A Review of Literature on Cloud Brokerage Services

Dr. J. Akilandeswari
Professor and Head, Department of Information Technology
Sona College of Technology,
Salem, India.

C. Sushanth
PG Scholar, Department of Information Technology
Sona College of Technology,
Salem, India.

ABSTRACT

Cloud computing is kinetically evolving areas which offer large potential for agencies of all sized to increase efficiency. Cloud Broker acts as a mediator between cloud users and cloud service providers. The main functionality of cloud broker lies in selecting best Cloud Service Providers (CSP) from requirement set defined by cloud user. Request from cloud users are processed by the cloud broker and suited providers are allocated to them. This paper gives detailed review of cloud brokerage services and their method of negotiating with the service providers. Once the SLA is specified by cloud service provider, the cloud broker will negotiate the terms according to the user's specification. The negotiation can be modeled as a middleware, and its services can be provided as application programming interface.

Keywords

Cloud computing, broker, mediator, service provider, middleware.

1. INTRODUCTION

A cloud refers the interconnection of huge number of computer systems in a network. The cloud provider extends service through virtualization technologies to cloud user. Client credentials are stored on the company server at a remote location. Every action initiated by the client is executed in a distributed environment and as a result, the complexity of maintaining the software or infrastructure is minimized. The services provided by cloud providers are classified into three types: Infrastructure-as-a-Service (IaaS), Software-as-a-Service (SaaS), and Platform-as-a-Service (PaaS). Cloud computing makes client to store information on remote site and hence there is no need of storage infrastructure. Web browser act as an interface between client and remote machine to access data by logging into his/her account. The intent of every customer is to use cloud resources at a low cost with high efficiency in terms of time and space. If more number of cloud service providers is providing almost same type of services, customers or users will have difficulty in choosing the right service provider. To handle this situation of negotiating with multiple service providers, Cloud Broker Services (CBS) play a major role as a middleware. Cloud broker acts as a negotiator between cloud user and cloud service provider. Initially, cloud provider registers with cloud broker about its specification on offerings and user submits request to broker. Based on type of service, and requirements, best provider is suggested to the cloud user. Upon confirmation from the user, broker establishes the connection to the provider.

2. CLOUD BROKERAGE SERVICES (CBS)

Foued Jrad et al [1] introduced Intercloud Gateway and Open Cloud Computing Interface specification (OCCI) cloud API to overcome lack of interoperability and heterogeneity. Cloud users cannot identify appropriate cloud providers through the assistance of existing Cloud Service Broker (CSB). By implementing OCCI in Intercloud Gateway, it acts as server for service providers and OCCI act as a client in abstract cloud API. Cloud Broker satisfies users of both functional and non-functional requirements through Service Level Agreement (SLA). Intercloud Gateway acts as a front end for cloud providers and interacts with cloud broker. Figure 2.1 shows a generic architecture of the service broker.

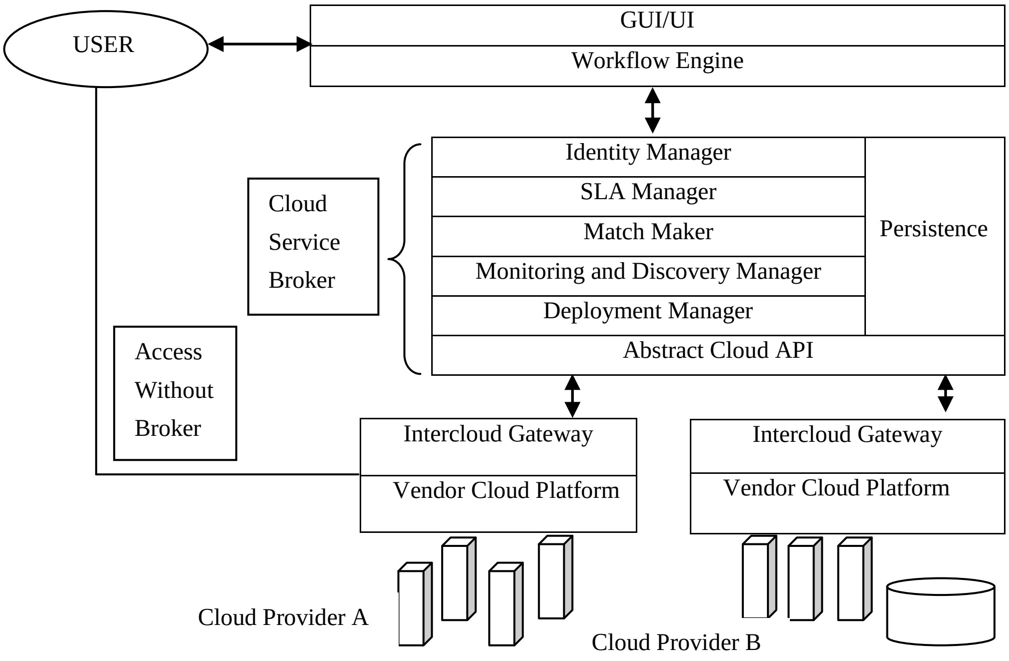

Figure 2.1 A generic architecture for Cloud Service Broker

Identity Manager handles user authentication through unique ID.SLA Manager is responsible for negotiates SLA creation and storing. Match Manager takes care of selecting suitable resources for cloud users. Monitoring and Discovery Manager monitor SLA metrics in various resource allocations. Deployment manager is in charge of deploying services to cloud user. Abstract cloud API provides interoperability.

The user submits a request to SLA Manager and it parses the request into SLA parameters which is given to Match Maker. By applying algorithm Match Maker find best suited solution and response is passed to the user. Upon user acceptance a connection is provided by service providers.

Table 2.1 Sample SLA parameters for IaaS

Functional	**Non-functional**
CPU speed	Response time
OS type	Completion time
Storage size	Availability
Image URL	Budget
Memory size	Data transfer time

Through this architecture, interoperability is achieved, but this cannot assure best matching cloud service provider to the client.

Tao Yu and Kwei-Jay Lin [2] introduces Quality of Service (QoS) broker module in between cloud service providers and cloud users. The role of QoS information is collecting information about active servers, suggesting appropriate server for clients, and negotiate with servers to get QoS agreements. The QoS information manager collects information required for QoS negotiation and analysis. It checks with the Universal Description Discovery and Integration (UDDI) registry to get the server information and contacts servers for QoS information such as server send their service request and QoS load and service levels. After receiving clients functional and QoS requirements, the QoS negotiation manager searches through the broker’s database to look for qualified services. If more than one candidate is found, a decision algorithm is used to select the most suitable one. The QoS information from both server and QoS analyzer will be used to make the decision. By using this architecture load balancing factor of server is maintained for a large number of users, but not efficient in delivering best suited provider to the client.

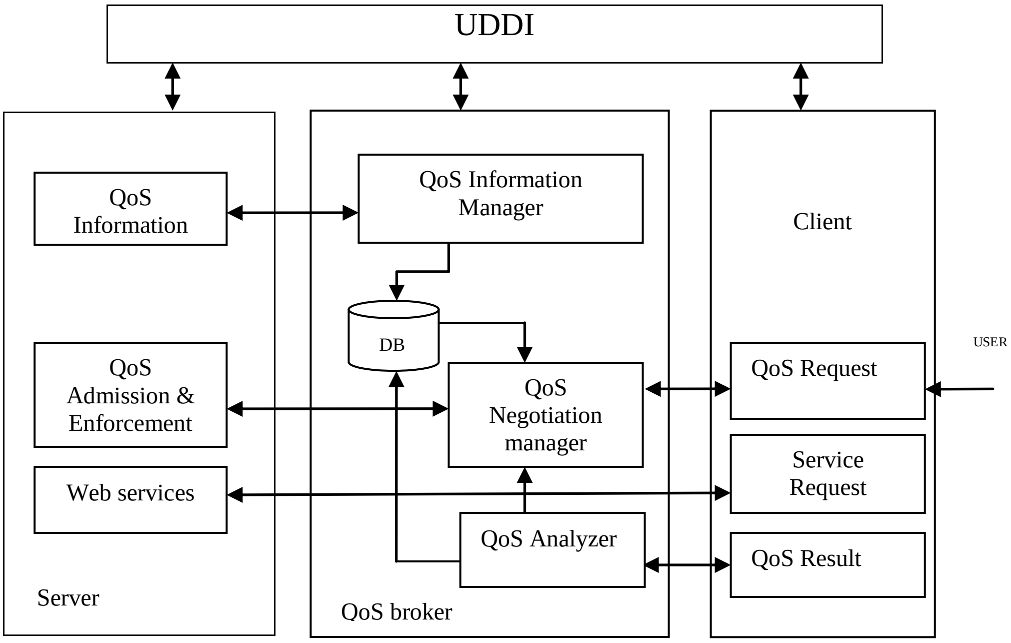

Figure 2.2 QoS based Architecture

HQ and RQ allocation algorithm is proposed to maximize server resource while minimizing QoS instability for each client. The HQ allocation algorithm is to evenly divide available resource among required client based on active clients. RQ assigns a different service level to client based on requirements.

Josef Spillner et al [3] provided solution is to subdivide resource reservation into either serial or parallel segments.

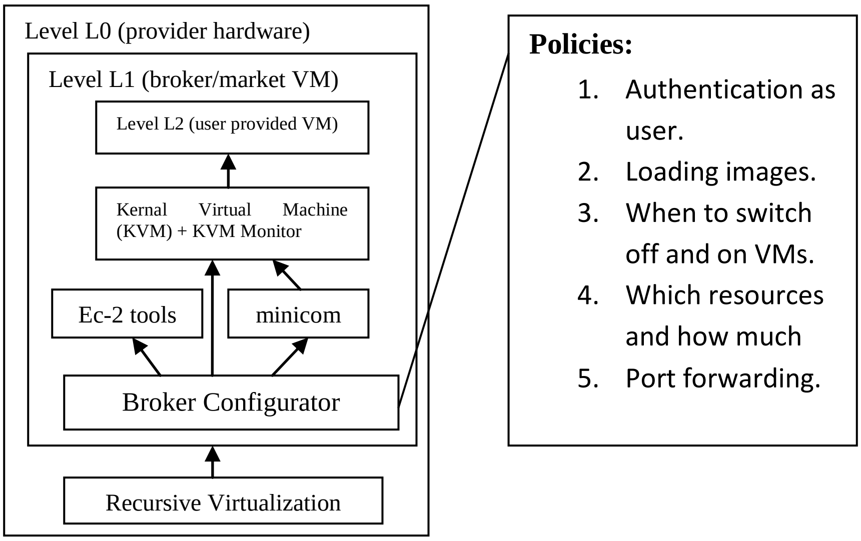
Level L0 (provider hardware)
Level L1 (broker/market VM)
Level L2 (user provided VM)
Kernal Virtual Machine (KVM) + KVM Monitor
Ec-2 tools
minicom
Broker Configurator
Recursive Virtualization
Policies:
1. Authentication as user.
2. Loading images.
3. When to switch off and on VMs.
4. Which resources and how much
5. Port forwarding.

Figure 2.3 Nested cloud with virtual machine

Nested virtualization provides services to cloud user. The outcome is a highly virtualizing cloud resource broker. The system supports hierarchically nested virtualization with dynamically reallocate capable resources. A base virtual machine is dedicated to enabling the nested cloud with other virtual machines is referred to as sub-virtual machine running at a higher virtualization level. The nested cloud virtual machine is to be deployed by the broker and offers control facilities through the broker configurator which turn it into a lightweight infrastructure manager. The proposed solution yields the higher reselling power of unused resources, but hardware cost of running virtual machine will be high to obtain the desired performance.

Chao Chen et al [4] projected objectives of negotiation are minimize price and guaranteed QoS within expected timeline, maximize profit from the margin between the customers financial plan and the providers negotiated price, maximize profit by accepting as many requests as possible to enlarge market share. The proposed automated negotiation framework uses Software–as-a-Service (SaaS) broker which is utilized as the storage unit for customers. This helps the user to save time while selecting multiple providers. The negotiation framework helps user to assist in establishing a mutual agreement between provider and client through SaaS broker. The main objective of the broker is to maintain SLA parameters of cloud provider and suggesting best provider to customer.

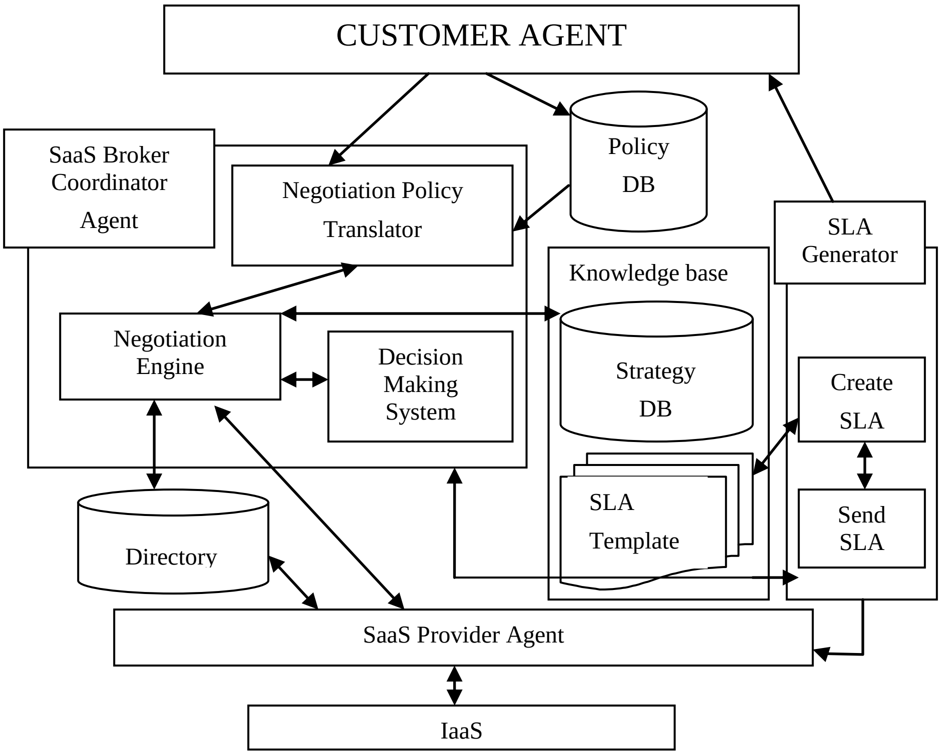

Figure 2.4 Negotiation Framework

Negotiation policy translator maps customers QoS parameters to provider specification parameters. Negotiation engine includes workflows which use negotiation policy during the negotiation process. The decision making system uses decision making criteria to update the negotiation status. The minimum cost is incurred for resource utilization. Renegotiation for dynamic customer needs is not solved.

Wei Wang et al [5] proposed a new cloud brokerage service that reserves a large pool of instances from cloud providers and serves users with price discounts. A practical problem facing cloud users is how to minimize their costs by choosing among different pricing options based on their own demands. The broker optimally exploits both pricing benefits of long-term instance, reservations and multiplexing gains. Dynamic approach for the broker to make instant reservations with the objective of minimizing its service cost is achieved. This strategy controls, dynamic programming and algorithms to quickly handle large demands.

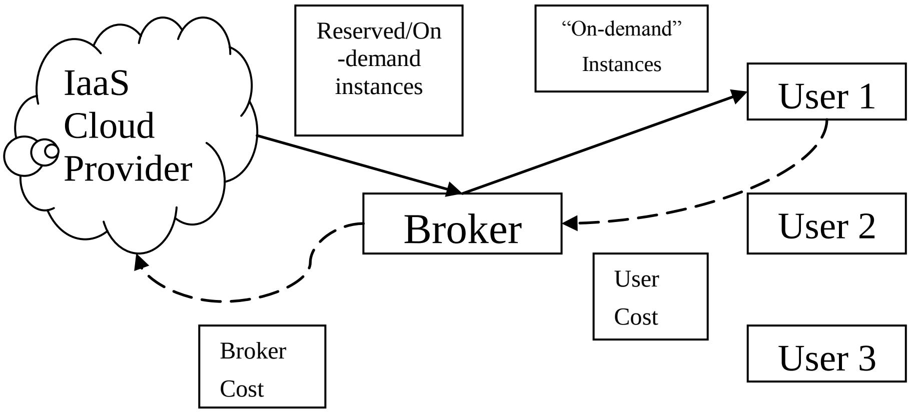

Figure 2.5 Cloud Broker Model

A smart cloud brokerage service that serves cloud user demands with a large pool of computing instances that are dynamically launched on-demand from IaaS clouds. Partial usage of the billing cycle incurs a full cycle charge, this makes user to pay more than they actually use. This broker uses single instance to serve many users by time-multiplexing usage, reducing cost of cloud user.

Dharmesh Mistry [6] proposed a cloud-based analytics solution as a service from a cloud broker which could considerably minimize costs for the client, while assisting Independent Software Vendor (ISV) to maximize profit. When data are arriving, it is divided and index is created and finally it is mapped to original values through analysis. Large organizations are purchasing such software as a SaaS instead of obtaining and hosting software internally. But for ISVs that constructed their business by the traditional model. The cloud broker acts as middleware between the ISV and cloud providers. ISV yields solution to meet customer demands from for existing services. The broker provides services such as entitlement, analytics, billing and payment, security and context provisioning. ISVs usually rely on pre-module licensing models and software audits to confirm that the appropriate number of users access the modules and functions for which the customer will be paid.

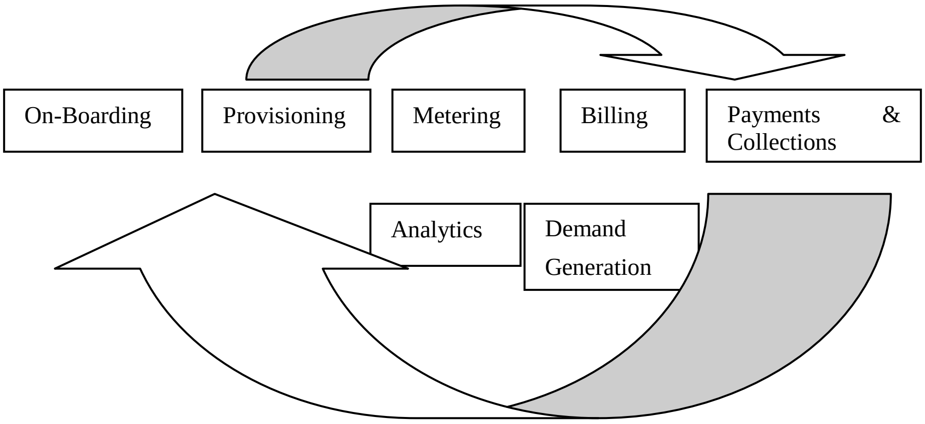

Figure 2.6 Mapping in Cloud Broker

An ISV can drive faster profit growth, while maintaining margins, and respond to market demand more quickly.

Lori MacVittie [7] introduces broker as a solution to integrate hybrid policy without affecting control in services. The integration between cloud and datacenter is done with cloud broker integration at the process layer. Brokers deploy vast amount of applications for customer through infrastructure defined by corporate enforced policies. Identity broker module communicates with datacenter through authorization and authentication mechanism. The real-time implementation of cloud broker is achieved by two types of architectures: Full-proxy broker and Half-proxy broker. In Full-proxy broker requests are processed through the tunneling and implemented in many ways such as VPN. In Half-proxy broker only validation of the request is done by broker, successive communication established directly. This model defines how the request can be handled in late binding. A cloud delivery broker can make decision, such as where to revert user upon request. Hybrid cloud must be able to describe capabilities such as bandwidth, location, cost, type of environment.

Sigma Systems [8] introduces cloud service broker which is responsible for order management, provisioning, billing integration and Single Sign–On (SSO). In the proposed architecture, the Cloud Service Broker allows service providers to offer their own SLA, which provides a single source for all applications to customers. Providers can establish and grow a single and a combined collection of services that match their set of services, and allow for unique grouping to meet their customers' needs. Cloud brokerage from Sigma Systems is available either as a managed service or can be deployed on-basis.

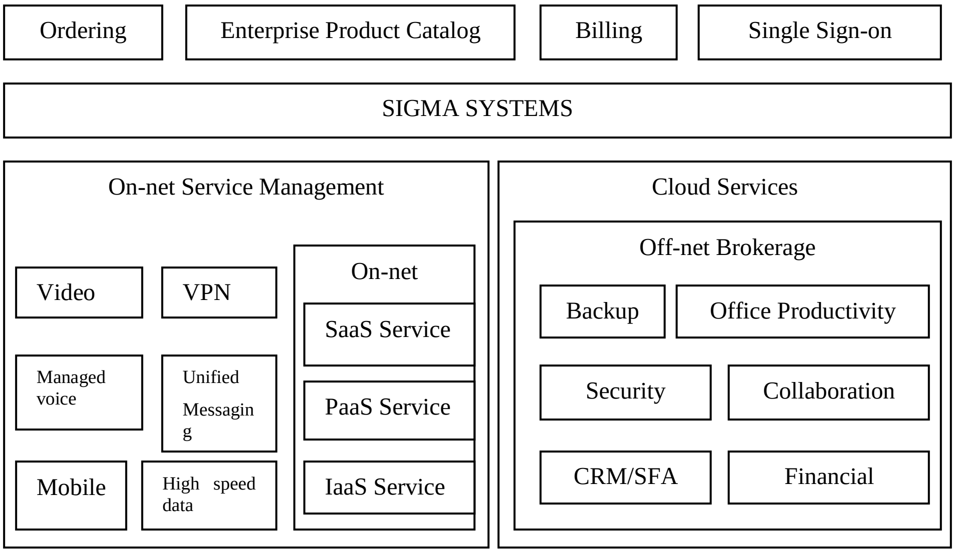

Figure 2.7 Sigma System model

The Sigma model allows service providers to create single and highly exciting packages by combining high-speed data and other complex network services with business and productivity-enhancing, SaaS based services.

Vordel [9] developed cloud service broker in order to allow organizations to apply a layer of confidence in their cloud computing applications. It agents the connection to the cloud infrastructure, relating governs controls for service usage and service uptime.

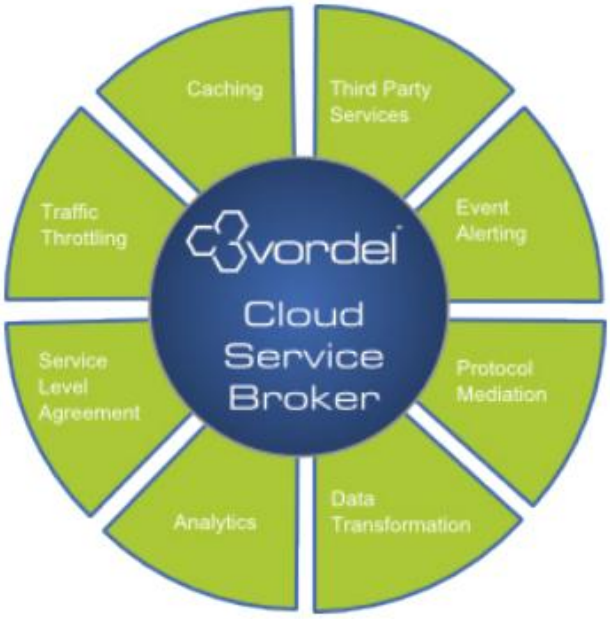

Figure 2.8 Services provided by Vordel

It records service type, time of day, and the identity of the user. All information sent to cloud services must be examined for disclosing data, in order to allow Data Loss Prevention (DLP). Caching protects the enterprise from inactivity linked with connecting to the cloud service. Service Level Agreement (SLA) monitoring observes the whole transaction throughput time. The Cloud Service Broker contains a pluggable structure which allows for modules to be added, such as modules to provide additional encryption algorithm.

Apostol T. Vassilev [10] introduced personal brokerage of Web service access which becomes part of the Web authentication structure, by network smart cards. This allows new Web services based on their characteristic properties of essential resistance, tough cryptography, connectivity and computing power. To enhance network, smart-card capabilities, particularly in the serious area of human-to-card interaction evidence, to bring further accessibility and personalization to Web security and privacy. In Single-Sign-On (SSO) systems users attempt to access services offered by a connected service provider using a web browser on the client system. The provider redirects the service request by directing the user’s browser to the Identity Provider’s (IDP) authentication page. To facilitate the redirection, the service provider issues a ticket that unites the user's digital identity once the authentication is complete.

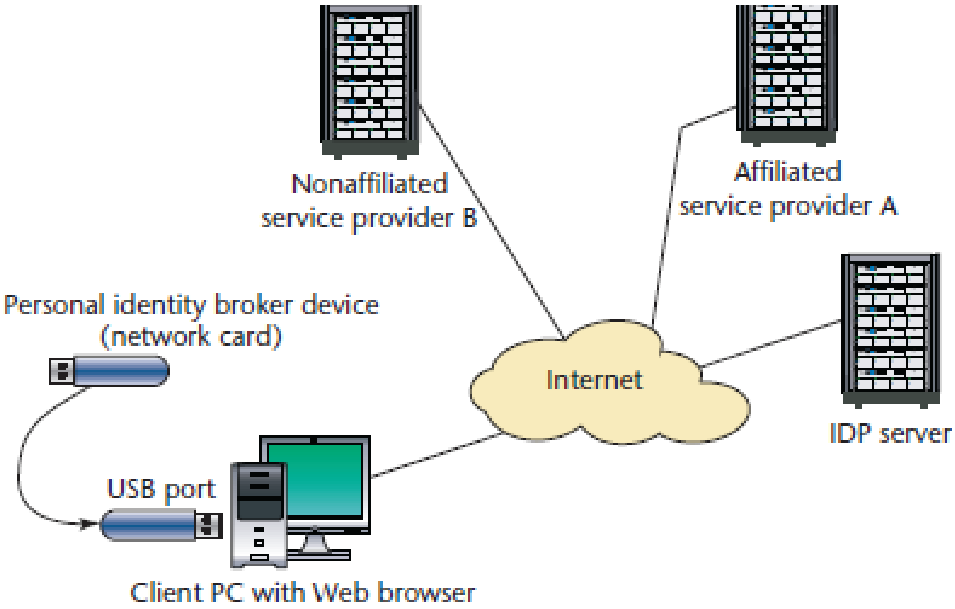

Figure 2.9 Personal Brokerage extensions of Federated service

Users use an IDP enforced method for authentication to prove their identity. If the authentication is well, the IDP declares the user's identity in the ticket sent back to the browser, which in turn sends it to the service provider. Users can then access the requested services. The existing IDP-enforced authentication method is by means of a user name and key. Because the entire united system of Web services only requires one username and password license, SSO systems are convenient for the user. At the equal time, such credentials become a major target for hackers because it gives them access to many private user resources at once. Presently network traffic between users' browsers and remote servers is secured by ubiquitous standard security protocols for information exchange, based on Secure Socket Layer (SSL) and Transport Layer Security (TLS).

Muhammad Zakarya and Ayaz Ali Khan [11] found that Distributed Denial of Service (DDoS) attack is identified as a major threat in present time, which we overcome by new cloud environment architecture and Anomaly Detection System (ADS). These ADS improve computation time, QoS and high availability. Each cloud is separated as regional areas known as GS. Each GS is protected by AS/GL. Developed ADS are installed in cloud node or AS and router. A tree is maintained at every router by making every packet with path modification strategy, so the attacker of node is easily found. ADS have two phases detection of malicious flow confirmation algorithm to drop attack or pass it.

Randomness or Entropy is given by,

$$\mathbf{H(X) = -\sum_{x \in X}^{n} p(x) \log p(x)} \qquad \mathbf{(2.1)}$$

Where 0< H(x) < log (n), p(x) probability of x

$$\mathbf{P(x) = m_i/m} \qquad \mathbf{(2.2)}$$

Where m_i is number of packet with value x and m is total number of packets

Normalized entropy is calculated to get overall probability of captured packet in specific time

$$\textbf{Normalized entropy} = \mathbf{(H / \log n_0)} \qquad \mathbf{(2.3)}$$

For detection of DDoS attack, decide a threshold value. An edge router collects the flow of traffic for a specific time window w. Find probability p(x) for each packet node. Calculate link entropy of all active nodes separately. Calculate H(x) for router, if normalized entropy less than identified malicious attack flow then system is compromised. For confirmation of attack flows, decide a threshold value and compare with entropy rate.

Srijith K. Nair et al [12] describes the concept of cloud bursting, cloud brokerage, framework of power brokerage based on service OPTIMIS. When a private cloud need to access external cloud for a certain time for computation, then the process is called cloud bursting. Internal cloud in the company needs to verify SLA requirements to measure performance. Cloud bursting environment, architecture being developed by OPTIMIS with following capabilities need common management interface, set of monitoring tools, global load balancer, and categorized providers. Cloud brokerage model was created by cloud service providers for the cloud management platform. The cloud management platform is responsible for activities such as policy enforcement, usage monitor, network security, platform security. Cloud API mediates consumer interaction with cloud broker. The SLA monitoring unit is responsible for monitoring all SLA and violations. Identify and access module records of serviced customer and generate one time token. Audit unit inspects broker platform and capabilities. Risk management prioritizes risks based on events. Network/platform security provides overall security through IDS. The user send storage request to cloud portal. Then portal forwards id and password for Identity and Access Management (IAM), it verifies and grants access along with criteria. Cloud portal converts identity and access rights to external token, containing criteria and request, which is encrypted and sent to Broker IAM. Broker IAM decrypt using portal public key and verifies integrity which in turn generates one time access token.

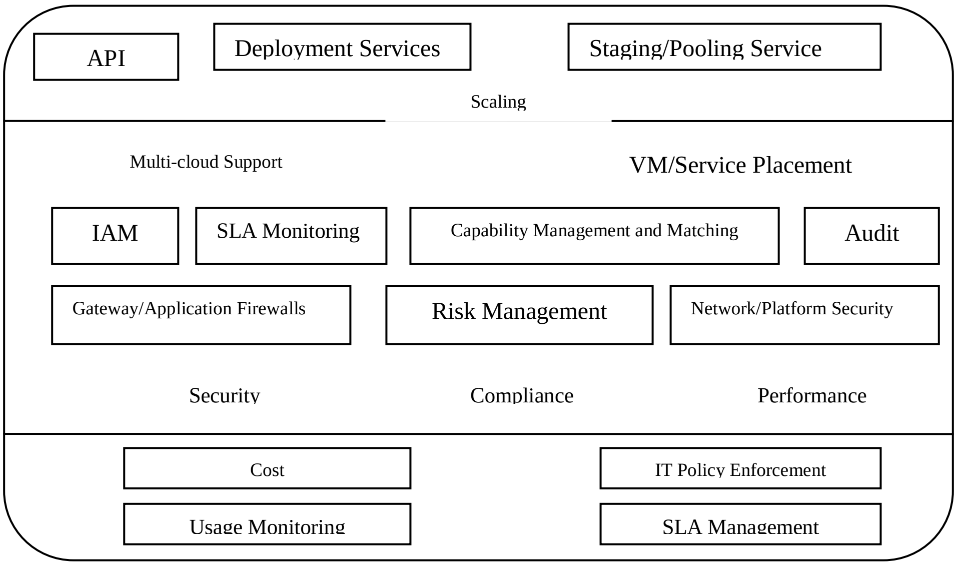

Figure 2.10 Functional Requirements for Cloud Service Broker

This token contains Uniform Resource Identifier (URI) which is again forwarded to portal and discard old token. Cloud portal decrypts using the private key of broker and forward to the respective user. The user sends data to Application Programming Interface (API) which checks the strength of token and grant access to upload data. This uploaded data in service provider sends the position of data through secret key. This ensures confidentiality and integrity.

Mark Shtern et al [13] described AERIE architecture. When organization changing to public cloud infrastructure they have problem with control and security and must contain best model for deployment. This project suggests reference architecture for virtual private cloud built on cross provider platform on-demand compute instance, that reduce levels of trust on infrastructure providers. Inner instance is started from outer instance. Together inner and outer instance forms a nested instance. An outer instance runs an agent which ensures that it has not been modified. These agents establish connection with the controller using novel key exchange algorithm. A standard security application is implemented to preserve integrity of outer instance. Traffic from public internet is made to pass through security bulwark. A load balancing DNS service is capable of detecting inaccessible host from available solution. Each instance has an image which contains encrypted image to launch inner instance. Trusted

Instance Agent (TIA) conducts key exchange with controller to establish HTTPS connection using novel algorithm. The controller checks validity of certificate in image. To maintain integrity it employs Intrusion Detection System, if any, violations are met a virtual channel is terminated.

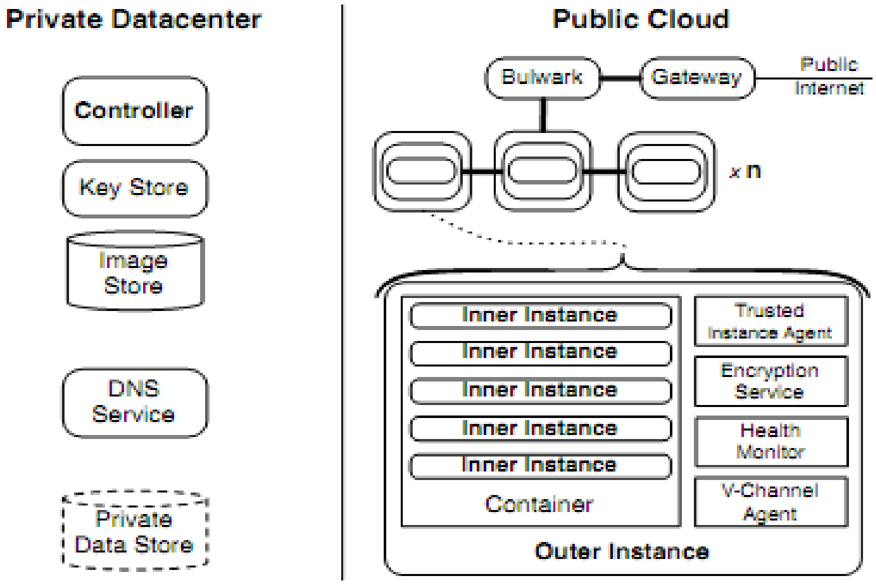

Figure 2.11 AERIE Architecture

Przemyslaw Pawluk et al [14] introduce cloud broker service which enables the deployment and runtime management of cloud application using multiple providers. Service Measurement Index (SMI) is a possible approach to facilitate the comparison of cloud business. An attribute is then expressed as a set of Key Performance Indicators (KPI) which specifies requested data acquired from every metric. After initial deployment, decision to add/remove resources is made by cloud manager. Application manager controls run time management of application according to the model. A Resource Acquisition Decision (RAD) involves. We will use the following scenario as a running example the selection of n resources from a set of m providers. The Broker is responsible for solving the RAD problem. It must also connect to the set of selected providers to be used and acquire the collection of resources. Topology Descriptor File (TDF) is used to identify the application topology to be deployed on the cloud. Each cloud provider describes details of environment variables in the TDF. The chosen nodes are instantiated through a translation layer. Cloud Manager and Broker make use of monitoring information, the former to make ongoing elasticity decisions and the latter to assist in the decision process. The broker selects the set of all possible specifications that satisfy the objectives stated in the desired models named in the TDF. Next, as a result of multi-criteria optimization process, a set of equivalent specifications is selected.

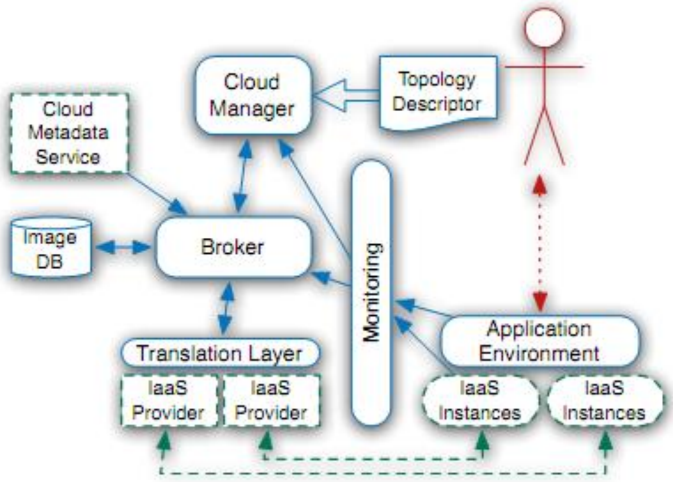

Figure 2.12 Cloud Management Frameworks

From this set, one is selected and the appropriate instance is acquired from the provider. In the situation where there are no suitable specifications suits the objectives, the broker makes an attempt to relax objectives by identifying the closest specification in each direction. Next, the optimization step is performed over the resultant set of relaxed results. The RAD problem can be formulated as a multi-criteria optimization problem.

Paul Hershey et al [15] presented System of Systems (SoS) method which is responsible for activities such as QoS monitoring, management and response for cloud providers that delivers computing as a service. Various metrics are considered to calculate performance and security of SoS.

Delay is the sum of delays in lower level domain of cloud. There is an infrastructure component delay. Hence delay is given by

$$\mathbf{D_{sos} = p1D_g+p2D_b+p3D_S+p4D_i} \qquad \mathbf{(2.4)}$$

P_i- parameter that is dependent on the infrastructure components used.

D_j – delay experienced in each layer.

Throughput at system level is defined as the number of transactions that are completed per unit time

$$\mathbf{T_1 = n \text{ x } Transaction\ Throughput} \qquad \mathbf{(2.5)}$$

$$\mathbf{T_S = m \text{ x } T_1} \qquad \mathbf{(2.6)}$$

$$\mathbf{T_B = q \text{ x } T_S} \qquad \mathbf{(2.7)}$$

Where m, n, q are number of transactions at lower domain needed to complete transaction at higher domain. Authentication metric is a logical conjunction of each level in EMMRA.

Table 2.2 Metrics Categories

Category	**Metric**
Performance	Delay Delay Variation Throughput Information Overhead
Security	Authentication Authorization Non-repudiation Integrity Information Availability Certificate and Accreditation Physical Security

$$\mathbf{A_{sos} = A_{G\,\wedge}A_B{\wedge}A_S{\wedge}A_I} \qquad \mathbf{(2.8)}$$

Authorization is a bottom-up metric and it is applied at each level. Authorization at IaaS level can be given as,

$$\mathbf{Auth_I = \min \{\Pi\ P_I\}} \qquad \mathbf{(2.9)}$$

P_I is permission to perform actions I at IaaS level. The min operator is used to indicate least privilege level that is granted to the user.

3. CONCLUSIONS

The development of a cloud brokerage services framework is getting momentum since its usage is pervasive in all verticals. The works till now do not consider the scenario of more than one cloud service provider providing the same level of requirements to the user. This scenario will induce an ambiguity for the users to choose an appropriate provider. The Cloud Broker Services will act on behalf of the user to choose a particular service provider for providing service to the user. If Cloud Broker Service becomes a standard middleware framework, many chores of cloud service providers can be taken by CBS.

REFERENCES

[1] Foued Jrad, Jie Tao, Achim Streit, *SLA Based Service Brokering in Intercloud Environments.* Proceedings of the 2nd International Conference on Cloud Computing and Services Science, pp. 76-81, 2012.

[2] Tao Yu and Kwei-Jay Lin, *The Design of QoS Broker Algorithms for QoS-Capable Web Services*, Proceedings of IEEE International Conference on e-Technology, e-Commerce and e-Service, pp. 17-24, 2004.

[3] Josef Spillner, Andrey Brito, Francisco Brasileiro, Alexander Schill, *A Highly-Virtualising Cloud Resource Broker*, IEEE Fifth International Conference on Utility and Cloud Computing, pp.233-234, 2012.

[4] Linlin Wu, Saurabh Kumar Garg, Rajkumar Buyya, Chao Chen, Steve Versteeg, *Automated SLA Negotiation Framework for Cloud Computing*, 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, pp.235-244, 2013.

[5] Wei Wang, Di Niu, Baochun Li, Ben Liang, *Dynamic Cloud Resource Reservation via Cloud Brokerage*, Proceedings of the 33rd International Conference on Distributed Computing Systems (ICDCS), Philadelphia, Pennsylvania, July 2013.

[6] Dharmesh Mistry, *Cloud Brokers can help ISVs Move to SaaS*, Cognizant 20-20 Insight, and June 2011.

[7] Lori MacVittie, *Integrating the Cloud: Bridges, Brokers, and Gateways*, 2012.

[8] Sigma Systems, *Cloud Brokerage: Clarity to Cloud Efforts*, 2013.

[9] Vordel white papers, Cloud Governance in the 21st century, 2011.

[10] Apostol T. Vassilev, Bertrand du Castel, Asad M. Ali, *Personal Brokerage of Web Service Access* IEEE Security & Privacy, vol. 5, no. 5, pp. 24-31, Sept.-Oct. 2007.

[11] Muhammad Zakarya & Ayaz Ali Khan, *Cloud QoS, High Availability & Service Security Issues with Solutions*, International Journal of Computer Science and Network Security, vol.12 No.7, July 2012.

[12] Srijith K. Nair, Sakshi Porwal, Theo Dimitrakos, Ana Juan Ferrer, Johan Tordsson, Tabassum Sharif, Craig Sheridan, Muttukrishnan Rajarajan, Afnan Ullah Khan, *Towards Secure Cloud Bursting, Brokerage and Aggregation*, Eighth IEEE European Conference on Web Services, pp.189-196, 2010.

[13] Shtern. M, Simmons. B, Smit. M, Litoiu. M, *An architecture for overlaying private clouds on public providers*, Eighth International Conference and Workshop on Systems Virtualization Management, pp.371, 377, 22-26 Oct. 2012.

[14] Przemyslaw Pawluk, Bradley Simmons, Michael Smit, Marin Litoiu, Serge Mankovski, *Introducing STRATOS: A Cloud Broker Service*, IEEE Fifth International Conference on Cloud Computing, pp.891-898, 2012.

[15] Hershey. P, Rao. S,Silio. C.B., Narayan. A, *System of Systems to provide Quality of Service monitoring, management and response in cloud computing environments*, 7th International Conference on System of Systems Engineering (SoSE), vol., no., pp.314, 320, 16-19 July 2012.

This paper may be cited as:

Akilandeswari, J. and Sushanth, C., 2014. A Review of Literature on Cloud Brokerage Services. *International Journal of Computer Science and Business Informatics, Vol. 10, No. 1, pp. 25-40.*