PIK, Vol. 36, pp. 153–159  DOI 10.1515/pik-2012-0036

# Optimization of Elastic Cloud Brokerage Mechanisms for Future Telecommunication Service Environments

**Konrad Campowsky**

Fraunhofer Institute FOKUS
Kaiserin Augusta Allee 31
10589 Berlin
Germnay
konrad.campowsky@fokus.fraunhofer.de

**Giuseppe Carella**

Technische Universität Berlin
Straße des 17. Juni 135
10623 Berlin
Germnay
giuseppe.carella@tu-berlin.de

**Thomas Magedanz**

Technische Universität Berlin
Straße des 17. Juni 135
10623 Berlin
Germnay
thomas.magedanz@tu-berlin.de

**Florian Schreiner**

Fraunhofer Institute FOKUS
Kaiserin Augusta Allee 31
10589 Berlin
Germnay
florian.schreiner@fokus.fraunhofer.de

## Abstract

Cloud computing mechanisms and cloud-based services are currently revolutionizing Web as well as telecommunication service platforms and service offerings. Apart from providing infrastructures, platforms and software as a service, mechanism for dynamic allocation of compute and storage resources on-demand, commonly termed as "elastic cloud computing" account for the most important cloud computing functionalities.

Resource elasticity allows not only for efficient internal compute and storage resource consumption, but also, through so called hybrid cloud computing mechanisms, for dynamic utilization of external resources on-demand. This capability is especially useful in order to cost-efficiently cope with peak-workloads, allowing service providers to significantly reduce usually required over-provisioned service infrastructures, allowing for "pay-per-use" cost models.

With a steadily growing number of cloud providers and with the proliferation of unified cloud computing interfaces, service providers are given free choice of flexibly selecting and utilizing cloud resources from different cloud providers. Cloud brokering systems allow for dynamic selection and utilization of cloud computing resources based on functional (e.g. QoS, SLA, energy consumption) as well as non-functional criteria (e.g. costs).

The presented work focuses on enhanced cloud brokering mechanisms for telecommunication service platforms, enabling quality telecommunication service assurance, still optimizing cloud resources consumption, i.e. saving costs and energy.

Furthermore this work shows that by combining cloud brokering mechanisms with standardized telecommunication service brokering mechanisms an even greater benefit for telecommunication service providers can be achieved as this enables an even better cost-efficiency since different user segments can seamlessly be served by allocating different cloud resources to them in a policy-driven manner.

## 1 Introduction

Cloud hosting and computing mechanisms have gained broad attention in recent years attracting steadily increasing numbers of service providers by providing means to optimize storage and compute resource consumption and to allow for outsourcing of infrastructure and service management costs.

Elastic cloud computing, defined as the capability of cloud platforms to dynamically up- and down-scale resources according to current load situations, is one of the most important mechanisms of a Infrastructure as a Service (IaaS) cloud infrastructure. It allows for efficient resource utilization, enabling pay-per-use cost models strongly related also Green-IT and autonomic computing mechanisms.

By utilizing converged, all-IP, access-network-independent service and session control platforms such as the IP Multimedia Subsystems (IMS) [2] an increasing number of telecommunication service providers are currently consolidating their service infrastructures towards converged, Next Generation Network (NGNs) service delivery platforms

(SDP). Although these SDPs are sought to greatly reduce new telecommunication service's time-to-market, significant upfront investments into IT service infrastructures and SDPs are usually still required. Thus in many cases service roll-outs are still risky enterprises with an uncertain return on invest. With cloud computing mechanisms applied to IMS-based service infrastructures, IMS service providers are charged on a pay-per-use basis, significantly lowering the risk of unsuccessful investments. By efficiently combining service and cloud brokering mechanisms NGN service providers are empowered to efficiently provide different service qualities to different user segments allowing for additional cost saving strategies.

This work is mainly motivated by the following rationale. Firstly, current elastic cloud computing mechanisms (such as Amazon's Elastic Compute Cloud [1], Rackspace, CloudSigma and ElasticHosts) are cloud provider specific, thus they are not empowering service providers to flexibly deploy and release resources across multiple cloud provider platforms. Therefore these solutions per se do not support dynamic and seamless migration of services between multiple cloud provider infrastructures and platforms thus fostering cloud provider lock-ins, rather than empowering service providers to exploit the increasing competition in the cloud provider market. Furthermore they are not sensitive to and not aware of network performance metrics (between core network and cloud service platforms), which, to a lesser degree affects typical web applications, but to a significant degree affects real-time communication service's quality (voice / video, conferencing, messaging).

Secondly, IMS-/NGN-based telecommunication service platforms can indeed be deployed on multiple cloud platforms. Even the cloud-based deployment and hosting of an IMS itself, including the idea of offering a cloud-based IMS as a service is currently investigated and no more a far out vision. By combining cloud brokering mechanisms with standardized NGN service brokering mechanisms, telecommunication service providers are empowered to flexibly provide a broad range of services that can be shaped to specific customer segments. This stems from the fact that state-of-the-art NGN service brokering mechanisms already provide means for policy-based, user-centric access to multiple service endpoints; endpoints of cloud-based services. These service brokering solutions already provide means for exposure of telecommunication services hosted on multiple cloud platforms. They are already capable of providing a broad range of service qualities (from best-effort to highly reliable service qualities) based on finely granular cost models for different user segments. Therefore by combining service brokering mechanisms with cloud brokering mechanisms low-risk service roll-outs, pay-per-use costs models and differentiated service provisioning can readily be enabled.

This work presents the design and a brief evaluation of a cloud brokering system, the FOKUS Cloud Broker Engine (CBE), for IMS-based services, as shown in Figure 1, capable of simultaneously interworking with multiple cloud platforms, furthermore capable of dynamically up- and downscale cloud resources and able to migrate cloud resources across multiple cloud platforms on-demand. Furthermore in combination with a service brokering solution, this work shows the applicability and benefits of such flexible service provisioning strategies.

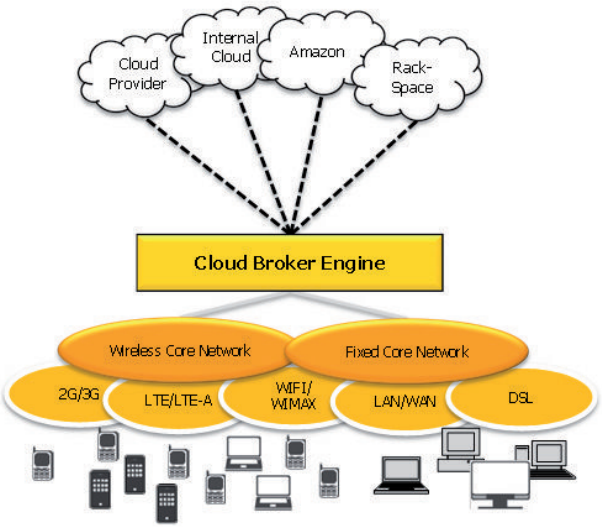


**Figure 1** Cloud Broker Engine.

The remainder of this paper is structured as follows. Section 2 provides the necessary IMS, SDP and cloud computing background as well as a related research section. Section 3 presents the design of the CBE solution, while Section 4 provides insights into the implementation and evaluation of the CBE. Section 5 describes the integration of the CBE with the FOKUS Broker. Finally Section 6 concludes the paper providing an outlook on next steps.

## 2 Background and Related Work

The elastic provisioning of IMS-based telecommunications services is still a widely unexplored and still challenging issue. To fully understand our novel solution, this section briefly introduces the IMS, IMS-based SDPs and state-of-the-art cloud computing mechanisms and provides an overview of related research which addresses QoS-aware elastic cloud brokering mechanisms similar to our work.

### 2.1 IP Multimedia Subsystem

The IP Multimedia Subsystem (IMS) [2] is a session control platform, standardized by the 3GPP, for NGN telecommunication services. It is capable of converging fixed, mobile and Internet services. It uses the Session Initiation Protocol (SIP) [3] for controlling multimedia telecommunication sessions. Through the IMS technology Internet services, like web services, e-mail, instant messaging or social networks can easily be combined with telecommunication services like VoIP, presence and videoconferencing. Notwithstanding its relevance for telecommunication service providers, the deployment of IMS-based services usually requires significant upfront investments, especially for services shared by a large amount of users such as voice- and video-streaming and -conferencing, location, presence and messag-

ing services. For the work presented here we utilized the Open IMS Core [4], Fraunhofer FOKUS' reference implementation of the 3GPP IMS, the heart of the Open IMS Playground [5], a unique testbed for NGN telecommunication systems and services.

## 2.2 IMS-Based Service Delivery Platforms

After the Open Mobile Alliance (OMA) had already specified several IMS service enablers (like presence, group- and list service enablers) they realized the need for an overarching service brokering functionality for controlling access to these service enablers from 3rd parties. This led to the specification of the OMA Service Environment (OSE) [6] as a common abstraction layer for IMS-based service environments. The core component of the OSE is Policy Evaluation, Enforcement and Management (PEEM) component which intercept service requests from a foreign domain as well as from any other service requestor and applies certain rules (policies) to them. Based on these policies not only authorization and authentication mechanisms are triggered, but also user-specific charging rules and can be applied. Furthermore different sets of services can be exposed to different user segments. For the work presented here, we utilized the FOKUS Broker, an integral component of FOKUS' Open SOA Telco Playground [7], an OMA compliant implementation of the OSE, which has previously been described and specified in [8] and [9].

## 2.3 Cloud Computing and Cloud Management Systems

For efficiently and economically operating telecommunication service platforms, cloud computing mechanisms are continuously becoming more and more relevant for telecommunication service providers, by offering the possibility to dynamically scale resource utilization, including "hybrid" cloud platform models which allow for on-demand outsourcing of required computing capacities. Generally speaking three cloud service types can be distinguished: Infrastructure as a Service (IaaS) (e.g. Amazon S3, Microsoft SQL Azure, Amazon EC2, Zimory), Platform as a Service (PaaS) (e.g. Google App Engine, Windows Azure) and Software as a Service (SaaS) clouds (e.g. Google Docs, Salesforce CRM). This work focuses on IaaS solutions, being agnostic to the actually hosted application, but still providing mechanisms such as elastic resource scaling. There are several open-source cloud management solutions already available like OpenNebula [10], Eucalyptus, Nimbus and Open Stack. The reason for choosing OpenNebula is mostly related to its support for a standardized cloud management application programming interface (API), the Open Cloud Computing Interface (OCCI) [11] and the support for a broad range of hypervisors (i.e. Xen/KVM/VMWare) and its capabilities for interworking with Amazon EC2 [1].

## 2.4 Related Work

Although a significant amount of research was recently conducted in the field of cloud computing mechanisms, only limited efforts have been made so far focusing on cloud-based IMS infrastructures and services.

Some of those implement a very specific solution for deploying a specific IMS service on a cloud platform, like in [12] a presence service is implemented in a cloud computing environment, not taking into account the QoS for the final user. In [13] an approach similar our approach is being described, which focuses on Web Services (instead of IMS services), which is also limited by using un-weighted round-

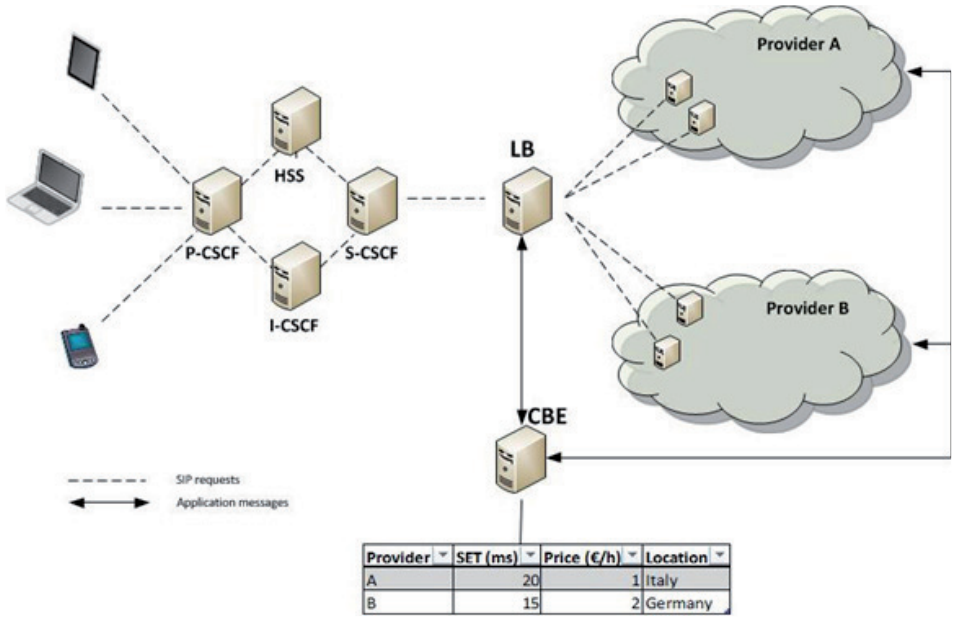


**Figure 2** Cloud Brokering Architecture.

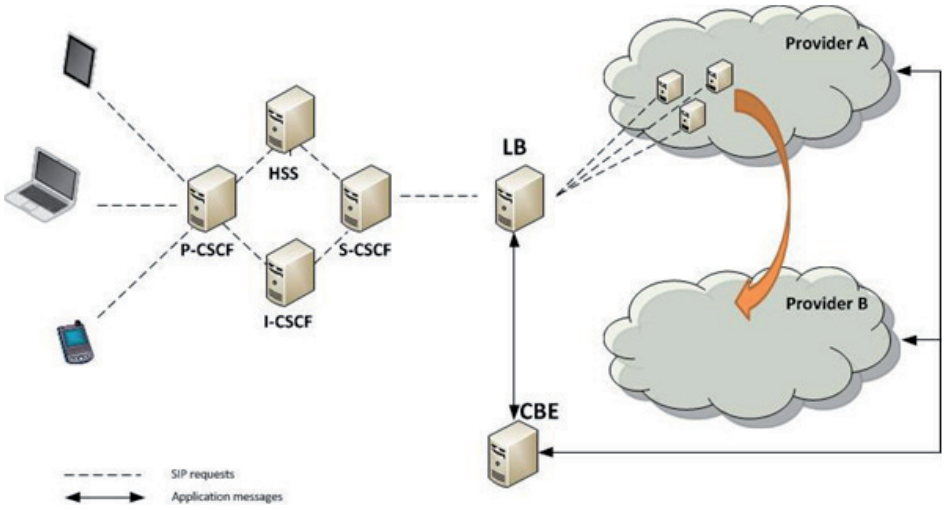


**Figure 3** IMS-based cloud service migration.

robin load-balancing algorithms and which is limited in terms of analyzed monitoring data. For our solutions (utilizing multiple distributed cloud platforms) this solution would be inappropriate as QoS could not be assured appropriately. Authors of [14] describe a solution for simplifying the deployment and scaling of cloud applications. This "profile-based" approach is similar to our approach but it takes into account only the CPU utilization of a given VM. In [15] the possibility to deploy two different cluster-based services on top of a virtualized infrastructure is being analyzed. Authors are using, similarly to our solution, a hybrid cloud infrastructure, but they do not consider real-time monitoring data to scale their services automatically.

## 3 Design of the Cloud Broker Engine

As explained in the previous sections the main novelty of this work is to be able to provide cloud-based IMS services that can flexibly be hosted on top of multiple cloud platforms at assured QoS levels optimizing resource consumption. In order to achieve this goal, we realized a system capable of monitoring and managing VMs across different cloud platforms.

The CBE consists of three main components that dynamically compose a pipeline: a monitoring aggregator, a rules engine, and a cloud management system. First the monitoring aggregator receives monitoring data from distributed VMs, networks and load-balancing systems, aggregating, preprocessing and forwarding them to a rules engine that analyzes those data and sends commands to the cloud manager and load-balancer(s) based on current situation of the pool of utilized resources, network performance and service execution times (SET); finally, the cloud manager translates those commands into specific cloud platform control invocations.

Furthermore we implemented a web-based interface allowing service providers to specify service and cost related Key Performance Indicators (KPIs), which control the decisions of the rules engine. For instance, for serving premium customers, a service provider would be interested more in the QoS and reliability of services rather than in the price-per-hour of a cloud resource. By introducing different weighting schemas for different KPIs, the behavior of the CBE is controlled. Every time a new VM is needed in order to serve an increasing load, the VM is deployed on the currently optimal cloud platform based on the previously specified KPIs.

Another capability of the CBE allows for dynamic migration of services from one platform to another cloud platform with a better ranking score. In order to choose the optimal provider, we use an algorithm that takes into account both the performance parameters specified by the user as well as the KPIs provided by the service provider, matching the latter against real-time measurement about resource utilization, network parameters and SETs.

### 3.1 Service Elasticity

Considering that each VM corresponds to an IMS AS, elastic mechanism allow to scale up whenever the CPU utilization of a current set of running resource reaches a specific threshold (a scale down in the opposite case). We utilize a monitoring tool to receive real-time data about the CPU and RAM consumption of every single VM instance. Each VM contains a monitoring agent, which reports the current status of the VM to the monitoring server(s).

Figure 2 illustrates a complete cloud-based IMS service infrastructure, where ASs are deployed on top of two cloud platforms. Users send requests to the IMS, which, based on the user service profile, routes these requests to the appropriate IMS ASs. By utilizing a layer 7 load balancer (LB) requests are distributed across different ASs. This work uses a weight-based round-robin algorithm to optimize load distribution across different VMs.

Whenever the CBE receives a trigger message it interrogates the monitoring system in order to check the situation of the entire group of VMs. When each VM is in trigger state (reached a given threshold) the CBE deploys and boots a new

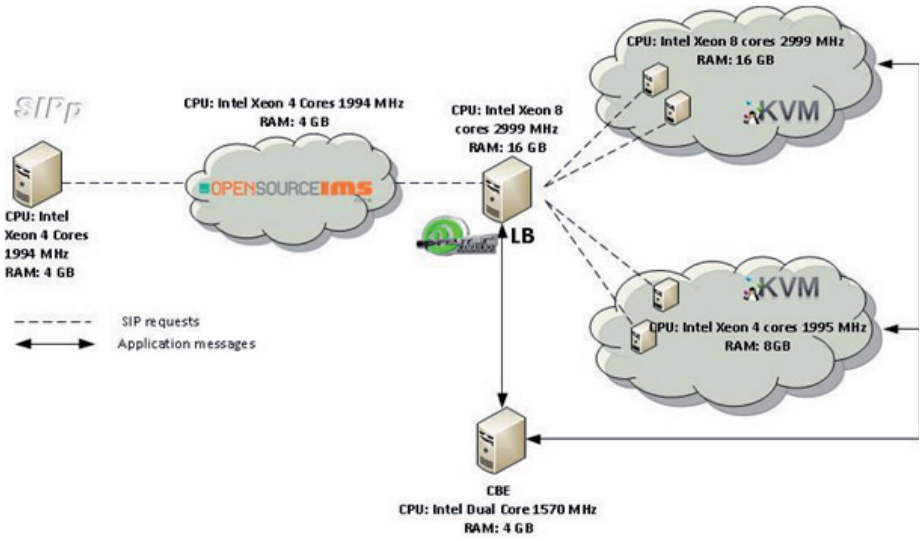


**Figure 4** Cloud Broker Engine Evaluation Setup.

VM in the optimal cloud platform chosen through the optimization algorithm.

### 3.2 Service Migration

The term "service migration" refers to a mechanism which allows switching services from one cloud provider to another. In our architecture the IMS and the load balancer are both located in the operator premises, and between them and the application servers, there is a standard, best-effort internet connection. We developed a passive monitoring infrastructure that is able to collect metrics (delay, jitter, packet-loss and actively also bandwidth) related to the network performance. Figure 3 shows the architecture of the IMS-based service migration.

Basically the CBE periodically receives information about the service execution time (SET) and additional network parameters between the load balancer and the ASs. In this scenario the SET is represented by the time between sending a SIP request from the load balancer to the backend application server, and receiving the 200 OK response. Whenever the averaged SET for a certain provider reaches a defined threshold the CBE automatically looks for an alternative provider with a better QoS and migrates the service to that particular one.

## 4 Implementation and Evaluation of the Cloud Broker Engine

For the evaluation of the CBE we deployed a testbed as depicted in Figure 4. We utilized the Open Source IMS Core [19] as a SIP-based session control platform. For the service layer we utilized Cipango [16] as SIP servlet container. We implemented a SIP User Agent Server (UAS) servlet and packaged both into the utilized VM images. We utilized OpenNebula [10] as a cloud manager and Kernel-based Virtual Machine (KVM) as hypervisor solution. For the monitoring tool we utilized the Zabbix [17] network management system designed to monitor various network, services and resource related metrics. For the load balancer we utilized the dispatcher module from Kamailio [18] an open source SIP proxy server, call router and user agent registration server.

We utilized IMS Bench SIPp [19] an IMS traffic generator to test system scalability. In addition, we deployed the Network Emulation tool Netem [20], to network performance deteriorations such as a delay between the load balancer and the domain A. After receiving a trigger from a VM located in a cloud platform affected by a network problem, the rules engine sends a command to the cloud manager, which requires 100s to migrate the service from one domain to another.

The evaluation showes that an optimal allocation of cloud resources can be achieved by applying elastic mechanisms and also that by combining elastic mechanisms and service migration mechanisms QoS levels can be assured across cloud platforms. The current system's resource allocation delay amounts to 45s and service migration delay in the given scenario of 100s has been measured and found to be acceptable for many applications, however they can still be significantly improved.

## 5 Integration of Cloud Broker Engine with FOKUS Broker

The benefit of integrating the CBE into an OSE becomes obvious. Users of Web-based telecommunication services as well as 3rd party developers are provided with a single point for accessing telecommunication service. As an example for such an OSE, we chose the FOKUS broker which allows for flexible profile based service composition. Integrating the CBE was a rather straightforward matter. In addition to de-

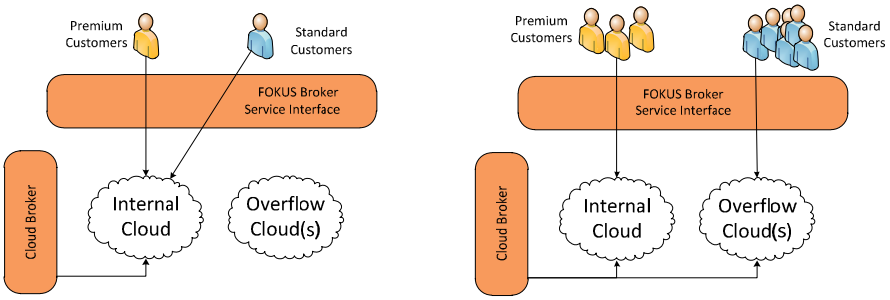


**Figure 6** Policy-based, differentiated cloud service exposure during normal operation (left) and during heightened demand (right).

ploying a service within the broker's internal runtime environment, the FOKUS broker is also able to integrate external services. We leveraged this mechanism to integrate the load-balancer endpoint provided by the CBE as such an external service as shown in Figure 5. A slight adaptation of the broker's user interface made this process fully transparent to the user.

The invocation of backend services and service enablers is being conducted in a manner that is transparent to users, optionally based on a specific user profile. Thus, service brokering mechanisms allow for exposure and invocation of one and the same service, hosted on multiple (differently expensive) cloud-based platforms, at different service qualities. By doing so, telecommunication service providers are empowered to flexibly allocate different cloud-based service resource to different user segments, where budget users are provided access to best-effort cloud resources and premium customers are served by highly reliable cloud resources as shown in Figure 5.

The same scenario allows for risk-free deployment of new services/service trials, which can initially be hosted on external cloud infrastructures and migrated to internal cloud platforms if service-uptake is successful.

We integrated the CBE with the FOKUS Broker [8]. The interworking between the FOKUS Broker and the CBE was tested in several scenarios, amongst which the successful operation of a SIP-based mediaserver, the "SIP Express Media Server" [21] hosted on top of multiple internal and external cloud platforms can be conceived as an adequate proof-of-concept, since not only SIP signaling, but also media transport performance was verified. It could be shown that continuously increasing media-streaming requests, far beyond the capabilities of a single mediaserver instance, can be served without experiencing VoIP quality deterioration, by gradually applying aforementioned up-scaling mechanism which continuously allocates resources first internally and subsequently on external clouds (published work is currently being reviewed).

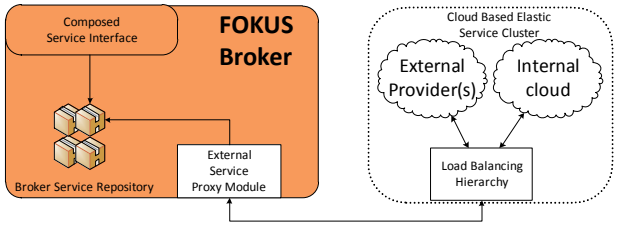


**Figure 5** Integration os the FOKUS Broker with the CBE.

## 6 Conclusion and Future Work

As shown, being able to distribute cloud resources across different cloud provider platforms yields a number of very interesting advantages for telecommunication service providers. Our work in the field of QoS-aware cloud service brokering mechanisms for telecommunication systems provides several promising results.

We were not only able to show that an optimal and efficient allocation of cloud resources can be achieved by applying elastic mechanisms, but also that by combining elastic mechanisms and service migration mechanisms QoS levels can be assured.

Furthermore, it was shown that by tightly integrating cloud brokering mechanisms with policy-based service brokering mechanisms an even greater benefit for telecommunication service providers can be achieved. On the one hand these combined systems allow for flexible allocation of resources for serving different user segments with different levels of service qualities. On the other hand by being able to flexibly operate service environments in flexible hybrid manner telecommunication service providers are able to cost-efficiently utilize internal and external cloud resources, greatly reducing the risk of unsuccessful infrastructure investments, but still being able to assure acceptable QoS levels.

We are currently evaluating the combined FOKUS Broker and the CBE solution in large-scale scenarios involving several cloud provider platforms in parallel. One of these multi-site cloud infrastructures is currently provided by the EU FP7 project BonFIRE [22], where several cloud platforms are made available across several European countries. Further improvements to the system currently involve enhancements of dynamic load and cloud performance prediction mechanisms. Furthermore in order to proof the commercial utilization of the overall system, we are currently conducting tests on the commercial platforms of Amazon EC2 later also Rackspace, CloudSigma and ElasticHosts.

## References


[1] Amazon. Amazon web services. 2011; Available from: http://aws.amazon.com

[2] 3GPP. TS 23.228. IP Multimedia Subsystem (IMS).

[3] J. Rosenberg et al., "SIP: Session Initiation Protocol," IETF RFC 3261.

[4] The Open Source IMS Core, http://www.open-ims.org

[5] Open IMS Playground: http://www.open-ims.org

[6] Open Mobile Alliance (OMA). OMA Service Environment. Approved Version 1.0.4, 01 Feb 2007.

[7] The Open SOA Telco Playground: http://www.opensoaplayground.org

[8] N. Blum, I. Boldea, T. Magedanz, U. Staiger and H. Stein, A Service Broker Providing Real-Time Telecommunications Services for 3rd Party Services, COMPSAC, 2009, 85–91.

[9] N. Blum, T. Magedanz and F. Schreiner, "Definition of a Service Delivery Platform for Service Exposure and Service Orchestration in Next Generation Networks", Ubiquitous Computing and Communication Journal, **3(3)** (2008).

[10] The OpenNebula Project: http://www.opennebula.org.

[11] Open Cloud Computing Interface (OCCI): http://occi-wg.org

[12] Wei Quan, Jun Wu, Xiaosu Zhan, Xiaohong Huang and Yan Ma, "Research of presence service testbed on cloud-computing environment", IC-BNMT, 2010.

[13] W. Iqbal, M. Dailey and D. Carrera, "Sla-driven adaptive resource management for web applications on a heterogeneous compute cloud", Cloud Computing, ser. Lecture Notes in Computer Science, M. Jaatun, G. Zhao, and C. Rong, Eds. Springer Berlin / Heidelberg, **5931** (2009), 243–253.

[14] Y. Jie, Q. Jie, and L. Ying, "A profile-based approach to justin-time scalability for cloud applications", 2009, 9–16.

[15] R. Moreno-Vozmediano, R. S. Montero and I. M. Llorente, "Elastic management of cluster-based services in the cloud", ACDC '09: Proceedings of the 1st workshop on Automated control for datacenters and clouds, New York, NY, USA, 2009, 19–24.

[16] Cipango SIP/HTTP Servlets Application Server: http://www.cipango.org/

[17] Zabbix, Open Source distributed Monitoring System: http://www.zabbix.com

[18] Kamailio SIP Server, www.kamailio.org

[19] IMS Bench SIPp, Open Source IMS benchmarking tool, http://www.sipp.sourceforge.net/ims_bench

[20] Linux Network Emulation, http://www.linuxfoundation.org/collaborate/workgroups/networking/netem

[21] The SIP Express Media Server: http://www.iptel.org/sems

[22] EU FP7 BonFIRE Project: http://www.bonfire-project.eu