Towards Secure Cloud Bursting, Brokerage and Aggregation

Srijith K. Nair[1*], Sakshi Porwal[2], Theo Dimitrakos[1], Ana Juan Ferrer[3], Johan Tordsson[4], Tabassum Sharif[5], Craig Sheridan[5], Muttukrishnan Rajarajan[6] and Afnan Ullah Khan[1]

[1]*BT Innovate and Design,* [2] *University College London,* [3]*Atos Origin,* [4]*Umeå University,* [5]*Flexiant,* [6] *City University*

Abstract—**The cloud based delivery model for IT resources is revolutionizing the IT industry. Despite the marketing hype around "the cloud", the paradigm itself is in a critical transition state from the laboratories to mass market. Many technical and business aspects of cloud computing need to mature before it is widely adopted for corporate use. For example, the inability to seamlessly burst between internal cloud and external cloud platforms, termed cloud bursting, is a significant shortcoming of current cloud solutions. Furthermore, the absence of a capability that would allow to broker between multiple cloud providers or to aggregate them into a composite service inhibits the free and open competition that would help the market mature. This paper describes the concepts of cloud bursting and cloud brokerage and discusses the open management and security issues associated with the two models. It also presents a possible architectural framework capable of powering the brokerage based cloud services that is currently being developed in the scope of OPTIMIS, an EU FP7 project.**

Keywords: Cloud Computing, Cloud Brokering, Cloud Federation, Cloud Bursting

I. Introduction

Cloud computing has emerged as one of the most promising and challenging technologies of our time. This new paradigm utilises two separate technological development—utility computing and service oriented architecture—to provide the users (individuals, SMEs and enterprises) with a highly scalable, pay-per-use, everything-as-a-service model for IT delivery. Some of the properties that characterise the cloud computing service delivery model are scalability/elasticity, on-demand service provisioning, shared resource pooling, multi-tenancy hosting, utility pay-as-you-use pricing and abstraction of lower layers [1]. These characteristics give rise to several business drivers that make cloud computing an attractive service delivery model from a customer's point of view. They include capital expenditure reduction, increased IT agility, faster return on investment, removal of barriers to entry as well as a more robust and resilient infrastructure leading up to better business continuity.

From a deployment model point of view, a cloud computing taxonomy can be divided into the following types:

Public clouds: where the IT capabilities that are offered by cloud providers to any customers over the internet.

Private clouds: where IT capability is offered to a select group of consumers who are part of an enterprise. The cloud service provider may be an internal IT organisation (i.e., the same organisation as the consumer) or a third party.

Hybrid clouds: in which the environment is created through the usage of a combination of private and public cloud offerings by an organisation.

Internal clouds: is a subset of the private cloud model, where the cloud is an IT capability offered as a service by an IT organisation to its own business.

External clouds: is IT capability offered as a service to a business that is not hosted by its own IT organisation. An external cloud can be public or private, but must be implemented by a third party.

From a point of view of architectural service layers based on the services provided using the cloud model, the ecosystem can be broadly divided into three:

Software as a Service (SaaS): forms the top layer featuring a complete application provided in a multi-tenant environment. One prominent example of SaaS is Salesforce [2].

Platform as a Service (PaaS): providing a development and deployment middleware layer. Key players include the Microsoft Azure platform [3] as well as Google App Engine [4].

Infrastructure as a Service (IaaS): the lowest layer delivering services like compute storage and network. One prominent example of IaaS is Amazon EC2 service [5]. The work reported here mainly deals with the IaaS service delivery model.

Financial support for this work is provided by the European Commission's Seventh Framework Programme ([FP7/2001-2013]) under grant agreement no. 257115.

* Corresponding author. Address: pp13, GF, Orion Building, Adastral Park, Ipswich, IP53RE, UK.srijith.nair@bt.com

While a lot has been said about the cost benefits of moving into a cloud-based delivery model for IT resources as well as the technical merit of using a multi-tenanted cloud infrastructure, there has so far been a lack of enterprise-level uptake at a large scale. Privacy and security concerns are widely considered as major obstacles for cloud adoption by corporations and governments at large [6]. The widely publicised security issues associated with the cloud are major challenges that need to be addressed before wider adoption from enterprises can be expected. Some of the security issues that concern a generic cloud service offering include:

- Privacy of data: It has always been a sensitive topic as far as IT services are concerned. The complexity of the problem is compounded in the cloud services due to their unique operational model involving multi-tenanted architecture that makes extensive use of shared resources.
- Confidentiality and integrity of data: while cloud provider aims to provide data confidentiality and integrity, they do not offer any actionable guarantee regarding this and leak of non-encrypted data remains a big risk.
- Integration with existing infrastructure: even when an enterprise moves its IT services to the cloud, in effect de-perimeterising its infrastructure, it still needs to integrate these services with the rest of the existing IT infrastructure. An improperly integrated Identity and Access Management (IdAM) system can present practical difficulties during the implementation stage.
- Platform vulnerabilities: the potential threat/attack surface exposed by a cloud service depends on the delivery model of the provider. SaaS offerings' potential breach area are more concentrated on the actual application security issues (like database security, message-level security) and operational issues (like accounts management, user rights managements). PaaS services inherit most of the SaaS attack vectors in addition to having specific ones like Web Service security and value-adding platform services security. A IaaS service exposes a plethora of attacks vectors that range from the non-unique application, database and message-level security to infrastructure specific issues like virtual machine management, operating system hardening, virtual machine security etc

While security issues do present a clear hurdle for wider adoption of cloud services, equally inhibiting the adoption of cloud computing is the lack of maturity of the delivery and operational models for cloud based services.

This paper looks at some of the work done in rectifying these shortcomings as a part of the OPTIMIS [7,8] project. The rest of the paper is organised as follows. Section 2 provides the motivation behind the OPTIMIS project. Section 3 describes the cloud bursting model and the main capability requirements associated with it while Section 4 describes the various models that form the cloud brokerage service model. Section 5 proposes a functional architecture that implements the model identified in the previous section. Section 6 and 7 looks at the specific case of use of such a broker based cloud service model for both storage and compute services. Section 8 concludes the paper with future directions for this work.

II. Motivation

As recognized in previous literature [9], cloud technologies and models have yet to reach their full potential and many of the capabilities have not yet been developed and researched to a level that allows their exploitation to a full degree. For example, there is currently a lack of efficient automation of the processes underpinning the management of internal clouds and interaction between public-private clouds.

The EU FP7OPTIMIS [7] project aims to rectify this situation by tackling the issue of optimisation over the whole cloud service life-cycle to enable a cloud ecosystem where both the internal clouds and cloud customer-provider relationships are managed optimally by incorporating economical and ecological sustainability, risk and trust. The high-level objective of the OPTIMIS project is to enable as open, scalable, dependable and virtualized cloud service ecosystem that will improve the delivery of adaptable, reliable, self-service, secure, elastic, auditable, and sustainable IT services. The overall aim is to allow organizations to automatically and seamlessly externalize services and applications into trustworthy and auditable cloud services.

The OPTIMIS project aims to offer functionality substantially beyond that of current cloud infrastructures. Contemporary solutions, both research projects and commercial products, have focused on providing functionalities at levels closer to the infrastructure, e.g., improved performance for virtualization of all, compute, storage, and network resources, as well as necessary raw functionality such as virtual machine migrations and server consolidation [10]. However such focus provides solutions that address only partial problems such as packaging a software stack in a virtual machine image. In order to move from a basic cloud service infrastructure to an improved cloud service ecosystem, there is a great need for supporting higher-level concerns in a comprehensive manner. OPTIMIS aims to enable the cloud service eco-system by developing open, scalable and dependable service platforms and architectures that allow flexible and dynamic provision of advanced services.

The advancements towards an open cloud service ecosystem targeted by OPTIMIS are best explained by means of use case scenarios that showcase selected capabilities that are being developed by the OPTIMIS consortium in a likely commercial context.

This paper focuses on the security of the cloud bursting and cloud brokerage / aggregation use-cases. Related issues of how the heterogeneity of the difference cloud environments can be abstracted away using a combination of commonly agreed upon APIs as well as a higher level programming environment that the application developers can develop against, thought important, is not discussed in this paper. Similarly, a use case focusing on cloud-based application development, though part of OPTIMIS, falls out of the scope of this paper.

III. Cloud Bursting

Consider the case of a company owning their own cloud infrastructure, a private cloud and willing to use resources from an external cloud provider, for certain time intervals and given certain circumstances triggering this use. Such a capability, termed cloud bursting, would enable the organization to scale out their infrastructures and rent the resources from a third-party provider if and when needed, in a seamless manner. The renting of the external resources exponentially improves the elasticity of the company's IT infrastructure and allows them to confront the fluctuations on demand dynamically. In this section we look into the salient features and requirements for implementing a cloud bursting capability.

In order for the cloud bursting mechanism to work, the internal cloud of the company has to provide mechanisms to detect and determine their own-status: this is, to verify to what degree the provisioned services fulfil the established service level agreements (SLAs) and auxiliary requirements like energy consumption.

In addition to this, the private cloud's management mechanism also has to determine the nature of the services running in the internal cloud – its criticality, data privacy concerns, regulatory compliance issues and other requirements or constraints that have to be considered before migrating services and data to an external cloud provider. In order to benefit from speed, agility and economies of scale, this decision making step has to be driven by a set of policies. Such policies could identify, for example, that a particular compute service uses a non-proprietary algorithm and only process public or non-sensitive data is a suitable candidate for bursting to a public cloud when the internal cloud environment is constrained. In the presence of an optimised cloud bursting infrastructure, such activities should be completed with the minimum cost for the private cloud owner. However, it can also be the case that due to regulatory constraints, for example, the services running on the private cloud infrastructure are not allowed to migrate to a public cloud under any circumstance, regardless of the level of assurance that the public cloud provider may offer.

The cloud bursting orchestration layer has to be able to make these decisions based on the various SLA requirements and metadata associated with the applications and the data used by them. The policy-based service management mechanism allows cloud infrastructure to scale out, selecting resources according to the customer's requirements without human intervention. The internal cloud management system will be able to take such decisions both on the basis of service level, and of other non-functional properties such as security, cost, etc., that require synthesis of information from within and outside the internal cloud. An example could be to pro-actively force the migration of non-critical services whose SLAs are being met, in order to assure that a critical service that cannot be migrated receives all resources it requires to fulfil its SLA.

To ensure the desired functionality in a dynamic environment, it is necessary to implement several auxiliary management capabilities, in addition to security and Quality of Service (QoS) optimisation, including, for example, the correct management of license instances in the external cloud.

In addition, the external cloud has to be able to satisfy the over-capacity needed by the internal cloud with very short advance warning and it should also be able to load-balance its workload in such a way as not to suffer unacceptable service deterioration to existing customers. The external cloud should also execute the workloads dynamically, billing the resources by usage.

In order to provide a working cloud bursting environment, the architecture being developed by OPTIMIS makes use of the following capabilities, among others:

– Common management and operational interface – Since the cloud ecosystem consists of multiple cloud providers, each with its own management and operational interfaces, it is required that intermediary services (like the broker introduced in the next section) manage the compatibility of providers of cloud infrastructure through the use of appropriate API adapters and a management layer and that the individual providers adhere to and conform to a set specifications
– A set of monitoring tools for capacity needs deployed on each collaborating cloud that can also be used to measure SLA parameter readings.
– A common agreed format to specify SLA requirements and capabilities.
– Providers can be categorised and rated according to service offering parameters as well as non-functional aspects such as risk, trust, cost and carbon usage.

IV. Cloud Brokerage Model

As a generic definition, a Cloud service broker creates a governed and secure cloud management platform to simplify the delivery of complex cloud services to cloud service customers. They enable customers to realize the full potential that a cloud provider has to offer by creating a trusted, governed and secured cloud management platform between the provider and the consumer of the cloud services. They also help enforce the correct IT policies and effectively handle SLAs between cloud providers and cloud service consumers.

As such however, the term cloud service broker is a loosely defined one and is meant to serve the needs of several different models.

Enterprise use of multiple cloud providers: in this model, an enterprise organisation makes use of services provided by various cloud providers to fulfil an internal process. The main benefit of using the broker in this scenario is its ability to integrate more than one provider. For examples the process could be a one-off marketing initiative that uses CRM data from a SaaS provider like Salesforce exposed through their API, data captured within internal systems and data stored in cloud storages like S3 [11] or databases like SimpleDB [12]. In addition, the enterprise also decides to use an IaaS service from Sun to perform the necessary processing of the gathered data. OPTIMIS examines the generic requirements of such a process composition from the enterprise's view point and the necessary technical resources necessary to realise such an orchestration. It covers the broad range of topics from service development to VM management and placement.

From the point of view of the cloud service providers, extra capabilities need to be exposed to enable the enterprise to fulfil all orchestration requirements. These include, among others, common requirements such as trust establishment, federated identity management, access management; Web

Service based API calls for managing and operating the resources etc.

Brokering multiple providers to provide a SLA-based tiered pricing model: in this scenario, an enterprise approaches a cloud broker with a given set of functional and SLA-based requirements and the cloud broker then picks up the best match in terms of the functions as well as variables like pricing, SLA parameters and other non-functional requirements like compliance and certification capabilities. The cloud broker could provide just the broker service or it could use its middle-man status to provide seamless and federated identity management, access management, policy enforcement, and audit capabilities to the enterprise, as shown in the illustration below.

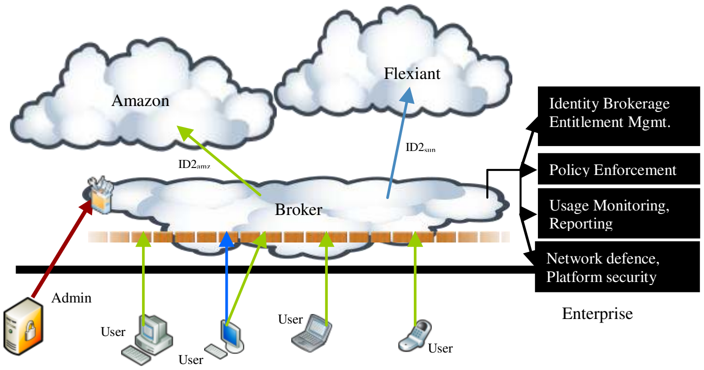
Amazon
Flexiant
$ID2_{amz}$
$ID2_{sun}$
Broker
Identity Brokerage Entitlement Mgmt.
Policy Enforcement
Usage Monitoring, Reporting
Network defence, Platform security
Enterprise
Admin
User
User
User
User

Fig. 1. The Cloud Broker as an entity that broker between multiple cloud providers to provide an SLA-based tiered pricing model.

Cloud aggregation ecosystem (CAE): this scenario offers the potential to treat both IT and business functions as a series of interconnected cloud services. CAE offers a means to architect a Service Oriented Infrastructure (SOI) on the cloud that is built on the fusion of: (a) composition of loosely coupled services based on an evolution of Service Oriented Architecture (SOA) principles applied to services that reside on cloud platforms (b) distributed management of ICT resources applied on federations of cloud platforms and (c) a network resource management based on a federated Operational Support System (OSS) architecture built on top of an in-cloud Network as a Service (NaaS) offering.

This scenario adds the capability to incrementally build new service offerings by mixing together reusable functions (common capabilities) provided by off the shelf components and 3rd party cloud platforms in a new offering. More specifically, CAE refers to the federation of a set of distributed virtual hosting environments for the execution of an application, integrating value-adding services (VAS) with these hosting environments, and providing a single (logical) access point to this aggregation. In the simplest form, the CAE can be thought of as an extension of the SLA-based brokerage model with the additional ability of the cloud provider to in turn use other cloud services to provide its offering. From the perspective of the application consumer, these federations are transparent and constitute an integral part of the service being offered.

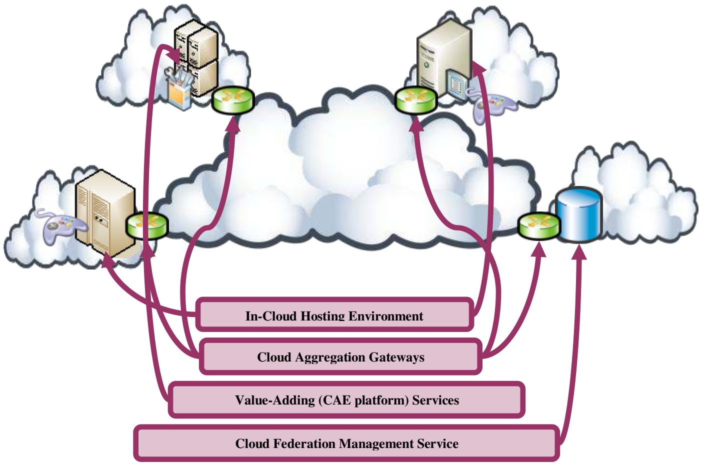
In-Cloud Hosting Environment
Cloud Aggregation Gateways
Value-Adding (CAE platform) Services
Cloud Federation Management Service

Fig. 2. The conceptual layering of the various services provided by the cloud aggregation ecosystem model.

V. SECURE CLOUD BROKER ARCHITECTURE

Now that the various forms of cloud brokerage has been introduced and its salient features explained, in this section we propose an architecture that can be used to implement the broker model involving the brokering of multiple providers to provide a SLA-based tiered pricing model to the customers of the broker.

From the discussion above the requirements of a cloud service broker can be stated as the ability to:

- Ensure data confidentiality and integrity to service customers.
- Effectively match the requirements of cloud consumer with the service provided by the provider.
- Negotiate with service consumers over SLAs.
- Maintain performance check on these SLA's and take actions against SLA violation.
- Effectively deploy services provided by the cloud provider to the customer.
- Manage the API so that provider doesn't learn anything about the identity of the service consumer.
- Securely transfer customer's data to the cloud.
- Enforce access control decisions uniformly across multiple clouds.
- Scale resources during load and provide effective staging and pooling services.
- Securely map identity and access management systems of the cloud provider and consumer.
- Analyze and take appropriate actions against risks.
- Handle cloud burst/spill-over situations effectively.

The functional components needed by the broker to provide these broker capabilities are specified in Figure 3 below.

The *Cloud API* is a mechanism through which consumers can interact with the cloud broker for performing cloud related actions including creating and managing cloud resources like compute and storage components. The *Deployment Service* unit handles the deployment of services which cloud provider offers to the service consumer. To exemplify, through this unit the broker requests the cloud provider to start a virtual machine which the consumer can use for compute purpose or to create a storage space for user's data.

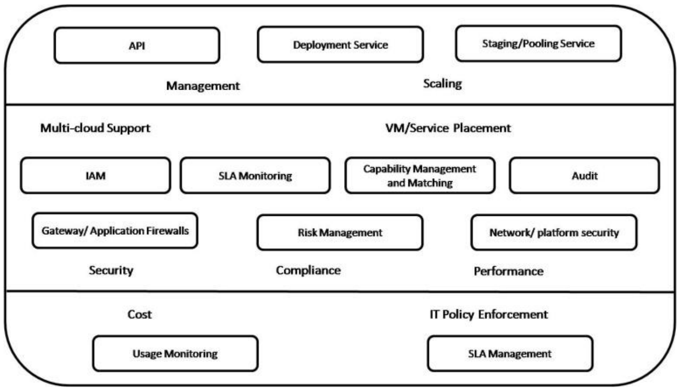

Fig. 3. Functional components of a cloud service broker providing brokerage services between multiple cloud providers.

The *Staging/Pooling Service* unit is to handle cloud burst and spill over situations. Since Cloud services are offered on pay-as-you-go basis, and are highly scalable, service consumers can demand extra resources and the broker need to ensure that a suitable cloud provider is in place to handle the request. The broker interacts with the respective cloud provider on which service consumer's applications and data are currently residing in a proactive manner to demand more resources. In case the cloud provider cannot meet the demand for the extra resources, the staging and pooling unit will find out another suitable match from the pool of underlying cloud providers and request the resource.

Identity and access management unit is a vital unit as it not only verifies its own employees but also keeps record of all the service consumers - required enterprise details, assigned cloud providers, type of service like storage or compute and classification criteria. The classification criteria is decided during the SLA negotiation phase between the broker and the consumer and based on it, broker grants access to the employees of the consumer enterprise. As will be seen later on, the enterprise's cloud management entity forwards a storage or compute request to the broker on behalf of its employee along with the classification criteria. Identity and access management unit then generates a one-time token on the basis of assigned classification criteria.

The *SLA monitoring* unit constantly monitors all the SLAs negotiated by the SLA management unit. It checks to see if there are any impending SLA violations and if any, to take the specified preventive measures. The *Capability Management and Matching* unit keeps tab of all the capabilities provided by the various providers like its delivery and deployment model, security mechanism, fee structure, IT functionalities and all other necessary details. Whenever a service consumer approaches the broker for cloud services, this unit matches the consumer's functional and SLA requirements with the services each cloud provider offer and finds the most appropriate and suitable match. This unit is also referred by the staging and pooling unit during sudden spike in demand of storage space. The *Audit* Unit periodically audits broker's platform as much as possible using capabilities provided by the cloud provider. Audit is performed to ascertain the validity and reliability of information and to provide an assessment of internal controls. *Firewall* blocks malicious traffic to and from the various components exposed by the interfaces of the cloud broker service.

The *Risk Management* unit identifies, assess and prioritize risks on the basis of effects of events. It targets to minimize, monitor and control the probability of such unfortunate events. The strategies to manage the risks include avoiding the risks, reducing negative effect of risks and accepting some or all of the consequences of a particular risk. In addition to the firewall module, the *Network/ Platform Security* unit manages the overall security of the broker's platform. Provisions and policies are adopted to prevent unauthorized access, misuse, modification or denial of network or network accessible resources. It provides a protection at the boundaries of the platform by keeping out intruders. Furthermore, the intrusion detection system which is a part of this unit focuses on protecting data from malware, virus, worms, or trojans.

The *Usage Monitoring* unit monitors the usage of the services by the cloud consumer, and generates monthly bills for them. The *SLA Management* module controls all the SLAs in place between the broker and consumer.

VI. STORAGE IN A BROKERED CLOUD MODEL

Given the component architecture in the previous section, we now look at how some of these components work together in an operational view of the system. For this, we here describe the sequences of steps taken by the various parties involved in order to provide a cloud based storage service to an enterprise end user using a brokered cloud model. For the sake of simplifying the presentation, it is assumed that the SLA based negotiations has taken place and the cloud provider bas already been chosen by the broker based on the criteria.

The main entities involved in the process are the end user within an enterprise, the enterprise system's internal identity and access management services, the enterprise cloud system represented by the cloud portal, the external broker service and the third party cloud providers.

First we look at the process, show in Figure 4 that is followed when the end user is in need of a cloud service in order to save data into the cloud. For ease of explanation, the data transmitted in the packets of a specific step is mentioned in brackets.

Step 1 (user credentials and storage request) - User sends a storage request to Enterprise Cloud Portal with its user id and password to store data on the cloud.

Step 2 (user id, password and storage request) - Cloud portal then sends the same request with user id and password to enterprise IAM. The IAM verifies the user through its identity information and if user is authentic, it checks the type of request user has sent. If the enterprise IAM can verify that the user is legitimate, it grants access rights as per the access management / classification criteria, and if it cannot, then it responds in negative.

A. *Storage of data in a brokered storage service*

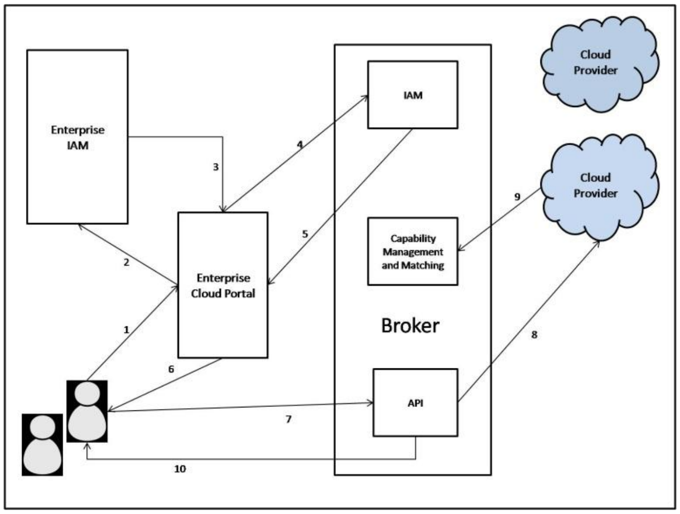

Fig. 4. Steps involved in a user performing a data storage operation on a cloud service using a Broker.

Step 3 (Authentication and authorization response) - After authenticating the user, IAM gives access rights to the user based on the type of request along with the classification criteria. If IAM authorizes the request, then the Enterprise Cloud Portal converts the identity and access rights to an external token containing the classification criteria and storage request sent by the user, encrypts it and forwards it to the Broker IAM. If the response is in negative, cloud portal informs the user that it's request has been denied.

Step 4 (encrypted external token and storage request packet) – Cloud portal forwards external token along with the storage request packet to the broker IAM, which then decrypts the packet using the cloud portal's public key. This verifies the integrity of the packet received from the cloud portal. On the basis of the decrypted packet, broker's IAM module generates a one-time access token. This one-time access token will allow the user to get access to the cloud assigned to the respective enterprise. Along with the token, the broker IAM issues a URI through which user can get access to broker's API. Furthermore, broker encrypts both the one time access token and URI and sends it back to the enterprise cloud portal.

Step 5 (Encrypted one time access token and URI packet) – After validating the packet coming from enterprise cloud portal, broker IAM sends encrypted one time access token and URI to cloud portal. Enterprise cloud portal decrypts it, using the private key of the broker and forwards it to the respective user. Presence of cloud portal keeps the identity of user anonymous to the broker.

Step 6 (One time access token and URI) - Cloud portal forwards the decrypted one time access token and URI of the broker's API to the user.

Step 7 (URI, one time access token and data)- Using the URI sent by cloud portal, user access the broker's API and sends to it the token and the storage data to be pushed on to the cloud. The broker's API validates the user using the token (broker's API internally checks the validity of the token with broker's IAM). If the token is correct, broker's API stores the data onto the cloud.

Step 8 (Data) - Broker's API verifies the user through the one-time access token and pushes the data sent by the user onto the cloud. In this step, broker's API only forwards the data to the cloud without any other information to prevent the cloud provider from knowing anything about the identity of the enterprise. Cloud provider only knows that the data came from the broker.

Step 9 (ID) – Cloud provider stores the data in a "bucket" inside the cloud and gives the externalised information about the position of the data to the broker's Capability Management and Matching (CMM) unit. This helps broker to generate retrieval data access token. This type of information that the cloud provider passes on to the broker about the location of the data varies from cloud to cloud. The cloud provider also encrypts the location information packets with its secret key before forwarding it to the broker.

Step 10 (UID)-Broker maps the location information provided by the cloud provider with a unique identity. For example if broker receives location information from the cloud provider as (CL1, BK15), it then maps it to a unique id like (B1 15). This is done so as to abstract away the cloud provider specific service details. This internal mapping is kept within the broker's database. The broker then forwards the UID to the user through Broker's API as an acknowledgment. The user stores this UID and uses it for future retrieval request.

One of the main advantages of this architecture is that none of the internal information of the enterprise is revealed at broker's or cloud provider's platform. At the same time, the cloud provider specific details are also abstracted away from the enterprise or the end user.

B. *Retrieval of data in a brokered storage service*

Now that the data has been stored at the cloud provider, this part explains how it can be retrieved by the user at a later stage using the broker service.

Step 1 (user credentials, UID and retrieval request) - User forwards a retrieval request to Enterprise Cloud Portal with its user id, password and UID to retrieve data from the cloud.

Step 2 and *Step 3* is the same as the first architecture, UID is not provided to Enterprise IAM as it doesn't have any information about it.

Step 4 (encrypted external token, UID and retrieve request packet) – Enterprise cloud portal forwards the encrypted external token, UID and retrieve request packet to the broker. Broker decrypts the packet using cloud portal public key. On the basis of the UID and the retrieve request, the broker generates a one-time access token along with a URI. The broker forwards the encrypted packet of one time access token and the URI to the cloud portal.

Step 5 (Encrypted one time access token and URI packet) – Enterprise cloud portal decrypts the packet and forwards the decrypted content to the user.

Step 6 (One time access token and URI) – Cloud portal forwards one time access token and URI of the broker's API to the cloud.

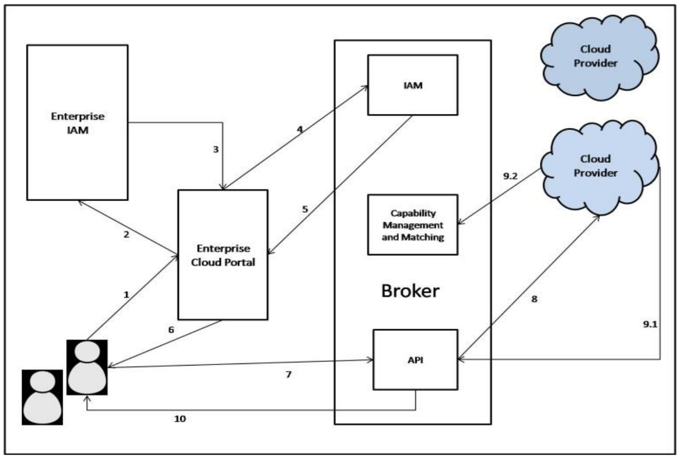

Fig. 5. Steps involved in a user performing a data retrieval operation on a cloud service using a Broker.

Step 7 (URI and one time access token) – User access broker's API using the URI sent by the cloud portal and feeds in the token. This token slightly differs from the token generated in storage request and the broker's API internally checks the location of the data stored on the cloud.

Step 8 (ID) – The broker internally translates the ID to the cloud provider specific format (CL1, BK15) and forwards the encrypted ID to the cloud provider. The cloud provider decrypts the ID using its secret key and sends the requested data to broker's API. In addition, it informs the broker if all or parts of the stored data have been retrieved.

Step 9.1 (Data) – Cloud provider retrieves the data stored in the cloud on the basis of the ID provided by the broker's API and forwards it to the broker's API.

Step 9.2 (ID) – Cloud provider also informs the broker if the request was for just access of the data or deletion of all the data or part of the data. If all the data is removed from the cloud, the broker only sends the data to the user and if not, it sends the UID along with the data to the user.

Step 10 (Data and UID) – Through broker's API, the user receives the requested data and UID of the remaining data.

VII. COMPUTE IN A BROKERED CLOUD MODEL

Just as with the storage use case, when the enterprise user wishes to use a compute platform in the cloud by starting up a new virtual machine, the service broker plays an intermediary role, but unlike the storage case, the nature of the compute example necessitates the direction access of the cloud provider platform by the enterprise end use. There are several steps involved in starting up the compute environment as well as for the user to actually access the environment and eventually shut it down when there is no more use for the environment. These steps are described below.

Step 1 (user credentials and compute request)- User forwards the compute request along with its user id and password to enterprise cloud portal.

Step 2 (user id, password and compute request) – Enterprise cloud portal then sends this request, the user id and password to enterprise IAM. IAM authenticates the user on the basis of its user id and password. Furthermore, IAM checks that the user is entitled to and accordingly grants access permission to the user.

A. *Compute instance creation*

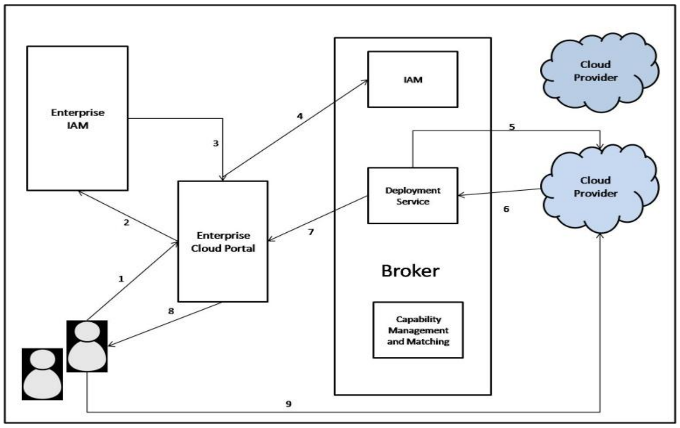

Fig. 6. Steps involved in a user creating a compute environment on a IaaS cloud service through a Broker service.

Step 3 (Authentication and authorization response) - On the basis of user's identity, IAM verifies the user and checks the role the user is assigned to. After authenticating the user it gives access rights along with classification criteria. If IAM responds positively, cloud portal converts the packet received in step 2 to an external token and encrypts it together with the compute request and forwards it to the broker. If the response of IAM is a negative, cloud portal informs the user about the denial of request.

Step 4 (encrypted external token and compute request packet)- Cloud portal forwards the encrypted external token and compute request packet to the broker, broker decrypts the packet using its public key. On the basis of the compute request, broker contacts cloud provider to start a virtual machine instance.

Step 5 (encrypted request to launch virtual machine instance) – After validating that the packet came from genuine source, broker requests the chosen cloud provider to start a virtual machine instance. *Step 6* (Encrypted ID, elastic IP address and key packet) - Cloud provider acts on the request of the broker and launches VM. Further, it replies back to the broker with encrypted packet containing ID (information about the location of the VM), dynamic IP address of the VM and key details using which the user can SSH into the VM.

Step 7 (Encrypted UID, elastic IP address and key packet) – Broker decrypts the packet received from the cloud provider using its secret key. It then maps the ID with a UID in the same manner as in the storage architecture and forwards an encrypted packet containing UID, dynamic IP address of the VM and the access key, to the enterprise cloud portal.

Step 8 (UID, elastic IP address and key) – Enterprise cloud portal decrypts the packet received from the broker using its private key. It then forwards the decrypted packet to the

respective user.
Step 9 (IP address and key) – User stores the UID for future reference and with the help of dynamic IP address and key, it SSH to the virtual machine. Thus a connection is established between the virtual machine and the user.

B. Compute instance shutdown

Once the compute platform has been used by the enterprise user, it needs to be shutdown. The following steps help perform this action.

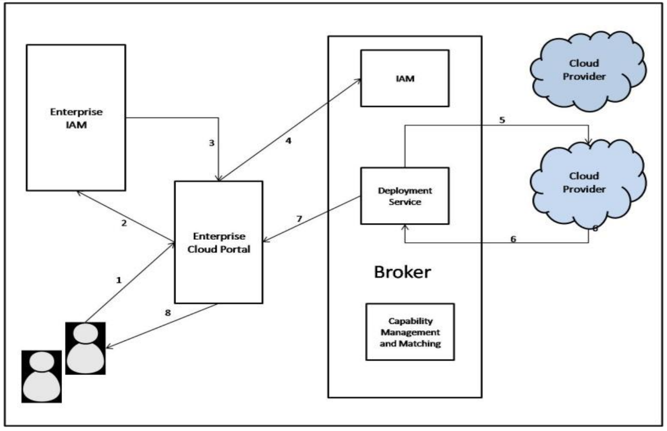

Fig. 7. Steps involved in a user shutting down a compute environment on a IaaS cloud service through a Broker service.

Step 1 (User credentials, UID, and VM shut down request): User sends the request to shut down the VM instance to the cloud portal, along with user id, password and the UID assigned by the broker.
Step 2 and *Step 3* works the same way mentioned in architecture 3, with the UID not being provided to Enterprise IAM as it doesn't have any knowledge about it.
Step 4 (encrypted packet of external token, UID and VM shut down request): Enterprise cloud portal informs the cloud broker to shut down the VM instance initiated with the respective UID.
Step 5 (encrypted VM shut down request and ID packet): Broker decrypts the packet with its public key. After that broker checks using the translated UID, the identity of the cloud provider and the ID of the VM issued by the cloud provider it sends an encrypted packet containing ID and VM shut down request to the respective cloud provider.
Step 6 (Acknowledgment) Cloud provider decrypts the packet using its secret key and on the basis of the ID it shuts down the VM. Further, it sends an acknowledgment to the broker with the timestamp.
Step 7 (Acknowledgment): Broker informs cloud portal that VM instance has been shut down.
Step 8 (Acknowledgment): Cloud portal informs the same to the user.

These steps show how the architecture is able to let the cloud broker seamlessly mediate between the enterprise (and its user) and the cloud provider, while at the same time allowing the user to directly access the compute environment at the provider when it is required.

VIII. Conclusion and Future Work

The cloud delivery model has been suggested as the panacea for complex as well as inflexible IT systems that can be ill afforded by small and large enterprises alike. However the lack of mature operational models like that of cloud bursting and cloud brokerage has meant that the cloud capabilities have not yet been developed and researched to a level that allows their exploitation to a full degree. This has contributed to a lack of uptake of the cloud computing model in all but the simplest of enterprise IT setups. In this paper we introduced the concepts of cloud bursting, cloud brokerage and cloud aggregation and identified the capability requirement of the entities. The cloud brokerage model was examined in further detail, identifying the steps necessary to provide an efficient cloud broker service in storage and compute use case scenarios.

As future work we plan to implement the steps identified in this paper with the focus on the capability to perform the SLA and security requirements matching correctly between user requirements and what the cloud provider can provide. We also plan to provide a similar treatment to the cloud bursting and cloud aggregation models.

References

[1] NIST Definition of Cloud Computing v15, http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc
[2] CRM - salesforce.com, http://www.salesforce.com/
[3] Windows Azure Platform, http:///www.microsoft.com/windowsazure/
[4] Google App Engine, http://code.google.com/appengine/
[5] Amazon Elastic Compute Cloud(Amazon EC2), http://aws.amazon.com/ec2/
[6] "An SME perspective on Cloud Computing", ENISA, Nov 20, 2009
[7] OPTIMIS - Optimized Infrastructure Services, http://optimis-project.eu/
[8] "OPTIMIS: a Holistic Approach to Cloud Service Provisioning", Ana Juan Ferrer. Francisco Hernandez, Johan Tordsson, Erik Elmroth, Csilla Zsigri, Raul Sirvent, Jordi Guitart, Rosa M. Badia, Karim Djemame, Wolfgang Ziegler, Theo Dimitrakos, Srijith K. Nair, George Kousiouris, Kleopatra Konstanteli, Theodora Varvarigou, Benoit Hudzia, Alexander Kipp, Stefan Wesner, Marcelo Corrales, Nikolaus Forgo, Tabassum Sharif, Craig Sheridan, Submitted for publication.
[9] EU Expert Group, The Future of Cloud Computing, http://cordis.europa.eu/fp7/ict/ssai/docs/cloud-report-final.pdf
[10] "VMware vMotion for Live Migration of Virtual Machines", http://www.vmware.com/products/vmotion/
[11] Amazon Simple Storage Service (Amazon S3), http://s3.amazonaws.com/
[12] Amazon SimpleDB, http://aws.amazon.com/simpledb/