Interoperability in the Cloud

Need for Interoperability

Cloud interoperability refers to the ability for diverse cloud environments to work together in a common way. In order to truly realize the benefits offered by cloud computing, organizations must be able to communicate and collaborate across various cloud providers. Interoperability between different cloud distributions (open source and commercial) helps the developers in developing platform independent applications that work seamlessly on any cloud platform without any major modifications. The other major reason for interoperability is to enable federation of trusted private cloud environments. Federating trusted private cloud environments could increase the total available resource pool for each participant in a significant way. Federation could be within an organization (multi-site organization) or between different trusted organizations wishing to work together for a common cause [1].

Essential Requirements for Cloud to Cloud Communication

The following is a high level outline of some of the most important requirements for two or more private cloud systems to communicate and share resources in a secure and efficient way.

- Resource naming, discovery and access.
    - Resources could be virtual machines, storage space, jobs, data etc.
- Distributed network management.
    - Virtual network must be created between the federated private cloud systems.
- Security, auditing and monitoring [2].

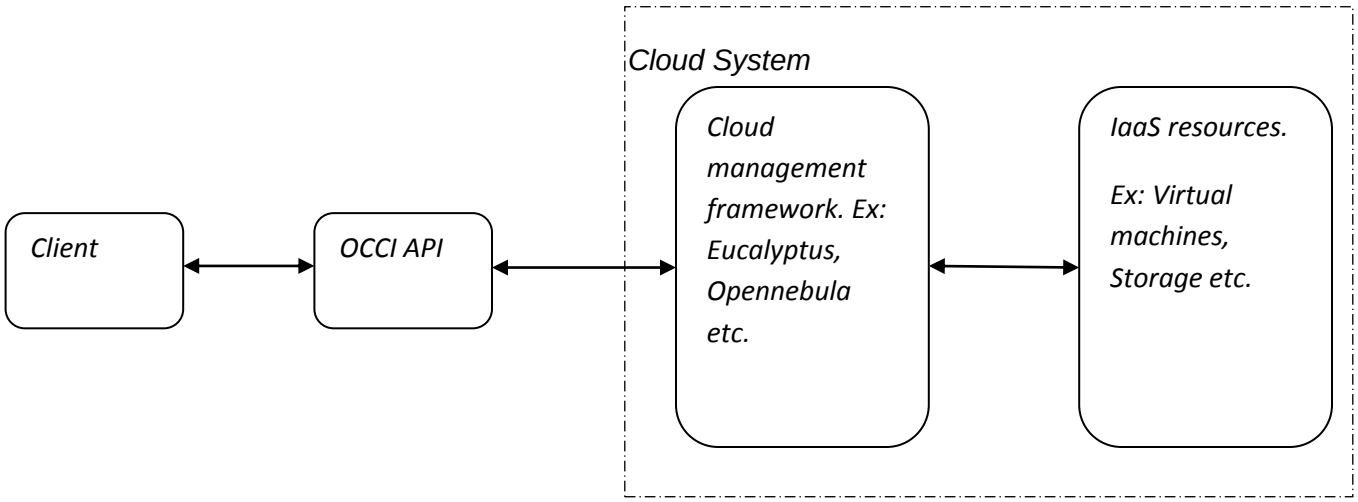
Current Interoperability Efforts

Interoperability issue in the cloud is currently being addressed from two different approaches. One approach is through unifying the interfaces of diverse cloud environments facilitating different cloud environments to collaborate with one another. The other approach is for simplifying the development and deployment of applications by devising a vendor independent way of accessing the resources. The following organizations/projects are involved in cloud interoperability.

- OCCI

  Open Cloud Compute Interface is a protocol and REST based API for remote management of IaaS resources. The API helps in building interoperable tools for tasks like deployment, autonomic scaling, monitoring etc. OCCI acts as a service

frontend to a cloud system's management framework. The following figure shows a high level view of a cloud system and the role of OCCI.



The resources are addressed using a *uniform resource identifier* (URI). The framework also supports linkage of resources and a set of actions on the resources. The resources are managed by basic operations like create, retrieve, update and delete. Currently three types of resources are considered in the specification; *storage, network* and *compute* resources. Each resource also has a set of attributes in order to provide additional details about the resources. The following is a list of attributes defined for the resource types.

a. <u>Compute resource</u>

Compute resources are used for processing information. Typically these resources denote computational resources in terms CPU or virtual machine instances in a cloud setup.

The specification provides a set of attributes for the compute resource type.

1. occi.compute.architecture – CPU architecture of the instance.
2. occi.compute.core – Number of cores assigned to the instance.
3. occi.compute.hostname – DNS hostname for the instance.
4. occi.compute.speed – CPU clock frequency.
5. occi.compute.memory – Maximum RAM allocated to the instance.
6. occi.compute.state – Current status of instance (possible values are *active*, *inactive*, *suspended*) [3].

b. Storage resource
   Storage resource type is used to denote disk storage resources in a cloud environment.
   The specification provides a set of attributes for storage resource type.
   1. occi.storage.size – Storage size of instance.
   2. occi.storage.state – Current status of instance (possible values are *online, offline, degraded*) [3].

c. Network resource
   Network resource type provides details about the networking in a cloud system.
   The specification provides a set of attributes for network resource type.
   1. occi.network.vlan – VLAN identifier.
   2. occi.network.label.
   3. occi.network.address – IPv4 or IPv6 address.
   4. occi.network.gateway – Gateway IP address.
   5. occi.network.allocation – IP address allocation mechanism.
   6. occi.network.state – Current status of instance (possible values are *active, inactive*) [3].

- Jclouds/Deltacloud/Libcloud
  In order to provide a vendor independent way of handling resources of a cloud provider, jclouds, libcloud and deltacloud provide open source client libraries for interacting with different cloud environments. These open source projects support majority of the cloud service providers. The supported cloud service providers are Amazon ec2, Eucalyptus, Gogrid, Opennebula, Rackspace, Microsoft azure, vCloud etc. Deltacloud provides a REST based API for remotely accessing and performing basic operations on resources provided by the cloud service provider.

- Unified Cloud Interface
  Unified cloud interface project is an attempt to build an interface that unifies various cloud API's. UCI uses a resource description framework to describe data and resources in the cloud. This project has not been that active off late.

Future Roadmap

As described in the document, interoperability issue could be addressed by providing a vendor independent framework compatible and interoperable with all the major cloud service providers. The other aspect of interoperability could be to explore the scenario where a private cloud may share resources like computational power, data or storage capabilities building a virtual cloud over remotely located cloud systems. Currently,

opennebula project has included an implementation of OCCI in its current release. Eucalyptus and open stack distributions could also have an OCCI implementation in their future releases. Evaluation of OCCI functionalities in opennebula could be a good starting point in building an API for interoperability. The next step could be to remotely access opennebula resources from a different private cloud environment like eucalyptus.

References

[1] Guilherme Sperb Machado, David Hausheer, Burkhard Stiller: Considerations on the Interoperability of and between Cloud Computing Standards.

[2] Standardized Cloud: Available at: http://www.elasticvapor.com/2008/08/standardized-cloud.html

[3] Open Cloud Computing Interface – Infrastructure, Available at: http://forge.ogf.org/sf/wiki/do/viewPage/projects.occi-wg/wiki/Infrastructure