Document Identifier: DSP2042

Date: 2015-02-26

Version: 1.0.0

Cloud Infrastructure Management Interface Use Cases

Supersedes: None

Document Type: DMTF Informational

Document Class: Informative

Document Status: Published

Document Language: en-US

Contents

Figures

Foreword

This document contains a set of use cases that are candidates to be addressed by the next major functional revision of the CIMI (Cloud Infrastructure Management Interface) specification.

This document has been developed as a result of joint work with many individuals and teams, including:

Enrico Ronco	Telecom Italia (Editor)
Eric Wells	Hitachi Ltd.

Contributors:

Winston Bumpus	VMware Inc.
Mark Carlson	DMTF Fellow
Jacques Durand	Fujitsu
Robert Freund	Hitachi
Ali Ghazanfar	ZTE Corporation
Jie Hu	ZTE Corporation
Iwasa Kazunori	Fujitsu
Dies Koper	Fujitsu
Larry Lamers	VMware Inc.
John Leung	Intel Corporation
Arturo Martin de Nicolas	Ericsson
Ryuichi Ogawa	NEC
Shishir Pardikar	Citrix Systems Inc.
John Parkem	DMTF Fellow
Federico Rossini	Telecom Italia
Alan Sill	Open Grid Forum
Marvin Waschke	DMTF Fellow
Martin Wiggers	Fujitsu
Daniel Wilson	Ericsson

Cloud Infrastructure Management Interface (CIMI) Use Cases

1 Introduction

The Cloud Management Working Group (CMWG) has started the process of developing a new version of the Cloud Infrastructure Management Interface (CIMI) specification to address the next generation of issues facing Infrastructure as a Service (IaaS) providers. To this goal, CMWG members have prepared a number of use cases that typify these issues and that the CMWG has agreed should be addressed. This document collects those use cases for publication to inform both DMTF members and the industry at large and to solicit feedback on the functions the CIMI specification should provide.

1.1 Document structure

To ease readability, the use cases have been grouped into broad categories as follows:

- Business Continuity/Disaster Recovery (BC/DR group)
  Enhanced provisioning of IaaS elements (Machines, Volumes, Networks, etc.)
- Service Level Objective Management (SLO Group)
  Provisioning to maintain agreed service levels
- Log / Metadata Management (Log/Met Group)
  Efficient management of event and reporting data throughout the cloud environment
- Multicloud Management (Multicloud Group)
  Including federation, brokering, and intercloud scenarios
- Open Virtualization Format Management (OVF Group)
  Package lifecycle management, aligned with the DMTF OVF Standard
- Resource Group Management (Res-MGM/Ctrl Group)
  Enabling providers to manage pooled resources

A use case may belong to more than one category, but each is described only once under the "major" group to which it belongs.

Use cases are described using a common template, which includes the following sections:

- Description
  A brief textual description of the use case
- CIMI rationale
  Justification, in terms of industry needs, as to why the use case should be supported
- Dependencies
  Interdependencies with other use cases, standards and technologies
- CIMI challenges
  Areas where the existing CIMI specification needs enhancement or modification
- Business Actor(s)
  The various parties involved in implementing the use case
- Process flow
  The sequence of operations that Business Actors perform to implement the use case

- Variations
  Additional or alternative use cases that are similar to the one described
- Detailed description
  Further explanatory and technical details of the use case

1.2 Disclaimer

As more IT functions are developed using IaaS and cloud adoption increases further, the CMWG envisions issues occurring that currently present few challenges. The objective of these use cases is to highlight these future needs of cloud providers and consumers and not just to address current issues.

Consequently, the CMWG reserves the right to develop the CIMI specification in the manner it sees as best fulfilling industry needs. Therefore these uses cases may not actually be supported by the CIMI specification, or may be supported in a manner that differs from that described in this document. The CIMI specification may also support use cases that are not described in this document, but which the CMWG has determined need to be addressed.

The CMWG encourages any and all feedback on these use cases and any others that the reader feels should be supported by the CIMI specification. Non-DMTF members can provide feedback via the DMTF website: http://dmtf.org/contact

2 References

The following documents provide additional background information that the reader should find helpful in understanding these use cases.

DMTF DSP0243, *Open Virtualization Format Specification 2.1.0*
http://www.dmtf.org/sites/default/files/standards/documents/DSP0243_2.1.0.pdf

DMTF DSP0262, *Cloud Audit Data Federation (CADF) -Data Format and Interface Definitions Specification version 1.0.0*,
http://dmtf.org/sites/default/files/standards/documents/DSP0262_1.0.0.pdf

DMTF DSP0263, *Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol 1.1.0*
http://www.dmtf.org/sites/default/files/standards/documents/DSP0263_1.1.0.pdf

DMTF DSP2017, *Open Virtualization Format White Paper 2.0.0*
http://www.dmtf.org/sites/default/files/standards/documents/DSP2017_2.0.0.pdf

DMTF DSP2027, *Cloud Infrastructure Management Interface (CIMI) Primer 1.1.0*
http://www.dmtf.org/sites/default/files/standards/documents/DSP2027_1.1.0.pdf

DMTF DSPIS0101, *Interoperable Clouds 1.0.0*
http://www.dmtf.org/sites/default/files/standards/documents/DSP-IS0101_1.0.0.pdf

DMTF DSPIS0102, *Architecture for Managing Clouds1.0.0*
http://www.dmtf.org/sites/default/files/standards/documents/DSP-IS0102_1.0.0.pdf

DMTF DSPIS0103, *Use Cases and Interactions for Managing Clouds 1.0.0*
http://www.dmtf.org/sites/default/files/standards/documents/DSP-IS0103_1.0.0.pdf

NIST Special Publication 800-145, Peter Mell and Timothy Grance, *The NIST Definition of Cloud Computing*, Sept. 2011
http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

NIST Special Publication 500-292, Fang Liu, Jin Tong, Jian Mao, Robert Bohn, John Messina, Lee Badger and Dawn Leaf, *NIST Cloud Computing Reference Architecture*, Sept. 2011
http://collaborate.nist.gov/twiki-cloud-computing/pub/CloudComputing/ReferenceArchitectureTaxonomy/NIST_SP_500-292_-_090611.pdf

The reader is also encouraged to review the latest Work-In-Progress documents for the related DMTF standards available at: http://www.dmtf.org/standards/cloud

3 Terms and definitions

Some terms in this document have a specific meaning beyond their normal English meaning. The majority of those terms are defined by the CIMI specification, which should be referred to for exact definitions. However, a number of terms are given informal definitions below to aid the reader:

3.1

Business Actors

The various logical parties involved in a use case, synonymous with the actors defined in the NIST Reference Architecture [SP500-292].

3.2

Cloud

Synonymous with "cloud computing" as defined in section 2 of the NIST Definition of Cloud Computing [SP800-145].

3.3

Cloud Entry Point, CEP

The top-level representation of the cloud service defined by the CIMI model. The CEP implements a catalog of Resources that can be browsed and queried by the Cloud Service Consumer. [DSP0263]

3.4

Cloud Brokering, Cloud Federation

Processes by which services from two or more Cloud Providers are aggregated and presented to a Cloud Consumer as a single service. A Cloud Broker may be one of the Business Actors involved in a use case.

3.5

Cloud Service Consumer, Cloud Consumer, Consumer

Actors that receive services from a Cloud Service Provider. This group includes both consumer administrators and the end users of the service. A Cloud Service Consumer is equivalent to the "Cloud Consumer" actor defined in the NIST Reference Architecture [SP500-292].

3.6

Cloud Service Provider, Cloud Provider, Provider

Actors that provide cloud services to Cloud Service Consumers. This group includes various managerial and operational roles within the Cloud Providers organization. A Cloud Service Provider is equivalent to the "Cloud Provider" actor defined in the NIST Reference Architecture [SP500-292].

3.7

Disk, Machine, Network, System, Volume, etc.

When capitalized these terms refer to the specific Resource of the same name, as defined by the CIMI specification [DSP0263].

3.8

Infrastructure as a Service (IaaS)

A cloud computing service model defined in section 2 of the NIST Definition of Cloud Computing [SP800-145].

3.9

Resource

A representation of an entity managed by the Cloud Service Provider that the Cloud Service Consumer can access or operate using the CIMI specification [DSP0263].

3.10

Template

A CIMI Resource that represents the set of metadata and instructions used to instantiate some other Resource (e.g., a `MachineTemplate` is used to create a `Machine`) [DSP0263].

4 CIMI candidate use cases

4.1 Business Continuity/Disaster Recovery management use cases

4.1.1 Realizing Business Continuity on a `Machine`

(Either in active-passive or in active-active mode)

BC/DR-001	**Realizing Business Continuity on a `Machine`** **(either in active-passive or in active-active mode)**
Description	A consumer creates a machine for which Business Continuity must be provided. Two scenarios are supported and can be chosen by the Consumer: active-passive cluster or active-active cluster.
CIMI rationale	Business Continuity is becoming ever more important as enterprises move their data centers to the cloud. Enhancing CIMI to encompass such functionality can be an enabler to adoption and a differentiating factor for providers. Providing a cluster of machines for Business Continuity implies an interaction between the software layer (application and operating system) and the underlying layers (hypervisor and firmware); the software layer is managed by the machine user, the underlying layers by the machine administrator. In an “in house” scenario, the setup of the Business Continuity functionality for a given application is easily achievable because all the actors can readily communicate. In a cloud environment, CIMI can be viewed as the correct medium to enable this interaction.
Dependencies with other use cases, standards, and technologies	A Disaster Recovery use case is described in clause 4.1.2 of this document. Possible relations with service level objective (SLO) theme are in clause 4.2 of this document. Specific middleware/vendor platforms have their own solutions to ensure Business Continuity.
CIMI challenges	CIMI will need to define a cluster entity model; as a first working hypothesis, a set of attributes that qualify the Machine and its configurations could be sufficient. It is necessary to determine and formalize the replication of the Machine status to its backup node and to define a protocol for the replication. Analysis should be performed to identify whether the CIMI information model should also formalize the policies that determine failover.
Business Actor(s)	Cloud Service Consumer, Cloud Service Provider(s)

BC/DR-001	**Realizing Business Continuity on a `Machine`** **(either in active-passive or in active-active mode)**	
Process flow	Step Description	Data Required
	1: The Consumer wants to create a `Machine` in Business Continuity mode and chooses a configuration from those offered by the Provider, selecting either an active-passive or an active-active cluster configuration.	
	2a: For active-passive scenario: The Consumer sends the POST command by passing the chosen configuration. 2b: For active-active scenario: step 2a + the Consumer and also passes the number of nodes as inline parameters.	
	3: The Provider creates the selected cluster (either active-passive or active-active). (Specifically, the Provider creates a running `Machine`, connects it to a load balancer, and configures the virtualization layer to monitor the status of the running `Machine`).	
	4: The Consumer configures the `Machine` by installing applications, etc.	
	5: When the configuration is complete, the Consumer communicates the status to the Provider and indicates that the `Machine` should be placed in a cluster (e.g., executes a command that transfers the `Machine` status to the backup `Machine`).	
	6: Sometime during execution the `Machine` might stop running (e.g., a request sent from the load balancer to the `Machine` doesn't receive a response).	
	7a: For active-passive scenario: The Provider immediately instantiates and executes the backup `Machine`. 7b: For active-active scenario: The Provider immediately switches every incoming request to the backup `Machine`.	
	8: For active-passive scenarios only: When the backup `Machine` completes the startup process, the load balancer switches requests to the backup `Machine` and "normal" operations resume.	
Variations	The same use case can be applied to "Systems" resources created by the Consumer.	

BC/DR-001	**Realizing Business Continuity on a `Machine`** **(either in active-passive or in active-active mode)**
Notes	Note 1: The Business Continuity functionality provided by this use case has to be built in collaboration with the Consumer: essentially the Provider makes available Machine configurations that allow Consumers to build their own Business Continuity features. Thus CIMI is needed to facilitate the interactions between the Consumer and Provider. Note 2: Given that this use case covers two different scenarios (active-passive and active-active clusters), there may be specific requirements for each use case. In particular the synchronization mechanisms between primary and backup nodes may be different for each scenario. E.g., for the active-active scenario, the Provider can make available periodic synchronization, while in the active-passive scenario, the Consumer might need to perform manual synchronization.

<u>Detailed description</u>:

The following text provides a more detailed explanation of the use case but is *not* intended to be a complete technical implementation. Readers should be aware that the description inevitably refers to terms defined by CIMI and assumes a working familiarity with the specification.

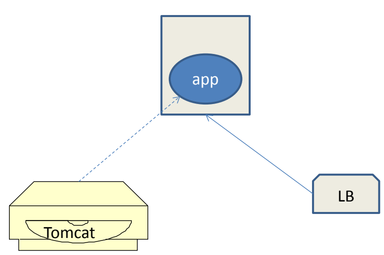
Some steps are highlighted within the Active-Passive and Active-Active scenarios.

Active-Passive scenario

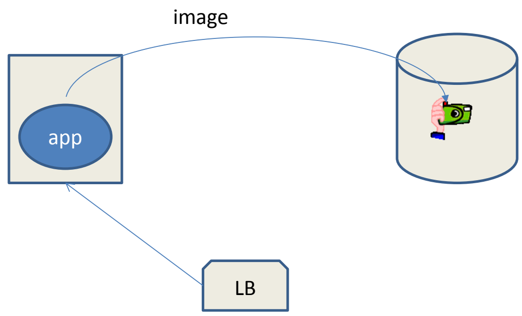

```
GET /machineConfigs HTTP/1.1
HTTP/1.1 200 OK
Content-Type: application/json
{ "resourceURI":
"http://schemas.dmtf.org/cimi/1/MachineConfigurationCollection",
"id": "http://example.com/machineConfigs",
"machineConfigurations": [
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineConfiguration",
"id": "http://example.com/configs/tiny",
"name": "tiny",
"description": "a teenie tiny one",
"created": "2012-01-01T12:00:00Z",
"updated": "2012-01-01T12:00:00Z",
"cpu": 1,
"memory": 4000000,
"disks" : [
{ "capacity": 50000000 }
]
«highAvailability" :
{«type": «passive» }
},
```


Figure 1 − Active-Passive scenario – Example cluster configuration

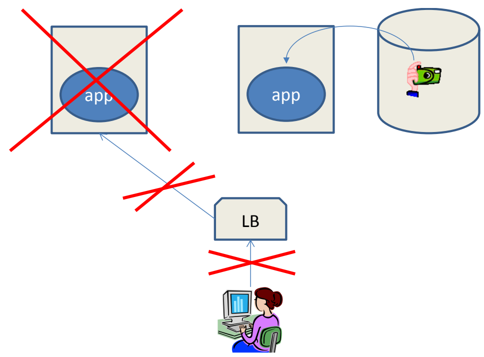
Figure 2 shows an example operation of an Active-Passive scenario.



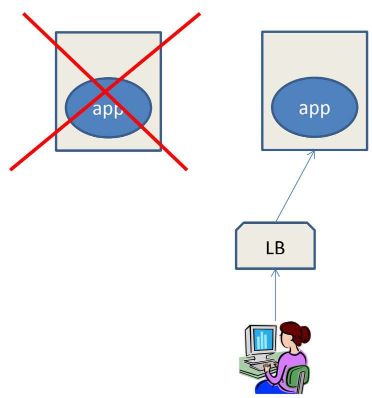
1. Normal operation: The load balancer forwards requests to the primary `Machine`.



2. Configuration for the backup `Machine` is created.



3. When the primary `Machine` fails, the backup `Machine` is created and started.



4. The load balancer forwards requests to the backup `Machine` and business operations are restored.

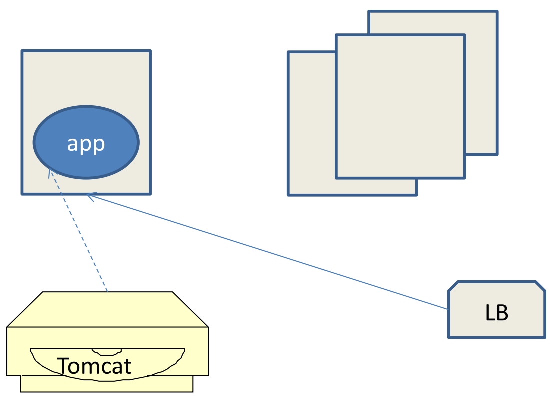
Figure 2 – Active-Passive scenario – Example operation

Active-Active scenario

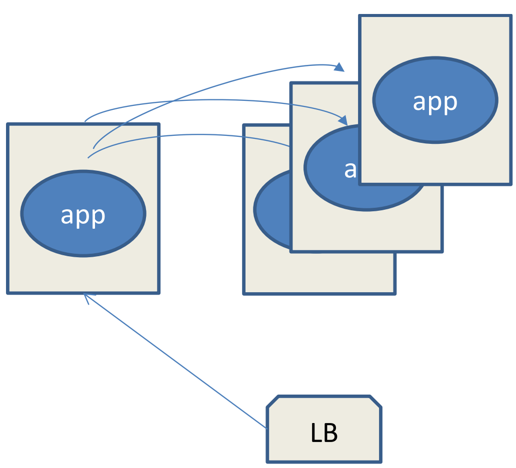

```
GET /machineConfigs HTTP/1.1
HTTP/1.1 200 OK
Content-Type: application/json
{ "resourceURI":
"http://schemas.dmtf.org/cimi/1/MachineConfigurationCollection",
"id": "http://example.com/machineConfigs",
"machineConfigurations": [
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineConfiguration",
"id": "http://example.com/configs/tiny",
"name": "tiny",
"description": "a teenie tiny one",
"created": "2012-01-01T12:00:00Z",
"updated": "2012-01-01T12:00:00Z",
"cpu": 1,
"memory": 4000000,
"disks" : [
{ "capacity": 50000000 }
]
«highAvailability" :
{«type": «active» , «node»: 3}
},
```


Figure 3 − Active-Active scenario – Example cluster configuration

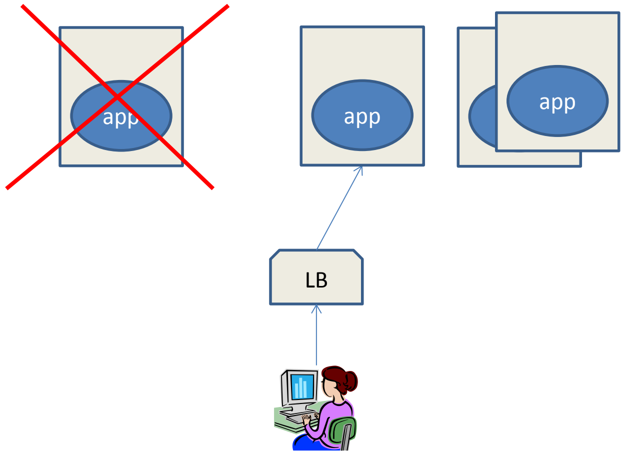
Figure 4 shows an example operation of an Active-Active scenario.



Active-Active Scenario - steps 3 - 4



Active-Active Scenario - step 5



Active-Active Scenario - steps 6 – 7

Figure 4 − Active-Active scenario - Example operation

4.1.2 Realizing Disaster Recovery on a Machine

BC/DR-002	**Realizing Disaster Recovery on a `Machine`**
Description	A Consumer creates a Machine that requires Disaster Recovery functionality.
CIMI Rationale	Disaster Recovery is becoming ever more important as enterprises move their data centers to the cloud. Enhancing CIMI to encompass such functionality can be an enabler to adoption and a differentiating factor for providers. The Consumer may want an “absolute” reliability guarantee for specific applications, even if they are cloud based, and hence Disaster Recovery becomes a necessary feature. Providing a cluster of `Machines` for Disaster Recovery implies an interaction between the software layer (application and operating system) and the underlying layers (hypervisor and firmware); the software layer is managed by the `Machine` user, the underlying layers by the `Machine` administrator. In an “in house” scenario, the set-up of Disaster Recovery functionality is easily achievable because the actors can readily communicate. In a cloud environment CIMI can be viewed as the correct medium to enable this interaction.
Dependencies with other use cases, standards, and technologies	• Business Continuity use case is described in clause 4.1.1 of this document. • Possible relations with SLO theme is described in clause 4.2 of this document. • Specific middleware/vendor platforms have their own solutions to ensure disaster recovery.
CIMI challenges	• CIMI needs to define a cluster entity model; as a first working hypothesis a set of attributes that qualify `Machine` and its configurations could be sufficient. • It is necessary to determine and formalize the replication of the `Machine` status to its backup node and define a protocol for the replication. • Analysis should be performed to identify whether the CIMI information model should also formalize the policies that determine failover.
Business Actor(s)	Cloud Service Consumer, Cloud Service Provider(s)

BC/DR-002	**Realizing Disaster Recovery on a `Machine`**	
Process Flow	Step Description	Data Required
	1: The Consumer chooses a `Machine` with a Disaster Recovery configuration from those offered by the Provider. The Consumer also selects Disaster Recovery options for any `Volumes` to be connected to the `Machine`.	
	2: The Consumer sends POST commands (`Machine` and `Volumes`) by passing the chosen configurations.	
	3: The Provider creates the primary `Machine` and `Volumes` and also remote copies for the Disaster Recovery configuration.	
	4: The Consumer configures the primary `Machine` and `Volumes` by installing applications, etc.	
	5: When the configuration is complete, the Consumer indicates the status to the Provider, who copies this configuration to the remote `Machine` and `Volumes`.	
	6: For every update on the primary `Volumes` connected to the primary `Machine`, the update is sent also to the remote `Volumes`.	
	7: Sometime during operation, the primary `Machine` might stop running (e.g. the data center has a severe issue).	
	8: The Provider starts the remote `Machine` and routes all incoming requests to the backup `Machine`.	
Variations	The same use case can be applied to “Systems” resources created by the Consumer.	
Notes	The Disaster Recovery functionality provided by this use case has to be built in collaboration with the Consumer: essentially the Provider makes available Machine and Volume configurations that allow Consumers to build their own Disaster Recovery features. Thus CIMI is needed to facilitate the interactions between the Consumer and Provider.	

4.2 Service Level Objective management use cases

4.2.1 Introducing SLO concepts in CIMI

SLO-001	**Introducing SLO concepts in CIMI**	
Description	The Provider is able to advertise Service Level Objectives (SLOs) that can be applied to specific resources. The Consumer is also able to request the creation of a resource (e.g., machine) that meets specific SLOs.	
CIMI Rationale	The CIMI rel. 1.1 specification introduced basic functionality related to SLOs. Enhancing the support for SLOs will allow the Provider to offer services that allow Consumers to optimize cost/performance benefits. For example, the Provider could offer "basic" Machines with limited performance at low cost and "premium" Machines with specific guaranteed performance at a higher cost (e.g., bronze, silver, gold type services).	
Dependencies with other use cases, standards, and technologies	• The adopted solution could benefit from output of the NIST cc_tax study group. • ISO / IEC JTC1 SC38 • ISO/IEC 17826:2012 Information Technology Cloud Data Management Interface (CDMI)	
CIMI challenges	• Defining SLOs in a manner that is applicable to different platforms. • Determining how performance can be measured to verify SLOs are being met.	
Business Actor(s)	Cloud Service Consumer, Cloud Service Provider(s)	
Process Flow	Step Description	Data Required
	1: The Provider advertises the ability to create specific resources (`Machines`, `Volumes`, etc.) with given SLOs.	
	2: The Consumer selects a specific SLO for a resource and asks the Provider to create such a resource.	May require preexisting "out of band" agreement on SLOs between the Provider and the Consumer.
	3: The Provider creates the resource as requested.	
	4: Ongoing monitoring by the Provider, the Consumer or both, determines whether or not the specified SLO is being met.	
Variations	The proposed use case is related to creating a machine with a given SLO. Other variations of this use case include creation of other "core" resources (such as volumes, networks, systems, etc.) with specific SLOs.	
Notes		

4.2.2 Assigning a common SLO to a resource in multiple clouds

SLO-002; Multicloud-007	**Assigning a common SLO to a `Machine` in multiple clouds**	
Description	The Consumer, a client of two different Providers, creates a Machine in the cloud of one Provider with a specific SLO. Subsequently the Machine is moved to the cloud of the other Provider maintaining the same SLO.	
CIMI Rationale	SLO management is fundamental to guarantee interoperability between clouds and to utilize IaaS as a commodity. The Consumer needs be able to obtain the “same” level/kind of service from different providers.	
Dependencies with other use cases, standards, and technologies	The “Introducing SLO concepts in CIMI” use case described in clause 4.2.1 of this document.	
CIMI challenges	Define an extended metrics system to evaluate the quality of the “same” service offered by different providers. Also to define mechanisms to “homogenize” different but “similar” metrics supported by different providers.	
Business Actor(s)	Cloud Service Consumer, Cloud Service Provider(s)	
Process Flow	Step Description	Data Required
	1: The Consumer is a client of two different Providers that share common SLOs (e.g., they both use the same metrics for “Availability”).	
	2: The Consumer selects a configuration for a `Machine` to be created with the desired SLO.	Providers shared/common SLOs.
	3: The Consumer requests that the first Provider create the `Machine` in the first cloud with the chosen SLO.	
	4: The Consumer moves the `Machine` from the cloud of the first Provider to the cloud of the second Provider maintaining the target SLO.	
Variations	The proposed use case is dependent upon being able to create a `Machine` with a given SLO. Other variations of this use case include creation of other “core” resources (i.e., `Volumes`, `Networks`, `Systems`, etc.) with specific SLOs.	
Notes		

<u>Detailed description</u>:

The following text provides a more detailed explanation of the use case but is *not* intended to be a complete technical implementation. Readers should be aware that the description inevitably refers to terms defined by CIMI and assumes a working familiarity with the specification.

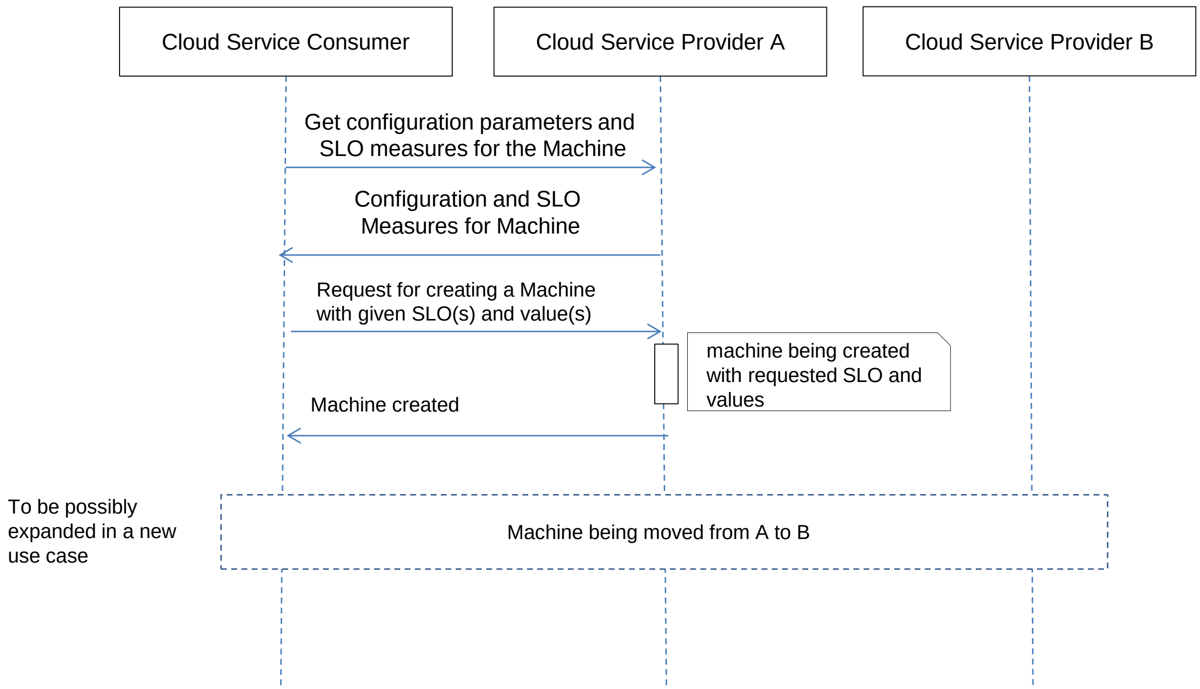
288 

Figure 5 – Creating and moving a `Machine` with a specified SLO

289 ### 4.2.3 Auto-scaling functionality

BC/DR-003 **Res-Mgm/Ctrl-002**	**Auto-scaling functionality**
Description	A Consumer creates a cluster or System that consists of several Machines, all of which are connected to the same network. The Consumer installs software in one Machine enabling it to act as a load balancer for a task, so the load is uniformly distributed across all the Machines. The Consumer defines scaling criteria and expects the Provider to observe and monitor resource utilization to automatically perform scaling actions: Examples: • Adding a new Machine when average CPU load across all Machines exceeds X% during t amount of time. The Machine cluster and the load balancer are updated to include the new Machine. • Adding a new Volume to a Machine or increasing the Volume size when the usage exceeds Y% of its nominal capacity. • Adding network capacity when the measured bandwidth utilization exceeds Z%. • Corresponding cases to decommission resources when the utilization goes below defined thresholds.

<table>
<tr><th>BC/DR-003
Res-Mgm/Ctrl-002</th><th colspan="2">Auto-scaling functionality</th></tr>
<tr><td>CIMI Rationale</td><td colspan="2">• Auto-scaling (up/down and in/out) of virtualized resources is gaining increasing attention by cloud Consumers and Providers. A cloud orchestrator is expected to support this functionality.
• Enhancing CIMI with orchestrator related functionality will increase its value and positioning potential.
• Even without an orchestration role, auto-scaling of infrastructure resources would enhance CIMI's value as a more powerful IaaS management interface.
• An auto-scaling function can be used to automate the offloading of cloud infrastructure in multicloud scenarios that are target of the next CIMI release.</td></tr>
<tr><td>Dependencies
with other use cases, standards, and technologies</td><td colspan="2">Synergies with the Business Continuity use case presented in clause 4.1.1 of this document:
• Machine cluster
• Load Balancer
• Use of policies to determine failure, recovery, scaling condition
• Action upon recovery/failure in business continuity and upon load threshold crossing in auto-scaling</td></tr>
<tr><td>CIMI challenges</td><td colspan="2">At the very least, the following new functionality would be needed in CIMI to support auto-scaling:
• Consumer defined scaling criteria (or policies) related to:
  – CPU load
  – Volume and Disk usage
  – Network and network interface bandwidth utilization
• Bandwidth management: Consumer capability to configure bandwidth of networks and network interfaces
• Provider capability to autonomously add or remove resources to/from a System when scaling criteria is met
• Provider capability to autonomously start application software, e.g., via an image, on a new Machine
• Provider capability to inform the Consumer (via eventLog) of inability (temporary or permanent) to perform further scaling actions as requested by the Consumer
• Notion of Machine cluster: a group of Machines all running the same software and homogeneous load distribution</td></tr>
<tr><td>Actor(s)</td><td colspan="2">Cloud Service Consumer, Cloud Service Provider</td></tr>
<tr><td rowspan="2">Process Flow</td><td>Step Description</td><td>Data Required</td></tr>
<tr><td>1: The Consumer creates a System with a number of Machines connected to the same Network. Each Machine has its own Volume attached. The Consumer selects desired CPU characteristics, Volume size, and Network interface bandwidth.</td><td>Network interface bandwidth</td></tr>
</table>

BC/DR-003 Res-Mgm/Ctrl-002	Auto-scaling functionality	
	2: Either as part of the `System` instantiation or as a separate update operation after the instantiation is concluded, the Consumer defines the following scaling criteria:	Scaling criteria as new properties
	2a: A new `Machine` should be instantiated and added to the `System` if the average load on all Machines exceeds 70% for more than five minutes.	CPU load as scale-out criteria (CPU percentage, average across `Machines` or measured on a single `Machine`, length of measurement period, etc.)
	2b. Additional storage, e.g., 50% more, should be added to a `Volume` when its usage exceeds 80% of capacity.	`Volume` usage as scale-up criteria (Volume usage, one time threshold crossing or averaged during a period, amount of additional storage to be allocated, etc.)
	2c: Additional bandwidth should be added to a `Network` interface, if the average bandwidth utilization of the interface exceeds 80% for more than 30 minutes. Note 1: One method for the Provider to fulfill the requirement would be to use link aggregation to add bandwidth capacity. Note 2: Bandwidth utilization criteria on other `Network` segments, e.g., between L2 switches, may be also defined.	Bandwidth utilization as scale- up criteria (bandwidth utilization, measurement period, etc.)
	3: The `System` is started. The `MachineTemplates` may also contain application software (an image) to be started. One of the `Machines` contains a load balancer function to distribute the load evenly among all the `Machines` in the `System`.	Application software
	4: The usage for a `Volume` attached to a `Machine` exceeds 80%.	
	5: The size of that `Volume` is automatically increased by 50%.	
	6: The CPU load, averaged over all Machines, exceeds 70% for at least five minutes.	
	7: A new `Machine` using the same `MachineTemplate` is created automatically, connected to the same `Network` and added to the `System`. The `Machine` and the application software are started.	
	8: The measured bandwidth utilization of a `Network` interface on a `Machine` exceeds 80% for more than 30 minutes.	
	9: Additional bandwidth is automatically added to the `Network` interface, e.g., using Ethernet link aggregation.	

BC/DR-003 **Res-Mgm/Ctrl-002**	**Auto-scaling functionality**
Variations	Additional capacity, (`Machine`, `Disk`, `Volume` and `Network`), may be allocated in a different cloud (cloud offloading as a multicloud scenario). Criteria and support for scaling down/in can also be provided.
Notes	

4.3 Log/Metadata management use cases

4.3.1 Authorization metadata management

Log/Met-001	**Authorization Metadata Management**	
Description	There are cases where two or more administrators manage and operate cloud system resources. In these cases authorization data to specify the mapping of administrators or associated users to resources and allowable operations is necessary. (i.e., access policies or Access Control Lists - ACLs).	
CIMI Rationale	Many enterprises (i.e., cloud Consumers) assign multiple administrators for cloud management and governance based on Separation of Duties (SoD) and workload reduction practices. For example, an enterprise may want different cloud resources for different departments and assign an administrator for each department. There would also be a “super-administrator” to supervise and manage shared resources for all the departments. In general, authorization rules are defined outside of CIMI and the resource description is usually in non-CIMI format. Usually administrators need to specify access policies or ACLs to be applied to resources and maintain consistency between policies and resources via a cloud management console. CIMI is required to have the capability to maintain consistency of the policy-resource mappings, even across different platforms and Providers. For example, if resource migration occurs between data centers or clouds, the associated policies must be updated and/or migrated accordingly.	
Dependencies with other use cases, standards, and technologies	• Use case “Multicloud System configuration” presented in clause 4.4.6 of this document • Other technologies: Interface to transfer authorization data from a cloud Consumer to a cloud Provider needs to be specified (for example, via ID management software)	
CIMI challenges	For a Consumer enterprise, current ways of authorizing deployment/ operation of cloud systems tend to be local (by department) or user-group based. CIMI needs to be able to adopt similar levels of granularity for authorization. Possible extensions to CIMI could be: • Mapping granularity of authorization metadata (user groups, roles, access policies etc.) to CIMI objects. • Interfaces between CIMI and authorization software to exchange authorization metadata.	
Business Actor(s)	Cloud business manager, Consumer administrators	
Process Flow	Step Description	Data Required

Log/Met-001	**Authorization Metadata Management**	
	1: Cloud business manager of Company A assigns Alice an administrator role for Company A resource operation. Alice defines two user groups and associated roles as follows: • Dept1 group for Department 1 users • Dept2 group for Department 2 users • Dept1 role to operate resources used by Department 1 • Dept2 role to operate resources used by Department 2 She then assigns Bob and Carol with roles Dept1 and Dept2, respectively.	
	2: Alice deploys a shared `Volume` for the two groups. Then she updates the access policy file with regard to authorization rules for Dept1, Dept2 and the shared resources. (The file is a non-CIMI resource but could be referenced by CIMI. In case of OpenStack, it is in the Nova service.)	Policy example • Dept1 resource: All operations are permitted to Admin, Dept1. • Dept2 resource: All operations are permitted to Admin, Dept2. • Shared `Volume`: All operations are permitted to Admin. Read/Update operations to specified sectors are permitted to Dept1, Dept2.
	3: Bob creates Dept1 resources. He attaches the shared `Volume` to his `Machines`. Then he starts running his `Machines`. Carol creates Dept2 resources. She attaches the shared `Volume` to her `Machines`. Then she starts running her `Machines`. While running, Dept1 resources and shared `Volume` sectors assigned to Dept1 are not visible to Dept2. Likewise, Dept2 resources and shared `Volume` sectors are not visible to Dept1.	
	4: When Dept2 resources need to be migrated to a different data center, Alice and Carol perform the operations necessary for migration (details described in another use case). In the migration preparation Alice copies the access policy for Dept2 resources and embeds them in the migration metadata. (Possibly part of OVF metadata, or other standards could be used.)	
Variations		

Log/Met-001	Authorization Metadata Management
Notes	CIMI extension considerations are based on authorization models of the two prevalent cloud platforms, OpenStack and Amazon Web Services (AWS). In the OpenStack model roles are assigned to users and access policies are stored in a policy file (see Figure 7, Figure 8). In the AWS model, policies are attached to resources and user entities (see Figure 10). A significant issue is how CIMI should handle authorization data such as users and roles. In the following figures we assume that user/role management is done by ID and Access Management (IAM) software outside of CIMI, and that CIMI has an IAM interface to exchange authorization data.

Detailed description:

The following text provides a more detailed explanation of the use case but is *not* intended to be a complete technical implementation. Readers should be aware that the description inevitably refers to terms defined by CIMI and assumes a working familiarity with the specification.

In the following figures, some CIMI extension possibilities are provided for the use case with two different authorization models (OpenStack-like and AWS-like).

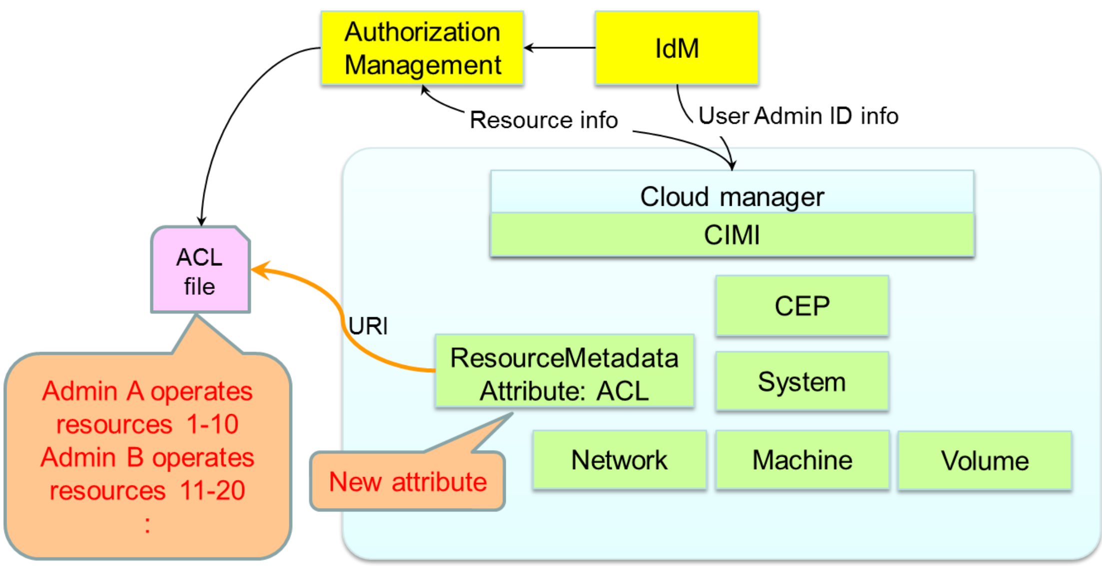


Figure 6 − Authorization metadata management example

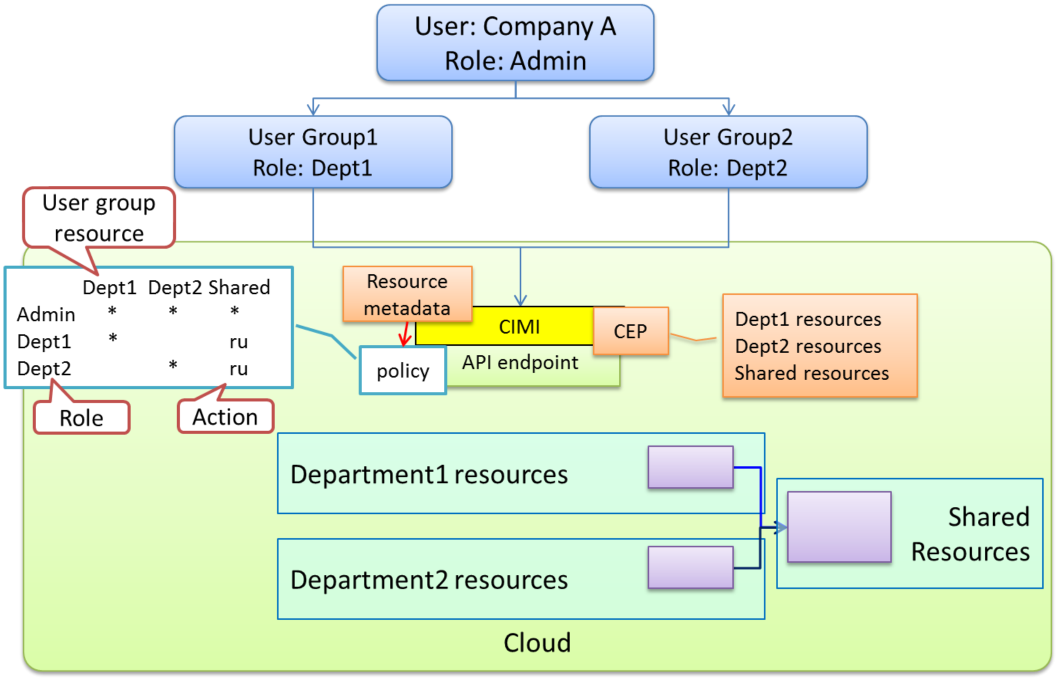


Figure 7 − OpenStack authorization model with Resource Metadata extension

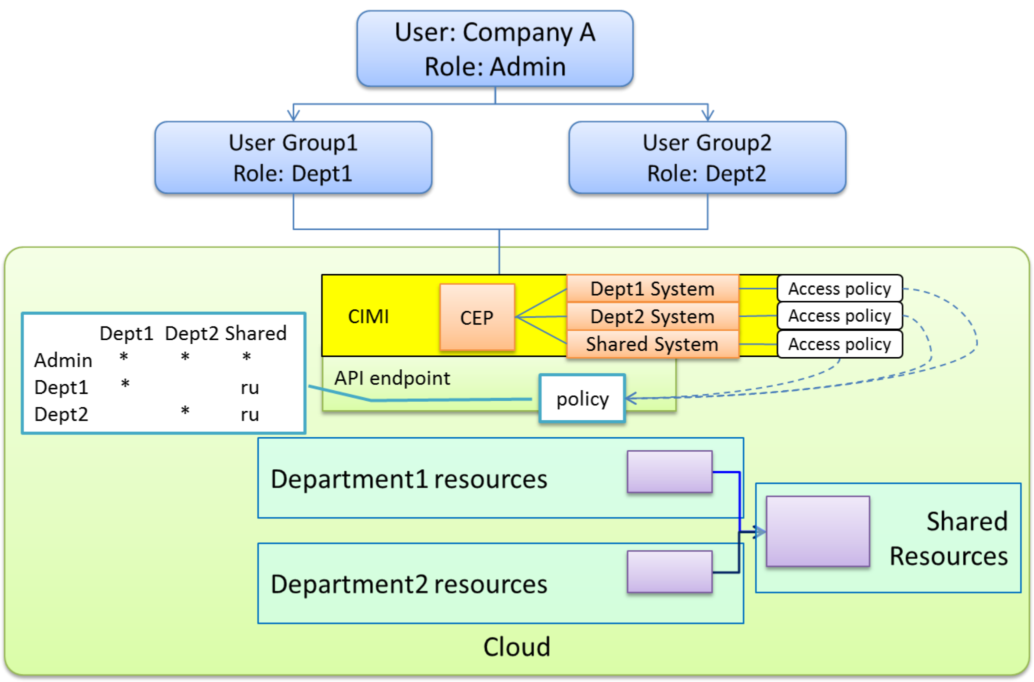


Figure 8 − OpenStack use case with System extension (Access Policy attribute)

In Figure 8 the CEP has three `System` objects corresponding to Department1, Department2, and Shared resources. Each `System` has a new object, `AccessPolicy` that refers to an access policy file in the API endpoint module. This configuration allows group by group (i.e., `System` by `System`) policy mapping.

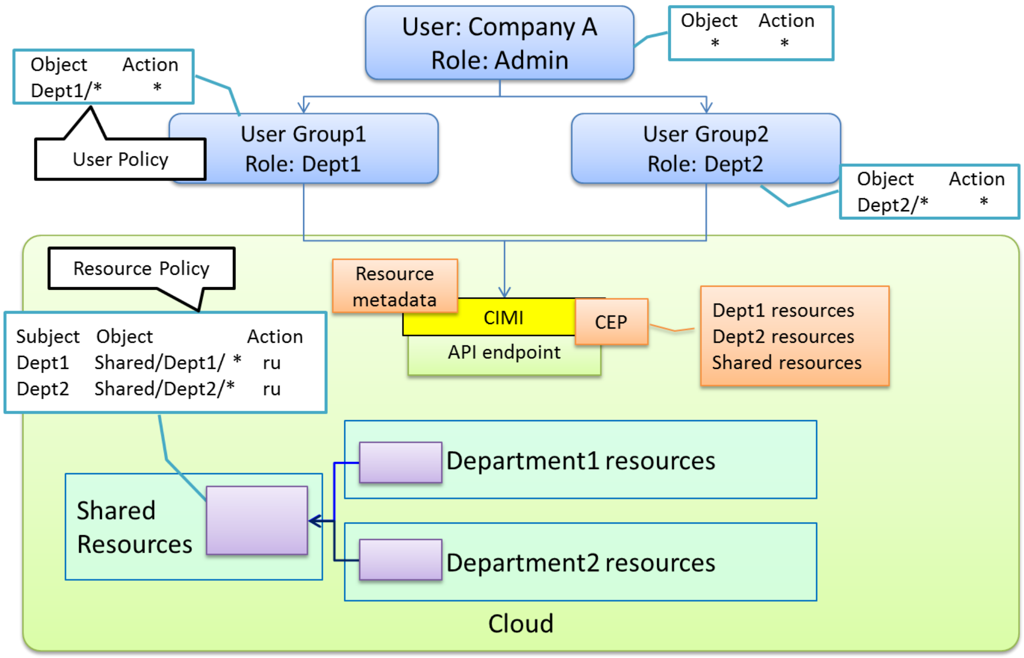


Figure 9 – AWS authorization model use case

In this case we have two types of access policy: one is associated with user groups or roles (shown as User policy) and another one is associated with resources (shown as Resource policy). User groups (Company A, Department1, Department2) have their own User policies that are not directly reachable from CIMI, and Shared resource has its own Resource policy. Simple resource metadata extension is not appropriate for this case.

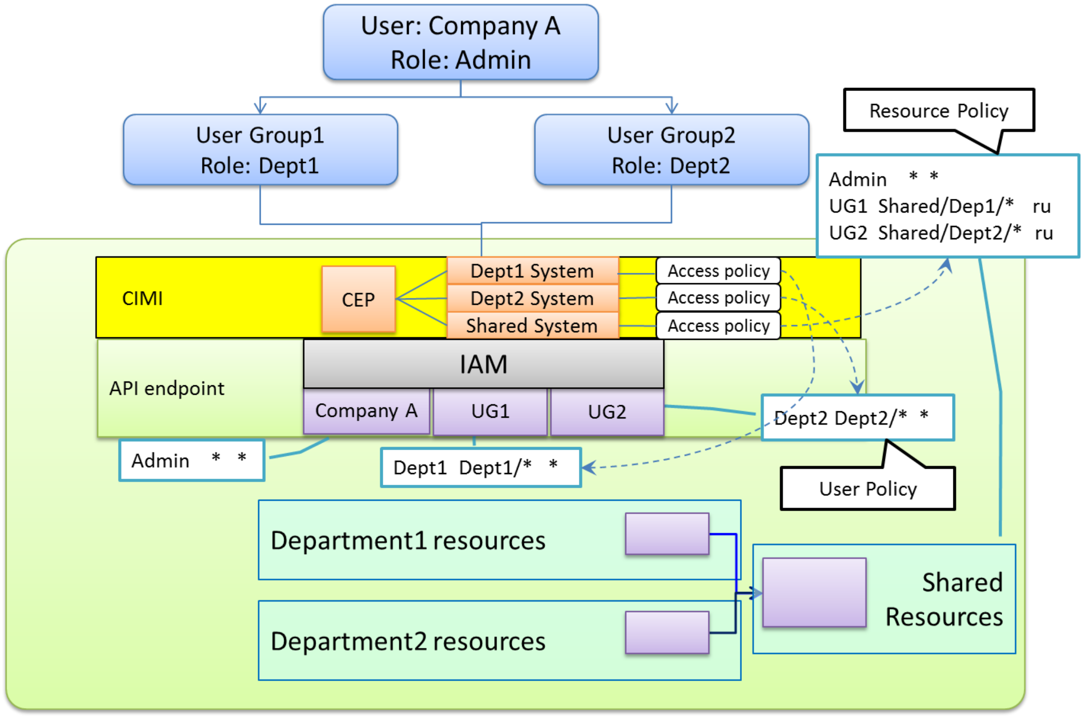


Figure 10 – AWS use case with System extension (Access Policy attribute)

To cope with user policies, CIMI needs an interface to the IAM component so that it can access user groups and associated policies, and map a user group to System objects used by the group. Figure 10 shows three `System` objects corresponding to Department1, Department2, and Shared resources, having `AccessPolicy` attributes to specify corresponding policy contents. Each policy includes access rules applied to resources belonging to the `System`.

The figure shows a somewhat mid-grained policy example, but it could also be fine grained. Other resources (`Machine`, `Volume`, etc.) can have `AccessPolicy` attributes if necessary.

An example of an `AccessPolicy` object definition is shown below. It has a `policyDocument` attribute that is a reference to a file of authorization metadata (policy by reference).

Name	`AccessPolicy`	
Type URI	http://schemas.dmtf.org/cimi/1/AccessPolicy	
Attribute	**Type**	**Description**
`enabled`	*Boolean*	Indicates whether any access policy is specified for the associated Cloud Entry Point. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
`policy Document`	*Ref*	Is the reference to the content of this access policy. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; mutable

Another example of `AccessPolicy` definition is shown below. In this example the `policyDocument` attribute has authorization metadata content or a query (queries) to retrieve authorization metadata (policy by value).

Name	AccessPolicy	
Type URI	http://schemas.dmtf.org/cimi/1/AccessPolicy	
Attribute	**Type**	**Description**
enabled	*Boolean*	Indicates whether any access policy is specified for the associated Cloud Entry Point. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
policy Document	*String*	Is the document of this access policy. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; mutable

Consumers are allowed to specify their own `AccessPolicy` objects, and also to choose an `AccessPolicyTemplate` object supplied by a Provider. An example of an `AccessPolicyTemplate` object definition is shown below. An `AccessPolicyTemplate` object has a name and a template document containing the access policy. For example, cloud Providers can prepare a “root” access policy template that allows complete access to any resources. Consumers would then choose this template to be adapted for their super users.

Name	AccessPolicyTemplate	
Type URI	http://schemas.dmtf.org/cimi/1/AccessPolicy	
Attribute	**Type**	**Description**
name	*String*	The displayed name of this access policy template. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
policy Document	*Ref*	The contents of this access policy template. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only

4.3.2 Log data management

<table>
<tr><th>Log/Met-002</th><th colspan="2">Log data management</th></tr>
<tr><td>Description</td><td colspan="2">This use case describes procedures for retrieving multiple event logs, including CIMI event logs.</td></tr>
<tr><td>CIMI Rationale</td><td colspan="2">CIMI event logs for <code>Machines</code> are generally considered to be limited to VM logs. (Usually guest OS and application event logs are not disclosed to Providers). Consumer administrators may need to access both CIMI and non-CIMI logs for cloud operations, and use cases to clarify that these procedures are useful.

Because CIMI event logs are collections of resource-specific logs, Consumer administrators may need to compile these from various sources (e.g., sort logs based on timestamps, or output events in CADF format) to transfer them to log management software.</td></tr>
<tr><td>Dependencies with other use cases, standards, and technologies</td><td colspan="2">• CADF Standard
• Use case “Aligning Monitoring and Auditing with CADF” presented in clause 4.3.3 of this document.</td></tr>
<tr><td>CIMI challenges</td><td colspan="2">• It is not clear what comprises a <code>System</code> event log. It may be better to define <code>System</code> logs as the collection of the logs attached to resources belonging to the <code>System</code>.
• Other challenges are still open; conversion to CADF for integrated analysis with non-CIMI logs could be an issue.</td></tr>
<tr><td>Business Actor(s)</td><td colspan="2">Consumer administrator</td></tr>
<tr><td rowspan="2">Process Flow</td><td>Step Description</td><td>Data Required</td></tr>
<tr><td>Case 1: <u>Machine overload</u>

A Consumer administrator finds that a <code>Machine</code> has become overloaded (via monitoring Meter data).

Reviewing the CPU usage rates for the Machine (CIMI state events) for the previous 24 hours indicates the symptoms started at time x.

The event logs of the <code>Machine’s</code> guest OS (non-CIMI, performance monitoring logs) with timestamps between x ± 1 hour are then examined.

These logs allow the identification of a problematic application (e.g., Web server).</td><td>CIMI Meter data
CIMI State Events
Non-CIMI performance logs</td></tr>
</table>

Log/Met-002	Log data management	
	Case 2: Emergency alert An Intrusion Detection System (IDS) issues an emergency alert (CIMI alarm events). The alert indicates the address where an anomaly has occurred. The Consumer administrator uses the address key to determine the corresponding `Machine`. Event logs for the `Machine` (CIMI state events) and the guest OS (non-CIMI logs) for the previous 24 hours are retrieved and examined for anomalies. If no anomalous event is found, further logs can be retrieved according to the timeline. The event logs of other `Machines` (CIMI and non-CIMI) can be examined for similar incidents.	CIMI Alarm Events CIMI State Events Non-CIMI Event logs
	Case 3: Sudden shutdown The Consumer administrator finds an unexpected application shutdown on a `Machine` (CIMI state event/alarm events). Retrieving the event logs of the guest OS (non-CIMI) on the `Machine` for the 3 hours prior to the shutdown shows a file system fault. The administrator then retrieves the event logs of the `Volumes` used by the `Machine` (CIMI state/alarm events) and identifies an unavailable `Volume`.	CIMI Alarm Events CIMI State Events Non-CIMI Event logs
	Case 4: Log transfer The Consumer administrator creates a job to execute retrieval of all CIMI logs once per day (at a specified time) and save them to a specified `Volume`. The job is run periodically so that logs are collected and transferred to Log Management software on a daily basis.	
Variations	In case 4, conversion to CADF could be included, Because it can be done in CIMI client also, it is not a strong requirement.	
Notes	It is assumed that time-based filtering capability is provided.	

4.3.3 Aligning Monitoring and Auditing with CADF

<table>
<tr><th>Log/Met-003</th><th colspan="2">Aligning Monitoring and Auditing with CADF</th></tr>
<tr><td>Description</td><td colspan="2">The Cloud Audit and Data Federation (CADF) specification has defined an event and logging format and model, the primary focus of which is to support cloud auditing functions. However the specification also supports other logging and monitoring functions (operations monitoring, metering, lifecycle history, alarms and errors).
A CIMI Provider should provide CADF compliant audit logs for use by audit tools.
CIMI metering and monitoring logs should also use the CADF format, to allow unified tooling and to avoid reformatting CIMI events into the CADF format.</td></tr>
<tr><td>CIMI Rationale</td><td colspan="2">• CADF is supports a general cloud audit function, yet it is also well-suited for operations monitoring (as in CIMI). If cloud Providers support CADF for audit, CIMI implementations would be simplified if they reuse CADFs logging feature instead of implementing their own.
• Many CIMI events have audit relevance. These events will not need to be mapped to CADF if they are already generated in that format.
• Clouds with their own APIs are likely to support CADF in the future. For example, CADF has been adopted in the Keystone authentication component of OpenStack and is being considered for the monitoring component, Ceilometer, and possibly the NOVA (compute)component. If CIMI adopted the CADF format this would make it easier to map a CIMI front-end to work with OpenStack.</td></tr>
<tr><td>Dependencies with other use cases, standards, and technologies</td><td colspan="2">This will introduce dependencies:
• With the CADF standard.
DSP0262 - Cloud Auditing Data Federation (CADF) - Data Format and Interface Definitions Specification
• Possibly with the forthcoming CADF profile for OpenStack, when using CIMI with OpenStack
DSP2038 - Cloud Audit Data Federation - OpenStack Profilehttp://members.dmtf.org/apps/org/workgroup/cadf/download.php/77597</td></tr>
<tr><td>CIMI challenges</td><td colspan="2">• CIMI events and logs need be replaced, and a CADF profile for CIMI needs to be defined.
• CIMI logs are per-Resource only. This may need to be reconsidered and a global CEP [CADF] log used instead or in addition. Because CADF events refer to specific Resource targets, it is always possible to select a subset of these that relate to a specific CIMI Resource.
• The serialization rules of CADF are different from those in CIMI, for arrays / collections. A decision needs to be made whether to keep CIMI serialization rules or adopt CADF rules.
• It may be that the CIMI specification has to leave some details of events and logs to a profile (CADF profile), in case different back-end Clouds have to be accommodated that have different content requirements for events and logs.</td></tr>
<tr><td>Business Actor(s)</td><td colspan="2">Will benefit:
• Auditors and Audit tool vendors (Consumer side, Provider side)
• CIMI developers
• Developers of CIMI adapters to existing Clouds</td></tr>
<tr><td rowspan="2">Process Flow</td><td>Step Description</td><td>Data Required</td></tr>
<tr><td>N/A</td><td></td></tr>
<tr><td>Variations</td><td colspan="2"></td></tr>
</table>

Log/Met-003	**Aligning Monitoring and Auditing with CADF**
Notes	A CADF-CIMI mapping has already been outlined in CADF specification 1.0 Appendix D.

4.4 Multicloud management use cases

4.4.1 Support for multiple operations in one job

<table>
<tr><th>Multicloud-001
Res-Mgm/Ctrl-001</th><th colspan="2">Support for multiple operations in one job</th></tr>
<tr><td>Description</td><td colspan="2">Assumptions:
• Some cloud Providers using CIMI as a Consumer northbound interface must make complex decisions regarding the placement of virtual resources (e.g. `Machines`, `Networks`, `Disks`, etc.) onto sub-clouds.
• CIMI `Systems` consisting of multiple interacting Resources must be analyzed as a unit if optimal sub-cloud assignments are to be made.
Thus:
• Cloud Providers need a methodology to receive all updates to a CIMI `System` as a single operation so they can be analyzed and processed as a unit.
• Cloud Providers need a methodology to understand the capabilities available to them from sub-cloud Providers.</td></tr>
<tr><td>CIMI Rationale</td><td colspan="2">The CIMI ambition is to move beyond a simple Hypervisor/Virtualization-Platform interface to support multicloud environments and complex multi-resource systems.</td></tr>
<tr><td>Dependencies
with other use cases, standards, and technologies</td><td colspan="2">• Interaction with OVF.</td></tr>
<tr><td>CIMI challenges</td><td colspan="2">Currently CIMI does not provide a general mechanism for combining multiple atomic operations into a single macro operation. `SystemTemplates` cover the case where a new `System` and all its contained components are created in one operation but modifications to `Systems` must be made by individual operations.
Currently CIMI does not provide a mechanism for a Provider to learn about the capabilities provided by sub-cloud Providers.
Currently CIMI does not support an adequate mechanism for specifying SLOs. (See section *4.2 Service Level Objective management use)*.</td></tr>
<tr><td>Business Actor(s)</td><td colspan="2">Cloud Service Consumer, Cloud Service Provider(s)</td></tr>
<tr><td rowspan="3">Process Flow</td><td>Step Description</td><td>Data Required</td></tr>
<tr><td>1: Cloud Provider acquires knowledge of the capabilities available from sub-cloud Providers and stores this in a database.</td><td>The capabilities and current utilization of each sub-cloud.</td></tr>
<tr><td>2: When a `System` request (create, modify, etc.) is received from a Consumer, analyze the required capabilities/SLOs and map these to available sub-clouds.</td><td></td></tr>
</table>

Multicloud-001 **Res-Mgm/Ctrl-001**	**Support for multiple operations in one job**
Variations	Many factors drive variations: • Action of the `System` request: create, change, and remove. • Available virtualization platforms/sub-clouds and their capabilities/utilization. • Relationships between the cloud receiving the request and its sub-clouds: – Contained with detailed knowledge – Shared with detailed knowledge – Shared with little or no knowledge • Complexity of `System`: number of Resources, SLOs and the ability to scatter over different sub-clouds (e.g., availability of inter-data-center networking).
Notes	None

<u>Detailed description</u>:

The following text provides a more detailed explanation of the use case but is *not* intended to be a complete technical implementation. Readers should be aware that the description inevitably refers to terms defined by CIMI and assumes a working familiarity with the specification.

4.4.1.1 Architecture

The general assumption is that the generic architecture of a Cloud Provider is given below:

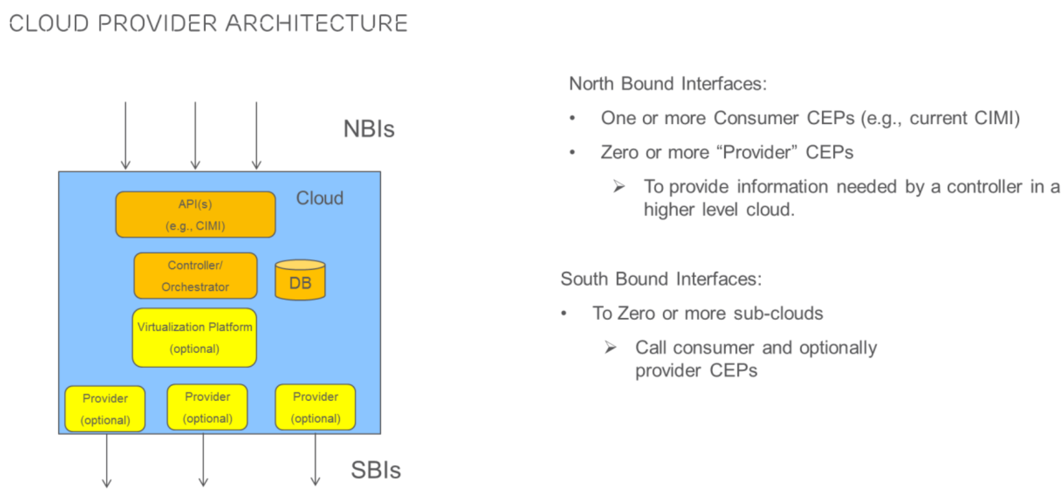


Figure 11 – Proposed Cloud Provider architecture

Each Cloud must have:

- A northbound interface
- Some sort of logic (controller/orchestration) that decides where to place `Machines`, `Disks`, `Networks`, etc.
- Targets for Resource creation:
  - Internal virtualization platform, and/or
  - Other clouds

4.4.1.2 Basic use cases based on cloud relationships

1. A single cloud has strong visibility into a subcloud.
   – Subcloud's Provider CEP provides a detailed view of available Resources (compute, storage, and network).
   – Top-cloud needs information about the subcloud's Resources (e.g., location, capabilities, utilization) to make good delegation decisions (i.e., place part of a `System` in that subcloud).
2. Many clouds have strong visibility into a subcloud.
   – Subcloud's Provider CEP provides a detailed view of available Resources.
   – May need to hide some details for privacy?
3. Many clouds use a subcloud but have only weak visibility.
   – Subcloud's Provider CEP provides only limited information advertising general capabilities or maybe nothing.
   – Top-cloud may just be instructed to use a subcloud without any knowledge of its capabilities.

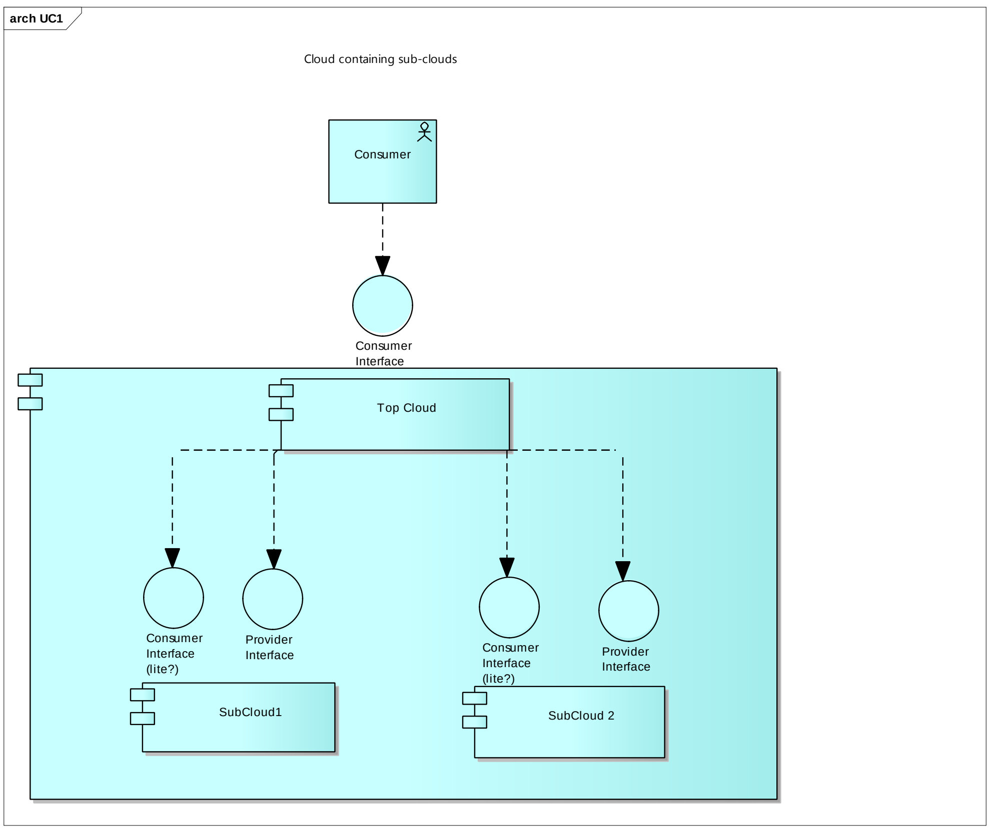


Figure 12 − Cloud containment scenario

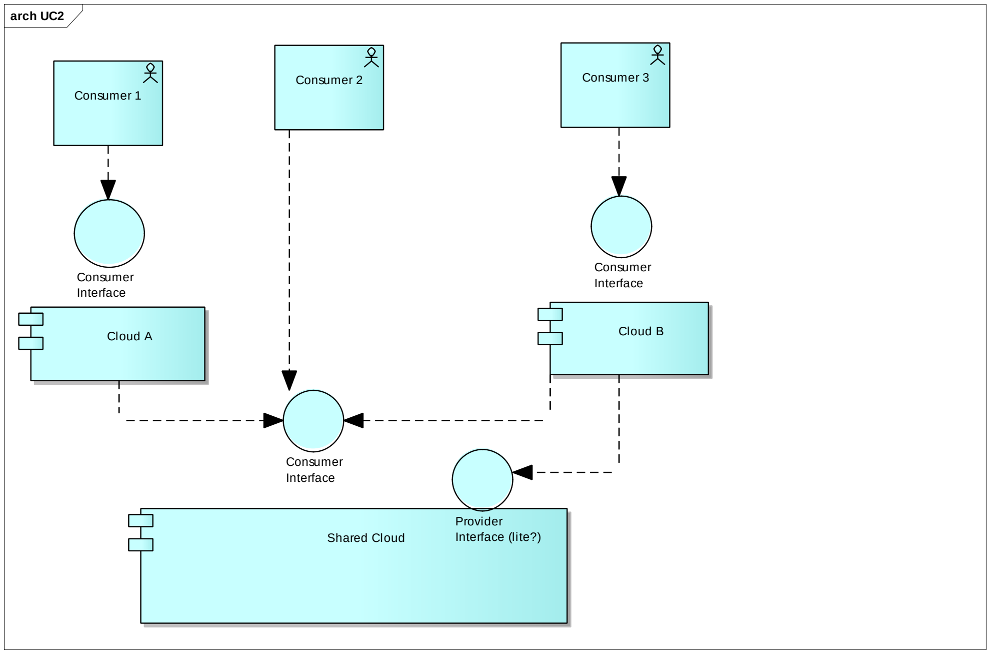


Figure 13 – Shared cloud scenario

4.4.1.3 Example

Assume there is a `System` that consists of several `Machines` and a `Network` with some maximum bandwidth. Placement of the `Machines` in various data centers was carefully analyzed when the `System` was built based on various factors including bandwidth requirements, bandwidth availability and the ability to support the `Machines` (CPU and storage).

A change request is made asking to add another `Machine` (with affinity for an existing `Machine`) and increase the `Network` bandwidth. Consider:

- Both changes may set off a chain reaction of reassignment: There may be no room for the new `Machine` near the existing `Machine` and the new bandwidth requirement might not be supportable between the currently used data centers. Thus the change must in effect completely redo the original `System` request.
- It is very inefficient to make the changes one at a time because subsequent changes may undo the work done previously.
- It is possible there is no way to accommodate all the requested changes.
- The most obvious resolution would be for the Consumer to indicate all the required changes to the `System` and pass it to the Provider as one operation.

4.4.2 Federation and multibrokering

Multicloud-002	**Federation and Multibrokering**
Description	A Consumer client of two different Cloud Providers (A and B) wants access to a unified cloud datacenter and to manage Resources allocated in the datacenters of both Providers. The Consumer wants to manage Resources as if they were from a single Provider, but also wants: • to be aware of which is the Provider for every Resource, • `System` Resources to be spread between the two Providers, • Systems within either cloud to reference Resources of both Providers. Either Provider (A or B) can act as “broker” on behalf of Consumer C and operate on Resources allocated to C within the other Provider’s datacenter.
CIMI Rationale	The use case should be supported by CIMI because both the federation and multcloud management themes are in scope of the next CIMI release according to the revised DMTF CMWG charter. Federation and brokering are facilitators for wide adoption of cloud computing technologies.
Dependencies with other use cases, standards, and technologies	• Some common aspects with use case Multicloud System configuration in clause 4.4.6 of this document • Relations with Identity Federation technologies and standards: – IEEE Standard for Intercloud Interoperability and Federation (SIIF) by the P2302 working group in IEEE. – OpenAuth (also known as OAuth) and/or OpenID (from OpenID Foundation)
CIMI challenges	Identity Federation: there must be a mechanism that enables each Cloud Provider involved in the “federation” to recognize and accept credentials presented by the Consumer. Some user/ID metadata may need to be handled by CIMI. There is a distinction to be made between: • ID management needed for the connection of two application components across “Clouds”, and • ID management needed for administrative rights on underlying Resources across “Clouds”. On the basis of a brokering request given by the Consumer, a Cloud Provider must be able to “discover” Resources in clouds managed by other Providers that are allocated to that Consumer. The Provider must also to send information on resources allocated to Consumers in its own cloud to other Providers involved in the “federation”. Existence of an effective mechanism to enable a unified vision and management of all Resources allocated to a Consumer, maintaining specificity and control for single Resources in different clouds. The role of CEPs in multicloud management: • In some cloud brokerage cases it may be preferable to hide the multicloud aspects from the Consumer. Thus only one CEP should be provided to the Consumer. • In other cases, one CEP per cloud should be accessible for the Consumer. Some type of CEP subordination may be desirable in these cases. Aspects of concurrent management of Resources should be considered: Provider B could perform “maintenance” activities on a Resource in Cloud B that is allocated to Consumer C at the same instant Consumer C tries to access that specific Resource in Cloud B via Cloud A (the broker).
Business Actor(s)	Cloud Service Consumer, Cloud Service Provider(s)

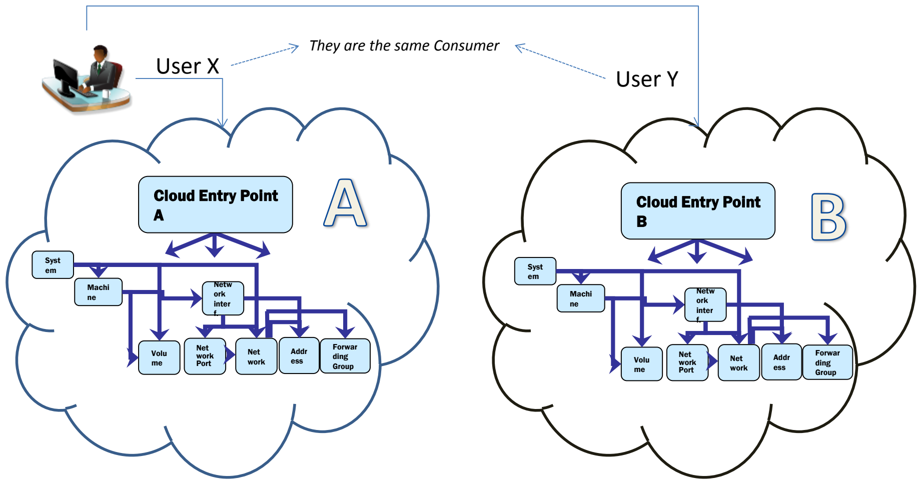
Multicloud-002	**Federation and Multibrokering**	
Process Flow	Step Description	Data Required
	0: Consumer “C” is a client (has accounts) with both Cloud A and Cloud B	Each Provider (A and B) has an account related to Consumer C.
	1: Provider A starts a “conversation” with Provider B to federate Cloud A and Cloud B. The process executes with success.	Accounts, privileges, authorizations, roles
	2: Consumer C sends a “brokering request” to Provider A to make A the broker for C towards Provider B. The process executes with success.	Credentials of C
	3: Provider A discovers Resources within Provider B that are assigned to Consumer C (in a way to be determined Provider A obtains knowledge from Provider B on B’s Resources “dedicated to A”).	
	4a: Consumer C accesses the federated cloud via Cloud A, and creates a `System` in Cloud A.	None
	4b: Consumer C accesses the federated cloud via Cloud A and creates a `Machine` in Cloud B.	None
	4c: Consumer C accesses the federated cloud via Cloud A and adds the `Machine` in Cloud B to the `System` in Cloud A.	`System` in A and `Machine` in B ready to be linked
Variations	The federation step might fail. In this case steps from 2 onwards are not executed. The brokering request from Provider A to Provider B might fail. The process is blocked until the issue is resolved. If Providers A and B are not already federated, at reception of brokering request from Consumer C, Provider A must start the federation process with Provider B for C’s Resources. This use case could lead to a hybrid cloud scenario, where a Consumer has a private cloud (e.g., Cloud A) and extends his cloud to a public cloud (Cloud B), generating a hybrid cloud.	
Notes		

Detailed description:

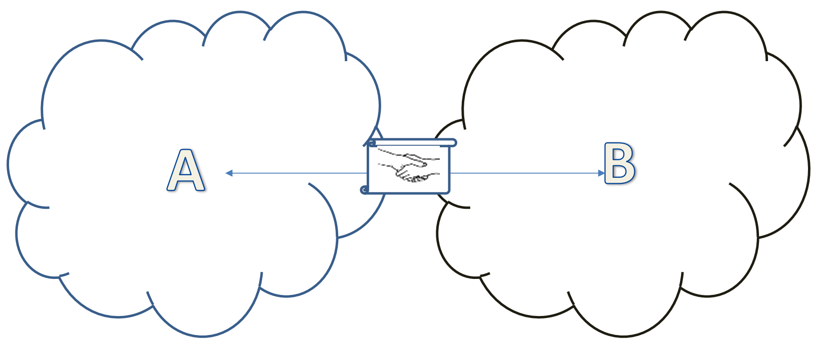
The following text provides a more detailed explanation of the use case but is *not* intended to be a complete technical implementation. Readers should be aware that the description inevitably refers to terms defined by CIMI and assumes a working familiarity with the specification.

Some steps are highlighted:

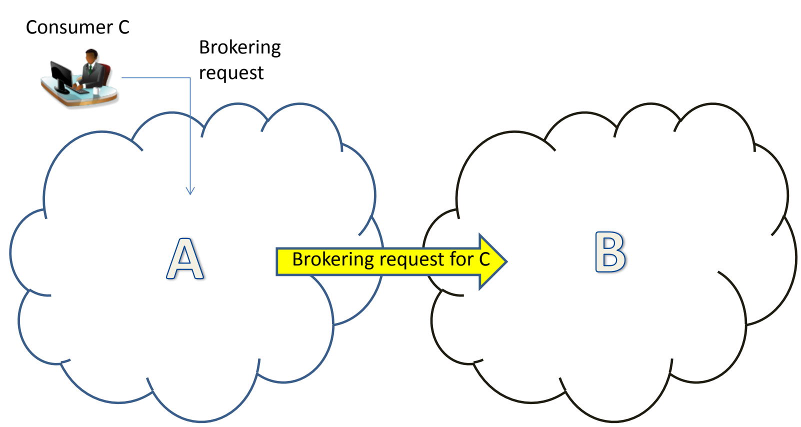
Step 0: The Consumer C is a client (has an account) of Provider A and Provider B.



Step 1: Provider A starts a “conversation” with Provider B to federate Cloud A and Cloud B. The process executes with success.

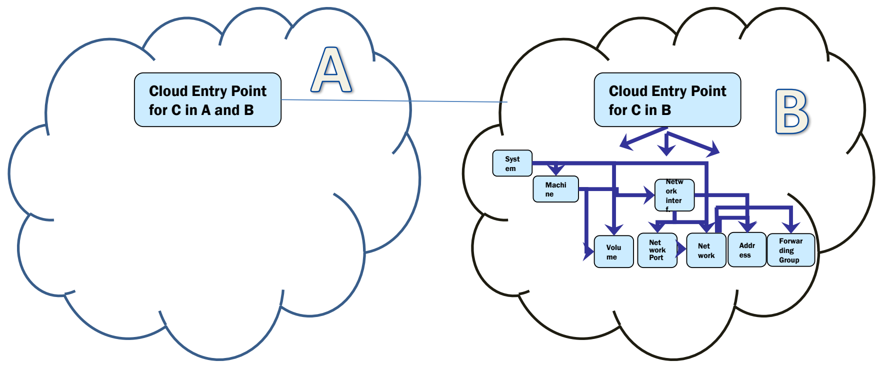


Step 2: Consumer C sends a “brokering request” to Provider A to make A the broker for C towards Provider B. The process executes with success.

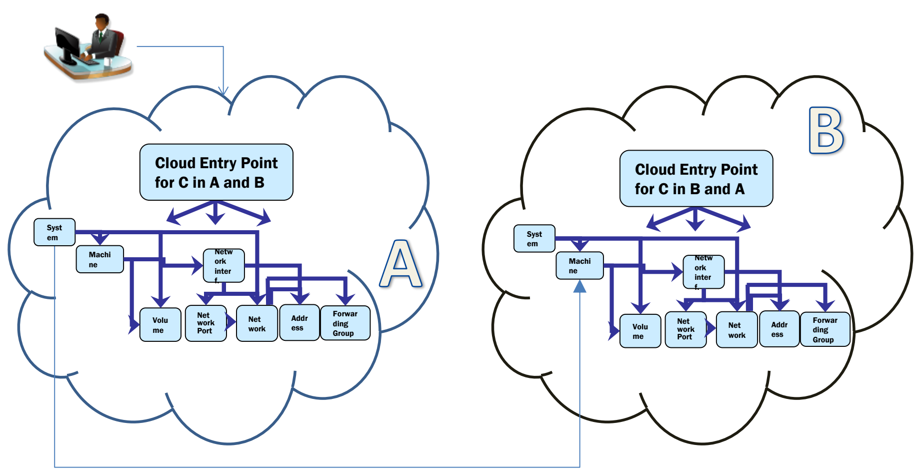


Step 3: Cloud A Provider discovers resources within Cloud B that are assigned to Consumer C.

- Note: in a way to be determined Cloud Provider A obtains knowledge from Provider B on B's Resources "dedicated to A".



Step 4: Consumer C accesses the federated cloud via Cloud A; then, as an example, creates a `System` in Cloud A, a `Machine` in Cloud B and then adds the `Machine` in Cloud B to the `System` in Cloud A (`System` in Cloud A will reference the `Machine` in Cloud B).



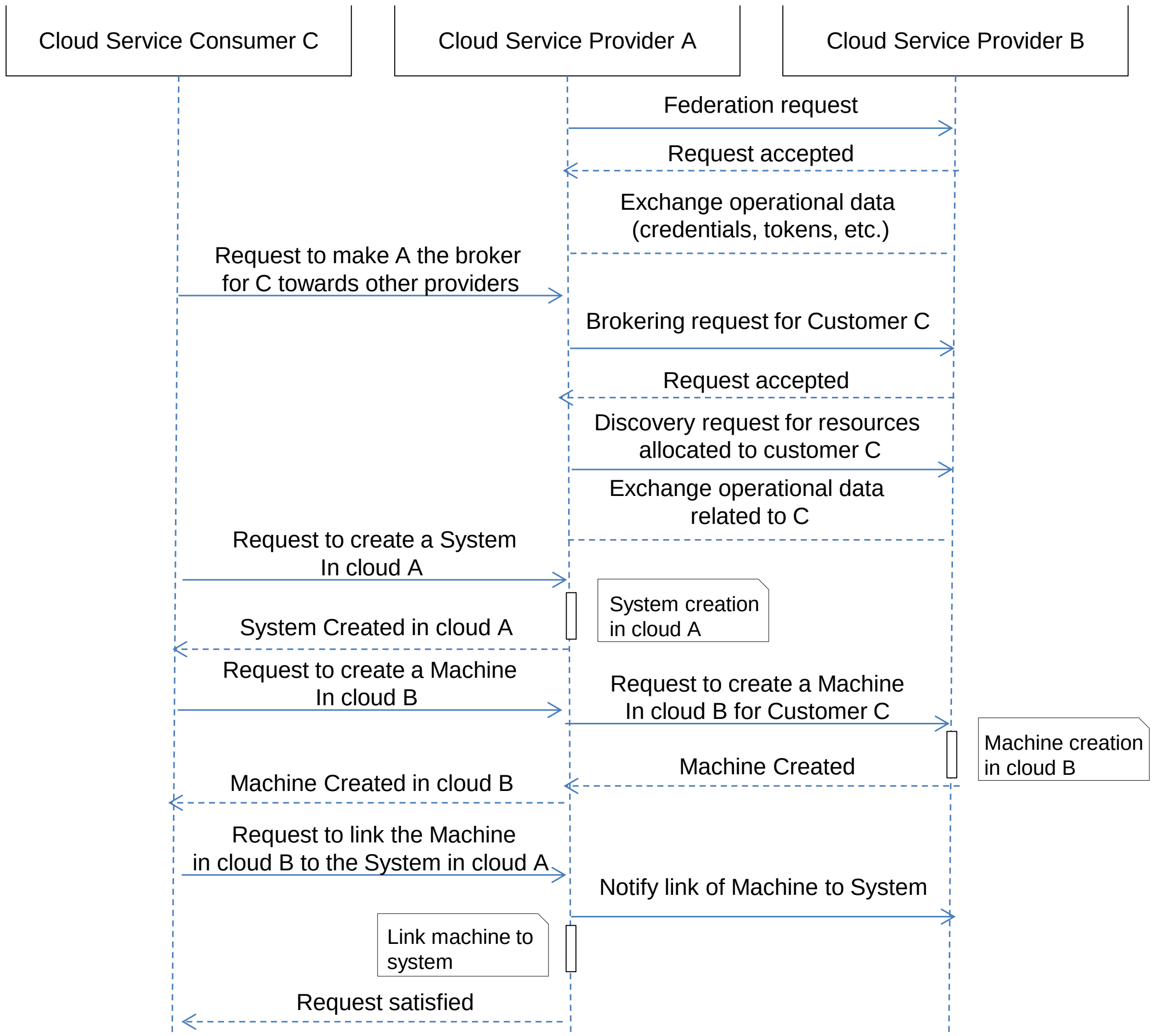


Figure 14 – Federation and Multi Brokering use case "high level view" sequence chart

4.4.3 Resource placement in a multicloud environment

<table>
<tr><th>Multicloud-003</th><th colspan="2">Resource placement in a multicloud environment</th></tr>
<tr><td>Description</td><td colspan="2">A CIMI Provider needs to be aware of federation between clouds that has different end points managed by different Providers.
In this case the CIMI Provider needs some indication as to in which backend cloud Resources needs to be provisioned. (This function could be called “cloud placement”).
CIMI needs to have knowledge of Resource placement across clouds.
CIMI currently does not model multiple clouds.
• Subcase 1: Provisioning is governed by placement policy.
Provisioning indicators could be in the form of placement policies (family of subcases). A concrete example would be that, due to latency constraints, a Consumer needs a web server to run in the same cloud as a database resource associated with that web server. In this case the Consumer facing CIMI Provider may decide which backend cloud to use to provision the database, and then once this choice is made, this decision will constrain where to put the web server.
NOTE CIMI is not required to have semantic knowledge of these components (web server, database). CIMI only needs to be aware of affinity rules and placement policies across clouds (i.e., <code>Machine</code> X needs to be provisioned in the same cloud as <code>Machine</code> Y). Such rules may be provided by a higher level orchestration layer (e.g., TOSCA). The use case does not involve higher level semantics.
• Subcase 2: Provisioning is governed by placement indicators that explicitly describe the destination cloud for the Resource being provisioned.
NOTE Subcase 2 can be seen as simpler subset of Subcase 1.
The Template author decides which backend cloud must be used to provision Resources. The use case concerns to how to convey this knowledge to the Provider so that each time the Template is used provisioning is determined according to the same rule.</td></tr>
<tr><td>Rationale</td><td colspan="2">The use case should be supported by CIMI to support effective multicloud management.
By means of this proposed interface enhancement, the Consumer is able to manage two (or more) federated clouds as if they were one.</td></tr>
<tr><td>Dependencies with other use cases, standards, technologies</td><td colspan="2">Depends on an enhancement to CIMI architecture to accommodate federated cloud environments.</td></tr>
<tr><td>CIMI challenges</td><td colspan="2">• Currently there is no place in CIMI for a Consumer to indicate that specific Resources (e.g., <code>Machines</code> in a <code>System</code>) can/should reside on different clouds.
• Today in CIMI there is no way to convey or represent either placement indication (statically determined, similar to the way other template data is handled) or placement policies (a more general case, handling dynamic placement).</td></tr>
<tr><td>Business Actor(s)</td><td colspan="2">Cloud Service Consumer, Cloud Service Provider(s)</td></tr>
<tr><td rowspan="3">Process Flow</td><td>Step Description</td><td>Data Required</td></tr>
<tr><td>1: The Consumer communicates to the Provider which cloud, from a set of those subscribed to, a given Resource is to be placed.</td><td></td></tr>
<tr><td>2: The Provider provisions the Resources as requested.</td><td></td></tr>
</table>

Multicloud-003	**Resource placement in a multicloud environment**
Variations	
Notes	

4.4.4 Extending an existing network to multiple clouds

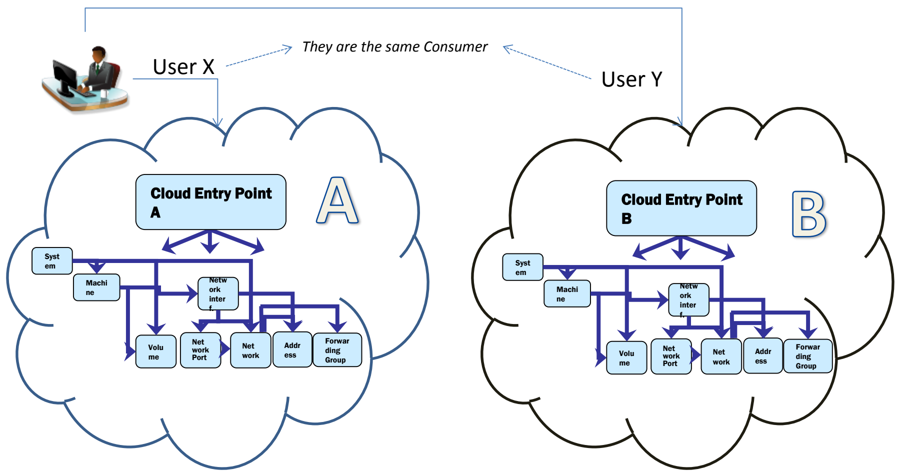
Multicloud-004	**Extending an existing network to multiple clouds**	
Description	A Consumer client of two or more different Cloud Providers wants to "extend" a private `Network` within one cloud by adding `Machines`; the added `Machines` will belong to different clouds.	
CIMI Rationale	Multicloud management is in scope for the next version of CIMI according to the revised DMTF CMWG charter. For CIMI to effectively address multicloud management, enhancements to the networking functionality currently present in CIMI are necessary.	
Dependencies with other use cases, standards, technologies	• Relations with work carried out within DMTF NSMWG. • Possible (TBD) relations with ETSI NFV initiative. • Outputs from DMTF NSMWG.	
CIMI challenges	• Integrate virtual network technologies in CIMI and enhance the CIMI network model. • Harmonize the potentially different levels of networking service delivered by all the Providers.	
Business Actor(s)	Cloud Service Consumer, Cloud Service Provider(s)	
Process Flow	Step Description	Data Required
	0: Consumer "C" is client (has an account) for Cloud A and Cloud B;	Each Provider (A and B) has an account related to Consumer C.
	1: The two Providers agree to share virtual networking (virtual network and virtual network functions) within their hardware infrastructure: the two Providers will interoperate to provide the Consumer with a unified service.	Network and commercial parameters to enable networking federation: DNS, router and proxy addresses, etc.
	2: The Consumer creates a private `Network` on Cloud A and a `Machine` in Cloud B.	
	3: The Consumer defines a `NetworkInterface`.	
	4: The Consumer adds the `Machine` in Cloud B to the `Network` in Cloud A by assigning the `NetworkInterface` to the `Machine`.	
Variations	The address assigned to the `Machine` may be static or of DHCP type.	
Notes		

<u>Detailed description</u>:

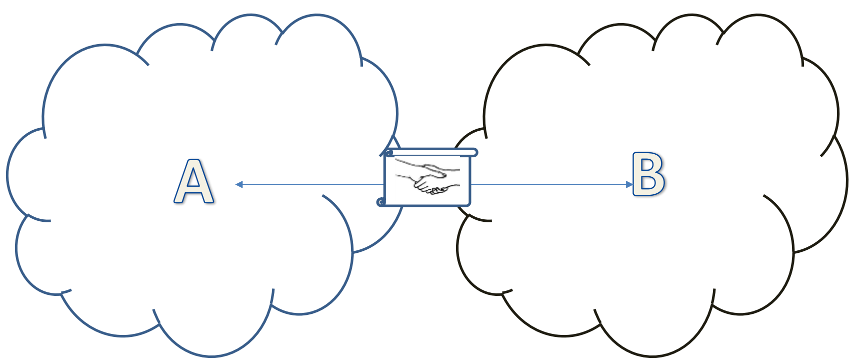
The following text provides a more detailed explanation of the use case but is *not* intended to be a complete technical implementation. Readers should be aware that the description inevitably refers to terms defined by CIMI and assumes a working familiarity with the specification.

Some steps are highlighted:

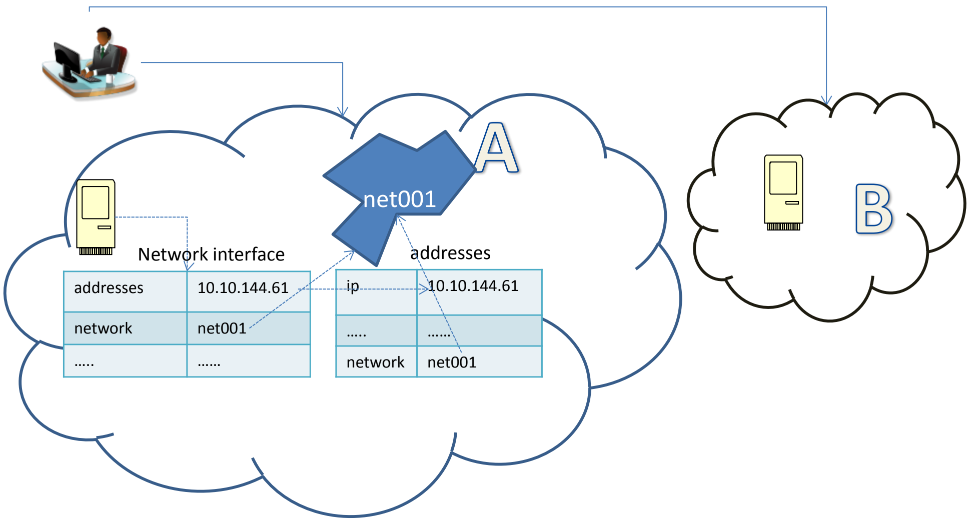
Step 0: Consumer "C" is client (has an account) of Cloud A and Cloud B.



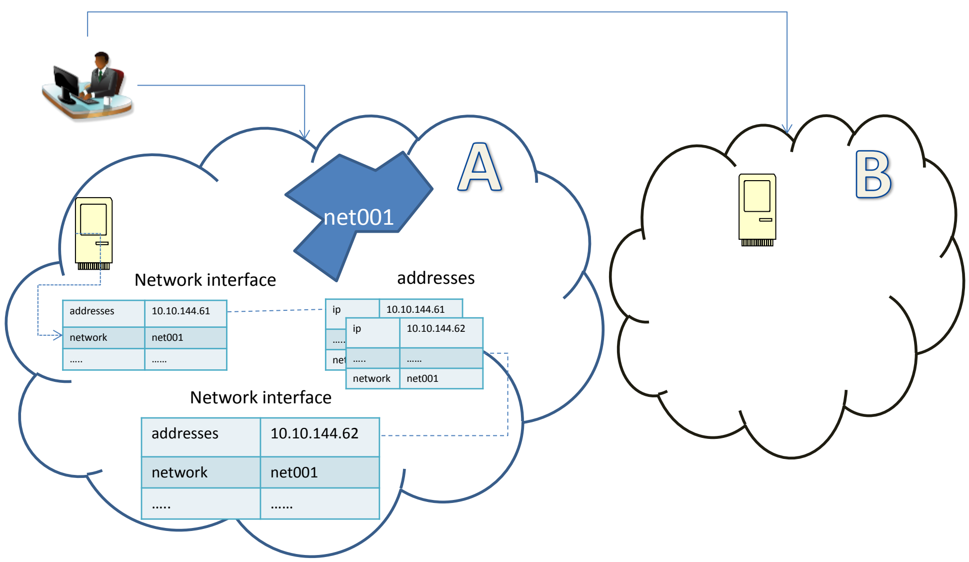
Step 1: The two Providers agree to share virtual networking (virtual network and virtual network functions) within their hardware infrastructure: the two Providers will interoperate to provide the Consumer with a unified service.



Step 2: The Consumer creates a private `Network` on Cloud A and a `Machine` in Cloud B.



Step 3: The Consumer defines a `NetworkInterface`.



Step 4: The Consumer adds the `Machine` in Cloud B to the `Network` in Cloud A by assigning the `NetworkInterface` to the `Machine`.

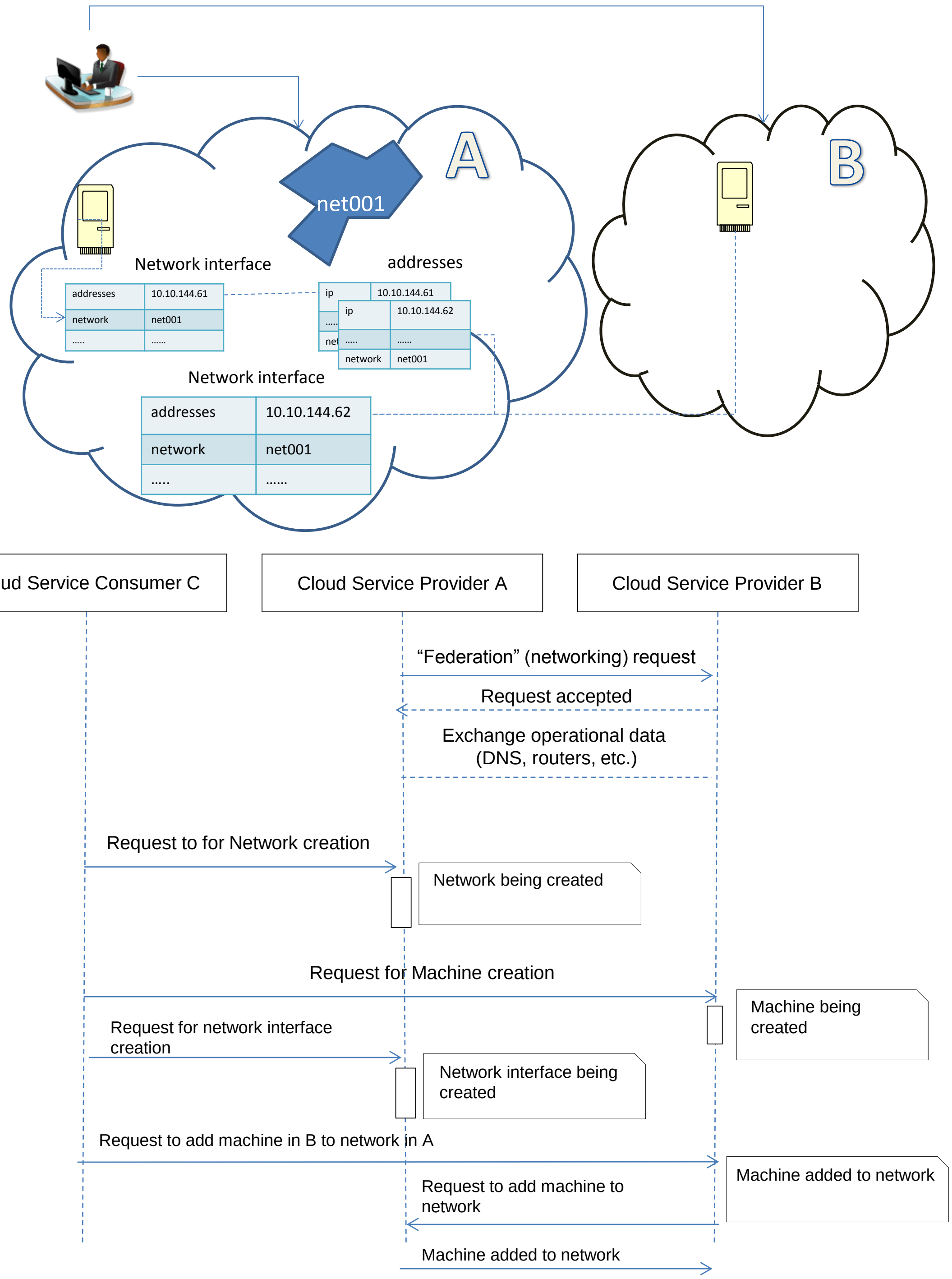


Figure 15 – Extending an existing network to multiple clouds use case “high level view” sequence chart

4.4.5 Creating an intercloud network

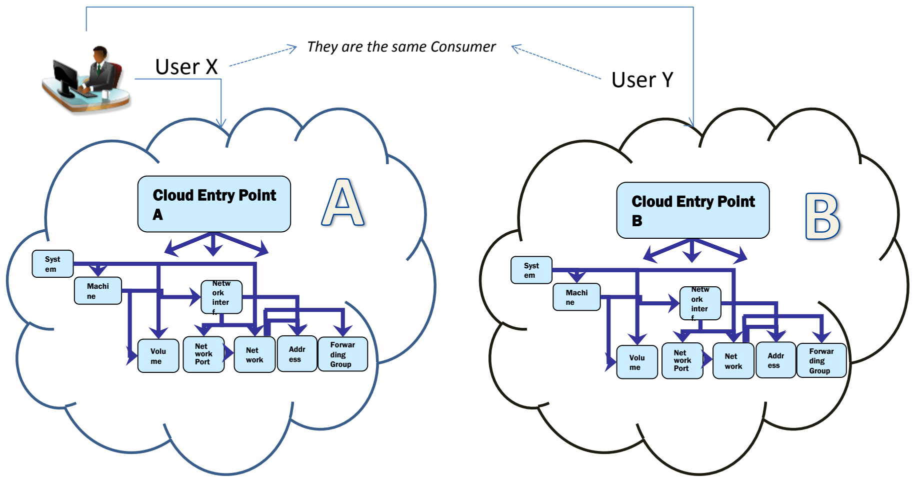
Multicloud-005	**Creating an intercloud network**	
Description	A Consumer client of two different cloud Providers (A and B) wants to create a private Network shared between the two Providers (there are segments of the Network supported by each Provider) and add machines to it; the machines will belong to both the clouds.	
CIMI Rationale	Multicloud management is in scope for the next version of CIMI according to the revised DMTF CMWG charter. For CIMI to effectively address multicloud management, enhancements to the networking functionality currently present in CIMI are necessary.	
Dependencies with other use cases, standards, technologies	• Relations with work carried out within DMTF NSMWG. • Possible (TBD) relations with ETSI NFV initiative. • Outputs from DMTF NSMWG.	
CIMI challenges	The two Providers can deliver networks with different SLOs. It is necessary to define a model that allows the Consumer to choose in a coherent way the service level for the network interfaces.	
Business Actor(s)	Cloud Service Consumer, Cloud Service Provider(s)	
Process Flow	Step Description	Data Required
	0: Consumer C is a client (has an account) for Cloud A and Cloud B. Provider A and Provider B are "federated" for networking functionality (they have agreed to share virtual networking functions).	
	1: The Consumer requests the creation of a shared private `Network` on both Cloud A and Cloud B.	
	2: The `Network` is created with appropriate service levels.	
	3: The Consumer adds two `Machines` to the `Network`, one in Cloud A and one in Cloud B.	
Variations	The addresses assigned to the `Machines` may be static or of DHCP type.	
Notes	It is possible to define two models: Characteristics and constraints of the federated `Network` are reconciled from the networking infrastructures of the two Providers (i.e., for each parameter the supported value will be the lowest common provided by both Providers). Network segments with different characteristics (e.g., SLO, QoS) could be present; The Consumer will be made aware of such differences and will be guided in defining the correct interface when adding `Machines` to the `Network`. Changes in characteristics of the interface could dictate movement of `Machines` from one Provider to the other (if all resources are federated).	

<u>Detailed description</u>:

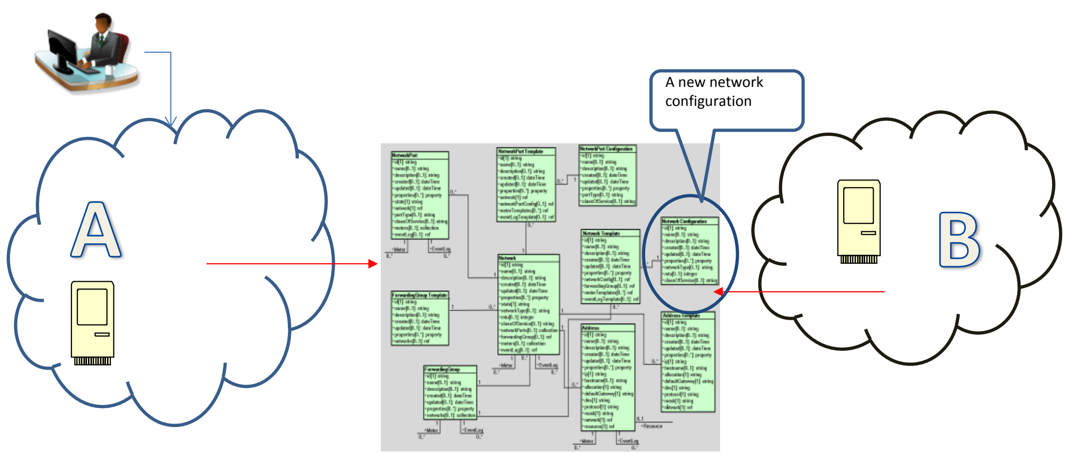
The following text provides a more detailed explanation of the use case but is *not* intended to be a complete technical implementation. Readers should be aware that the description inevitably refers to terms defined by CIMI and assumes a working familiarity with the specification.

Some steps are highlighted:

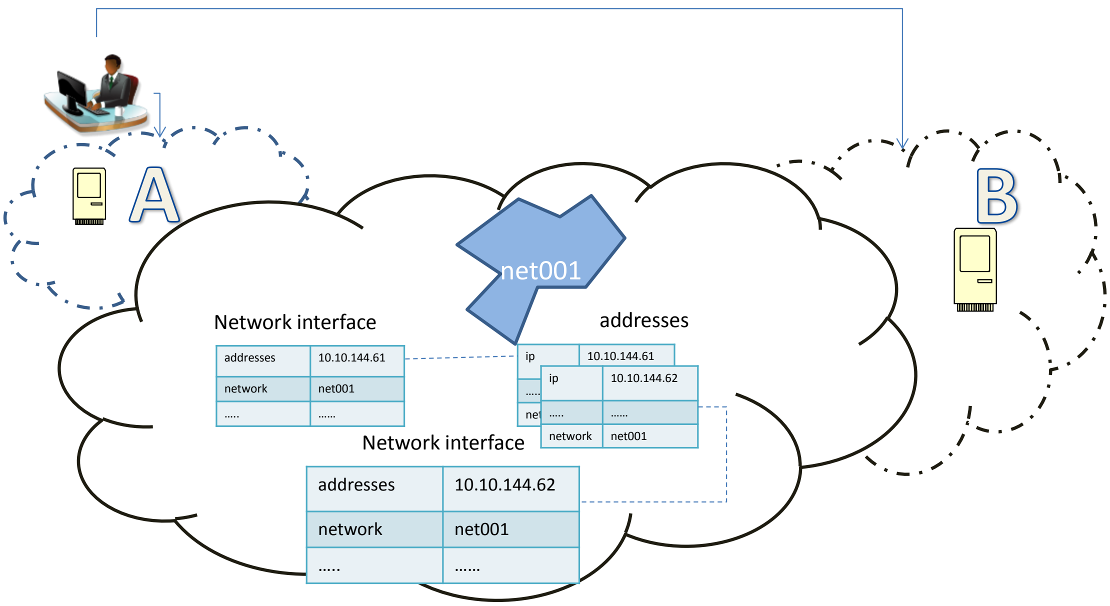
Step 0: Consumer C is a client (has an account) for Cloud A and Cloud B. Providers A and B are “federated” for networking functionality (they have agreed to share virtual networking functionality).



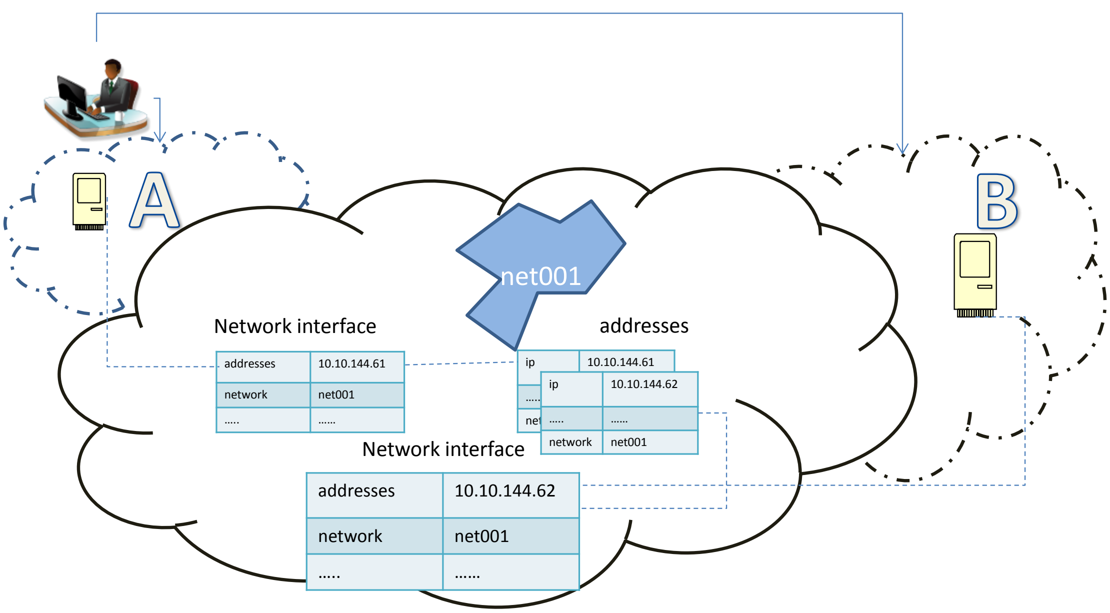
Step 1: The Consumer requests the creation of a shared private `Network` on both Cloud A and Cloud B. Provider A shows the Consumer C possible configurations that reconcile the `Network` characteristics of both Providers.



Step 2: The `Network` is created with appropriate service levels.



Step 3: The Consumer adds two `Machines` to the `Network`, one in Cloud A and one in Cloud B.



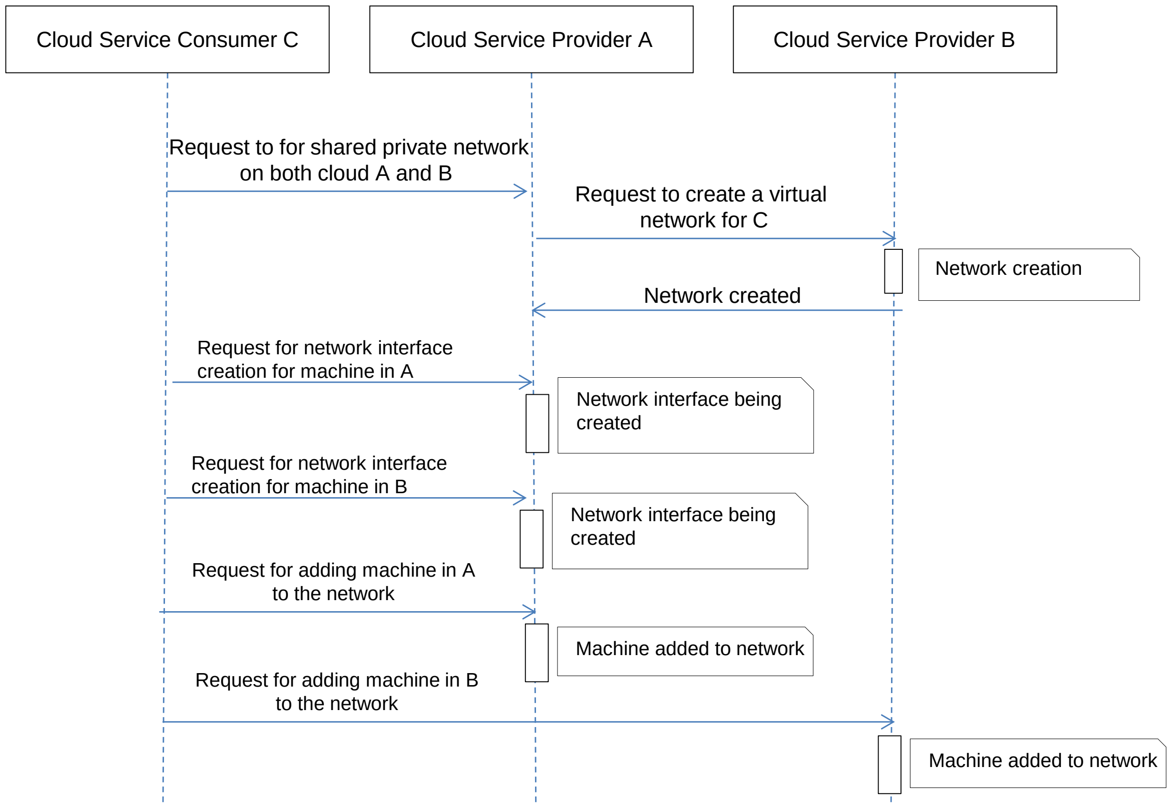


Figure 16 − Intercloud network use case "high level view" sequence chart

4.4.6 Multicloud System configuration

Multicloud-006	**Multicloud System configuration**
Description	This use case describes a machine-storage configuration across two clouds (or datacenters).
CIMI Rationale	Some `Systems` in a cloud may use extra storage, such as Resources located in another cloud. For example, suppose we have a `Machine` in one cloud (or datacenter) using a database, and a new `Machine` to provide the same service is newly deployed in another cloud (or datacenter), but the database cannot be duplicated. The deployment and operation of such `Systems` needs to be done in an integrated manner because the storage is in the same cloud.
Dependencies with other use cases, standards, and technologies	• Use case *Federation and multibrokering* presented in clause 4.4.2 of this document. • Use case *Resource placement in a multicloud environment* presented in clause 4.4.3 of this document. • Use case *Authorization metadata management* presented in clause 4.3.1 of this document.

Multicloud-006	**Multicloud System configuration**	
CIMI challenges	Assume that there is an agreement on resource deployment between Clouds A and B, so that a Cloud A consumer can deploy Cloud B Resources and use them in an integrated manner. Current CIMI does not assume that a Resource of a cloud is connected to (or included in) a Resource of another cloud. For this use case, CIMI needs be checked from viewpoints such as: Namespace rules should be consistent to specify Resources of different clouds. The current CIMI specification does not specify normative URI expressions, so some conversion or normative rules for interprovider Resource connectivity would be necessary. Many-to-one relationship of Resource connectivity should be considered. Suppose resource B of Cloud B is connected to Resource A of Cloud A. How should manage the integrity of Resource B operation (such as update/delete) by different clouds be managed? Re-definition or enhancement of the Cloud Entry Point would be needed if it includes connected Resources of different clouds (related to hybrid-cloud management use case). For example, should the CEP exactly specify a Resource in another CEP connected to its governing Resources, or the CEP where the connected resource belongs?	
Business Actor(s)	Consumer administrator	
Process Flow	Step Description	Data Required
	While operating System A of Cloud A, the Consumer administrator of Cloud A can refer to the CEP of authorized Cloud B Resources and deploy Resource B.	CEP of System B in Cloud B
	After Resource B deployment in Cloud B, the Consumer administrator connects Resource B to Resource A of Cloud A. CEP and `System` entities of Cloud A are automatically updated.	CEP of `System` A in Cloud A
Variations	Many-to-one connectivity (such as shared storage) can happen within one cloud, so it would be better to separate the many-to-one connectivity case from multicloud use cases.	
Notes		

Detailed description:

The following text provides a more detailed explanation of the use case but is *not* intended to be a complete technical implementation. Readers should be aware that the description inevitably refers to terms defined by CIMI and assumes a working familiarity with the specification.

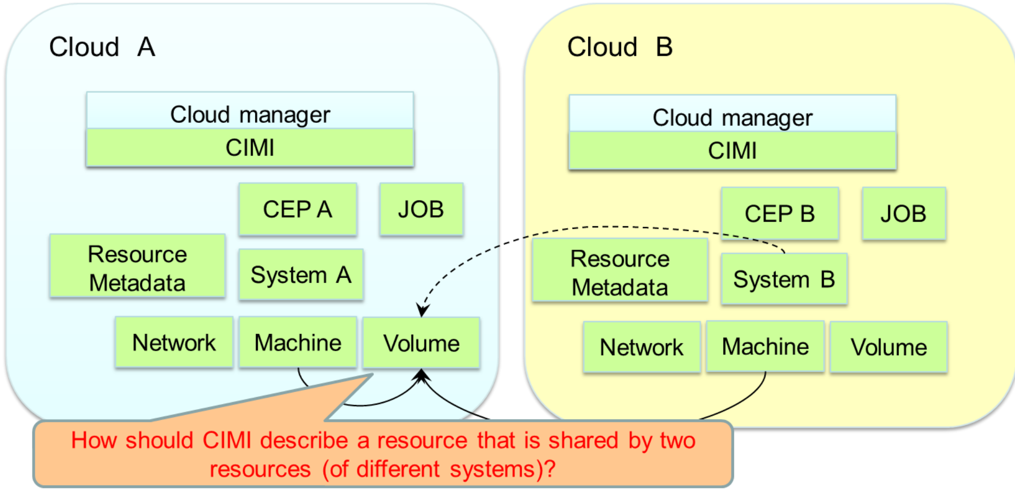


Figure 17 − Volume shared by different machines of different clouds (many-to-one connectivity)

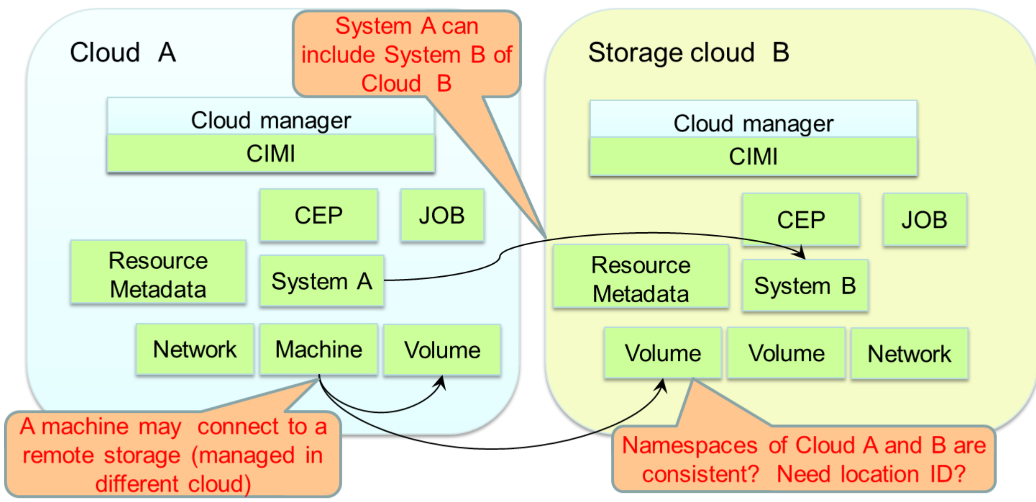


Figure 18 − Additional volume deployed in another cloud in an on-demand manner

4.4.7 Assigning a common SLO to a machine in multiple clouds

This use case can be associated to multiple categories. Refer to clause 4.2.2 for a detailed description.

4.5 OVF import/export use cases

4.5.1 OVF life cycle - import

<table>
<tr><th>OVF-001</th><th colspan="2">OVF Life Cycle – Import</th></tr>
<tr><td>Description</td><td colspan="2">Align CIMI and OVF into a unified lifecycle.</td></tr>
<tr><td>CIMI Rationale</td><td colspan="2">Cloud Consumers have a large number of software solutions available in OVF packages today.
OVF is a widely accepted international standard and supporting this use case makes CIMI useful to a wider base of Cloud Providers and Cloud Consumers.
OVF packages are designed to securely verify that the package is from the named source without modification. This is desirable for CIMI.</td></tr>
<tr><td>Dependencies with other use cases, standards, and technologies.</td><td colspan="2">• Dependence on other use cases to be determined
• Depends on OVF Standards DSP0243, DSP8027, DSP8023</td></tr>
<tr><td>CIMI challenges</td><td colspan="2">Mapping between OVF constructs and CIMI constructs; managing the OVF package as a single workload entity.</td></tr>
<tr><td>Business Actor(s)</td><td colspan="2">Cloud Service Developer
Cloud Service Consumer Administrator
Cloud Service Provider Business Manager</td></tr>
<tr><td rowspan="2">Process Flow</td><td>Step Description</td><td>Data Required</td></tr>
<tr><td>1: Import an OVF Package into the CIMI CEP making it available for use:
• Create a `SystemTemplate` from the OVF descriptor.
• Add a reference in `SystemTemplate` to the OVF package.</td><td>OVF descriptor
Virtual disks</td></tr>
<tr><td>Variations</td><td colspan="2">Some steps may be omitted and starting points may vary.</td></tr>
<tr><td>Notes</td><td colspan="2">There may be other methods of OVF package deployment; CIMI might need a way to discover running virtual systems in the Consumers environment. Authoring an OVF package is part of the OVF life cycle, but CIMI does not support OVF authoring.</td></tr>
</table>

Detailed description:

The following text provides a more detailed explanation of the use case but is *not* intended to be a complete technical implementation. Readers should be aware that the description inevitably refers to terms defined by CIMI and assumes a working familiarity with the specification.

OVF users are accustomed to describing complex systems of virtual machines, storage, and networks in an OVF descriptor, packaging the descriptor with disk images, and other files, and using the package to deploy the system on a virtualization platform. They may keep the OVF package in a library and deploy the same package repeatedly. OVF packages are also used to package systems for sale or for transfer between organizations. OVF packages are also designed to securely verify that the package is from the named source and delivered without modification, a desirable characteristic for a commercial package.

This use case proposes that participants in the OVF package ecosystem can use clouds with CIMI interfaces without rebuilding existing OVF packages to comply with CIMI. Because OVF is an established international standard, this will increase the potential CIMI user base.

4.5.2 OVF life cycle - export

<table>
<tr><th>OVF-002</th><th colspan="2">OVF Life Cycle – Export</th></tr>
<tr><td>Description</td><td colspan="2">Align CIMI and OVF into a unified life cycle. Export an OVF package,</td></tr>
<tr><td>CIMI Rationale</td><td colspan="2">Cloud Consumers need a way to move CIMI <code>Systems</code> between Cloud Providers. The OVF package is a way to do this in an interoperable manner. Cloud Consumers expect to be able to request that a hypervisor generate an OVF package that is a current snapshot of a set of virtual systems. Cloud Consumers expect the same capability for a <code>System</code> instantiated by CIMI. This enables the transfer of CIMI <code>Systems</code>, including multicloud ones, between Cloud Providers.</td></tr>
<tr><td>Dependencies with other use cases, standards, and technologies.</td><td colspan="2">• Dependence on other use cases to be determined
• Depends on OVF Standards DSP0243, DSP8027, DSP8023</td></tr>
<tr><td>CIMI challenges</td><td colspan="2">Mapping between OVF constructs and CIMI constructs.</td></tr>
<tr><td>Business Actor(s)</td><td colspan="2">Cloud Service Developer
Cloud Service Consumer Administrator
Cloud Service Provider Business Manager</td></tr>
<tr><td rowspan="2">Process Flow</td><td>Step Description</td><td>Data Required</td></tr>
<tr><td>1: Export the Cloud Consumer’s workload as an OVF Package.</td><td></td></tr>
<tr><td>Variations</td><td colspan="2">Some steps may be omitted and starting points may vary.</td></tr>
<tr><td>Notes:</td><td colspan="2">Generating an OVF package from a deployed and modified <code>System</code> is a common method of authoring OVF packages.</td></tr>
</table>

<u>Detailed description</u>:

The following text provides a more detailed explanation of the use case but is *not* intended to be a complete technical implementation. Readers should be aware that the description inevitably refers to terms defined by CIMI and assumes a working familiarity with the specification.

OVF users are accustomed to describing complex systems of virtual machines, storage, and networks in an OVF descriptor, packaging the descriptor with disk images, and other files, and using the package to deploy the system on a virtualization platform. They may keep the OVF package in a library and they may request an OVF package from the virtualization platform that represents that running system at the moment the package is requested. The resulting OVF package can be the starting point for further rounds of development by editing the new package to add new features.

This use case proposes that an OVF user should have the same experience when interacting with a cloud via CIMI.

4.6 Resources groups management and control use cases

4.6.1 Support for multiple operations in one job

Refer to clause 4.4.1 of this document for a description of the use case.

4.6.2 Auto-scaling functionality

Refer to clause 4.2.3 of this document for a description of the use case.

ANNEX A
(informative)

Change log

Version	Date	Description
1.0.0	2015-02-26	