performance. We have proposed a framework to manage quality based on fuzzy-logic for such composed services delivered on the cloud in Joshi et al. (2011).

4 VCL cloud broker

We have developed the VCL cloud broker application to enable users to search an image inventory and reserve images that meet their requirements and security policies. It consists of a web interface (Figure 2) that enables users to easily define the VCL image constraints by choosing predefined values from dropdown fields. The application then discovers the images that will match the specified constraints. The application also automatically relaxes constraints iteratively based on their priority till a match is found. The user can select the discovered images to view its detailed information and reserve the desired image. This application is based on the cloud services lifecycle described in the previous section. We have also developed an ontology in OWL 2 DL profile to define

VCL images (Joshi, 2012). This ontology can be referred to declare VCL image structure to external semantic-based cloud brokers that are discovering cloud resources.

4.1 Broker architecture

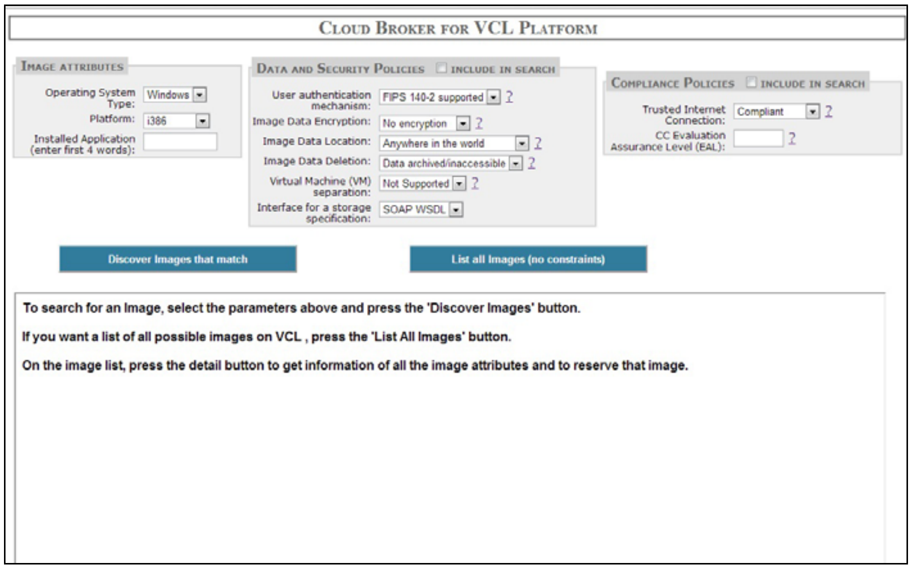
We used web technologies like Perl and PHP (*PHP: Hypertext Preprocessor*, 2012) to build the VCL broker prototype. We have incorporated actual enterprise policies related to data storage and security that are practiced by large organisations. We have used the policies defined in the Use Case 3.9 (NIST Cloud Computing Use Case 3.9, 2012) identified by the NIST cloud computing initiative.

We developed this application as an extension to the existing VCL architecture described in related work. The two major changes done to the VCL architecture were:

1 *Web-based user interface:* A new portal interface (see Figure 2) was designed for the VCL broker. It enables users to define their VCL image requirements as well as specify the data, security and compliance policies that the VCL image should meet. The priority of the fields was pre-determined by our NIST collaborators. The fields were laid out in order of decreasing priority. For instance, the priority of the data location constraint was determined to be lower than the priority of the data encryption constraint.

2 *Database server:* VCL's database server has been developed on MYSQL and Linux platform It holds all data related to VCL reservations, access controls, machine and environment inventory, log history, etc. We added additional tables and views in the MYSQL database to build the VCL broker. The tables store the information relating to the image data and security policies and compliance policies. Views were created to join the data spread across the image, OS, platform, image-security tables. These views were used for querying images in the user interface.

Figure 2 User interface of the VCL cloud broker tool (see online version for colours)


Cloud Broker for VCL Platform
Image attributes
Data and Security Policies
include in search
Compliance Policies
include in search
Discover Images that match
List all Images (no constraints)

Operating System Type: Windows
Platform: i386
Installed Application (enter first 4 words):

User authentication mechanism: FIPS 140-2 supported
Image Data Encryption: No encryption
Image Data Location: Anywhere in the world
Image Data Deletion: Data archived/inaccessible
Virtual Machine (VM) separation: Not Supported
Interface for a storage specification: SOAP WSDL

Trusted Internet Connection: Compliant
CC Evaluation Assurance Level (EAL):

To search for an Image, select the parameters above and press the 'Discover Images' button.

If you want a list of all possible images on VCL , press the 'List All Images' button.

On the image list, press the detail button to get information of all the image attributes and to reserve that image.

4.2 Web-based user interface

Users can specify their service requirements as well as the data, security and compliance policies that the service should meet using an easy to use web-based interface (Figure 2). There are three main categories of VCL image attributes – core attributes, data/security policy attributes and compliance policy attributes. We describe these in detail below.

Most fields in the interface have an associated 'help' description that describes the attributes and enables users to determine which option to select. The fields in each section are placed in descending order of priority. The priority for the data security and compliance policy fields was determined in conjunction with our NIST collaborators.

4.2.1 Key attributes

The core attributes are the mandatory attributes that the VCL image should meet.

1 *Operating system type:* the type of operating system supported by the image. The three choices are Windows, Linux and Unix.

2 *Platform:* specifies the platform of the VCL image. The three choices are i386, i386_lab and ultrasparc.

3 *Installed application:* the users can search for the application that they want installed on the VCL image by typing the first four letters of the application.

4.2.2 Data and security policy attributes

The data and security policy attributes specify the policies of the consumer organisation with regards to their data on the cloud. The policies that we selected are specific to the NIST Cloud Computing User Case 3.9 (2012). The attributes in this section are listed below.

1 *User authentication mechanism:* specifies whether FIPS 140-2 is to be supported by the cloud provider or not. NIST issued FIPS (Federal Information Processing Standard – Publication 140-2) is a US Government computer security standard used to accredit cryptographic modules.

2 *Data encryption:* specifies if the consumer wants the data to be encrypted when stored on the cloud.

3 *Data location:* specifies the constraint the consumer may have regarding the location of the cloud.

4 *Data deletion:* enables consumer to specify their data deletion policies for the cloud – whether the data is deleted or merely made inaccessible, secure wipe is supported or not. While in the cloud environment it may be difficult to ensure the intended data deletion; adding this policy to the service SLA will make the cloud provider liable if the deletion method specified is not followed. So the onus to ensure appropriate data deletion procedure will be on the cloud provider and not the cloud user.

5 *VM separation:* specifies whether the cloud provider supports separate VMs to store consumer's data on the cloud. Some organisations may desire separate VMs on the cloud to ensure more security.

6 *Interface for a storage specification:* The consumer can specify in their requirements whether they want SOAP protocol or representational state transfer (REST) interface support.

4.2.3 Compliance policy attributes

In the tool, we have included two compliance policies specified by NIST and majority of the US federal agencies. They are listed below. The compliance policy constraints have lower priority than the data/security policy constraints and hence are relaxed after cloud instance constraints, but before the data/security constraints during service negotiation.

1 *Trusted internet connection (TIC):* TIC initiative is mandated in an Office of Management and Budget (OMB) Memorandum (M-08-05) meant to optimise individual external connections, including internet points of presence currently in use by the Federal government of the USA.

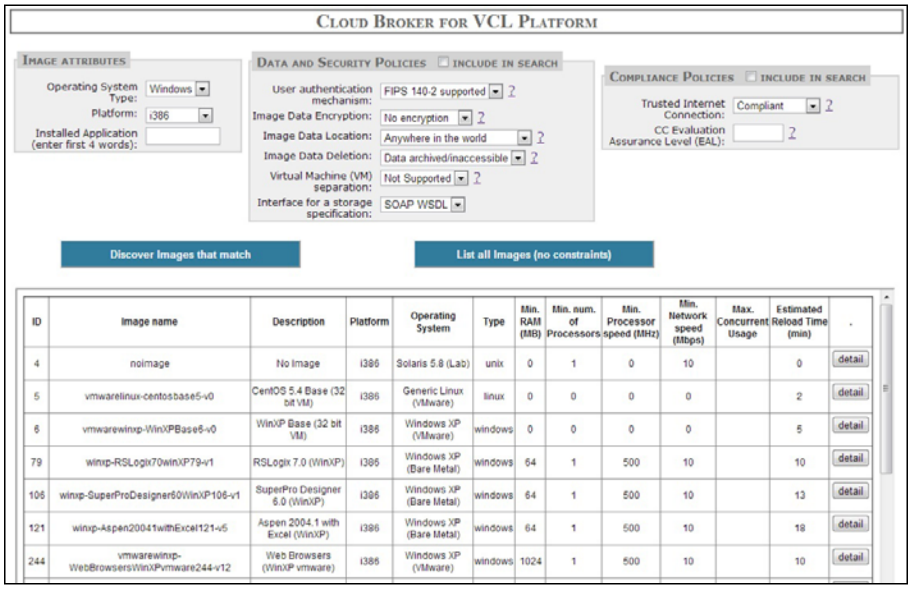
2 *CC evaluation assurance level (EAL) levels:* The EAL (EAL1 through EAL7) of an IT product or system is a numerical grade assigned following the completion of a common criteria security evaluation, an international standard in effect since 1999. This attribute accepts EAL number from 1 to 7.

4.3 Application workflow

VCL cloud broker allows the users to discover the VCL image that best matches their requirements and constraints. Users can also get a list of all existing images in the inventory by clicking the 'list all images' button. Figure 3 illustrates this option.

If users are searching for images containing some specific applications or adhering to specific security policies, they can specify their constraints by selecting them from the drop down list. The users press the 'discover images that match' button to begin searching for images that meet all the specified constraints. If no image matches the specified constraints, then the tool automatically relaxes the requirements one by one and generates a new query to search the image inventory after every relaxation. The order of constraint relaxation for this prototype was determined by the NIST team collaborating with us who specified the priority of each constraint. After each constraint relaxation, the tool executes the new query to discover images that match the new constraint set. When a match is found, the tool returns the details of that image along with a list of constraints not met. Figure 4 illustrates the scenario where three constraints that were not met were automatically relaxed by the application before searching the image inventory again. Users can view all the attributes of a VCL image by clicking the 'details' button next to the list (see Figure 5) and decide on the VCL image that will best meet their needs. Users click the 'reserve image' button to send a reservation request.

Figure 3 Clicking on the list all images button lists all possible images in the (see online version for colours)



CLOUD BROKER FOR VCL PLATFORM

IMAGE ATTRIBUTES

Operating System Type:	Windows
Platform:	i386
Installed Application (enter first 4 words):	

DATA AND SECURITY POLICIES ☐ INCLUDE IN SEARCH

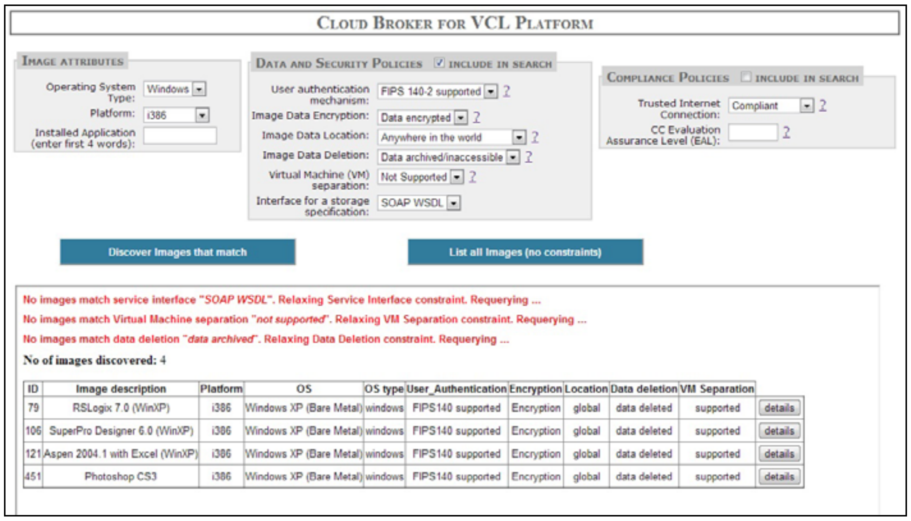
User authentication mechanism:	FIPS 140-2 supported
Image Data Encryption:	No encryption
Image Data Location:	Anywhere in the world
Image Data Deletion:	Data archived/inaccessible
Virtual Machine (VM) separation:	Not Supported
Interface for a storage specification:	SOAP WSDL

COMPLIANCE POLICIES ☐ INCLUDE IN SEARCH

Trusted Internet Connection:	Compliant
CC Evaluation Assurance Level (EAL):	

ID	Image name	Description	Platform	Operating System	Type	Min. RAM (MB)	Min. num. of Processors	Min. Processor speed (MHz)	Min. Network speed (Mbps)	Max. Concurrent Usage	Estimated Reload Time (min)	.
4	noimage	No Image	i386	Solaris 5.8 (Lab)	unix	0	1	0	10		0	detail
5	vmwarelinux-centosbase5-v0	CentOS 5.4 Base (32 bit VM)	i386	Generic Linux (VMware)	linux	0	0	0	0		2	detail
6	vmwarewinxp-WinXPBase6-v0	WinXP Base (32 bit VM)	i386	Windows XP (VMware)	windows	0	0	0	0		5	detail
79	winxp-RSLogix70winXP79-v1	RSLogix 7.0 (WinXP)	i386	Windows XP (Bare Metal)	windows	64	1	500	10		10	detail
106	winxp-SuperProDesigner60WinXP106-v1	SuperPro Designer 6.0 (WinXP)	i386	Windows XP (Bare Metal)	windows	64	1	500	10		13	detail
121	winxp-Aspen20041withExcel121-v5	Aspen 2004.1 with Excel (WinXP)	i386	Windows XP (Bare Metal)	windows	64	1	500	10		18	detail
244	vmwarewinxp-WebBrowsersWinXPvmware244-v12	Web Browsers (WinXP vmware)	i386	Windows XP (VMware)	windows	1024	1	500	10		10	detail

Figure 4 Clicking on the discover images button lists only images that meet the specified constraints (see online version for colours)



CLOUD BROKER FOR VCL PLATFORM

IMAGE ATTRIBUTES

Operating System Type:	Windows
Platform:	i386
Installed Application (enter first 4 words):	

DATA AND SECURITY POLICIES ☑ INCLUDE IN SEARCH

User authentication mechanism:	FIPS 140-2 supported
Image Data Encryption:	Data encrypted
Image Data Location:	Anywhere in the world
Image Data Deletion:	Data archived/inaccessible
Virtual Machine (VM) separation:	Not Supported
Interface for a storage specification:	SOAP WSDL

COMPLIANCE POLICIES ☐ INCLUDE IN SEARCH

Trusted Internet Connection:	Compliant
CC Evaluation Assurance Level (EAL):	

No images match service interface "*SOAP WSDL*". Relaxing Service Interface constraint. Requerying ...

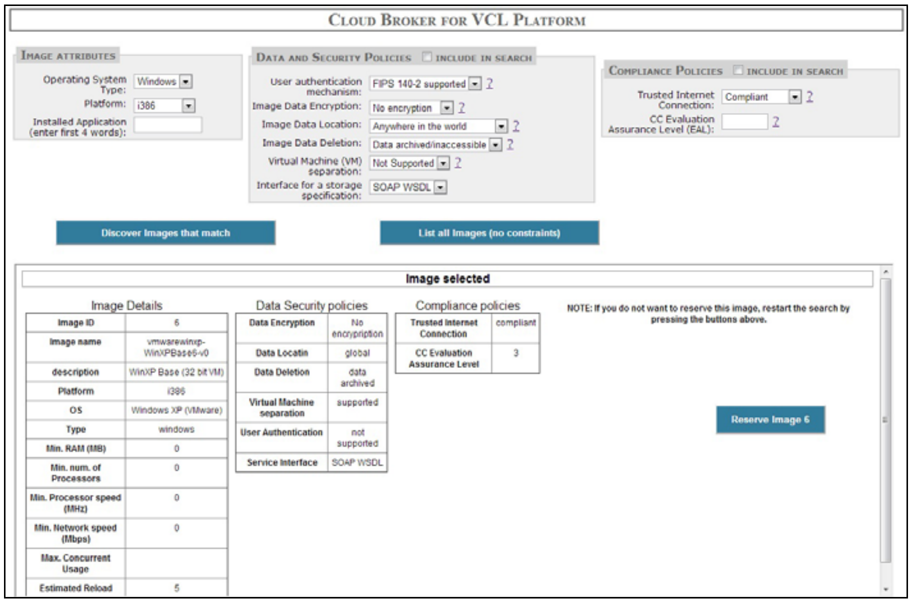
No images match Virtual Machine separation "*not supported*". Relaxing VM Separation constraint. Requerying ...

No images match data deletion "*data archived*". Relaxing Data Deletion constraint. Requerying ...

No of images discovered: 4

ID	Image description	Platform	OS	OS type	User_Authentication	Encryption	Location	Data deletion	VM Separation	
79	RSLogix 7.0 (WinXP)	i386	Windows XP (Bare Metal)	windows	FIPS140 supported	Encryption	global	data deleted	supported	details
106	SuperPro Designer 6.0 (WinXP)	i386	Windows XP (Bare Metal)	windows	FIPS140 supported	Encryption	global	data deleted	supported	details
121	Aspen 2004.1 with Excel (WinXP)	i386	Windows XP (Bare Metal)	windows	FIPS140 supported	Encryption	global	data deleted	supported	details
451	Photoshop CS3	i386	Windows XP (Bare Metal)	windows	FIPS140 supported	Encryption	global	data deleted	supported	details

Figure 5 When user presses the details button, all the attributes of the VCL image are displayed (see online version for colours)



CLOUD BROKER FOR VCL PLATFORM

IMAGE ATTRIBUTES

Operating System Type:	Windows
Platform:	i386
Installed Application (enter first 4 words):	

DATA AND SECURITY POLICIES ☐ INCLUDE IN SEARCH

User authentication mechanism:	FIPS 140-2 supported
Image Data Encryption:	No encryption
Image Data Location:	Anywhere in the world
Image Data Deletion:	Data archived/inaccessible
Virtual Machine (VM) separation:	Not Supported
Interface for a storage specification:	SOAP WSDL

COMPLIANCE POLICIES ☐ INCLUDE IN SEARCH

Trusted Internet Connection:	Compliant
CC Evaluation Assurance Level (EAL):	

Image selected

Image Details

Image ID	6
Image name	vmwarewinxp-WinXPBase6-v0
description	WinXP Base (32 bit VM)
Platform	i386
OS	Windows XP (VMware)
Type	windows
Min. RAM (MB)	0
Min. num. of Processors	0
Min. Processor speed (MHz)	0
Min. Network speed (Mbps)	0
Max. Concurrent Usage	
Estimated Reload	5

Data Security policies

Data Encryption	No encryption
Data Locatin	global
Data Deletion	data archived
Virtual Machine separation	supported
User Authentication	not supported
Service Interface	SOAP WSDL

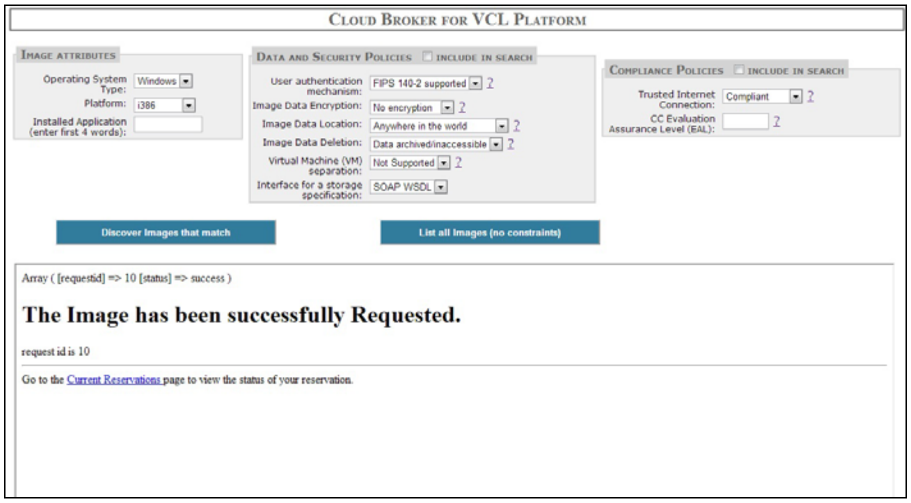
Compliance policies

Trusted Internet Connection	compliant
CC Evaluation Assurance Level	3

NOTE: If you do not want to reserve this image, restart the search by pressing the buttons above.

When the user presses the ‘reserve image’ button on the tool, a request is sent to the VCL admin module to reserve the image. If the request is successfully sent, then the tool returns the request ID to the user (see Figure 6). Users can then see the status of the reservation by selecting the ‘current reservations’ button on the VCL scheduling application under their account. Figure 7 is a screen shot of the VCL web interface showing the successful reservation ready for connection.

Figure 6 The image is successfully reserved (see online version for colours)



CLOUD BROKER FOR VCL PLATFORM

IMAGE ATTRIBUTES

Operating System Type:	Windows
Platform:	i386
Installed Application (enter first 4 words):	

DATA AND SECURITY POLICIES ☐ INCLUDE IN SEARCH

User authentication mechanism:	FIPS 140-2 supported
Image Data Encryption:	No encryption
Image Data Location:	Anywhere in the world
Image Data Deletion:	Data archived/inaccessible
Virtual Machine (VM) separation:	Not Supported
Interface for a storage specification:	SOAP WSDL

COMPLIANCE POLICIES ☐ INCLUDE IN SEARCH

Trusted Internet Connection:	Compliant
CC Evaluation Assurance Level (EAL):	

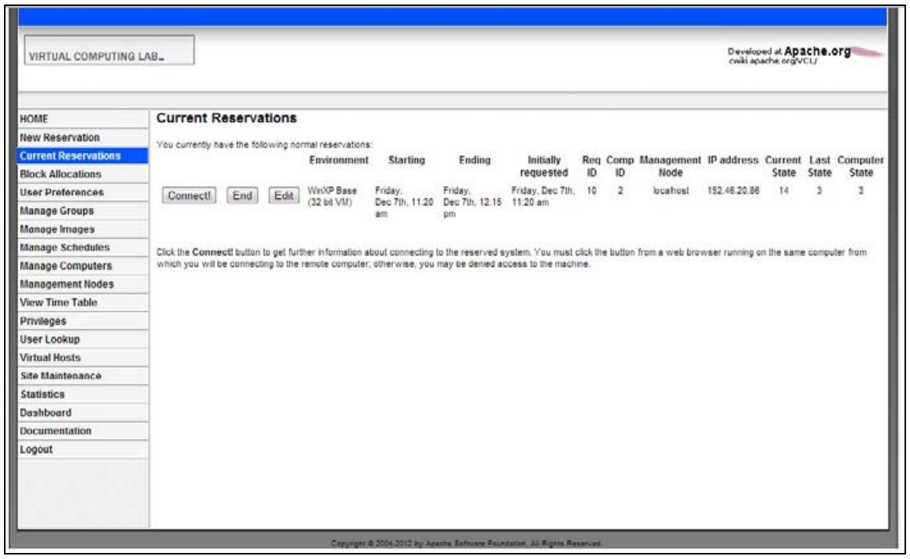
```
Array ( [requestid] => 10 [status] => success )
```

The Image has been successfully Requested.

request id is 10

Go to the Current Reservations page to view the status of your reservation.

Figure 7 The successful request is visible in the current reservations screen in the user's VCL account (see online version for colours)



Current Reservations

You currently have the following normal reservations:

	Environment	Starting	Ending	Initially requested	Req ID	Comp ID	Management Node	IP address	Current State	Last State	Computer State
Connect! End Edit	WinXP Base (32 bit VM)	Friday, Dec 7th, 11:20 am	Friday, Dec 7th, 12:15 pm	Friday, Dec 7th, 11:20 am	10	2	localhost	152.46.20.86	14	3	3

Click the **Connect!** button to get further information about connecting to the reserved system. You must click the button from a web browser running on the same computer from which you will be connecting to the remote computer; otherwise, you may be denied access to the machine.

Copyright © 2004-2012 by Apache Software Foundation, All Rights Reserved.

5 Conclusions and future work

We have described an integrated methodology to automate IT services lifecycle on the cloud. To the best of our knowledge, this is the first such effort, and it is critical as it provides a holistic view of steps involved in deploying IT services. Our methodology, described in Section 3, complements previous work on ontologies for service descriptions in that it is focused on automating the processes needed to procure services on the cloud. The methodology can be referenced by organisations to determine what key deliverables they can expect at any stage of the process. We also hope that it will enable the academia and the industry to be on the 'same page' when referring to IT services on the cloud.

The VCL cloud broker application successfully demonstrated how our methodology can be used to significantly improve the existing image reservation and inventory management capability in VCL by incorporating data security and compliance constraints that many large organisations are looking for in cloud platforms. We have successfully executed this application on our environment and are in the process of releasing this application to multiple VCL users to analyse how this scales up. As part of our ongoing work in this area, we are also working on enhancing the application to be able to discover images across multiple VCL cloud platforms.

References

Bianchini, D., De Antonellis, V., Pernici, B. and Plebani, P. (2006) 'Ontology-based methodology for e-service discovery', *International Journal of Information Systems, The Semantic Web and Web Services*, June–July, Vol. 31, Nos. 4–5, pp.361–380.

Black, J. et al. (2007) 'An integration model for organizing IT service management', *IBM Systems Journal*, July, Vol. 46, No. 3, pp.405–422.

Cardoso, J., Barros, A., May, N. and Kylau, U. (2010) 'Towards a unified service description language for the internet of services: requirements and first developments', *IEEE Intl. Conference on Services Computing*, IEEE Computer Society Press.

Hepp, M. (2008) 'GoodRelations: an ontology for describing products and services offers on the web', *Proceedings of the 16th International Conference on Knowledge Engineering and Knowledge Management (EKAW2008)*, Italy, Springer LNCS, Vol. 5268, pp.332–347.

Joshi, K. (2010) *OWL Ontology for Lifecycle of IT Services on the Cloud* [online] http://ebiquity.umbc.edu/ontologies/itso/1.0/itso.owl (accessed 7 December 2012).

Joshi, K. (2012) *OWL Ontology for Virtual Computing Lab Image* [online] http://ebiquity.umbc.edu/ontologies/itso/1.0/VCLImage.owl (accessed 7 December 2012).

Joshi, K., Finin, T. and Yesha, Y. (2009) 'Integrated lifecycle of it services in a cloud environment', *Proceedings of The Third International Conference on the Virtual Computing Initiative (ICVCI 2009)*, October, Research Triangle Park, NC.

Joshi, K., Joshi, A. and Yesha, Y. (2011) 'Managing the quality of virtualized services', *Proceedings of the SRII Global Conference*, March, San Jose.

Lassila, O., Swick, R. et al. (1999) 'Resource description framework (RDF) model and syntax specification', *World Wide Web Consortium*.

Martin, D. et al. (2005) 'Bringing semantics to web services: the OWL-S approach', *Lecture Notes in Computer Science*, Vol. 3387, pp.26–42, Springer.

Maximilien, E.M. and Singh, M. (2004) 'A framework and ontology for dynamic web services selection', *IEEE Internet Computing*, September/October, Vol. 8, No. 5, pp.84–93.

McGuinness, D., Van Harmelen, F. et al. (2004) 'OWL web ontology language overview, W3C recommendation', *World Wide Web Consortium*.

Moothoor, J. and Bhatt, V. (2012) *IBM VCL: A Cloud Computing Solution in Universities* [online] http://www.ibm.com/developerworks/webservices/library/ws-vcl/ (accessed 20 September).

NIST Cloud Computing Use Case 3.9 (2012) *Query Cloud-Provider Capabilities and Capacities* [online] http://www.nist.gov/itl/cloud/3_9.cfm (accessed 25 February 2012).

NIST Special Publication 500-292 (2011) *NIST Cloud Computing Reference Architecture*, November.

NIST Special Publication 800-145 (2011) *The NIST Definition of Cloud Computing*, September.

ONTOGRID Project (2005) *Paving the Way for Knowledgeable Grid Services and Systems*, EU FP6 project (FP6-511513) [online] http://www.ontogrid.net/ (accessed 23 September 2012).

Papazoglou, M. and Van Den Heuvel, W. (2006) 'Service-oriented design and development methodology', *International Journal of Web Engineering and Technology*, Vol. 2, No. 4, pp.412–442.

Paurobally, S., Tamma, V. and Wooldrdige, M. (2007) 'A framework for web service negotiation', *ACM Transactions on Autonomous and Adaptive Systems*, Vol. 2, No. 4, Article 14, November.

PHP: Hypertext Preprocessor (2012) [online] http://www.php.net/ (accessed 27 August 2012).

Prud'hommeaux, E. and Seaborne, A. (2008) *SPARQL Query Language for RDF, W3C Recommendation*, January 2008 [online] http://www.w3.org/TR/rdf-sparql-query/ (accessed 27 April 2011).

Ran, S. (2003) 'A model for web services discovery with QoS', *ACM SIGecom Exchanges*, Vol. 4, No. 1, pp.1–10.

Sbodio, M.L., Martin, D. and Moulin, C. (2010) 'Discovering Semantic Web services using SPARQL and intelligent agents', *Journal of Web Semant.*, November, Vol. 8, No. 4, pp.310–328.

Schaffer, H., Averitt, S., Hoit, M., Peeler, A., Sills, E. and Vouk, M. (2009) 'NCSU's virtual computing lab: a cloud computing solution', *Computer*, July, Vol. 42, No. 7, pp.94–97.

SLA (2009) *Whats in a Service Level Agreement?*, SLA@SOI [online] http://sla-at-soi.eu/?p=356 (accessed 30 July 2009).

Smith, R. (1980) 'The contract net protocol: high-level communication and control in a distributed problem solver', *IEEE Transactions on Computers*, Vol. C-29, No. 12, pp.1104–1113.

Van Bon, J. et al. (2008) *Foundations of IT Service Management based on ITIL V3*, Van Haten Publishing, Netherlands.

Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J. and Sheng, Q. (2003) 'Quality driven web services composition', *Proceedings of the 12th International Conference on World Wide Web*, pp.411–421.