An Openstack based Accounting and Billing Service for Future Internet Applications

Oladotun Omosebi[1], Stelios Sotiriadis[2], Nik Bessis[3]
[1]Department of Computing and Maths, University of Derby, Derby, United Kingdom
[2]School of Electronic and Computer Engineering, Technical University of Crete, Greece
[3]Department of Computing, Edge Hill University, UK
o.omosebi1@unimail.derby.ac.uk[1], s.sotiriadis@intelligence.tuc.gr[2], nik.bessis@edgehill.ac.uk[3]

Abstract— **In latest years, there is a corresponding growth for cloud computing services, especially for those that can be offered as modular components in order to construct Future Internet (FI) applications. In addition, the emergence of inter-clouds as the mean to achieve utilization of various services available from different providers highlights new research directions. Following the business benefits of the cloud services into more advanced and modular systems the requirements for effective accounting and billing with regards to the service usage and the accurate billing of consumers of services becomes challenging. This work focuses on cloud platforms, their platform segregations and services, and presents an accounting and billing service that can be used to rate, charge and bill consumers of within a cloud or an inter-cloud platforms. The solution is developed based on FIWARE platform, which promotes the usage of Cloud Computing in Europe by designing general purpose cloud services namely as Generic Enablers (GEs). Here we design and implement an Accounting and Billing GE that is built upon OpenStack platforms and hosted as an easily to deploy and use cloud service.**

Keywords: Future Internet, Cloud Computing, Cloud Services, Inter-Cloud, OpenStack, Accounting and billing in cloud, FIWARE

I. INTRODUCTION

In recent years, businesses have been re-accessing the gains coming from investing more in utilizing cloud resources for hosting business functions. Spending on cloud computing resources by businesses is projected to be in the billions within the near future [20]. Since, more and more businesses are leveraging the cloud to host essential business processes, it becomes necessary to ensure that cloud-based resources are properly monitored and their usage accurately charged. Beyond utilizing cloud resources to provide services to clients, the utilization of the resources by both individual and business users requires an efficient form of charging and billing. Specifically business functions are transformed to cloud-based business services and will require the monitoring of usage and accounting. Such cloud-hosted billing infrastructure could be a third-party service that is built upon the various cloud services, such as Infrastructure as a Service (IaaS) or Platform as a Service (PaaS) and Software as a Service (SaaS).

This work focuses on the accounting and billing of inter-cloud services [8], [22] based on the benefits derived from developing such a cloud-based billing platform. According to [18], among the terrific benefits is the fact that there would be no need for dedicated hardware and software licenses while all users could use the same version of the software.

The proposed solution is based upon FIWARE, which offers flexible building blocks referred to as Generic Enablers (GEs) developed specifically for Future Internet applications and services [23]. Generic Enablers (GEs) are software modules that provide specific business functions, along with protocols and interfaces for integration and communication. This highlights new requirements in terms of effectively accounting and billing applications composed from services belonging to different cloud service providers. According to [18] a basic requirement for an accounting system is to provide flexible monetization strategy e.g. involving the ease of upgrading from free trial to subscription or bundled product-based. Also a flexible pricing plan will provide automatic pre-scheduled activation and deactivation, including implementation of flat-rate monthly recurring fees. In addition, tracking and assignment of user access permissions based on service selection and billing and details on service usage experience and statistics, will allow to feed into decision-making systems and services.

With this in mind, this paper aims to develop a cloud service accounting and billing module that could be used within the FIWARE infrastructure. We will maintain particular emphasis on resource usage costing and billing, for cloud environments built on an OpenStack back-end cloud management infrastructure. The solution is implemented as a GE to query cloud resource management platforms that run on the OpenStack architecture. From hereon, section II discusses the motivation and the literature review. Section III covers essential requirements for a cloud-based billing solution, while section IV presents the model billing system, while identifying its elements. Section V presents and demonstrates the implementation of the solution, while section VI evaluates the solution and discusses the testing strategy based on a use-case model. Finally, section VII concludes with a discussion on challenges faced and expected future directions.

II. ANALYSIS OF RELATED APPROACHES

In order to ensure the commercial viability of cloud computing service, resource usage needs to be properly metered and charged in order to recoup investments. In order to explore opportunities to achieve optimum returns, there is necessity for providers to offer flexible payment models to customers that gives them freedom to modify service plans when needed [2]. This allows users to view and select models that best fit their purpose. Charging, billing and revenue sharing among inter-clouds also needs to be enabled in order to support interoperability among providers [21]. In particular, the work of [14] presents a survey of inter-cloud schedulers for service distribution among clouds.

With regards to billing models a popular billing model used in cloud computing is the pay-as-you-go model used in [10] [11], [12], [13] and [5]. In [12] authors emphasize openness and transparency in billing methods, indicating that service providers need to ensure that customers have access to a standard interface that provides them with information about the provider's "unambiguous accounting model", which indicates clearly all elements involved in computing resource consumption charges. While implementing an efficient billing infrastructure, the accurate measurement of resource usage and rating of events data, based on established pricing models need to be given high priority. In comparison to mobile telecom services, there are no open standardized metering methods for the measurement of cloud computing resource usage [12] and [17]. Thus, the function and control of accounting and billing of resource usage lies with the service provider. In order to provide a transparent billing system that enables the billed consumer to justify the bill presented, access to the resource usage information should be granted to the consumer [1].

Accounting and billing becomes a vital requirement especially by the emergence of FIWARE platform. FIWARE is an open cloud-based infrastructure built to enable development, deployment and testing of innovative cloud-based solutions for Future Internet (FI) applications and services. The platform provides a suitable environment for innovative information sharing, application deployment into a non-commercial environment and access to real data for testing [7]. FIWARE expands OpenStack by developing an environment for Future Internet providers to develop applications and services. Its simplicity, scalability and its rich features underline OpenStack. It consists of interrelated projects that enable control and configuration of resources such as processing, storage and networking through a data centre. It also enables configuration via a web-based dashboard, command-line tools, and a RESTful API, while supporting all types of cloud environments [19]. It is usually deployed as an IaaS based on several complementary services. In this work we are motivated by the openings arising the possibilities of applying accounting and billing in the concept of Future Internet. This approach will enable the adaptation of new standards and will offer new opportunities for cloud providers, web entrepreneurs and SMEs to commercialize their products.

III. DESIGN AN ACCOUNTING AND BILLING SERVICE FOR FUTURE INTERNET APPLICATIONS

This section presents the design of an Accounting and Billing Service for FI Applications. We first present an identification and analysis of key the stakeholders based on [16] as follows.

- A "Consumer" is the user that utilizes an application that consumes services offered by the subscriber. In some scenarios, the subscriber and consumer could be the same entity, where the application consumes a service offered by a service provider.
- A "Subscriber" is the user that obtains a service contract from the service provider and uses it to provide services to the consumer. This could be a company that purchases a bulk license from the service provider, while providing a license to each employee within the company. The subscriber receives bills based on cloud resource usage.
- A "Service Provider" is the user that will manage the services hosted on a cloud platform. This would represent the administrator in an accounting and billing system. The system under this role consists of the provisioning system and accounting systems. Thus user is responsible for ensuring that services are delivered based on contract agreements.
- A "Payment Provider & Partner Provider" is the user that would be relevant in a composite service delivery scenario. The service provider would collaborate with a partner who would provide a third party service focusing on managing usage accounting.

According to [1], the resource usage involves the following charging and billing models that characterize key requirements for our study. Firtsly, the Online (Real-time) and Offline resource usage that is charged as pay per use (real-time) according to a pricing structure. The interim usage events are captured by a system, and saved to storage accessible to a billing system. In real-time rating, the files are processed immediately. For offline usage, the event files are stored and processed at later stage. Secondly, the Flat Rate or Pay-Per-Use where the service offered could be charged based on a recurrent flat rate, irrespective of the resources used. The more popular pay-per-use model enables the subscriber to pay only for services used. This enables the service consumer to determine the usage of the actual resources.

Based on the aforementioned discussion, we summarize the requirements of this work per category. These are (i) data management services (for data that need to be stored and be accessible via API calls), (ii) charging and rating functions, (iii) billing generation and presentation services (billing functions available via API calls), and (iv) the billing notification service. More specifically, the "Data Management includes" User Accounting, Platform Resource, User resource usage, Pricing structure data and Interim usage costing and Usage bills. The "Charging and Rating Function includes" use post-paid and pay-per-use models. This it will retrieve interim usage data from an external system specialized to perform that function. The solution shall rate usage based on the built-in pricing structure. The "Billing Generation and Presentation includes" itemized bills that will be generated from the rated interim usage event data and monthly bills that will be generated based on the itemized bills. Finally, the "Billing Notification" will include a solution that is integrated with a message-broker in order to be able to notify users of the bills on a recurrent basis.

Next we discuss the analysis and design of the billing system and identify typical use cases. To ensure the adherence of the design to industry standards, the design of the billing system is done based on NGOSS's Business Process Framework (eTOM) model, a framework mostly used within the telecommunication environment [3]. While providing a blueprint for enabling and implementing successful business transformation, it identifies essential industry-accepted business processes required to be in place in order to achieve an efficient, effective and agile enterprise [3] and [4]. Figure 1 illustrates Identification of accounting and billing components based on eTOM functional areas.

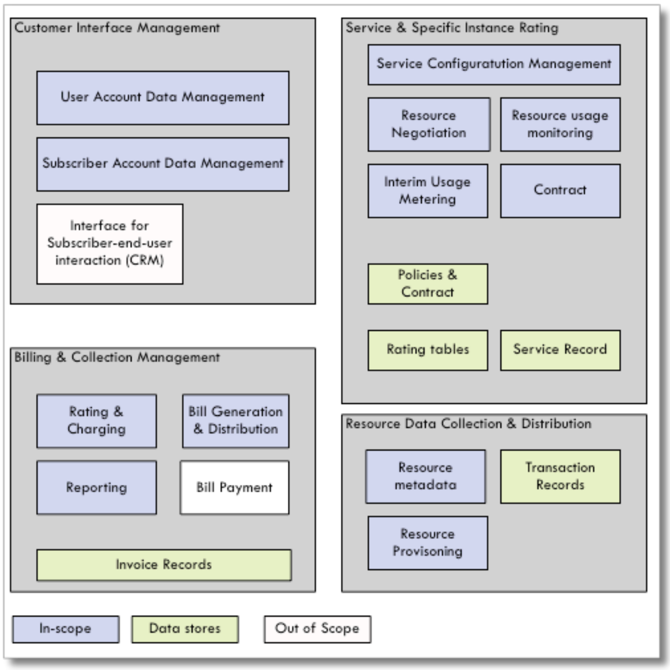

Fig. 1. Identification of accounting and billing components based on eTOM functional areas.

The accounting and billing component includes the Customer Interface Management, the Billing and Collection Management, the Service and Specific Instance Rating and the Resource Data Collection and Distribution that are described in the following subsections.

A. Customer Interface Management

This segment of eTOM level 2 covers elements that manage the interaction between the customer and the system, including contact and request management, customer analysis and reports, mediation and orchestration of customer interactions ([4]. In the scope of this project, these consist of the consumer and subscriber. In the case of a federated system, it would include the service provider. The following two components are identified under this category: a) consumer account data management: The consumer data needs to be stored to enable proper user authentication, authorization and service provisioning, these data is eventually used to generate bills against the appropriate entity and b) the subscriber account data management: Similar to consumer, the subscriber data also needs be stored for use by the service provider, in order to generate accurate usage events and to tie them to the appropriate trigger agent.

B. Service & specific instance rating

This section of the eTOM covers the mediation of usage records, including the validation, normalization, conversion and correlation of records obtained form resource usage. It also includes rating, analysis of usage records and generation of reports for other system processes [5]. The following elements fall under this category: a) Service Configuration Management: This function determines all activities involved in configuring the service for the customer and making it available for usage, b) Resource Negotiation: This is a unique aspect to this project and covers the automatic review of available options that will optimize resource usage for the customer based on the SLA, and also minimize service delivery expense for the service provider and subscriber, c) Resource Usage Monitoring: The resource ought to be monitored to ensure that it performs optimally according to the service level agreements held with the subscriber or end-user. The information acquired here also applies in the service provider's system upgrade and scalability strategy, d) Interim Usage Metering: The system shall provide a functionality to monitor instantaneous usage of resources in order to properly bill the user. The effectiveness of this function shall be based on the frequency of metering applied, taking into account the impact on latency of the system response to the user. This functionality relies on data stored per resource and users (subscribers and consumers), and e) Contract: This creates a bind between the system and the consumer, and if necessary, between consumer stakeholders. It specifies the services that will be offered, as well as the agreed service level agreements (SLAs) binding to that service.

C. Resource data collection & distribution

This section of the eTOM covers information and data collection management, process information sharing, distribution of data among system instances and processes and audit data collection and distribution [4]. The following elements fall under this category a) the Resource Metadata: the system needs to manage data on all available resources for effective provisioning, monitoring, billing and audit and b) the resource provisioning: this covers the provisioning of the requested resources for the consumer.

D. Billing & collection management

This section covers the management of customer billing, payments and debt collection [6]. The following elements fall under this category: a) the Rating & Charging: Provides the function of rating and charging usage event logs. It also handles the process of updating the subscriber's account balance, b) the Bill generation and distribution: After the usage has been rated and all charges applied, this function ensures that an accurate bill is generated and sent to the appropriate party by the pre-defined communication channel, c) the Reporting: This provides access to bills and analytics.

IV. DESIGNING THE ACCOUNTING AND BILLING OPENSTACK SERVICE

This section discusses the analysis and design of the billing service. The module is distributed across two cloud nodes of the FIWARE Lab[1] system. In particular, the Accounting and Billing Service deployed in the FIWARE Lab offers the following characteristics. The graphical user interface presents an accessible interface and enables both the user and the administrator to access the application features. The web server hosts the application and affords the users to access the application, enabling the users to consume the services provided by the module. The Accounting and Billing module encapsulates the cloud resource management logic. This exposes APIs that can be called from the web application. The first FIWARE node (called Intell-cloud) is an OpenStack environment that hosts the Apache web server on an Ubuntu instance, while also providing an OpenStack environment that can be queried and managed from the application. The Lab GEs provide services via public APIs, and are used by the Accounting and Billing module. Only the Oauth2 GE is used in this implementation. However, the Orion Message Broker GE is expected to provide messaging services, to notify users of their bills on a regular basis. Finally, the Accounting and Billing module also queries this environment, and enables the administrator to manage the cost of resources usage on the Lab OpenStack environment.

Figure 2 shows more details about the technical architecture of the Accounting and Billing module. This includes various elements like the Identity GE, the Accounting and Billing Module and the Application host. The Intelli-Cloud and the FI-LAB are the two OpenStack platform systems.

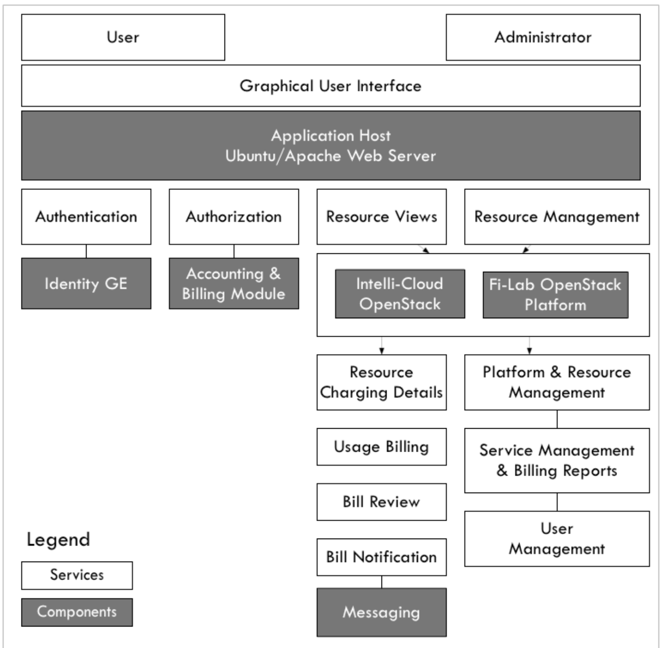

Fig. 2. Technical Architecture showing components of the service

The services of Figure 2 are discussed further below. The User and the Administrator represent the users of the application. The Graphical User Interface presents the HTML-based interface to the application, to enable access and administration. The Application Host is implemented based on Ubuntu operating system and Apache web server. These are hosted within the first FIWARE OpenStack node. Also, authentication is provided as a service by the module, and it is based on the Lab IdM Generic Enabler (GE), which provides Oauth2 authentication. All users must be authenticated against a Lab account before gaining access. Authorization is also provide based the module based on accounts already held within the module. Once a user is authenticated on Lab, an account is automatically created for the user in the module. An administrator can add more administrator accounts, and does not need to be authenticated via Lab. The Resource Views covers all the views to the platform resources. These include resource details and charges, usage details, current charges, and bill notification. These views are based on the selected platform prior to the user logging into the module. The details of the entered user account and platform selected are used to query the appropriate platform to retrieve resource and usage details. As shown in the diagram, this applies to both nodes of the FIWARE Lab. The Resource Management provides the administrator access to view platform resource details, and to allocate costs to the resources. The allocated costs will apply to all usage events generated by the user as they utilize the resource. Once the module retrieves the usage events from the platform, the costs are applied and used to generate the bills for usage. Resource management includes platform and resource management, service management and billing reports, and user management.

[1] https://www.fiware.org/lab/

V. IMPLEMENTATION OF THE ACCOUNTING AND BILLING SERVICE

In order to properly account for resource usage, stakeholders and services need to be identified, including service provider data, resources and properties, service provisioning and the accompanying rating, charging and billing. Table 1 summarizes the roles in the accounting and billing system.

TABLE I. ROLES FOR ACCOUNTING AND BILLING OF FUTURE INTERNET APPLICATIONS AND SERVICES IN CLOUDS

#	Actor	Description
1	Consumer (End-User)	Uses a thin client application to consume services offered by the subscriber.
2	Subscriber	Acts as the middleman between service offered by a service provider, and the consumer who uses the services to perform a business function.
3	Service Provider	Uses the system to implement an accounting system. Creates resource elements to be monitored and charged. Maintains a contract with a subscriber and consumes analytics and billing reports from the system
4	Third Party Payment Provider	This is an optional entity, with a dedicated responsibility for all rating, charging and billing functions, while the service provider handles only the resource information management and the service provisioning.

The next sections describe the classes and the functionality of the Accounting & Billing (AB) service.

A. *Class Diagram of the AB Service*

The class diagram for the AB service is presented in Figure 3. The functionalities provided by the class objects are grouped and will be exposed to external parties via the ABCoordinator. In detail, the "ABCoordinator" class manages all access to AB-GE methods and processes. It presents the single entry point for external parties trying to consume services in the GE. The communication channel for the GE can be modified on this class only, while it manages interaction with internal classes. This class and its routines are discussed in more detail in the following section. The "UserDetails" it manages user account details. The current user types include Consumer, Subscriber and the Service Provider types, all sub-classes to the Person class. This link between the Person and sub-classes is only used in illustration but not implemented in classes. The data stored in the UserDetails table differentiates the type of user. The "UserAccount" class manages the user's account details and is responsible for creating new accounts and adjusting account balances. The "Services" class manages all service requests and provisioning, using the sub-classes – ServiceRequest and ServiceProvisioning.

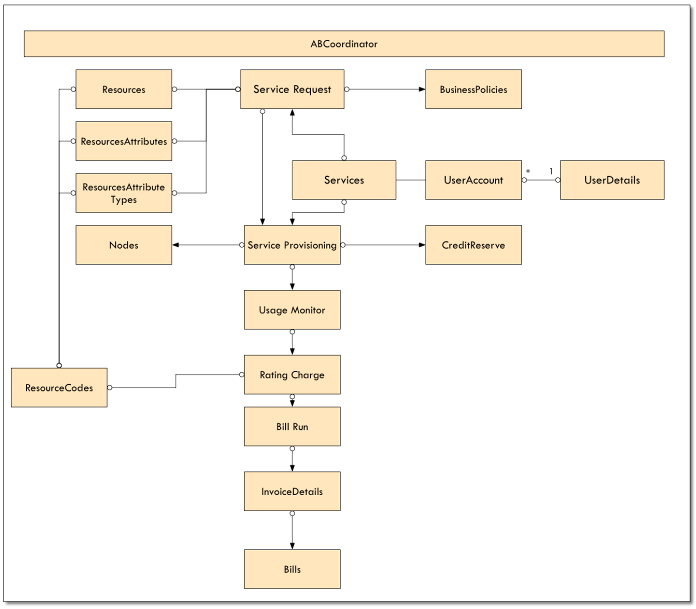

Fig. 3. Class entities and their links constituting the AB service

The "ServiceRequest" class manages all requests for resources, and creates a new record in the ServiceRequests table to represent every request submitted. The "ServiceProvisioning" provides access to all provisioned requests while relying on data already created in the ServiceRequests table. The "Resources" class manages all access to Resources, and provides access to all stored resources. A ServiceRequest object maintains a list of this object. The "ResourceAttributes" class manages the resource attributes, and provides access to the available system configured attributes. A Resource maintains a list of this object. The "ResourceAttribute Type" class manages all type variables for a resource attribute. A ResourceAttribute maintains a list of this object. The "Nodes" class manages all access to and management of nodes in which the GE is running. It provides access to the Nodes table and to methods that obtain the node name and characteristics. The "ResourceCodes" class provides access to the resource codes configured for the combination of resource, resource attribute and resource attribute types, which are eventually utilized in rating. The "BusinessPolicies" class manages access to the system business policies. It also provides access to data on service SLAs. The "UsageMonitor" hosts all methods that provide usage event data management. The "RatingCharge" class applies rating and charging function and generates the invoice records. The "InvoiceDetails" class manages all access and manipulations of the invoice details. The "Bills" class manages all access to bills, including the parent bill and itemized bills.

B. *Functionality of the AB Service*

The service is categorized based on functionalities, managed by the ABCoordinator, which provides a service request coordinator function. The ABCoordinator does this based on a set of functions. Each function aligns with a request and will specify the services that are called in one after the other to achieve the objective of the request. The different calls made less than one request must be maintained as atomic. A break along the processing of the function requires that all actions already executed be rolled back and appropriate feedback provided via the API call. The ABCoordinator's functions are summarized in Table 2.

TABLE II. INTERFACES AND CLASS DESCRIPTIONS.

#	Interface	Class/Description
1	GetAvailable Resources (ResourceSelection)	Resources – returns a JSON of all resources available
2	GetNewRequest (ResourceSelection)	Services – creates a new Request object to hold the details of the request the subscriber is about to create
3	AddResource (ResourceSelection)	Request – adds a new resource to the request, consisting of a Resource object
4	AddResourceAttribute (ResourceSelection)	Resource – adds a new resource attribute to the new resource object's resourceAttributes list.
5	GetTotalCost (ResourceSelection)	Request – to retrieve cost of items selected based on the entry in the Rating table. Thus, it calculates what has been selected against what exists in the Rating table
6	SaveResourceList (ResourceSelection)	Request – saves the in-memory Request object to the Request and RequestDetails tables
7	GenerateServiceRequest (ResourceSelection)	Service – to create a new ServiceRequest object in memory
8	AdjustAccountBalance (AccountTopup)	UserAccount
9	ValidateServiceRequest (Provisioning)	To ensure the requesting account has a active service request already created
10	ReserveCredit (Provisioning)	UserAccount – to reserve credit that would be charged if resources are eventually used
11	GenerateServiceProvisioning (Provisioning)	Called on the Service table to create a record for service usage
12	ValidateServiceProvisioning (ResourceConsumption & Monitoring)	Called on Service to confirm if there is a record for provisioning for this client
13	SaveInterimEvents (ResourceConsumption	Saves interim usage data to the UsageDetails table

	& Monitoring)	
14	ProcessEventFiles (ChargingAndRating)	UsageMonitor
15	AdjustAccountBalanceOnCredit (ChargingAndRating)	Called to adjust the user's account balance in a prepaid payment model

VI. CONCLUSIONS

This work focused on the development of a cloud-based accounting and billing service that could be used to query resources within an OpenStack-based cloud platform. The service retrieves users' resource usage data, enables the assignment of costs to its flavors, and bills users based on their usage logs. In particular, the service includes (a) OpenStack platform user authentication based on Osuth2, (b) in-built user authorization management, (c) dynamic menu generation for consuming channel, (d) administrator registration of new OpenStack platforms. Various APIs developed for the following: (a) configuration of platform flavour costs, (b) retreive user bills, (c) provide resource usage view, (d) provide users' platform's available resources such as servers, images and flavors, (e) provide user-oriented view of platform flavor costs, (f) provide interim usage costs for user based on their resource usage and (g) provide monthly bills based on user's usage. The future work of our research includes experimentation on the messaging optimization among clouds utilizing the framework in [9] and [14]. In addition, other aspects include provision of bill notifications by designing a module to be integrated with an OpenStack-based message context broker to provide automated billing alerts. Finally, context-awareness will increase efficiency of our solution. This could be achieved using a context-aware module to enable auto-configurations for adaptive administration.

REFERENCES

[1] Mihoob, C. Molina-Jimenez, S. Shrivastava, A Case for Consumer-centric Resource Accounting Models. In *IEEE 3rd International Conference on Computing Cloud (CLOUD), 2010*, IEEE Conference Publications, pp 506-512, DOI:10.1109/CLOUD.2010.44

[2] E. Elmroth, F.G. Marquez, D. Henriksson, D. P. Ferrera, Accounting and Billing for Federated Cloud Infrastructures. In *Eight International Conference on Grid and Cooperative Computing, 2009*, IEEE Computer Society, pp 268-275, DOI: 10.1109/GCC.2009.37

[3] TMForum, Business Process Framework (eTOM), 2014 http://www.tmforum.org/BusinessProcessFramework/1647/home.html

[4] TaskMap, Enhanced Telecommunications Operations Map, 2014, http://www.taskmap.com/Enhanced%20Telecom%20Operations%20Map%20-%20Customer%20Interface%20Management.html

[5] TaskMap, Enhanced Telecommunications Operations Map - Service & Specific Instance Rating, 2014, http://www.taskmap.com/Enhanced%20Telecom%20Operations%20Map%20-%20Service%20&%20Specific%20Instance%20Rating.html

[6] TaskMap, Enhanced Telecommunications Operations Map - Bill Payments & Receivables Management, 2014, http://taskmap.com/Enhanced%20Telecom%20Operations%20Map%20-%20Bill%20Payments%20&%20Receivables%20Management.html

[7] FIWARE - Open APIs for Open Minds, 2014, http://www.FIWARE.org/our-vision/

[8] Sotiriadis, S., Bessis, N., and Petrakis, E. (2014). An inter-cloud architecture for future internet infrastructures. In Pop, F. and Potop-Butucaru, M., editors, Adaptive Resource Management and Scheduling for Cloud Computing, Lecture Notes in Computer Science, pages 206-216. Springer International Publishing

[9] Bessis, N., Sotiriadis, S., Pop, F. and Cristea, V. (2013) "Using a Novel Message-Exchanging Optimization (MEO) Model to Reduce Energy Consumption in Distributed Systems", Simulation Modeling Practice and Theory, Elsevier, ISSN: 1569-190X

[10] R.A. Kumar, R.K. Mittal, A User-Centric Billing Model for Cloud Computing, in *International Conference on Cloud Computing, Technologies, Applications and Management (ICCCTAM), 2012*, IEEE Conference Publications, pp 132-138, DOI:10.1109/ICCCTAM.2012.6488086

[11] A. Mihoob, C. Molina-Jimenez, S. Shrivastava, A Case for Consumer-centric Resource Accounting Models, in *IEEE 3rd International Conference on Computing Cloud (CLOUD), 2010*, IEEE Conference Publications, pp 506-512, DOI:10.1109/CLOUD.2010.44

[12] Y. Xiao, C. Lin, Y. Jiang, X. Chu, X Shen, Reputation-Based QoS Provisioning in Cloud Computing via Dirichlet Multinomial Model, in *IEEE International Conference on Communications (ICC), 2010*, IEEE Conference Publications, pp 1-5, DOI: 10.1109/ICC.2010.5502407

[13] R. Barbera, E. Giorgio, S Hwang, G.M. Ricciardi, Gustav: A Lightweight Approach to CPU Accounting in *Distributed Systems, in International Conference P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2010*, IEEE Conference Publications, pp 24-29, DOI: 10.1109/3PGCIC.2010.9

[14] Sotiriadis, S., Bessis, N., Antonopoulos, A. and Anjum, A. (2013) "SimIC: Designing a new Inter-Cloud Simulation Platform for Integrating Large-scale Resource Management", 27th IEEE International Conference on Advanced Information Networking and Applications (AINA-2013), March 25-28, Barcelona, IEEE Computer Society, Washington, DC, USA, pp. 90-97

[15] Sotiriadis, S., Bessis, N. and Antonopoulos, N. (2011) "Towards inter-Cloud Schedulers: A Survey of meta-Scheduling Approaches", 6th IEEE International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC-2011), Barcelona, Spain, October 26-30, 2011, ISBN: 978-0-7695-4531-8, pp. 59-66

[16] M. Iwashita, S. Tanimoto, Y. Fujinoki, Approaches to analyze requirements of billing management for cloud computing services, in *IEEE/ACIS 12th International Conference on Computer and Information Science (ICIS), 2013*, IEEE Conference Publications, pp 17-22, DOI: 10.1109/ICIS.2013.6607810

[17] Kanter T., Pettersson S., Forsstrom S., Kardeby V. Norling R., Walters J., Osterberg P., (2009), Distributed Context Support for Ubiquitous Mobile Awareness Services, *Fourth International Conference on Communications and Networking in China, 2009*. ChinaCOM 2009. Digital Object Identifier: 10.1109/CHINACOM.2009.5339728, 2009 , Page(s): 1- 5

[18] Smith D., (2013), When is a cloud billing system not a cloud billing system- http://www.cerillion.com/Blog/April-2013/When-is-a-cloud-billing-system-not-a-cloud-billing/

[19] Openstack Cloud Software, OpenStack Architecture, 2014, http://docs.openstack.org/icehouse/install-guide/install/apt/content/ch_overview.html

[20] Forbes 2014, Cloud Computing: United States Businesses Will Spend $13 Billion On IT, http://www.forbes.com/sites/tjmccue/2014/01/29/cloud-computing-united-states-businesses-will-spend-13-billion-on-it/

[21] Sotiriadis, S., Bessis, N., Anjum, A., Buyya, R., "An Inter-Cloud Meta-Scheduling (ICMS) Simulation Framework: Architecture and Evaluation," IEEE Transactions on Services Computing, vol.PP, no.99, pp.1,1doi: 10.1109/TSC.2015.2399312

[22] Sotiriadis, S. and Bessis, N., "An inter-cloud bridge system for heterogeneous cloud platforms", Future Generation Computer Systems, Available online 25 February 2015, ISSN 0167-739X, http://dx.doi.org/10.1016/j.future.2015.02.005

[23] Sotiriadis, S., Petrakis, G.M.E., Covaci, S., Zampognaro, P., Georga, E., Thuemmler, C. (2013) "An architecture for designing Future Internet (FI) applications in sensitive domains: Expressing the Software to data paradigm by utilizing hybrid cloud technology", 13th IEEE International Conference on BioInformatics and BioEngineering (BIBE 2013), November 10-13, Chania, Greece