SERVICE-ORIENTED CLOUD ARCHITECTURE SCHEMA TO BRIDGE GAP BETWEEN STUDENT, STAFF AND ACADEMIA

Ravulakollu Kiran Kumar
Department of Computer Science & Engineering,
School of Engineering & Technology,
Sharda University, Plot No. 32-34, Knowledge Park-III,
Greater Noida, NCR, INDIA-201306
kiran.ravulakollu@sharda.ac.in
http://www.sharda.ac.in

Pankaj Chejara
Department of Computer Science & Engineering,
School of Engineering & Technology,
Sharda University, Plot No. 32-34, Knowledge Park-III,
Greater Noida, NCR, INDIA-201306
pankaj.chejara@sharda.ac.in
http://www.sharda.ac.in

Abstract

Cloud Computing is a new resource platform that offers abundant amount of services for organizations to meet furthermore satisfies the needs without huge and prior investment. When it comes to the development of a cloud, architecture plays a critical role. Architectural design is the primary step to meet the client demands. The purpose of this paper is to design a feasible and low-cost architecture that can be adapted for small-scale academic organizations. However, the architecture is expected to provide reliable and qualitative services such as SaaS, PaaS considering risk analysis into account. During the process, customization is carried out at possible stages of cloud development, such that the chances of generating a low-cost architecture maximizes. Proposed architecture considers limitations on hardware, middle ware, open source platforms, additional modules and client requirements during the development process. Novelty in proposed architecture lies in addition of group handler layer to increase service efficiency. Finally the architecture proposed through this paper is able to justify a stable and feasible platform for motivating universities to bridge the gap between student, staff and academia with cloud technology.

Keywords- Academic Cloud; Cloud Architecture; Cloud Computing; Cloud Customization; Cloud Service Handler and Service Oriented Cloud

1. INTRODUCTION

Higher education is the starting pillar on which a prospective research and significant technical and social development can be directed and maintained. From the past few years, a huge transition is observed among various universities within the higher education level whose motivation targets the current research scope along with up-to-date information technology infrastructure in terms of industrial relations [Mircea (2010)]. Technical advancements that are fully developed in industries are returning home '*university*' for playing their role in futuristic innovations and development. This transition helps to migrate the services from traditional and complicated to subtle and time saving. Due to the recent financial crisis across various countries, there is a huge cut in the grants sanctioned to higher education, along with universities. Also there is a dramatic increase in the handling costs across various universities such as budgets, number of students, industry oriented courses, infrastructure and research accomplishment [Golden B, (2012)]. In this circumstance universities are internally looking to cut back possible intermediate and inter-dependent services such that the academic progress and education can be improved [Ojala, (2012)]. It is observed that there are factors that current universities are struggling with towards balancing the workload, such as:

- Increase of infrastructure costs
- Pressure of intake increase
- Students success followed by inflow
- Institutional performance
- Competition in stabilization and development
- Adaptation to industrial and commercial sectors

In the study carried out by Sasikala and Prema concerned with above factors that "adaption of cloud computing is not an optional transmission but a necessary aspect" that has to be undertaken by academic universities [Sasikala (2010)]. Also, according to Mircea and Andreescu, many of academic organizations are looking for alternatives such that, institutions workload can be effectively managed using IT services. In this context Cloud Computing is a suitable extension that can be affordable and feasible for providing enhanced services in both academic and research platforms [Mircea (2011), McCrea (2009)]. By adapting this solution much of the workload that is now shared across faculty member can be centralized and automated without any overheads. By doing so, the possibility of spending much time on delivering a strengthened lecture and carrying out qualitative research can be improved. An introduction of service-oriented architecture in a cloud environment is definitely the place for universities when it comes to balancing cost and service [Mircea (2011)]. Apart from the teaching and research aspect, cloud is useful for the following:

- IT service complexity can be reduced using Cloud Computing
- To support co-operative learning
- Social based strategies of computer technologies
- E-Learning services
- Centralized data storage facilities using virtualization
- Constant data access and monitoring services
- Reduction in software and application licensing and updating
- Personalized data centers for enhanced services

2. Risk Analysis

For most of the academic organizations, IT services are the major face factors of them. Institutions honoring the development in this aspect are nearly 70% while the remaining focus will be on the cost reduction [Kartz (2010)]. One-way of improving this tendency is by analyzing the risks involved in going for cloud computing. The solution (architecture) is expected to reduce any associated risks (overheads) involved with costs towards bridging the gap between cloud and university. Certain tasks associated with cloud can be transferred to providers (incase of public cloud), which can handle security, customization and other performance related issues [Patterson (2010)]. The research work, knowledge and results that are carried out at the university could be of target in terms of security and data protection. Hence when the cloud is up and running security should not be compromised from attackers. Also the protection of intellectual property rights plays a vital role in security context [Tout (2009), Catteddu. D (2009), Goldstein P. J. (2012)].

In order to overcome these risks, different data protection policies are adopted. These policies are adapted in correspondence with the academic regularities and constraints grounding research, employee and student. They are:

- Centralized data center – containing sensitive data along with eliminating unwanted and unrelated relations with in the organization to obtain greater latency for both applications and users
- Task and target mapping – data access should only be held and handled as per organization structure for greater performance and potential security [Mcirvine S. (2010)].

These can be generalized into following solutions:

- Firewall
- Identity Management
- Data Masking

When it comes to information handling on an open environment such as cloud, data encryption and decryption are the standard security methodologies adapted. Hence, the firewall that is securing cloud is expected to perform the key management techniques for packet inspection. By doing so, the sensitive data residing inside can be secured effectively. Overall the decision of adapting cloud environment in academic organizations such as universities is feasible and affordable [Mircea (2011)]. On a global scale this transmission is recommended for all universities whose vision lies far-sighted. Also this provides a gateway between qualitative research (faculty), qualitative study (student), university management (administration) and innovation.

3. Cloud Architecture

The primary purpose of this architecture is to enhance the quality and performance of cloud services in both platform and software. The architecture considers university policies and regulations along with organization culture into account for balancing between service, cost and adaptation [Tuncay E. (2010), Mahjoub M. (2011)]. Customization is adapted at various stages of this architecture to ensure the requirements of the organization are satisfied.

In this paper architecture is laid-out in 3-different aspects.

- Base Structure
- Layers: limited layer interface mapping, for effective source location
- Mapping & Suitability

The base structure of the cloud is deterministic based on type of cloud adapted. However, at this stage no much customization is needed unless a huge expansion is needed. But the layers are subjected to customization based on the load, size and connectivity issues.

A. Structure

The structure of the cloud and its services are significantly categorized for academic suitability. In this case, the cloud structure shown in Fig.1 is organized based on the kind of cloud, categorization of service and enhancements. The above-defined structure can be claimed as operational structure, since the architecture is originated from developed and successfully implemented cloud environments that are being used across the world [Patterson (2010), Wyld D. C. (2009), Mahjoub M. (2011)]. For universities private cloud is primary option at the initiation level, due to reliable and effective information security and service associated.

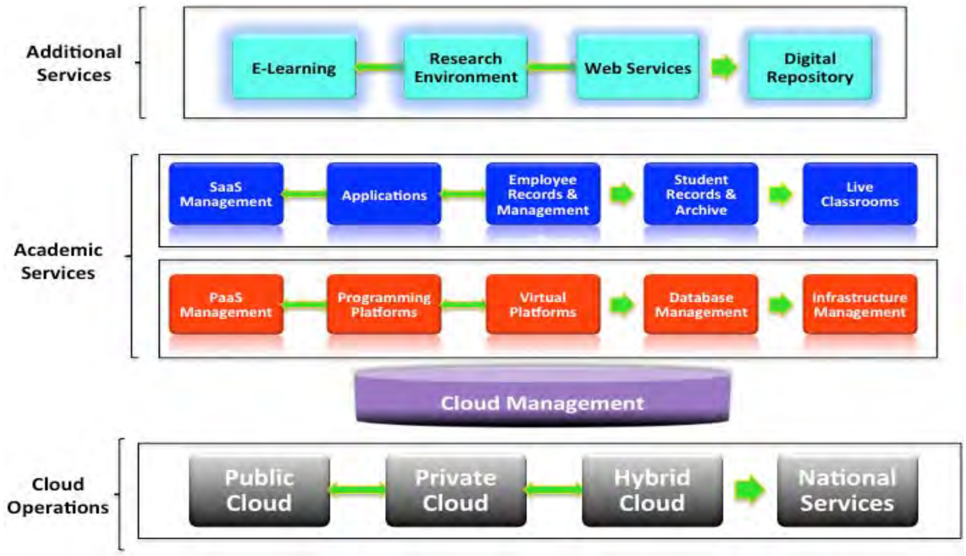

Fig1: Academic services based cloud structure that can fulfill the requirements of university in managing services across students, course and faculty.

Later, as the cloud grows (public cloud) certain services such as mobile and business motivation can be included. In this scenario, such public cloud environments will suffer from security issues. Considering these issues, the structure is made accommodative for hybrid cloud with the feature of customization at various levels for private and public cloud properties. This structure can be more rewarded only if the services are managed effectively using layer pattern for resource utilization.

Cloud management plays a vital role in managing cloud in terms of resource management, load management, service management and quality issues. Hence cloud manager is further classified in a set of layers for efficient usage.

B. Cloud Layers

The internal structure is organized into a series of layers as shown in Figure 2, through which the functionality of the cloud can be effectively achieved. The layers include cloud controller, cloud scheduler, node controller, resource locator and instance manager. All the layers are sandwiched against cloud controller for effective communication across the cloud. The layers are defined based on the task that is to be accomplished in the cloud management [Jadeja Y (2010), Eucalyptus]. And the layers are organized based on the connectivity against other layers and considering future expansion.

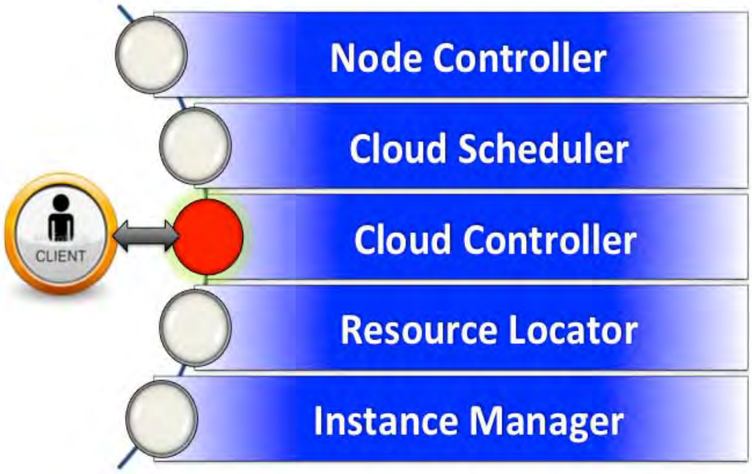

Figure 2: Layered architecture of cloud management module demonstrating the tasks that need to be accomplished during the client-cloud communication.

Each of the layers is equipped with a series of sub layers and modules signifying inter-layer connectivity, data management, and service balancing. Instance manager will comply with image generation along with number of instances to handle the client requests. Resource locator handles resources in the cloud at various virtual machines available along with the application transmission in and out of cloud. Cloud schedule engages in handling client requests and process them accordingly. The layer interacts with cloud controller and node controller for handing and managing the load [Zafarullahetet (2011)]. Node controller plays a major role in cloud load balancing. This layer is equipped with expansion levels, such that series of nodes can be used to grid up. These grids also help in instance generation from applications of SaaS and PaaS layers (academic services). Thus the layers form the internal structure of Cloud management through which cloud architecture can be effectively managed.

a) *Adaptation-to-Academia:* Modern cloud layers (discussed in pervious section) are always topped on their data centers. Usage of VMs across the data centers increases the interoperability of cloud usage [14]. In academic aspect, the partition of applications, software and research at architectural level with the facility of customization is feasible to a greater extent when compared with existing cloud platforms. Another issue that suffers from suitability is multi-tenancy. Deployment of Virtual Machines (VMs) and enabling sharing feature across the VMs will also correspond to the factors of efficient cost and hardware usage [Gao. B (2009)].

b) *Automation-to-Academia:* Adaptation of above features should enhance the efficiency of cloud platform in security, accessibility and adaptation of information processing and changes. When it comes to user, interface is the cloud. Hence, automation of cloud features should be initiated at all possible academic service levels defined in Figure 1. However, this automation can be effective with the customization of the services at its best for the efficient resource sharing and usage. The flexibility or automation integrated in the user interface will play a vital role in service access of cloud [Wei-Tek (2008)].

c) *Knowledge base-to-Academic:* Automation of services can only be efficient if the knowledge base created provides an efficient and less redundant information management. Knowledge base usually consist of research practices such that information confidentiality can be maintained and managed effectively across the cloud platform [Wyld (2009)]. When it comes to security, it depends on the organization and the rule-base adapted for this knowledge should be incorporated at this level. These policies should correspond to the university data protection act and also information infringement laws of the region, state and country. Knowledge base also comprises of lectures, presentations, notes and administration data. Knowledge base development considers academic, research and administration data along with information repository and knowledge sharing.

C. Mapping

Mapping across the layers and services in terms of hardware and software plays a critical role in cloud working based on functionalities such as capability, performance and efficiency [Abhishek Gupta (2011)].

Scalability adapted at different layers provides easy customization for services, which in this case is an added advantage. Similar is the case with services and grouping of users.

1) *User groups mapping:* By mapping working groups, application instances along with their generation, management and handling can be effective. This criterion is also successful in handling both security and instance management tasks of the particular application as shown in Figure 3. Additional advantage getting from this mapping will be speed of accessing. As the group size increases the chances of greater multi-tenancy is higher at both user and cloud levels

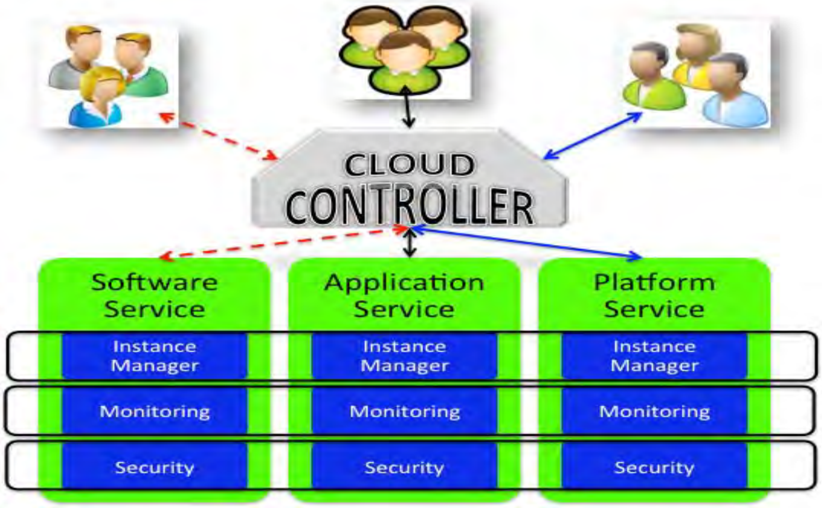

Figure 3: User groups based service mapping to the layers in the cloud. Instance manager plays a major role in handling instances of applications. With the grouping mechanism and introduction of instance manager, the service can be enhanced without compromising on load.

2) *Service groups mapping:* Service groups usually include the number of services that are associated with any particular application as shown in figure 4. The primary purpose is to entertain all possible services that applications can handle. In this aspect, a handler module is introduced within the service layer that can monitor the operational services from an application. Handler is made central and customizable, such that all the cloud applications can make use of it in assessing the service load. Group handler is included in the cloud service layer such that, the service load distribution can be made easy. Reason for introducing handler after instance manager, is to provide immediate services for client requests. Later the handler can group the services based on their usage and requirement.

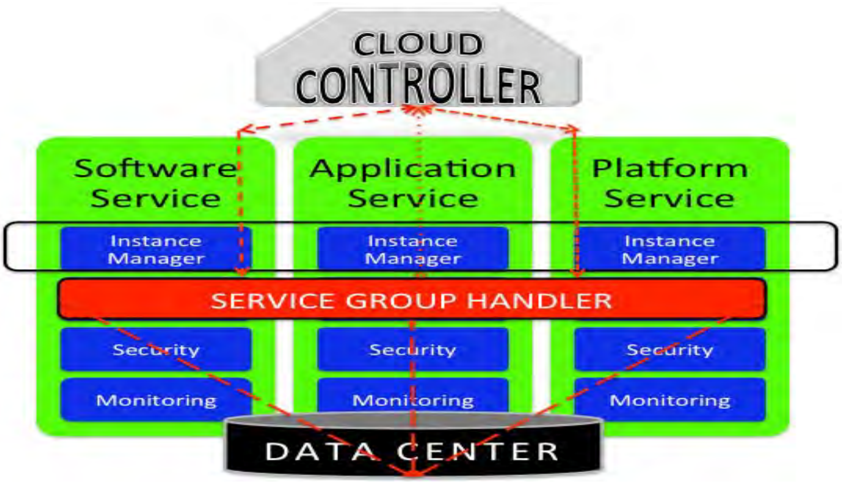

Figure 4: Service grouping signifying introduction of service group handler to group and handle service load across the cloud for all possible services.

3) *Application groups mapping:* Application mapping is associated with in the application. Number of services that an application is going to be associated with and mapping across the applications is also allowed to make effective use of resources and cloud handlers. A typical example (IT Services) of such application group mapping is detailed in the following figure 5.

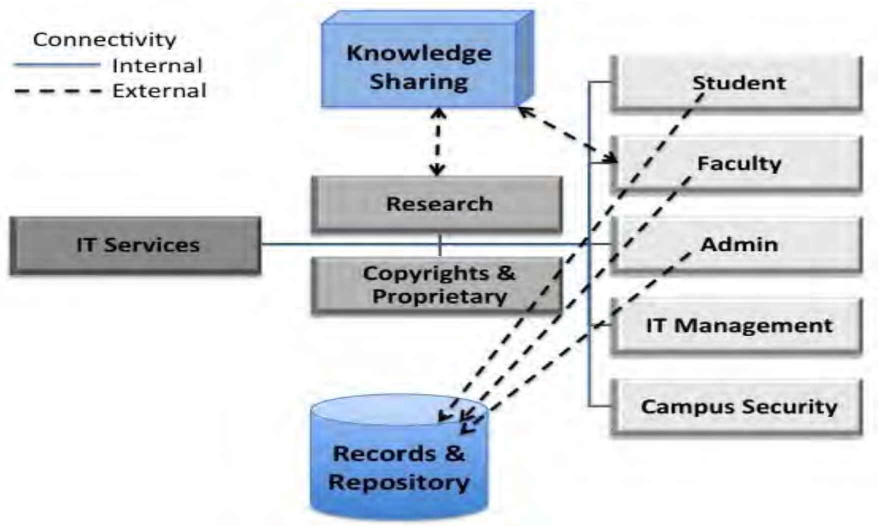

Figure 5: Application mapping signifying the services associated with 'IT services' both within application (internal) and among applications (external).

In this example, a quick view of services that can be offered by IT center of an academic organization is provided. All the rectangular boxes (users) are in direct connection (internal link) with IT. On the other hand, the service interconnectivity is also depicted on the image as 'external link' from other service providers such as knowledge sharing (other universities) and further more shown in the figure 5. This signifies that the mapping between the tasks, users and services is very essential from providing performance-oriented services.

D. Architectural Limitations

In cloud environments, limitations as usually associated with

- Amount of resources available & usage covering both physical and virtual machines,
- Quality of service or time taken by each request to process,
- Amount of services handled or maximum requests per application

So far, the cloud structure details the motive and intension behind the Project. In various stages what the cloud intends to do and what more can be done on an academic perspective is provided. In the following section, an architectural design of the discussed cloud structure is provided in detail.

4. Architectural Design

Division of cloud architecture services into layers makes this architecture a better option for developing customized cloud. Service layer provides an interactive interface for user as well as administrator. Administrator can customize cloud according to available resource and requirements. Inclusion of request scheduler and request dispatcher makes request processing efficient by separating process with respect to job for efficiency. Every user request is assigned a request id that is used at various layers. Requests are thus dispatched based on priority to group handler. In order to improve processing efficiency, requests are processed using grouping criteria set on the basis of rule base. These groups also minimize time required to identify application images along with resources. Group handler also provides initial level of security by applying security policies on groups. Each request is associated with a group id, which identifies group and thereby requests belonging to.

Resource manager provide optimal use of resource by maintaining resource information, usage and service requests in process and in queue. For maintaining effective services with the cloud environment, it is essential to have resource management. Resource manager also keep track of available service load so that an equivalent load can be distributed among cloud's resources. All the status of resource maintained in a registry and every time resource manager makes an entry into registry before allocating resources to request. Image manager is responsible for storage of images, allocation, updates and management. Image manager identify image for group instead of identifying for each request. Grouping of request save time needed to map request into image and make request processing faster.

Security manager provide second and core level of security, which apply security policies on every request separately. Since all requests finally associate with instance, these security policies are applied to instances. Security manager ensure accessibility of instance with valid resource and are not conflicting with other resource or instances. Finally instance manager manages aspects such as creation or execution of instance, destruction of instances. Twin layered security adapted in this architecture makes it more secure and efficient. In the following figure 6, the above-mentioned layers are schematically drafted.

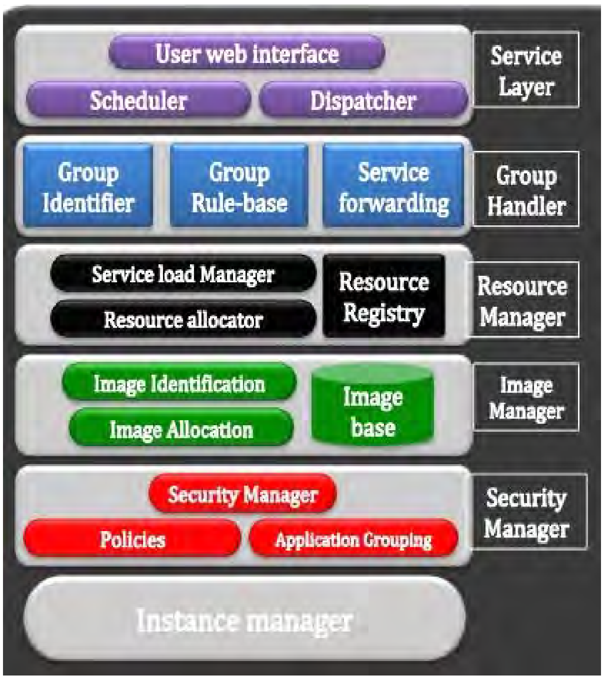

Figure 6:Proposed Layered Architecture of the Cloud consisting of all 6 layers.

The above provided cloud structure is redefined under this section in detail. A layered novel and generic, for academic institutions, architecture towards the development of a cloud environment is described that can be feasible to establish with minimal resources. This design provides the ease of development from localized cloud to wide cloud on a private platform such as academic institution. Each of the layers has definite tasks to process service requests and data moving from in and out of cloud. The advantage of using layered architecture lies in providing flexibility to developer in managing functionalities that the layer has to offer. The proposed architecture contains the following six layers:

1. Service layer
2. Group handler
3. Resource manager
4. Image manager
5. Security manager
6. Instance manager

A. Service Layer

Service layer is the abstract version of cloud services in terms of user. In user perspective, the concern is mainly on the services that cloud is supposed to offer rather than what is going on inside it. Service layer acts as an interface both for users and administrators from ‘what is going on’ inside cloud. Services concerning with platform, infrastructure and software (PaaS, IaaS, SaaS) are administered through this layer. Separation of connection handling mechanism using cloud’s interface with other layers, enables developer to isolate this layer's protocol development from underlying cloud architecture. Hence service layer is associated with user request handling mechanism.

For handling number of request efficiently, service layer is equipped with a request scheduler. Implementation of scheduler mechanism makes cloud service handling efficient, and independent of underlying cloud's layers. Greedy approach is most used (by default) to schedule user's request to provide immediate service. Request scheduler plays an important role in terms of performance of the architecture shown above. As requests are scheduled, dispatcher takes requests accordingly to their scheduler and dispatches them to next layer. Request dispatcher establishes connection between user's request and next layer. Request scheduler can also be considered as bridge between request scheduler and resource manager as indicated in the following figure 7.

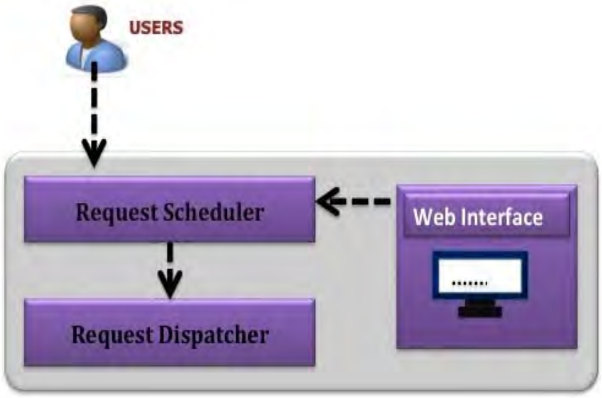

Figure 7**:**Service Layer block diagram representing the request transmission across the layer.

Division of request handler into request scheduler and request dispatcher increase performance by reducing load from scheduler. User is given opportunity to interact with cloud on a web-based interface to ease of the technical complexity with in cloud. This signifies, users should be aware of the services and their usage offered by the cloud environment rather than the technicalities with in it. Thus service layer can be considered as user interaction layer. Service layer's sole function is to provide better services from user point of view. Service layer has three sub modules

1. User web Interface
2. Request Scheduler
3. Request Dispatcher

Cloud Interface: Cloud interface provides an interaction platform for cloud in a web-browser environment. This web interface entertains both user and administrator interaction with cloud [Johnson D, 2010]. Administrator can do tasks such as:

- Creating new user account
- Managing existing account
- Uploading new application or operating system's image
- Start/Stop a service
- Accounting info maintenance
- Cloud Usage information

Request Scheduler: Service layer can expect a huge number of service requests arriving simultaneously. This number can vary from hundreds to thousands, depending on cloud service capacity. Request scheduler carries out scheduling of these requests and then handles them request-by-request. Round robin algorithm can be used to schedule incoming requests. All incoming requests are first buffered, and then scheduler picks these requests using scheduling algorithm. Capacity of buffer is defined by architecture/designer on the basis of load of cloud. After scheduling requests, they are stored in a queue ready for next step. Request dispatcher, associate with this queue dispatches request as per schedule and also considers the ongoing connections in the next layer.

Request Dispatcher: Request dispatcher acts as regulated transmitter of service requests to the next layer, group handler, of architecture. Request dispatcher take scheduled incoming request from scheduler. Dispatcher also contains the information about the occupancy of cloud, so as to efficiently manage the service requests. Frequency of picking scheduled requests for resource manager decides scalability of cloud architecture.

B. Group Handler

Group handler layer is associated with functionality to identify service domain from the user requests, depending on request type. Some cloud users take service from cloud for running application, while some use cloud services to store their data and access them later. For each request coming to cloud interface is mapped to an image, which in response is executed on virtual machine. Mapping of request to image is often time consuming. But if similar requests can be grouped before resource mapping, time required to process user's request could be comparatively reduced. Instead of mapping individual request to image, group of similar request is mapped to an image. Image identification phase identify image for group (using group id), although each request is processed separately but grouping make request processing faster when it comes to retrieval of service.

When it comes to processing, group handler forward requests' in a group to next layer. Each of the group requests is handled just like an individual request. When image manager receive request in form of group, image is identified for serving those request. However, image identification is performed for each group. Thus identified image is transferred from image repository to virtual machine, which is monitored by resource manager. Along with this image, number of requests' grouped is also sent to resource manager. When it comes to image initialization, the requests information will be helpful in determining the number of instances. For each request one instance is generated and that instance is associated with only that particular request.

In this cloud architecture, group handler layer also acts as primary level of security. Group handler initiates security constraints on group, which in turn carried to every request-associated within group. This mechanism is advantages in cases when a large number of instances required by an organization or people, all those instances are required to be inter-connected and communicate with each other. If all such requests can be combined into one group, group handler applies security policies on this group, which ensure that instance out of group, can't be accessed or communicated. Customization of such groups is possible with this policy and administrator has privilege to do so. Administrator can set rule so that services from same domain can be combined into one single group as shown in figure 8.

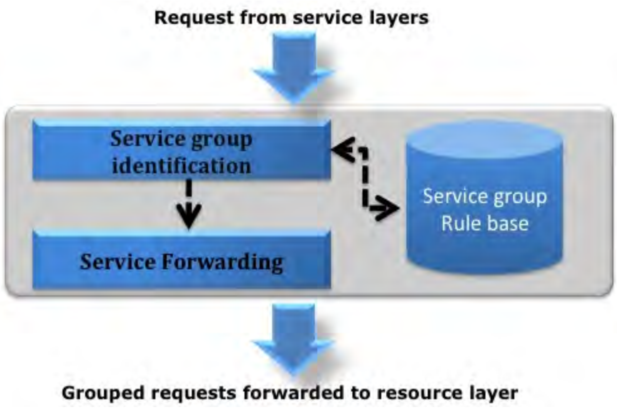

Figure 8: Group Handler Layer block diagram representing the request transmission across the layer.

Group handler has three following sub modules:

- Service Domain Identification
- Service grouping rule Base
- Service Forwarding

Service Domain Identification: Group handler initial step is identification of service domain. Group handler check request against rule base if requests satisfied grouping criteria then those requests are allowed for grouping. At this stage from the user request, information such as service name and request source is also obtained.

Service grouping rule base: Group handler also maintains grouping rule-base information, which is used while grouping of request into groups. Rule-base contains default rules along with specified sets that can be customized as per need and permissions. Grouping of these requests usually depends on their required service type. Administrator can change this rule base in order to customize grouping criteria. These rule are specified in following syntactic prototype:

<Request source, Type of Service, Service name, Security Requirement>

- Request source value can be same or different, which specify whether request coming from same source or different.
- Type of Service specify what services user want to invoke, which can be applications, software, platform such as operating system or infrastructure.
- Service name specifies name of service.
- Security requirement specify whether other user is authorized to access group's instance or not.

Service Forwarding: This phase involve sending grouped request with sufficient information to resource manager in next layer. Each group is forwarded with information such as number of requests; security policies associated with group, user-id of all requests present. This information will help resource manager to ensure proper allocation of sufficient resources for group. However, user information will help instance manager to bind instance with request along with security of cloud.

C. Resource Manager

Resource manager is intended for mapping of request to service. Since cloud environment may have limited resource, but with the help of virtualization it fulfills a large number of service requests. However, decision of assigning resources to service is entirely based on current availability of resources and service load. This architecture enables the resource manager for effectively handling resource distribution among various user requests thereby increasing the performance of service handling in cloud domains.

Resource management is carried out in three functionalities. Initially each resource available at the cloud should be registered by using a database. A registry function is used which provides a way to store current status of resource usage. In this registry current available virtual machine(s), storage available for storage manager, load on network are considered. This central database is regularly updated on a fixed time cycle, so that resource availability or resource load on the cloud can be obtained. This repository plays an important role for resource managing. If resource manager find's that allocation of more resources leads to decrement of performance, then resource manager can hold user's request (customize) without interrupting the cloud performance. Resource manager's goal is to offer better utilization of resources for handling as many services as possible, thereby increasing the efficiency of cloud.

Service-requests load handling consists the core functionality of resource manager layer as shown in the following figure 9. Resource repository contains information about what resources are in use and load on cloud.

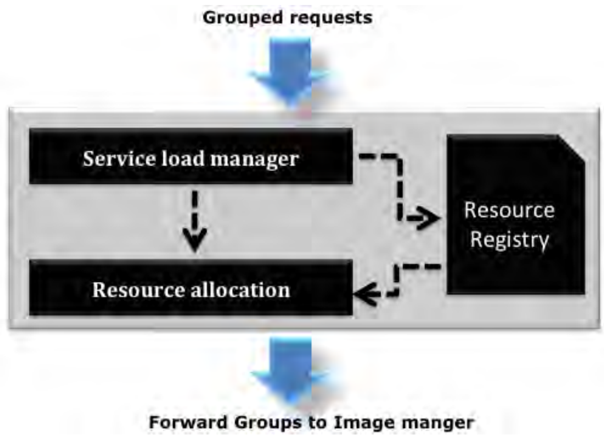

Figure 9: Resource Manager Layer block diagram representing the group request transmission across the layer with allocation of resources.

Based on this information, service load manager will take required decision, so as to determine to which machine request should be transferred. Thus obtained information, is then used for resource allocation. Then the service requests can be allocated to particular resource based on their availability or else set in queue. In terms of architecture, this layer has three sub modules.

1. Resource registry
2. Service Load Manager (SLM)
3. Resource allocator

Resource Registry: Resource registry maintains currently available resources. These resources include hardware, software and virtual machines. When service-request is assigned resource, an entry is made into resource registry. When those requests are completed, in that case also resource registry is updated. Registry is maintained on database server. On database server information regarding configured machines in cloud, their IP addresses, mac addresses are also stored.

Service Load Manager: Before resource allocation, service requests are passed through service load manager. Initially SLM carries out load check by requesting information from resource registry. The fetched details include resources states such as blocked, free, and waiting, based on which the current state of cloud usage is defined. On the availability of resource, request is transmitted to resource allocator else; Main role of SLM in cloud architecture is to provide a balanced distribution of load on cloud, so that cloud services can be efficient. Also, SLM module at architecture level will not suffer from increase of resources dynamically.

For each machine/resource deployed in cloud, load is determined by

- Number of requests handled by each machine
- Number of active requests
- Total CPU utilization

Architecture designer/developer determines and fixes maximum allowed load for each machine (depending on hardware), thereby providing customization of resources towards manageability. This information is contained within resource registry. Service load manager adapts limited timing queue technique to ensure the effective utilization within the load bounds. Service load manager make load distribution even on each machine and thereby to the cloud.

Resource Allocator: Resource allocator is responsible for allocating resources across the requests in the waiting queue. At this phase, decision of which resources are to be allocated to the service requests in the waiting queue is taken out. Once resources are finalized, then resource allocator updates the entry in resource registry and forward request to instance layer. Resource allocator then handle request to allocated resources. After handling user can start using resources.

D. Image Manager

In order to provide services of applications or platforms through cloud resources, application environments are supposed to be in ready to use form. This ready to use format or image is equivalent to installed application version. Image is a ready to use package of online running application. Such images uploaded to cloud by administrator have to be effectively managed for effective user service. Initially an indexing table is maintained for each uploaded image for effective lookup. This table is associated with attributes such as image ID, name and description. When it comes to locating the image, through these attributes it will be efficient. Upon locating the image, it has to be mapped through the application to image table. Upon mapping the image, image allocation is carried out. Image allocation involves initiating transfer of image to allocated user machine. From the user prospective, application is residing in the cloud. However, along with application images are also residing and they are given to the user, but not the original application. On the other hand, image manages handles the image management, processing and allocation as shown in the following figure 10.

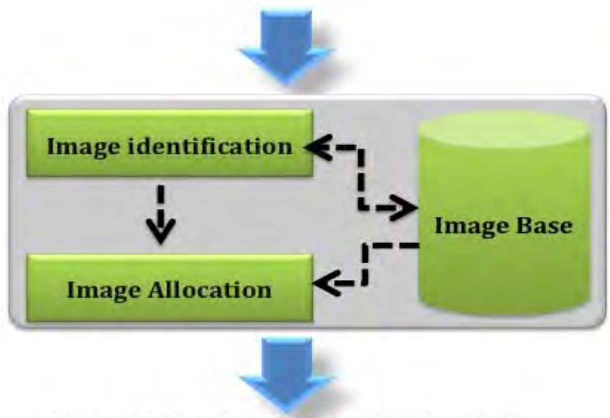

Figure 10: Image Manager Layer block diagram representing the group request transmission across the layer for mapping of image.

Hence image manager layer is organized with three modules to manage.

1. Image storage
2. Image identification
3. Image allocation

Image storage: Image storage is mainly associated with image repository maintained by database/storage server. Administrator defines applications, which can be provided as service, for which ready to use images' are needed. Administrator can customize building images using hypervisor either in physical or virtual machine. However, before building image for an application a virtual storage disk is created with defined processor type, memory and storage capacity. Usually depending on the application requirements the virtual storage attributes are set. However, storage is always categorized as a separate component in cloud, due to size and accessibility depending on size of users and load.

Image identification: Image identification purpose is to encapsulate image details from user. As and when user requests an application, mapping associated with identifying the specific image from the storage and towards the storage of application. Hence to improve the efficiency of this process, identification and mapping tables are maintained.

Image Allocation: Image manager is associated with cloud at server level, which enables effective look-up for client service requests. Image allocation initiates transfer of image from image storage to available physical or virtual machine. This module is also responsible for performance resource allocation task for the cloud at image storage, allocation and killing.

E. Security Manager

Now-a-days most organizations are shifting their services to cloud environments due to its minimal resource utilization, scalability and location transparency feature. To secure cloud from threats, an autonomous module for security is included in this cloud architecture. This layer is responsible for carrying out all security related activities throughout the cloud. Security manager is mainly associated with security of resources and their accesses. Since cloud services are highly targeted by many attackers on a regular basis for data / information, to secure cloud effectively from suspicious activities or attacks, security is maintained at layer as well along with group handler as shown in following figure 11.

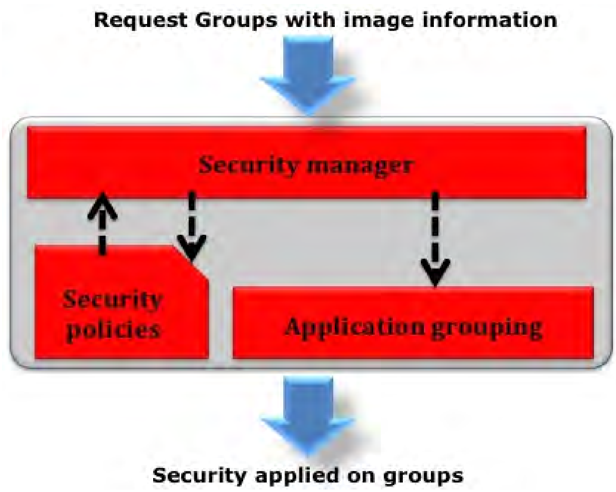

Figure 11: Security Manager Layer block diagram representing the group request transmission across the layer for mapping of image.

Security manager carries out the following criterion for maximizing cloud security:

- Instance access method
- Instance authentication
- Storage access control
- Cloud User privilege

Before accessing instance security manager ensure that user authenticate by providing user name and password assigned by cloud administrator. Only authenticated users can access application instance. Security manager can also customize instance usage, whether to run in isolation or not. For circumstance of high-level security, instances are required to run in isolation so that any attack from malicious user to cloud or other user's service, access can be denied.

Instance access methods strictly follow access control of user while obtaining access of instance. Instance execution is carried out by virtualization for effective resource utilization. So when instance access is given to user, amendments carried out often reflect on instance only. However the moment instance is killed the amendments are also cleared thus loosing data [Johnson D, 2010]. Hence, storage server solve this issue by providing storage for instances. After terminating virtual machine, data on storage server is retained for further use.

Security Policies: All security policies connected with the cloud environment are explicitly specified in rule-base file. Only administrator is authorized to access security files and customize accordingly. These policies depend on network security requirement, client requirement and attack control. Resource access constraints, privileges, data access control are components of policy file.

Application Grouping: Administrator usually applies set of constraints on a specific application for user access. Instead of applying security constraints on each and every user-access to application, all requests subject to same application are grouped without compromising on security. Security policies are then applied on such groups. These ensure same security on each access to application. Each request assigned a group id, this group id remain same for all requests for a specific application.

Security Manager: Security manager applies strict implementation of all security constraints specified in security policy file. Also, security manager is associated with monitoring resource access and maintain a log file for further reference.

F. Instance Manager

Instance management is final phase of request processing channel. Instances are generated whenever user-request comes for resources. Usually, a single instance is maintained per application in this layer to handle corresponding requests. However, as the number of requests for an application increase, its corresponding instances are generated accordingly [Eucalyptus, Avi kivity (2007)]. In this scenario, in order to manage and maintain state of an instance along with providing isolated environment for every such instance, instance manager is used in this layer. Activities involved in this layer are as follows:

- Instance initialization
- Instance destruction
- Instance isolation

Initialization of an instance mainly involves invoking virtual machine containing respective image of the instance. Each instance is equipped with a unique identification to differentiate. In order to maintain effective utilization of resources, instance manger maintains log containing address assignment given to all running instances. By doing so, the architecture will allow administrator to override any address assignment after starting instance for customization.

On completion of a service, the current instances corresponding to that service are subjected to destruction. Instance destruction involves release of occupied resources. Through the criteria of instance destruction, congestion among services in terms of instances can be avoided, and can maintain security and efficiency. Instance manager plays a vital role in managing instance isolation, thereby securing the cloud from attackers and un-authorized users.

Instance isolation criteria ensure that there is a complete isolation of every instance created from an image. Communication among instances monitored by instance manager. Instance isolation depends on type of user's request. If two instances are initialized to fulfill two different request then those instance will run in totally isolated environment, so that one instance can't not access other instance in terms of data and programs. From an attacker point, becoming an authenticated user is main break-through to any cloud environment, using which attacker tries to reach cloud users using instance communication channel. Due to involvement of instance manger, which prevents inter-instance communication, will provide security to cloud users. However, this architecture allows for customization of communication at various levels of service management. Hence the application of instance isolation is customizable for optimizing both security and performance.

5. Conclusion

In this paper, a generic low-cost based architecture is proposed specific to various types of academic institutions for moving to cloud-based environments. Initiated with a conceptual examination of feasibility and applicability towards the development of such cloud environments. Issues surrounded in existing clouds and risks associated with them in terms of workload, infrastructure and requirements are analyzed. The proposed architecture is motivated to improvise the service-time using user grouping and group handling criteria that are introduced in '*group handler layer*'. The layered architecture has successfully adapted customization feature at critical layers like, group handler, load manager instance manager and security manager and minimized the overall cost of the cloud with minimal resources. Finally the architecture proposed through this paper has justified a stable and feasible platform for motivating universities to bridge the gap between student, staff and academia with cloud technology.

References

[1] Abhishek Gupta; Jatin Kumar. (2011) *"Design and implementation of the workflow of an academic cloud"* 7th International Workshop, DNIS2011, Proceedings in Springer-Verlag Berlin Heidelberg 2011, Pp. 16-25

[2] Avi Kivity, Yaniv Kamay, Dor Laor, Uri Lublin, Anthony Liquori (2007) "*A.: kvm: the linux virtual machine monitor*" In Ottawa Linux Symposium, Pp. 225-230

[3] Catteddu D.; Hogben G. (2009) "*Cloud Computing: Benefits, Risks and Recommendations for Information Security*", European Network and Information Security Agency (ENISA). http://enisa.eurpoe.eu/act/rm/files/deliverables/cloud-computing-riskassessment

[4] Eucalyptus, Eucalyptus Cloud Architecture, "http://www.eucalyptus.com/eucalyptus-cloud/iaas"

[5] Gao B.; Guo C.; Wang Z.; An W.; Sun W. (2009). "Develop and Deploy Multi-Tenant Web-delivered Solutions using IBM middleware: Part 3: Resource sharing, isolation and customization in the single instance multi-tenant application". IBM Corporation

[6] Golden, B (2012) "What Cloud Computing Can Do for Higher Education", CXO Media Inc.

[7] Goldstein P.J (2009) "*Alternative IT Sourcing Strategies: From the Campus to the Cloud*", EDUCAUSE Center for Applied Research. [Online Retrieved Nov 10, 2012] http://www.educause.edu/library/resources/alternative-it-sourcing-strategies-campis-cloud%E2%80%93key-findings

[8] Jadeja Y. and Modi K. (2012) "*Cloud computing - concepts, architecture and challenges*" Proceedings of International conferences on computing, electronics and electrical technologies,Pp. 877-880

[9] Johnson D, Kiran murari; Murthy raju (2010) "*Eucalyptus Beginner's Guide UEC Ediiion*" v1.0 Published by CSS Citrix "CloudStack Installation guide"

[10] Karpagamet G. R. al. (2011) "Setting up of an Open Source based private cloud" IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 1. "*An Introduction to private cloud*" Whitepapers at amilelive.com http://tech.amikelive.com/whitepapers/An_Introduction_to_Private_Cloud-whitepaper.pdf

[11] Kartz, R.; Goldstein, P.; Yanosky, R. (2010) "*Cloud Computing in Higher Education*", EDUCAUSE. [Online, Retrieved Oct 20, 2012]

[12] Mahjoub M.; Mdhaffar A.; Halima R. B.; Jmaiel M. (2011) "*A comparative study of the current cloud computing technologies and offers*", Proceedings of First international symposium on Network Cloud Computing and Applications (NCCA), Pp. 131 –134

[13] McCrea, B. (2009) "*IT on Demand: The Pros and Cons of Cloud Computing in Higher Education*", Campus Technology. [Online, Retrieved Oct 2, 2012]

[14] Mcirvine S. (2010) "Making Cloud Computing Safe Trust, Security, Resiliency, Availability and Complexity", IBM Corporation.
[15] Mircea, M (2010) "*SOA, BPM and Cloud Computing: Connected for Innovation in Higher Education*", 2010 International Conference on Education and Management Technology (ICEMT 2010), November 2-4, Cairo, Egypt, ISBN: 978-1-4244-8617-5
[16] Mircea, M.; Andreescu, A. J. (2010a) "*Agile Systems Development for the Management of Service Oriented Organizations*", 11th International Conference on Computer Systems and Technologies, CompSysTech'10, Sofia, Bulgaria, 17-18 June 2010, ISBN: 978-1-4503-0243-2, 341-346
[17] Mircea M.; Andreescu A.I. (2011) "Using Cloud Computing in Higher Education: A Strategy to Improve Agility in the Current Financial Crisis", Communications of the IBIMA, Vol: 2011, Article ID: 875547.
[18] Mircea, M.; Ghilic-Micu, B.; Stoica, M. (2011a) "*Combining Business Intelligence with Cloud Computing to Delivery Agility in Actual Economy*", Journal of Economic Consumption and Economic Cybernetics Studies, In Press
[19] Ojala, A. (2012). "*Comparison of different revenue models in SaaS*". In E. Prakash (Ed.), Proceedings of 5th Computer Games, Multimedia & Allied Technology Conference (CGAT 2012) (pp. 120-123). Singapore: GSTF.
[20] Patterson, D. (2010) "*Cloud Computing and the RAD lab*", UC Berkeley Reliable Adaptive Distributed Systems Lab. [Online, Retrieved Oct 4, 2012]
[21] Sasikala, S.; Prema, S. (2010) "*Massive Centralized Cloud Computing (MCCG) Exploration in Higher Education*", Advances in Computational Sciences and Technology, 3 (2), 111-118
[22] Tout S.; Sverdlik W.; Lawver, G (2009) "*Cloud Computing and its Security in Higher Eduction*", Proc ISECON, Vol. 26 Tuncay E. (2010) "*Effective use of cloud computing in educational institutions*" Procedia-Social and Behavioral Science, Vol. 2, Issue. 2, Pp. 938 – 942
[23] Wei-Tek Tsai; Qian Huang; Jay Elston; Yinong Chen, (2008) "*Service Oriented User Interface Modeling and Composition*", International Conference on e-Business Engineering, Xi'an, Pp. 21-28
[24] Wei-Tek Tsai; Xin Sun; JanakaBalasooriya, (2010) "*Service-Oriented Cloud Computing Architecture*", IEEE Computer Society, 7th International Conference on Information Technology. Pp. 684-689
[25] Wyld D.C. (2009) "*Cloud Computing 101: Universities are Migrating to the Cloud for Functionality and Savings*", Computer Sight in Programming dated 23-11-09. [Cited on 20-11-12] http://computersight.com/programming/cloud-computing-101-universities-are-migrating-to-the-cloud-for-functionality-and-savings/
[26] Zafarullahetet (2011) "Digital Forensic for Eucalyptus" Proceedings of the 2011 Frontiers of Information Technology, Pp. 110-116