An Inter-Cloud Architecture for Future Internet Infrastructures

STELIOS SOTIRIADIS, Technical University of Crete, Greece
NIK BESSIS, University of Derby, United Kingdom
EURIPIDES G.M. PETRAKIS, Technical University of Crete, Greece

Iaan latest years, the concept of interconnecting clouds to allow common service coordination gained significant attention mainly because of the increasing utilization of cloud resources from Internet users. An efficient common management between different clouds is essential benefit, like boundless elasticity and scalability. Yet, issues related with different standards led to interoperability problems. For this reason, the definition of the open cloud-computing interface defines a set of open community-lead specifications along with a flexible API to build cloud systems. There releases like OpenStack, OpenNebula, Amazon Web Services, and VMWare Cloud for all kinds of cloud models such as Infrastructure, Software and Platform as a Service that expose APIs for inter-cloud communication. In this work we aim to explore an inter-cloud model by creating a new cloud platform service to act as a negotiator among OpenStack, FI-WARE datacenter resource management and Amazon Web Service cloud architectures, therefore to orchestrate communication of various cloud environments. The model is based on the concept of FI-WARE and will be offered as a reusable enabler with an open specification to allow service distribution.

Categories and Subject Descriptors: [**Cloud Computing**]: Clouds, Inter-Clouds, FI-WARE, OpenStack, Amazon Web Services—*Cloud Interoperability and Service distribution*

General Terms: Inter-Clouds

Additional Key Words and Phrases: Clouds, Inter-Clouds, FI-WARE, OpenStack, Amazon Web Services, RESTFul

1. INTRODUCTION

Cloud systems expose interfaces to communicate other clouds or services by forming an inter-cloud. In this setting, users describe their requirements in service level agreements (SLA) that are usually related with infrastructure resources as well as with the relevant services (software) offered from providers. Inter-clouds involve public clouds forming a collaborative environment for distribution and common management of cloud services. This represents the communication glue between the different providers and the different provision layers including Infrastructure, Platform, Network and Software as Services (IaaS, PaaS, NaaS and SaaS). This work vision is on connectivity between resource providers that develop clouds exposing interfaces e.g. following the Open Cloud Computing Interface (OCCI) standard.

Today, the area of inter-clouds has gained particularly interest in academia and industry. Various works like [11] demonstrate solutions and mechanisms to achieve inter-cloud service distribution by exploring various components. In our case the focus is on inter-clouds that emerge from the innovative area of Future Internet (FI) application development of FI-WARE [2] that offers services, called Generic Enablers (GEs). GEs provide essential functionalities, interfaces and APIs for various kinds of functionalities (e.g. authentication, Internet of Things device management, storage, cloud resource management, monitoring etc.). FI-WARE offers a cloud datacentre resource management service (DCRM) in order to control and manage IaaS cloud resources that is based on OpenStack. The inter-cloud service approach will serve as a GE that links various clouds that share characteristics derived from OpenStack API, FI-WARE Datacenter Resource Management GE (DCRM GE) Amazon Web

Services (AWS) [19], OpenNebula [21] and VM Ware cloud (VCloud) [20]. The service will be designed as easily deployable and configurable.

OpenStack is a platform architecture that provides a framework and APIs for cloud systems. It is an open source solutions that is based on open standards of OCCI. Lately, it is used widely (e.g. by FI-WARE, IBM etc.) to allow development of private or public clouds; it is simple to integrate and can be upgraded easily by providing an IaaS for managing datacentre resources [1]. The architecture defines an Inter-cloud as a Service (IC Service) that facilitates development of new IaaS based on the APIs of OpenStack, DCRM GE, AWS, OpenNebula and VCloud. This includes creation of an authentication mechanism to act as intermediate for all clouds. Based on this discussion, Section 2 presents the related works and motivation of this study while Section 3 defines the Inter-cloud Service architecture and services. The rest of the paper is organized as following, in Section 4 we present the proposed model by defining a range of services and the projected operations, in Section 5 we present the experimental prototype infrastructure, in Section 6 we demonstrate the draft inter-cloud collaboration already running clouds based on OpenStack and AWS. Finally in Section 7 we present the conclusions and future research steps.

2. MOTIVATION AND RELATED WORK

Inter-cloud has been characterized as the logical evolution of the Internet in terms of advanced service provision [6]. Today, various cloud vendors aimed to an interoperable cloud effort by jointly establishing federations of clouds. However, these vendor-oriented solutions do not base on future standards and open interfaces but in specific cloud settings as in [12]. A theoretical presentation of inter-cloud has been introduced by [12] from the view of federated perspective. They present a business model of a utility oriented inter-cloud system that includes a centralized coordinator per cloud for service dissemination.

In [11] a discussion is presented to demonstrate a broker that acts as an SLA resource allocator by combining components to achieve the agreed benchmark among users and providers. This is a generic view of brokers that generate challenges on how to manage the most effective resource allocation and scheduling. Here, we focus on OpenStack clouds and the issues arising from inter-cloud communication. Especially, when the number of OpenStack infrastructures, FI-WARE resource providers and their implementations increases, becomes more complex to control the various cloud resources. This includes decrease of the performance of the actual cloud infrastructures due to the utilization of many physical servers. In this study we aim to overcome the problem of vendor specific inter-clouds by focusing on the OCCI standard [4]. This means that cloud systems developers using such standard (FI-WARE, OpenStack, OpenNebula etc.[4]) will be able to utilize their interfaces to join an inter-cloud.

Lately, various FI-PPP programmes [13] (up to 16 EU funded projects) have been promoted to accelerate the development and adoption of Future Internet technologies in Europe, advance the European market for smart infrastructures, and increase the effectiveness of business processes through the Internet. All, base their developments in the FI-WARE cloud platform. Based on this, we develop an inter-cloud service that is deployed in the intellicloud [16] infrastructure of the Technical University of Crete (TUC) and could be offered as a GE service. Intellicloud is an experimental cloud infrastructure for designing cloud-based Internet applications.

3. THE INTER-CLOUD SERVICE ARCHITECTURE

The architecture is based mainly in OpenStack [14] as an open source platform for managing large-scale physical servers in a cloud-computing environment. This includes the transformation of the psychical resources to a virtual ones that could by delivered over the Internet as Virtual Machines (VMs). Using it, resource providers can controls large pools of compute, storage, and networking resources

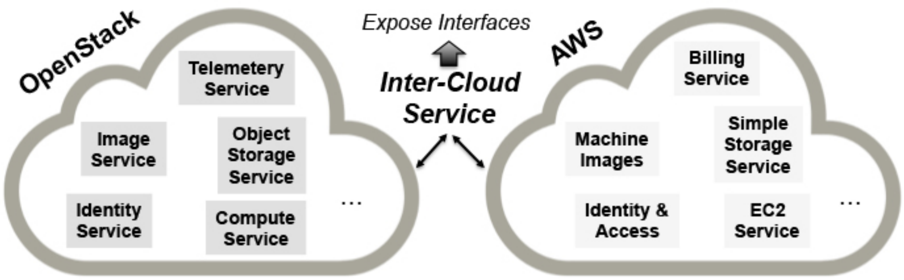

Fig. 1. The OpenStack and AWS architecture of an inter-cloud

throughout their datacentres through interfaces. OpenStack offers also a common API that works with RESTFul protocol and provides remote access to all local services.

The inter-cloud service is designed to be flexible, thus allowing interactions with other cloud that offer AWS, OpenNebula and VCloud API interfaces. These include services for on-demand self-service (users can access needed computing capacities), network access to cloud resources, elasticity to provision based on the user needs and monitoring of services according to the usage. With regards to the service models, OpenStack provides support for SaaS so consumers can deploy software and access it usually as a web-based service, PaaS to deploy applications through a programming language or tools and IaaS for instances, network connection and storage. Figure 1 shows the interactions and internal OpenStack and AWS services among two clouds. The service serves in between and exposes new interfaces for further communication. The key services offered by the Inter-Cloud Service are as follows.

(a) Identity Service: It provides an API as the mean for authentication and authorization by offering a service that generates access tokens for other OpenStack services. This includes a catalog of endpoints for all OpenStack services. Similarly, identical endpoints are defined in order to authorize inter-cloud communication.

(b) Image Service: It provides an API for management of images that are ready and pre-installed VMs that usually include operating systems along with some software configurations. Cloud administrators run this process, but users can also upload images from the inter-cloud service. Images are used to create new instances of services (containers of SaaS and PaaS). In the inter-cloud, images from all clouds are retrieved and shown in the common management space. Also, images could be transformed to more specicalized facilitators of software installations in order to be offered as blueprints (also known as snapshots).

(c) Compute Service: It provides an API for managing the whole lifecycle of instances (that are generated images) in an OpenStack environment. Key responsibilities include spawning, scheduling and decommissioning of VMs on demand [15]. Inter-cloud tenants could generate new instances and store them in their preferred location, this could be offered as an option in case of sensitive data storage as described [7].

(d) Network Service: It provides the network capabilities, e.g. to create new virtual networks and routers for network connectivity. Also, it provides the capability to build private networks of VMs and supports many popular networking vendors [14]. Within the OpenStack it builds the networks using virtual switches (e.g. by using OpenVSwitch [18] that is a multilayer virtual switch licensed under the open source Apache 2.0 license). In the inter-cloud, the network service will offer communication interfaces within the local clouds without exposing internal network topologies (in case of a private network). The level of accessibility will base only on the user privileges.

(e) Object Storage Service: It provides the storage service and retrieves arbitrary unstructured data objects [15]. It is highly fault tolerant with its data replication and scale out architecture. Its implementation is not like a file server with mountable directories as described in [15]. Inter-cloud storage service will be offered via a RESTful [9], HTTP based API.

(f) Database Service: It provides the relational database engine service of each OpenStack implementation of various clouds. OpenStack environments use the relational databases to store tenants and services data [5]. In the case of inter-cloud it will be accessible from super users that have administration right.

(g) Block Storage Service: It provides a storage service for additional storage capabilities as block storage to running instances [3]. The inter-cloud interface will allow to create extra blocks that could be attached to selected VM. This service will work for VMs and storage blocks that are from the same cloud provider.

(h) Orchestration Service: It provides orchestration capabilities for multiple composite cloud applications. It provides a template-driven engine that allows application developers to describe and automate the deployment of infrastructure [5]. This could assist on the deployment of multiple clouds at a later stage.

(i) Telemetery Service: It provides usage and performance data across the services deployed in an OpenStack clouds. In the inter-cloud will be the mean to manage billing, benchmarking, scalability, and statistical purposes. Also, the metering service will offer performance measures for messaging (for instance to oprimize RabbitMQ queues).

(j) User Interface Service: It provides the user interfaces that are web-based self-service portal to interact with underlying OpenStack services. Inter-cloud will use OpenStack horizon service.

4. THE INTER-CLOUD MODEL

This section demonstrates the model to connect inter-cloud IaaS environments that are geographically dispersed based on the OpenStack architecture of section 3. By using a common agreed standard in communication it solves issues reagarding interoperability among such systems. Figure 2 shows the inter-cloud services along with their key operations. The inter-cloud registry keeps a list of OpenStack architecture URLs that are used by the inter-cloud services. Also, OpenStack RESTFul APIs and schemes along with the advanced messaging queuing models (AMQ [17]) define explicitly a modular set of components and standard rules for connecting OpenStack infrastructure. This happens in a highly secure environment where access and resource utilization is controlled in many levels (users, roles, and projects) [5].

The model of the afforementioned topology provides services to facilitate inter-cloud communication in a common platform. The inter-cloud service could list all available services, instances, offer network capabilities, security and deployment of IaaS services. The model includes the following key operations. Firstly, the inter-cloud performs authentication for the specific tenant that uses it (at this stage the users require to have credentials to all clouds of the collaboration), this is the source of certification and serves as a RESTFul deployed service. Secondly, the model proposes a new platform service as a RESTFul Inter-Cloud as a Service (ICaaS) to interfacing to other clouds. The initial plan is to integrate OpenStack, DCRM GE and AWS clouds, yet the model will be expandable to OpenNebula and VClouds. The ICaaS is a platform service that offers the following capabilities:

(i) Offers a registry that contains the configuration (e.g. URL addresses of other OpenStack, AWS and FI-WARE DCRM infrastructures).

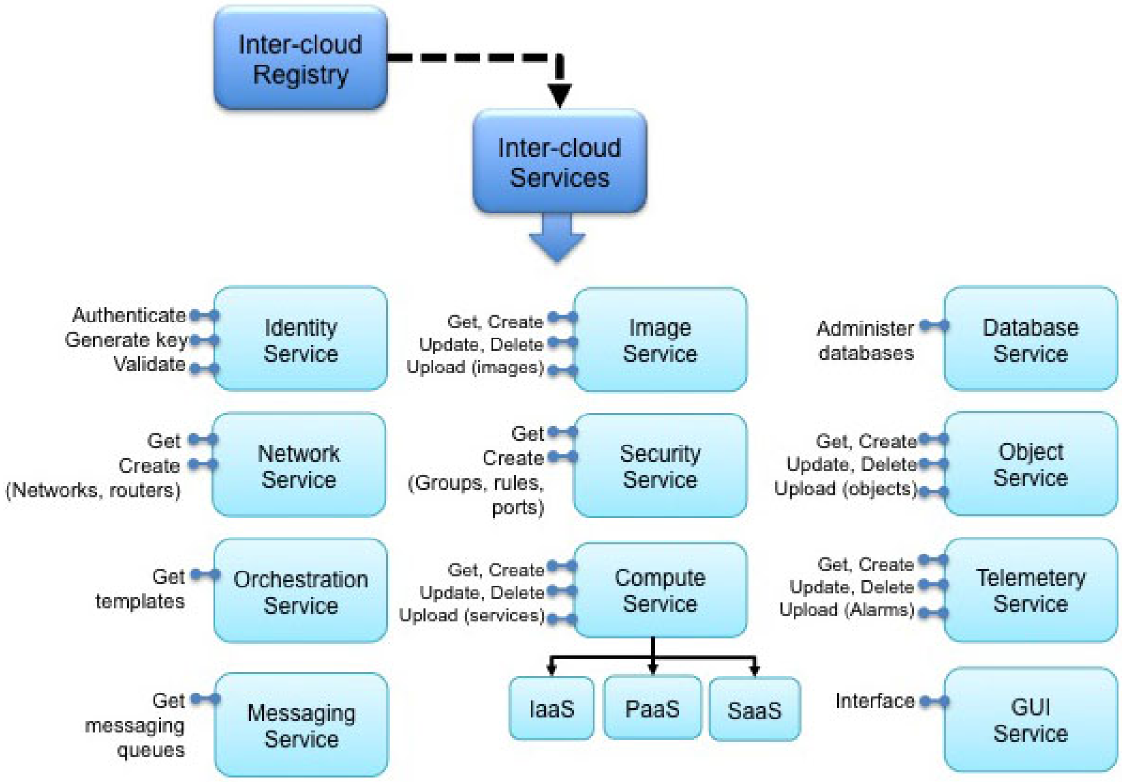

Fig. 2. The OpenStack model of an inter-cloud with available services and their key operations

(ii) Offers SaaS services that extent the OpenStack API to offer the next operations:

(a) ICaaS identity service: It is responsible to Authenticate, Generate, Validate. OpenStack tenants use it to revoke a token for access.
(b) ICaaS image service: To get, create, update, delete, upload an image of a cloud. The service returns a list of instances, while the post action allows information from the inter-cloud to be forwarded to each respectively (e.g. in case that a user wants to update an image description.).
(c) ICaaS database service: Offers an administrator service to Create a user (with credentials defined in the ICaaS) as well as to create flavors (variation of possible SLAs) to other clouds. This will be the mean to achieve new user creation in all clouds from the inter-cloud platform.
(d) ICaaS network service: Offers the option to get, create networks and routers. Usually, in case of a new user generation in the inter-cloud platform the administrators generate the networks, yet this could happen by them using the inter-cloud platform service.
(e) ICaaS security service: To get, create security groups, rules, ports, release floating IPs, associate floating IPs to linkes clouds. These are essential in order to offer the IaaS with all available options.
(f) ICaaS object service: To get, create, update, show, and delete account metadata and objects using the OpenStack service.
(g) ICaaS orchestration service: Offer a template-driven engine that allows application developers to describe and automate the deployment of infrastructure.

(h) ICaaS compute service: To get, create, update, show, and delete services (offered as IaaS VMs) in other clouds, this includes three kinds of services:
   i. The IaaS VM provisioning that relates with IaaS services. E.g. the ICaaS platform creates a VM instance in an inter-cloud using the available data (flavours, security, networks etc.).
   ii. The PaaS provisioning that relates with PaaS services hosted in a cloud and listed in the inter-cloud. E.g. the platform create a VM instance (blueprint or snapshot of already deployed operational environment) ready to be utilized by other. In this case developers could configure a platform (e.g. Eclipse) to work directly on the platform.
   iii. The SaaS provisioning that relates with already deployed instances that are listed in the platform,. These are deployed locally and are available to the platform users as VM instances e.g. accessible by their IPs.

(i) ICaaS messaging service: Operates with AMPQ servers of clouds in order to manage and optimize messaging queues. We use RabbitMQ that is a robust messaging mechanism for OpenStack services that is offered as open source. For inter-clouds will enable connectivity (asynchronous and decoupled) as it offers a common platform to send and receive messages.

(j) ICaaS telemetery service: Offers capabilities to list, create, gets details for, update, and delete alarms and meters.
   i. The ICaaS integrates a messaging model to optimize the interactions of the RESTFul components and their calls. This is a usually problem when scaling OpenStack with many servers, the systems tend to decrease performance as many calls are forwarded to the database from many callers thus leading to bottleneck.
   ii. The ICaaS integrates a performance metering service, the service collects internal cloud performance measures and allows inter-cloud platform administrators to define thresholds and performance parameters for monitoring purposes.

(k) ICaaS graphical user interface service: Offers the user friendly environment that combines aforementioned services in a web based interface to manage access and provision of services. The inter-cloud will use the horizon service will provide a portal for the inter-cloud in order to allow management of the VMs, floating IPs, security groups and public keys.

The modularity of the system is high, this means that the ICaaS services will be developed based on OpenStack API, will be hosted and deployed separately and will be available as open source instances of the intellicloud infrastructure of TUC [16]. Eventually, the services will be integrated into a platform that will offer the user friendly interfaces.

5. EXPERIMENTAL PROTOTYPE OF INTER-CLOUD

The experimental prototype is integrated using RESTFul API and implements some of the aforementioned services. It uses cURL [10] and allows the inter-cloud to transmit and receive HTTP requests and responses. The prototype offers a direct interaction with the various components provided by the OpenStack API, DCRM GE API and AWS API. The experiment is based on a single type of request (Identity Service of Section 4), that is executed on real-time, the so-called ”authentication, to get token from each cloud” to be used for further authentication of services among the following clouds.

(a) Intellicloud, architecture (OpenStack) of TUC (Crete, Greece)
(b) FI-LAB, architecture DCRM GE of FI-WARE (Sevilla, Spain)
(c) CloudLab: Experimental Cloud OpenStack of TUC (Crete, Greece)
(d) Amazon AWS Cloud (Oregon US)
(e) VMWare VCloud (UK)

Figure 3 demonstrates the prototype inter-cloud as a Service solution and the associated cloud environments. It should be mentioned that due to size limits the VCloud is not included in the analysis.

Fig. 3. The OpenStack model of an inter-cloud with available services and their key operations

The ICaaS exposes interfaces to internal procedures for further communication. It uses real time metrics to provide results. Especially, the ”real” metric is the wall clock time (the time needed from start to finish of the call), the ”user is the amount of CPU time spent in user-mode code (outside the kernel) within the process, the ”sys” is the amount of CPU time spent (inside the kernel within the process) and the ”factor” is the actual performance of the metric to the compared value (division of worst by best performance value). To demonstrate effectiveness we present the following 4 experimental studies (where for each of which we execute 10 requests namely as Req1 to Req10) as follows:

(Exp i) Demonstrates metric values of the ICaaS when is executed within the cloud (to be used as a benchmark)

(Exp ii) Demonstrates metric values of the ICaaS when is executed for two cloud systems

(Exp iii) Demonstrates metric values of the ICaaS when is executed for three cloud systems

(Exp iv) Demonstrates metric values of the ICaaS when is executed for four cloud systems

1st Experiment: Internal Calls

This demonstrates calls made from the ICaaS to the CloudLab system and FI-LAB infrastructures to collect measures of real, user and sys metrics that could be used as benchmarks. The results are provided in order to characterize a) the real-time responses of a cloud for calls that made internally (within the system, the calls are made from and to the CloudLab services) and b) the real-time responses of a cloud for calls that made externally (outside the system). The calls are made from CloudLab to the FI-LAB DCRM GE. Based on the comparison we extrapolate a fctor as generalized metrics. The factor for calls made within and outside is 54% (int/ext%). This means that case b) achieves 54% performance of the a) case. The results demonstrate realistic high performance (average 0.39 seconds with highest 0.435) with regards to the real-time calls; the average time of calls to the exposed cloud interfaces remain under 0.5 seconds.

2nd Experiment: External Call to CloudLab and FI-LAB cloud

This demonstrates the ICaaS calls that made from the service to CloudLab and FI-LAB infrastructures (the point of calls) for authentication at both endpoints. In this case, we compare the responses for the case of FI-LAB authentication (the benchmark of 1st experiment) in contradiction of the ICaaS. The new factor is 42%, yet the actual performance remains at realistic high levels as most of the calls (9 out of 10) have been completed less than 1 second. This could be considered as a fast response by taking into consideration the physical locations of datacenters (Greece and Spain) and the real-time execution of the requests. Figure 4 demonstrates the trend-lines, ICaaS shows a decreasing tendency for the real time metric as more requests are executed.

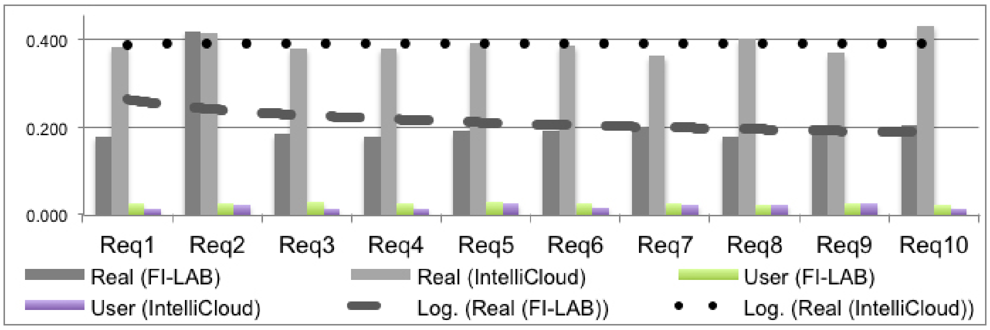

Fig. 4. Comparison of calls among IC and FI-LAB external

3rd Experiment: External Call to FI-LAB to CloudLab, FI-LAB and Intellicloud

We present an inter-cloud mechanism to compare performance of the same calls between: a) CloudLab and FI-LAB (as executed in Experiment 2) and b) CloudLab, FI-LAB and Intellicloud (three clouds request) Figure 5 shows that the fluctuation of the ICaaS for 2 and 3 clouds is at a value of 0.256 seconds. Also the real time increases slightly over the 1.2 sec. The point of calls is the FI-LAB.

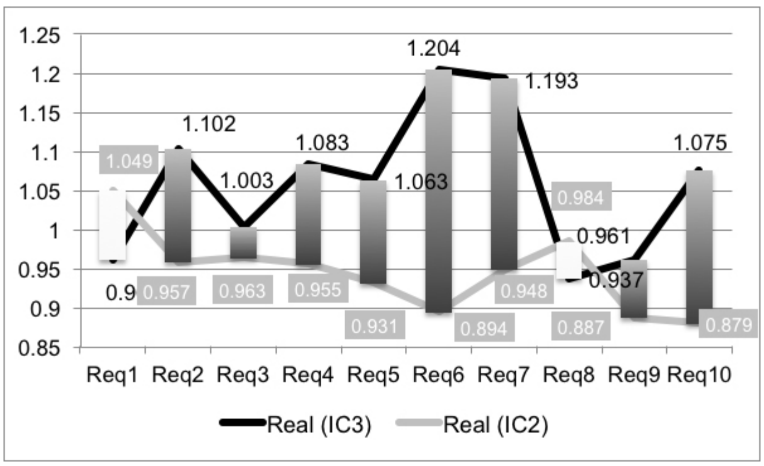

Fig. 5. Fluctuation of the IC service for 2 and 3 clouds

In this case the real time increases slightly over the 1 sec and again this could be considered as high performance measure for real-time responses. Compared to experiment 2, the factors in such case are in very high levels, as it achieves the 89% of the performance of experiment 2 (real time metric).

4th Experiment: External Call to CloudLab, FI-LAB, Intellicloud and AWS

The final experiment demonstrates an ICaaS for connection with four clouds. The inter-cloud sequence of calls include authentication in Intellicloud, CloudLab, FI-LAB, and Amazon AWS, for datacenters located in Greece, Spain and US. The factors for comparison of the four clouds are demonstrated bellow. The calls are made form an ICaaS service executed in FI-LAB. Table demonstrates the factors among the experiments.

	Factors 4 to 3 clouds	Factors 4 to 2	Factors 4 to 1
Real	53%	47%	20%
User	83%	75%	57%

In experiment 4, calls take less than 2 seconds to be completed, this means that we have an increase of averagely 0.8 seconds compared to experiment 3. This is due to the communication time between the ICaaS and the different regions of datacenters around the world. The diagram shows that the real time has been almost doubled; yet this remains slightly averagely under the 2 seconds, to our view a highly acceptable value by considering that the numbers of interactions have been increased along with the network bandwidth.

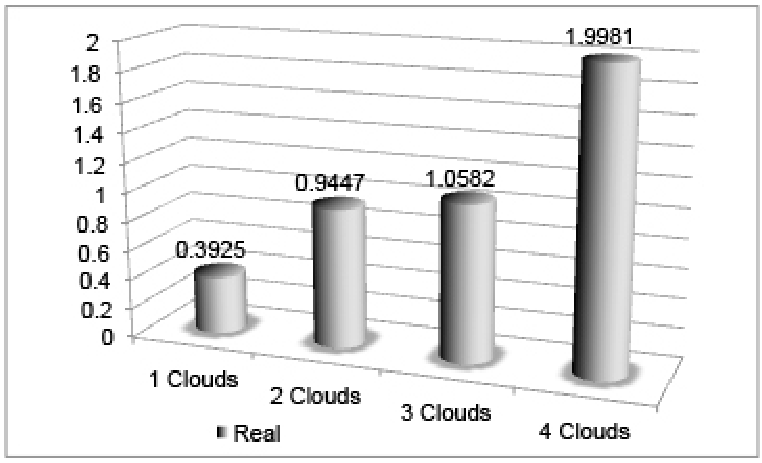

Fig. 6. Real times for combination of clouds (1-4 clouds)

6. CONCLUSIONS AND FURTHER WORK

This work presented an inter-cloud service architecture by utilizing OpenStack, FI-WARE DCRM GE and AWS APIs. The current version is draft and supports basic functionalities in order to integrate an inter-cloud GE that is deployed into a cloud. The proposed model integrates services to facilitate inter-cloud communication in a common platform in order to list all available services, instances, offer network capabilities, security and deployment of IaaS services. The experimental prototype demonstrates the basic configurations in order to develop an inter-cloud service. Since this is an ongoing effort more information could be found [16].

In future, we focus on the development of the proposed model in order to include all services and components, and to develop a graphical user interface to provide a common management platform that will be offered as a service instance. In addition, extra effort will be made to the characterization of performance metrics (e.g. by collecting the performance metrics of the servers) in order to provide optimization algorithms. This includes RESTFul APIs and schemes along with the AMPQ models. In particular, we aim to explore message models and flows of events in order to optimize performance metrics. Finally, we will explore OpenNebula, AWS and VCloud along with CloudStack RESTFul interfaces and APIs in order to expand the inter-cloud services.

ACKNOWLEDGEMENT

This work is part of the Future Internet Social Technological Alignment Research (FI-STAR) project, which is a Future Internet Private Public Partnership (FI-PPP) run by the European Commission. FI- STAR commenced on 1st April 2013 and will conduct early clinical and non-clinical digital-health use-case trials in European countries.

REFERENCES

[1] OpenStack Cloud Software, Available at: http://docs.openstack.org/api/openstack-image-service/2.0/content/upload-binary-image-data.html

[2] FI-WARE, Available at: http://www.fi-ware.org

[3] OpenStack API, Available at: http://docs.openstack.org/api/openstack-image-service/2.0/content/image-api-v2.0.html

[4] Open Cloud Computing Interface, Available at: http://occi-wg.org

[5] Jackson, K. and Cody Bunch, C. 2013. Openstack Cloud Computing Cookbook (2nd ed.). Packt Publishing.

[6] Sotiriadis, S., Bessis, N. Kuonen, P. and Antonopoulos, N. 2013. The Inter-cloud Meta-scheduling (ICMS) Framework. In Proceedings of the 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA '13). IEEE Computer Society, Washington, DC, USA, 64-73.

[7] 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

[8] Service API Examples Using Curl, Available at: http://docs.openstack.org/developer/keystone/api_curl_examples.html

[9] Richardson, L. and Ruby, S. 2007. Restful Web Services (First ed.). O'Reilly.

[10] Ward, S. and Hostetter, M. 2003. Curl: a language for web content. Int. J. Web Eng. Technol. 1, 1 (August 2003), 41-62.

[11] Petcu, D. 2013. Multi-Cloud: expectations and current approaches. In Proceedings of the 2013 international workshop on Multi-cloud applications and federated clouds (MultiCloud '13). ACM, New York, NY, USA, 1-6.

[12] Buyya, B., Ranjan, R. and Calheiro, N. R. 2010. InterCloud: utility-oriented federation of cloud computing environments for scaling of application services. In Proceedings of the 10th international conference on Algorithms and Architectures for Parallel Processing - Volume Part I (ICA3PP'10), Ching-Hsien Hsu, Laurence T. Yang, Jong Hyuk Park, and Sang-Soo Yeo (Eds.), Vol. Part I. Springer-Verlag, Berlin, Heidelberg, 13-31.

[13] Future Internet Public-Private Partnership (FI-PPP), Available at: https://www.fi-ppp.eu

[14] OpenStack Architecture, Available at: http://docs.openstack.org/training-guides/content/module001-ch004-openstack-architecture.html

[15] OpenStack Installation Guide, Available at: http://docs.openstack.org/trunk/install-guide/install/yum/content/ch_overview.html

[16] Intellicloud, Intelligence Systems Laboratory, Technical University of Crete, Greece, Available at: http://www.intelligence.tuc.gr

[17] Advanced Message Queuing Protocol, Available at: http://www.amqp.org

[18] Open vSwitch, Available at: http://openvswitch.org

[19] Amazon Web Services (AWS) API http://aws.amazon.com/documentation/

[20] VMWare VCloud API, http://pubs.vmware.com

[21] OpenNebula AIP, http://docs.opennebula.org/4.6/