

# **OGF OCCI-WG Deliverables**

**OCCI-WG**

---

# **OGF OCCI-WG Deliverables**

OCCI-WG

Publication date May 2009

---

# OCCI Core

## Introduction

The Open Cloud Computing Interface is an open community consensus API, initially targeting cloud infrastructure services or "Infrastructure as a Service (IaaS)". A "Resource Oriented Architecture (ROA)", it is as close as possible to the underlying HyperText Transfer Protocol (HTTP), deviating only where absolutely necessary. Each resource (identified by a canonical URL) can have multiple representations which may or may not be hypertext (e.g. HTML). Metadata including associations between resources is exposed via HTTP headers (e.g. the Link: header), except in the case of collections where Atom is used as the meta-model.

## Basics

### URL Namespace

A single URL entry point (and optionally, suitable credentials for HTTP based authentication schemes) is all a client requires. Each resource type or "noun" (e.g. compute, network, storage) is visible as a collection at the root level (e.g. `http://example.com/compute`) *Perhaps we should consider allowing them anywhere/everywhere as is the case today* and the resources themselves appear under these collections (e.g. `http://example.com/compute/123`). Implementors should also expose an AtomPub service document at the root to enable enumeration of resource types, supported formats and categories.

### Nouns, Verbs and Attributes

Interfaces expose "nouns" which have "attributes" and on which "verbs" can be performed. The attributes are exposed as key-value pairs and appropriate verbs as links, following HATEOAS principles.

### CRUD Operations

Create, Retrieve, Update and Delete (CRUD) operations map to the POST, GET, PUT and DELETE HTTP verbs respectively. HEAD and OPTIONS verbs may be used to retrieve metadata and valid operations without the entity body to improve performance. Additionally, all existing HTTP functionality is available for caching, proxying, gatewaying and other advanced functionality.

POST (Create)	POSTing a representation (e.g. OVF) to a collection (e.g. <code>/compute</code> ) will result in a new resource being created (e.g. <code>/compute/123</code> ) and returned in the Location: header. POST is also used with HTML form data to trigger verbs (e.g. restart)
GET (Retrieve)	GETting a resource (e.g. <code>/compute/123</code> ) will return a representation of that resource in the most appropriate supported format specified by the client in the Accept header. Otherwise "406 Not Acceptable" will be returned.
PUT (Update)	PUTting a representation (e.g. OVF) to a URL (e.g. <code>/compute/123</code> ) will result in the resource being created or updated. The URL is known or selected by the client (in which case UUIDs should be used), in contrast to POSTs where the URL is selected by the server.
DELETE (Delete)	DELETE results in the deletion of the resource (and everything "under" it, as appropriate).

## Web Linking

Linked cloud resources extend the World Wide Web (WWW) of linked hypertext (e.g. HTML) documents, only the links themselves are expressed out-of-band rather than inline.

## Collections

Operations that return multiple resources (e.g. categories, searches) are rendered as an Atom feed with an Atom entry per resource. Metadata that would normally appear in the HTTP headers appears in standard Atom elements with the entity-body itself being passed by reference or by value in the Atom content element.

## Versioning

Clients and servers should expose the protocol version (e.g. OCCI/1.0) via the User-Agent and Server HTTP headers respectively. Should second or subsequent versions of the descriptor format be required the version will be added to the Internet media type (e.g. application/occi2+xml).

## References

The following standards are referenced by this implementation.

- RFC 2616 Hypertext Transfer Protocol -- HTTP/1.1
- RFC 4287 The Atom Syndication Format
- RFC 5023 The Atom Publishing Protocol

Additionally RFC 2119 Requirement Levels are used throughout.

---

# OCCI Infrastructure

OCCI Infrastructure defines three nouns and various extensions relating to management of cloud infrastructure services (IaaS).

## Nouns

Cloud infrastructure can be modeled using three primary nouns: compute, network and storage.

### Compute

A compute resource is capable of conducting computations (e.g. a virtual machine).

**Table 1. Compute Attributes**

Attribute	Type	Description
occi.compute.arch	Enum (x86, x64)	CPU Architecture (e.g. x64)
occi.compute.cores	Integer	Number of CPU cores (e.g. 1, 2)
occi.compute.memory	Float (2 <sup>20</sup> bytes)	RAM in megabytes (e.g. 2048)
occi.compute.speed	Float (10 <sup>9</sup> Hertz)	Clock speed in gigahertz (e.g. 2.4)

### Network

A network resource is capable of transferring data (e.g. a virtual network or VLAN).

**Table 2. Network Attributes**

Attribute	Type	Description
occi.network.vlan-id	Integer (0..4095)	802.1q VLAN ID (e.g. 4095)
occi.network.vlan-tag	Token	Tag based VLANs (e.g. external-dmz)
occi.network.ipv4[].gateway	IPv4 Address	IPv4 gateway address (e.g. 192.168.0.1)
occi.network.ipv4[].netmask	IPv4 Address	IPv4 netmask address (e.g. 255.255.255.0)
occi.network.ipv4[].network	IPv4 Address	IPv4 network address (e.g. 192.168.0.0)
occi.network.ipv4[].cidr	Integer (0..32)	Netmask in CIDR notation (e.g. 24)

### Storage

A storage resource is capable of mass storage of data (e.g. a virtual hard drive).

**Table 3. Storage Attributes**

Attribute	Type	Description
occi.storage.persistence	Enum (transient, persistent, reliable)	Qualitative device persistence (e.g. transient)

Attribute	Type	Description
occi.storage.size	Integer (10 <sup>9</sup> bytes)	Drive size in gigabytes (e.g. 40)

## Extensions

Various extensions provide for more advanced management functionality such as billing, monitoring and reporting.

### State machine (state)

The state machine extension allows for the modeling of arbitrarily complex state machines and associated transitions (e.g. start, stop, restart).

---

# OCCI Registries

**Table 1. HTTP Status Codes**

Code	Description	Example
200 OK	Request completed successfully	Response is returned
201 Created	Request completed successfully, resource was created	Pointer to new resource returned
202 Accepted	Request accepted, processing not completed	Workload starting but not yet active
301 Moved Permanently	Resource has been assigned a new permanent URI	Workload migrated to another installation
302 Found	Resource resides temporarily under a different URI	Alias pointing to UUID can be updated
304 Not Modified	Conditional GET on resource that is unchanged	Client already has the latest version of the resource
400 Bad Request	Request could not be understood by the server due to malformed syntax	Client sent a representation that was unable to be understood
401 Unauthorized	The request requires user authentication	Client must retry with authentication
402 Payment Required	The server has refused to fulfill the request	Credit limit exceeded
403 Forbidden	The server understood the request, but is refusing to fulfill it	Attempt to access resource without permission
404 Not Found	The server has not found the resource	Feed or entry unknown
405 Method Not Allowed	The method specified is not allowed for the resource	Attempt to delete an immutable resource
406 Not Acceptable	The resource is not capable of requested content characteristics	Unsupported output format requested
409 Conflict	Request is in conflict with the current state of the resource	Resource updated by a third-party in the interim
410 Gone	Resource is gone, no forwarding address	Resource was deleted
500 Internal Server Error	Server encountered an unexpected condition	An unknown failure has occurred (e.g. out of memory)
501 Not Implemented	Functionality required to fulfill request is not implemented	A missing extension was called
502 Bad Gateway	An invalid response was received from an upstream server	The gateway received a malformed response from a node
503 Service Unavailable	Server is temporarily unable to handle the request	Server may be overloaded or down for maintenance
504 Gateway Timeout	No response was received from an upstream server	The gateway did not receive a response within the timeout period