Unified Master Service Catalogue Manager for Cloud

Prarthana Gupta

Abstract: ***Cloud computing takes the technology services and applications that are similar to those on the Internet and turn them into a self-service utility. It virtualizes systems by pooling and sharing resources and abstracts the details of system implementation from users and developers. Costs are assessed on a metered basis, multi-tenancy is enabled, and resources are scalable. Clouds can come in many different types, and the services and applications that run on the clouds are delivered by service provider. It is very important task to define services requirements and to choose service providers appropriately. Selection of a best service provider is a complex process, as it requires lots of efforts for analyzing providers policies, rules, regulations, validations security features etc. out of numbers of Providers.***

In this paper, we propose standardized way of representing services offered by the providers in a single unified master service catalogue, thus enabling faster and easier approach of searching a best suited cloud service provider. A master service catalog categorizes the service types along with some search options needed to be selected by customer such as customer's budget, term needed, availability requirement, monitoring facility required, security level requirement etc. All Information regarding service provider i.e. name, location, availability, scalability, elasticity, services types, security features, cost, rules, validations, policies are stored in a Real time database, and are linked with master service catalogue. Customers selected options are matched with the service providers' information as submitted by them in real time database, using search queries and thus optimal selection will be displayed consisting of list of some suited Service providers. Customer now selects best provider using payment gateway. This will help the customer to initiate the computing by selecting services easier and faster, thus reduces the burden of selection process.

Index Terms: Cloud computing, Cloud services, Infrastructure as a service, cloud architecture, on demand, pay-per-use.

I. INTRODUCTION

A. Background

Cloud Computing is an abstraction based on the notion of pooling physical resources and presenting them as a virtual resource. It is a new model for provisioning resources, for strategic applications, and for platform- independent user access to services. Cloud computing can be thought as ability to share computing resource among many different users [1].Cloud computing makes the long held dream possible with a pay-as -you-go, infinitely scalable, universally available system. With cloud computing, you can start very small and become big very fast. That's why cloud computing is revolutionary, even if technology it is built on is evolutionary.

B. Categorization of Services Types

Cloud Computing provides us three types of services as SaaS, PaaS and IaaS. Service choice depends on the type of user. With SaaS, the burden of software installation, maintenance, and support is practically eliminated from the client domain. From the vendor's viewpoint, SaaS establishes an ongoing revenue stream for the software developer. PaaS provisioning reflects a business model wherein one enterprise supplies PaaS hosting to one or more SaaS enterprises. In the provisioning package, the PaaS provider would supply advanced security measures and other necessary infrastructure facilities never actually seen, per se, by the cloud client. In this scenario, the SaaS enterprise is then solely responsible for cloud software development.

For example, end user prefers SaaS (Software as a Service) which works at application level, Developers prefer PaaS (Platform as a Service) which provides a platform, development environment for the developers and works at network level of the system, Businessmen prefers IaaS (Infrastructure as a Service) which satisfies the infrastructural need (e.g. storage) and works at Physical level of the system, there may be needs for hybrid types too.

C. Cloud Service Characteristics

Cloud service utilities are characterized by four key factors: necessity, reliability, usability, and scalability. Necessity refers to the idea that a preponderance of users depend on the utility to satisfy everyday needs. Reliability refers to the expectation that the utility will be available when the user requires it. Usability refers to the requirement that the utility is easy and convenient to use – regardless of the complexity of the underlying infrastructure. Scalability refers to the fact that the utility has sufficient capacity to allow the users to experience the benefits of an expandable utility that provides economy of scale. Certainly, modern Internet facilities for search operations that engage thousands of servers satisfy these characteristics.

The notion of "paying for what one uses" is a compelling argument for using the cloud for special or all computing needs. The proof may be in the details. The key question is whether the service should be based on a metered model or a subscription model. With the metered model, the usage is easily measured, monitored, and verified and lends itself to managerial control on the part of the user. In addition, metering can be applied to

Manuscript received on July, 2012
Prarthana Gupta, Department of Computer Science, Shriram Institute of Technology, Rajiv Gandhi Proudyogiki Vishwavidyalaya., Jabalpur, India.

differing levels of service. With the subscription model, usage is difficult to control and monitor and its adoption is favored by managers more concerned with convenience than with resource control.

A proper, but not necessarily definitive, conceptualization of cloud computing is to use office-class applications via your web browser over the Internet instead of having those applications reside on your "on premises" computer. In this instance, the service provider supplies the network access, security, application software, and data storage from a data center located somewhere on the Internet and implemented as a form of server farm with the requisite infrastructure. A service would have universal access through a web browser. In general, the cloud computing concept is not limited to single-function applications, such as those available with typical office suites, but could include comprehensive enterprise applications pieced together from components residing in varying Internet locations.

An Infrastructure as a Service (IaaS) cloud computing is computing concept in enterprises from viewpoints of both a server user who makes use of servers and a server administrator who prepares servers for server users and manages them.

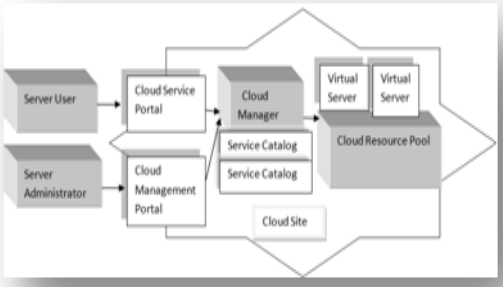

Figure1: IaaS Infrastructure as a Service.

Figure 1 shows a general IaaS cloud site. A server administrator creates virtual server templates, and enrolls them as service catalogs into a cloud manager in advance, using a cloud manager portal. Server users open a cloud service portal and select an appropriate service catalog when they would like to use servers. Then, the service catalog request is sent from the service portal to the cloud manager. The cloud manger creates a new virtual server on cloud resource pool by copying the template linked with the requested catalog, and informs the users that requested virtual server is ready. A server administrator manages virtual servers on the cloud management portal. Server users can immediately make use of virtual servers, exactly when they want without contacting a server administrator. A server administrator does not need to answer user's requests each time and can manage servers centrally.

A multi IaaS cloud site is appearing as cloud computing benefits spread .As the result, both a server user and a server administrator are annoyed because cloud sites are separated (Figure 2). First, depending on a user's requirement, he has to select an appropriate service provider having required cloud site and service catalog, using a different cloud service portal located on each cloud site. Second, a server administrator needs to manage virtual servers with different manners using a different cloud management portal on each cloud site.

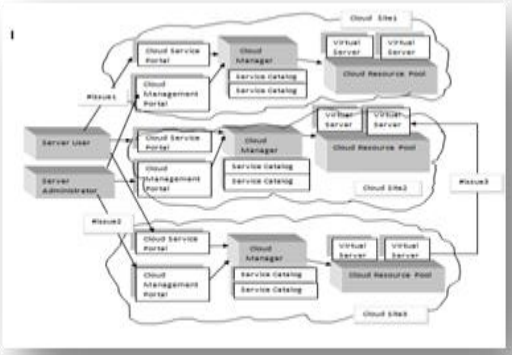

Figure2: Issues with Multi IaaS platform.

One of the important factors in managing a cloud is to ensure a way to manage IT assets and activities. Enter the service catalog (defined as part of the Information Technology Infrastructure Library [ITIL] Service Design best practice). The catalog is a component list that makes up internal and external services that are available to an organization. A typical service catalog includes such items as the definition of the service, its service level, which is entitled to use that service, and what components are required to execute that service. Clearly, a service catalog is required for organizations to manage services in a hybrid world across data centers, private and public clouds, as well as hosted environments. The service catalog is an essential tool for both cloud providers and customers that need a view into the assets they are using. Many cloud providers package a service catalog to help their customers work between their cloud and external resources.

II. CONCEPT

A. Our contribution

In this paper, we propose an approach by creating unified master service catalog manager a platform which enables the end users to choose required services using payment counter. Unified master service catalogue manager unifies the delivery and lifecycle management of public and private cloud services across the enterprise including cataloguing, provisioning, security, auditing, monitoring, reporting, metering, billing, administration, and user support thus enabling the customer to get proper information and benefit from the available services.

Initially the services are selected from different service catalogues located at different cloud sites as provided by their providers. This involves complex selection criteria, as analysis of service providers policies, rules, validations, security features, scalability

features, market value (it may not a fake), availability etc. For big organizations this hardly matters, but for small scale industries this cost matters a lot.

In order to overcome this problem we proposed an approach by representing Unified Service Master Catalogue Manager a platform. The key Cloud Services lifecycle management features that this Platform enables for enterprises include:

B. Unified Master Service Catalogue Manager comprises of following selection criteria:

- Categorization of services-service types
- Budget
- Availability Requirements
- Term Requirement
- Monitoring facility
- Interoperability
- Portability
- Validations
- Scalability
- Elasticity- Load Balancing elasticity and Elastic IP addresses, Elastic Storage facility.
- Bandwidth Requirement
- Billing Features.
- Reliability
- Security
- Instances Type Requirement

These features are discussed in details as Categorization of services:

Basic Services Types: Included are categorized as Application based Services, Technical Based Services, Hybrid Services .These categories itself includes number of services that a cloud can provide. According to customer needs and demands options are chosen from the given list. A master service catalog manager handles the services types and along with other measured criteria, displays an optimal service cloud provider to fulfill required service needs.

Application Based Services further can be classified into other services needs as shown:

Figure3. List of Application based Services.

Technical Services can be classified as:

Figure4. List of Technical based Services.

Hybrid Services includes mixed type of services may be application and infrastructural based, or may be other depends on the needs of customer.

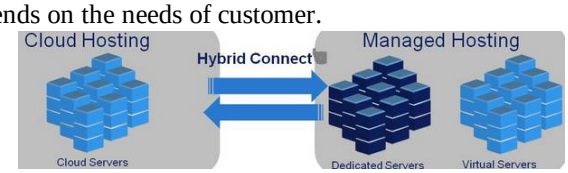

Figure5. Hybrid Service Types

Budget: Budget specifies the cost that customer could pay for the services.

Availability Requirements: Availability is the probability that the system is running without interruption at any point during scheduled time, which is a tough requirement but can be met .For many companies, particularly those with global operations, the scheduled time for business continuity reasons is twenty-four hours per day, and this practically means that at any time per day must be applied the algorithm

%Availability = System Usage Time / Scheduled Time.

System usage time (SUT) is also known as uptime, but the latter term is often employed in an ambiguous way. The equation is simple:

SUT = Scheduled Time – System Downtime.

For the global enterprise, scheduled time is equal to twenty-four hours per day. Local companies have a shorter time frame; they should, however, also account for legacy batch procedures (the old technology still alive), which may require twenty-four hours per day scheduled time for business continuity reasons anyway. A good part of system downtime is due to repair. Therefore, in connection to cloud computing, user organizations must carefully examine what the vendor puts into system downtime, because this will define the interruptions of access to the cloud's infrastructure (or on Demand software and platforms, depending on the nature of the SLA). Partial failures will be visible by the user organization only when they affect the response time hence the latter's importance.

Term Requirement: This specifies the time period of service as required by the customer. It may be 1 month, a quarter, Half year, 1 year, 2 year and 3 year or as specified by customer.

Monitoring facility: A cloud service provider must include a service management environment. A service management environment is an integrated approach for managing your physical environments and IT systems. This environment must be able to maintain the required service level for that organization. In other words, service management has to monitor and optimize the service or sets of services. Service management has to consider key issues, such as performance of the overall system, including security and performance. For example, an organization using an internal or external email cloud service would require 99.999 percent uptime with maximum security. The organization would expect the cloud provider to prove that it has met its obligations.

More than maintainability, the target is sustainability, and this includes not one but three overlapping processes: maintainability proper is the extent to which preventive maintenance can be performed without degrading availability repair ability refers to the speed with which a component failure can be detected and fully corrected and recoverability identifies what happens at the after-correction time of a component failure, specifically the speed with which system facilities can be restored, including recovery of data damaged as a result of failure.

Interoperability: Interoperability refers to cloud users being able to take their tools, applications, Virtual images, and so on and use them in another cloud environment without having to do any rework. Say one application runs in one environment and you need that application to operate with a partner's application in another cloud environment. If the right interoperability standards are in place, you can do this without needing multiple versions of this application. Simple Object Access Protocol (SOAP), representational State Transfer (REST), and Atom Syndication Format and Atom Publishing Protocol (both standards referred to as Atom) are all examples of widely used interoperability standards and protocols.

Portability: Portability lets you take one application or instance running on one vendor's Implementation and deploy it on another vendor's implementation. For example, you might want to move your database or application from one cloud environment to another.

Validations: confirming that a product or service meets the needs of its users. Validation is intended to check that development and verification procedures for a product, service, or system result in a product, service, or system that meets initial requirements, specifications, and regulations. For a new development flow or verification flow, validation procedures may involve modeling either flow and using simulations to predict faults or gaps that might lead to invalid or incomplete verification or development of a product, service, or system. A set of validation requirements, specifications, and regulations may then be used as a basis for qualifying a development flow or verification flow for a product, service, or system.

Scalability: The service needs to be available all the time (7 days a week, 24 hours a day) and it has to be designed to scale upward for high periods of demand and downward for lighter ones. Scalability also means that an application can scale when additional users are added and when the application requirements change.

Elasticity: enables you to increase or decrease capacity within minutes, not hours or days. One can acquire one, hundreds or even thousands of server instances simultaneously. As, all this is controlled with web service APIs, an application can automatically scale itself up and down depending on its needs.

Elastic IP addresses are static IP addresses designed for dynamic cloud computing. An Elastic IP address is associated with a particular account not with a particular instance, and one can control that address until chosen for explicit release. Elastic IP addresses allow us to mask instance or Availability Zone failures by programmatically remapping public IP addresses to any instance in an account.

Elastic Load Balancing: It enables you to achieve even greater fault tolerance in your applications, seamlessly providing the amount of load balancing capacity needed in response to incoming application traffic. Elastic Load Balancing detects unhealthy instances within a pool and automatically reroutes traffic to healthy instances until the unhealthy instances have been restored. Customers can enable Elastic Load Balancing within a single Availability Zone or across multiple zones for even more consistent application performance.

Bandwidth Requirement: Bandwidth can be defined as the net bit rate, information rate or physical layer useful bit rate, channel capacity, or the maximum throughput of a logical or physical communication path in a digital communication system. For example, bandwidth tests measure the maximum throughput of a computer network.

Billing features: A cloud environment needs a built-in service that bills customers. And, of course, to calculate that bill, usage has to be metered (tracked). Even free cloud services (such as Google's Gmail or Zoho's Internet-based office applications) are metered. This facility , if chosen by customer from master service catalogue manager will help in customizing billing, auditing, accounting and report management features of service providers.

One of the bigger challenges for an enterprise consuming many different Services from a number of external Cloud Service Providers and from internal IT, is how to consolidate billing, reconcile different payment terms, measure actual usage of Services by department and then enable accurate departmental charge backs to the appropriate internal cost centers.

Here are a few scenarios to help illustrate the complexity enterprises might face:

i. An enterprise uses many services from many different Cloud Service Providers each with its own billing cycle and invoice terms. For example, Service A is billed every 2nd of the month at Net 30 whereas Service B is billed on the 30th of the month at Net 45. Meanwhile, allocation to internal cost centers is done at the end of the month.

ii. Two departments within the enterprise have independently signed up for the same Service from an external Cloud Service Provider. Another department would now like to sign up for the same Service but IT has now been asked to centrally manage the administration, metering and provide chargeback statements for usage of the Service across all the departments.

iii. Internal IT creates a private IaaS Cloud offering and needs to allocate usage-based costs to departments. This Platform provides a comprehensive set of processes to handle the above scenarios and more.

Reliability: Reliability is not ability. It is the probability that a given system will operate without failure over a predetermined time period, under environmental and operational conditions defined in advance. Such conditions must be elaborated on the basis of studies that look at each system component and at the whole aggregate.

Security: Cloud security has to be a part of your company's overall security strategy. Most companies place a high priority on the testing and monitoring of threats to their data center, buildings, people, and information.

Even when cloud operators have good security physical, network, OS, application infrastructure), it is your company's responsibility to protect and secure your applications and information. Security services at both the application and the

infrastructure level must be a top consideration for organizations. Given the importance of security in the cloud environment, you might assume that a major cloud services provider would have a set of comprehensive service level agreements for its customers. In fact, many of the standard agreements are intended to protect the service provider not the customer. Therefore, a company really must understand the contract.

To avoid all these problems security features are well studied and only those service providers which are secure are included in the master service catalogue manager.

Instances Type Requirement: Service needs at specific times can be represented as instance type requirement. It can be on-demand, reserved, high CPU instance, high memory instance, high gpu instance etc.

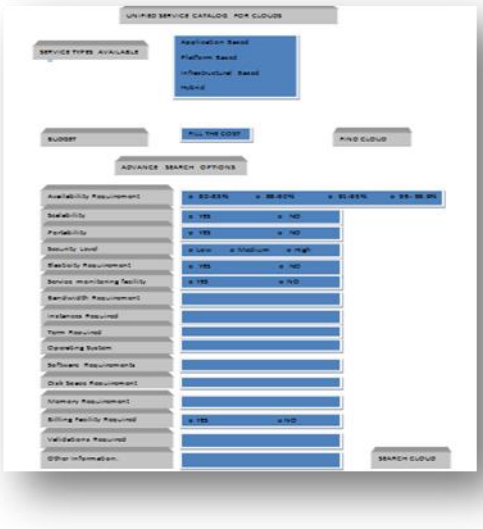

Figure6. Unified Master Service Catalog for cloud search

C. Service and user provisioning

Centralized Self-Service -The Platform enables a central portal for end-users within an enterprise from there they can select and self-provision a service. In a similar manner it provides a central portal for administrators to monitor and manage the on-boarding of end-users according to company policies. As a result, the Platform Significantly reduces the overall cost and of provisioning services to end-users while ensuring consistent policy enforcement across the organization..

III. DATABASE DESCRIPTION

Real time Database is created and, is linked to master service catalog manager for further updates using XML tags. Information regarding service provider must be stored in real time database using database commands(for e.g. by creating tables, inserting values, assigning keys, relationships and queries for further updates and selections) in any real time database management system. The form can be structured as consisting of following fields which must be entered by Service Provider are as

- Service Provider Name
- Location
- Types of Services provides: Application based (SaaS), Infrastructural based (IaaS), Platform Based (PaaS), Hybrid Type.
- Availability: generally it should be in between 80.00% to 99.9%.
- Portability
- Flexibility
- Elasticity: load balancing, elastic IP address, elastic storage capability.
- Scalability
- Security features: Multifactor authentication availability, firewall and switches, Network security, Certifications - ISO and others, etc.

- Cost Details- pricing models, CPU cycle/hr, hardware usage/hr, On-demand instances/hr, Maintenance / month, Monitoring/ month, Bandwidth Usage/ hr, Billing/Accounting/Auditing charges/ month, other charges.
- Instances Types available - Standard, Micro, High Memory, High CPU, High Cluster, High GPU etc.
- On Demand Instances Types
- Throughput
- Operating System Offered: Red Hat Linux, Gentoo Linux, Open Source Linux, Oracle Enterprise Linux, Sun Open Solaris, Windows Server /2003/2008 32-bit and 64 bit, Fedora , others.
- Software Offered: Application Development Environment based Software, Application Server Based Software, Batch Processes, Databases Software, Video Enhancing and Streaming Software, Web hosting Software, Others.
- Disk Space Availability: Small approx 300GB, Medium approx 600GB, Large approx 1000GB, Extra Large approx 2000GB.
- Monitoring Facilities
- Billing, metering, auditing, accounting facility.
- Interoperability
- Bandwidth Limit (in and out).
- Storage Capability
- Virtual Services
- Offering Type
- Validations and Regulations.
- Other Information.

A. Database Description

A computer database relies upon software to organize the storage of data. This software is known as a database management system (DBMS). Database management systems are categorized according to the database model that they support. The model tends to determine the query languages that are available to access the database. A great deal of the internal engineering of a DBMS, however, is independent of the data model, and is concerned with managing factors such as performance, concurrency, integrity, and recovery from hardware failures. In these areas there are large differences

between products.

A Relational Database Management System (RDBMS) implements the features of the relational model outlined above. In this context, Date's "Information Principle" states: "the entire information content of the database is represented in one and only one way. Namely as explicit values in column position (attributes) and rows in relations (tuples). Therefore, there are no explicit pointers between related tables."

Most relational DBMS's and some object DBMSs have the advantage that indexes can be created or dropped without changing existing applications making use of it. The database chooses between many different strategies based on which one it estimates will run the fastest. In other words, indexes are transparent to the application or end-user querying the database, while they affect performance, any SQL command will run with or without index to compute the result of an SQL statement. The RDBMS will produce a plan of how to execute the query, which is generated by analyzing the run times of the different algorithms and selecting the quickest. Some of the key algorithms that deal with joins are nested loop join, sort-merge join and hash join. Which of these is chosen depends on whether an index exists, what type it is, and its cardinality.

An index speeds up access to data, but it has disadvantages as well. First, every index increases the amount of storage on the hard drive necessary for the database file, and second, the index must be updated each time the data are altered, and this costs time. (Thus an index saves time in the reading of data, but it costs time in entering and altering data. It thus depends on the use to which the data are to be put whether an index is on the whole a net plus or minus in the quest for efficiency.)

A special case of an index is a primary index, or primary key, which is distinguished in that the primary index must ensure a unique reference to a record. Often, for this purpose one simply uses a running index number (ID number). Primary indexes play a significant role in relational databases, and they can speed up access to data considerably. Foreign Indexes can also play a very important role by creating a relationship in between tables, allowing easier selection and answering the queries.

In addition to their data model, most practical databases transactional databases attempt to enforce a database transaction. Ideally, the database software should enforce the ACID rules, summarized here:

• Atomicity: Either all the tasks in a transaction must be done, or none of them. The transaction must be completed, or else it must be undone (rolled back).

• Consistency: Every transaction must preserve the integrity constraints the declared consistency rules of the database. It cannot place the data in a contradictory state.

• Isolation: Two simultaneous transactions cannot interfere with one another. Intermediate results within a transaction are not visible to other transactions.

• Durability: Completed transactions cannot be aborted later or their results discarded. They must persist through (for instance) restarts of the DBMS after crashes.

Database security: denotes the system, processes, and procedures that protect a database from unintended activity. Security is usually enforced through access control, auditing, and encryption.

• Access control ensures and restricts who can connect and what can be done to the database.

• Auditing logs what action or change has been performed, when and by whom.

• Encryption: Since security has become a major issue in recent years, many commercial database vendors provide built-in encryption mechanisms. Data is encoded natively into the tables and deciphered "on the fly" when a query comes in. Connections can also be secured and encrypted if required using DSA, MD5, SSL or legacy encryption standard.

Enforcing security is one of the major tasks of the DBA. The databases are the most reliable source where data can be stored in an organized and secured form. The Database created must consist of following entities, attributes and data types as shown:

Service Provider		**Service Type**	
SP_ID	PKey (U)	S_Id	PKey
SP_Name	Text	SP_ID	FKey
Location	Text	Application based	Text
Availability	Text	Infrastructural based	Text
Portability	Text	Platform based	Text
Flexibility	Text	Hybrid type	Text
Scalability	Text		
Monitoring facility	Text	**Service Elasticity**	
Bandwidth limit	Num	E_Id	PKey
Interoperability	Text	SP_ID	FKey
Virtual-services	Text	Load balancing	Text
Storage capacity	Num	IP_Address	Text
Throughput	Text	Storage flexibility	Text
Security		**Reliability**	
Sec_Id	PKey	R_ID	PKey
SP_ID	FKey	SP_ID	FKey
Multifactor authentication	Text	Low	Text
ISO certification	Text	Medium	Text
Firewall network switches	Text	High	Text
Other Certification	Text		
		Operating System	
Cost		OS_Id	PKey
C_Id	PKey	SP_ID	FKey
SP_ID	FKey	Red_Ht_Ent_Linux	Text
CPU Cycle/hr	Num	Gentoo_linux	Text
Hardware Usage/hr	Num	Open_Source_linux	Text
On demand_Instances	Num	Oracle_Ent_linux	Text
Reserved_Instances/hr	Num	Sun_Open_Solaris	Text
Storage/GB	Num	Fedora	Text
Maintenance/month	Num	Windows Server2003/2008	Text
Bandwidth usage/month	Num	Others	Text
Billing Accounting Metering/month	Num		
Other charges/month	Num	**Memory**	
		Mem_Id	PKey
Disk Space		SP_ID	FKey
D_ID	PKey	Small_Instances	Text
SP_ID	FKey	Medium	Text
Small	Text	Large	Text
Medium	Text	Extra_large	Text
Large	Text		
Extra_Large	Text	**Software**	
		Soft_Id	PKey
Instance		SP_ID	FKey
Ins_Id	PKey	Application_Dev_Env_Software	Text
SP_ID	FKey	Application_Server_Software	Text
Standard	Text	Batch_Processing_Software	Text
High_Memory_Instances	Text	Database_Software	Text
High_CPU_Instances	Text	Video_Encoding_Streaming_Software	Text
High_Cluster_Instances	Text	Web_Hosting_Software	Text
On_Demand_Services	Text	Others	Text

B. Description of XML Tags

The data can be stored using following XML Tags and can be updated in real time by using any suitable database application program:

```
<? xml version="1.0" standalone="no"?>
<cloudsites _information>
    <service_provider >
        <sp_id>A001</sp_id>

        <record>

<sp_name>Cloud Containers
</sp_name>
```


Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

```
<location>USA</location>
<availability>99.9</availability>
<portability> Yes </portability>
<flexibility>Yes </flexibility>
<scalability> Yes</scalability>
<monitoring_facility>Yes</monitoring_
facility>
<bandwidth_limit> Yes </bandwidth_limit>
<interoperability> Yes </interoperability>
<virtual_services> Yes </virtual_services>
<storage_capacity>Extra large
</storage_capacity >

<service_type >
< s_id> S001 </s_id>
<application_based> Yes </application_based>
<platform_based> Yes </platform_based>
<infrastructural_based>Yes
</infrastructural_based>
<hybrid_type> No <hybrid_type>
</service_type>

<service_elasticity>
<e_id> E001</e_id>
<load_balancing>Yes</load_balancing>
<ip_addressing> Yes </ip_addressing>
<storage_flexibility> Yes </storage_flexibility>
</service_elasticity>

<security>
<sec_id>Sec001</sec_id>
<multifactor_authentication>Yes
</multifactor_aurhentication>
<ISO_certification> Yes </ISO_certification>
<network_firewall_switches_security>Yes
</network_firewall_switches_security>
<other_certification> No </other_certification>
</security>

< reliability >
<low> No </low>
<medium> No </medium>
<high> Yes</high>
</reliability>

<cost>
<c_id> C001 </c_id>
<cpu_cycle_per_hr > 10$ </ cpu_cycle_per_hr>
<hardware_usage_per_hr>5$
<hardware_usage_per_hr>
<on-demand-instances-hr>10$
<on-demand-instances-per-hr>
<reserved-instances-per-hr>20$
</reserved-instances-per-hr>
<storage-per-GB-day>5$
<storage-per-GB-per-day>
<monitoring-per-month>100$
</monitoring-per-month>
<bandwidth-use-per-1GB-per-month>10$
</bandwidth-use-per-1GB -per-month>
<billing-auditing-metering-per-month>100$
</billing-auditing-metering-per-month>
<other-charges-per-month>100$
</other-charges-per-month>
</cost>

<operating-system>
<Red-Hat-Ent-linux>Yes
</Red-Hat-Ent-linux>
< Gentoo-linux> No </Gentoo-linux>

<open-source-linux>Yes
</open-source-linux>
<oracle-ent-linux> Yes </oracle-ent-linux>
<sun-open-solaris> Yes <sun-open-solaris>
<fedora> No </fedora>
<windows-server-2003-2008>Yes
</windows-server-2003-2008>
<others> No </others>
</operating-system>

<disk-space>
<d-id> D001</d-id>
<small> Yes </small>
<medium> Yes </medium>
<large> Yes </large>
<Extra-large> Yes </Extra-large>
</disk-space>
<memory>
<mem_id> M001 </mem_id>
<small> 2 GB </small>
<medium> 8 GB </medium>
<large> 16 GB </large>
<extra-large> 32 GB </ extra-large>
</memory>

<instances_usage>
<instance-id> I001 </instance-id>
<standard> 10$ per hr </standard>
<high-cpu-instance>20$ per hr
</high-cpu-instance>
<high-cluster-instance>20$ per hr
<high-cluster-instance>
<on-demand-instance>50$ per hr
</on-demand-instance>
<other-instances>20$ per hr
</other-instance>
</instances-usage>

<software>
<sf_id> SO001 </sf_id>
<application-development-env-software>
Yes </ application -development-env-software>
```

```
<application-server-software>Yes
</application-server-software>
<batch-processing-software>Yes
</batch-processing-software>
            <database-software> Yes  </database-software>
             <video-encoding-and-streaming-software>  Yes
</video-encoding-and-streaming-software>
             <web-hosting-software>Yes
</web-hosting-software>
             < others> No </others>
          </software>
        </record>
      </service_provider>
      <service_provider>
      ...
      ...
     </service_provider>
   </cloud_sites_information>
   </xml >
```

IV. CONCLUSION

Customers select services from different service catalogues located at different cloud sites as provided by their providers, according to their requirements. This involves complex selection criteria, as analysis of service providers policies , rules , validations , security features, scalability features, market value (it may not a fake) , availability etc. For big organizations this hardly matters, but for small scale industries this cost matters a lot.

In this paper, we tried to overcome this problem by proposing an approach, by creating Unified master service catalog manager a platform which enables the end users to choose required services using payment counter. Unified master service catalogue manager unifies the delivery and lifecycle management of public and private cloud services across the enterprise including cataloguing, provisioning, security, auditing, monitoring, reporting, metering, billing, administration, and user support thus enabling the customer to get proper in formation and benefits from the available services.

Real time Database is created and, is linked to master service catalogue manager for further updates using XML tags. Information regarding service provider must be stored in real time database using database commands for e.g. by creating tables, inserting values, assigning keys, relationships and queries for further updates and selections in real time database management system, using available (any language) application based database program.

ACKNOWLEDGMENT

Thanks for continuous support and engagement from Mr. Brajesh Patel, Head – Department of Computer Science, Shriram Institute of Technology, Jabalpur, India. Your timely inputs and guidance have gone a long way in converting an idea into this research paper. Thanks to the department staff at Shriram Institute of Technology, Jabalpur, India and GS Moze College of Engineering, Pune, India for providing access to computing infrastructure and research resources, and journals that helped evolve the research approach in its early days.

REFERENCES

1. M. Armbrust and A. Fox et al. Above the clouds: A Berkeley view of cloud computing, Technical Report EECS-2009-28, UC Berkeley, 2009.
2. A. Li and X. Yang et al. Cloud Cmp: Shopping for a Cloud Made Easy, Proceeding of the 2nd USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 10), 2010.
3. T. Liu, Y. Li, and X. Li, A Collaborative Management as a Service Framework for Managing Internetware Systems, First Asia-Pacific Symposium on Internetware (Internetware 2009), 2009.
4. T. Wood and R. Shenoy et al. The Case for Enterprise-Ready Virtual Private Clouds, Proceeding of the 1st USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 09), 2009.
5. Elamparithi, M. Database Migration Tool (DMT) -accomplishments & future directions, Communication and Computational Intelligence (INCOCCI), 2010, p.481.
6. jgaard-Hansen, K. ; Huan Cong Nguyen ; Schwefel, H. Session mobility solution for client-based application migration scenarios, Wireless On-Demand Network Systems and Services (WONS), 2011, p.76.
7. Malkawi, M.; Abaza, M.; Knox, D.; Process migration in virtual memory multicomputer systems, Proceeding of the Twenty-Sixth Hawaii International Conference on System Sciences, 1993, Page(s): 90 98 vol.2.
8. Jamcracker, Inc., Unified Cloud Whitepaper, http://www.jamcracker.com/JSDNEnterprise.
9. RightScale Inc., Right Scale Cloud Management, http://www.rightscale.com/products/features/.

AUTHORS PROFILE

Prarthana Gupta She is currently pursuing Masters Degree in Computer Technology and Applications at Shriram Institute of Technology, India under Rajiv Gandhi Proudyogiki Vishwavidyalaya, India. Past experience includes five years of research and industry expertise in the field of Computer Technology, specializing in Cloud Enablement and Application in Heterogeneous Environments, Data Mining and Enterprise Data Modeling.

She has over four years of academic experience in teaching graduate students in the subjects of Data Mining, Database Management & Algorithm Design at Mumbai University, India & University of Pune, India. She also specializes in experiential learning methodologies by incorporating global best practices and emerging technologies into everyday learning for her students. She has successfully incorporated the use of technology enabled collaborative learning for two batches at graduate level.