

Open Cloud Computing Interface Specification

| | |
|--|----|
| 1. Introduction | 1 |
| 2. OCCI Core Specification | 1 |
| 2.1. Introduction | 1 |
| 2.2. Basics | 2 |
| 2.3. Connection | 3 |
| 2.4. Model | 3 |
| 2.5. Meta-model | 5 |
| 2.6. Extensibility | 8 |
| 2.7. Security Considerations | 8 |
| 2.8. Registration | 8 |
| 3. OCCI Infrastructure | 10 |
| 3.1. Kinds | 10 |
| 3.2. Extensions | 12 |
| 4. OCCI HTTP Binding | 12 |
| 4.1. Specification | 12 |
| 4.2. Example | 12 |
| 5. Contributors | 13 |
| 6. OCCI Legal Notices | 14 |
| 6.1. Intellectual Property Statement | 14 |
| 6.2. Disclaimer | 14 |
| 6.3. Full Copyright Notice | 14 |

Status of This Document

This document provides information to the Cloud and Grid community. Distribution is unlimited.

Copyright Notice

Copyright (c) Open Grid Forum (2009). All Rights Reserved.

Abstract

This document is the second deliverable of the Open Cloud Computing Interface working group. It describes a slim and extensible Interface for Infrastructure as a Service (IaaS) model based Clouds. The document consist of several modular parts which each can be used without the others. After a walkthrough the OCCI Core is described followed up by some renderings. Those parts can be seen as mandatory. For usage in IaaS based Clouds the Infrastructure part of OCCI is described next. Finally the Registries are summed up.

1. Introduction

1. An overview of the document.

2. OCCI Core Specification

2.1. Introduction

2. 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) has zero or more 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.

Tip

Some resources can be interacted with but not rendered due to the nature of the resource or prevailing security policies.

2.2. Basics

URL Namespace

3. The interface is defined by a single URL entry point which will either be a *collection*, contain *link(s)* to *collection(s)* (*in-band* and/or *out-of-band*) or both.

Kinds, Actions and Attributes

4. An interface exposes "kinds" which have "attributes" and on which "actions" can be performed. The attributes are exposed as key-value pairs and applicable actions as links, following HATEOAS principles (whereby state transitions are defined *in-band* rather than via rules).

CRUD Operations

5. 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. WebDAV definitions are used for MOVE and COPY. All existing HTTP features is available for caching, proxying, gatewaying and other advanced functionality.

POST (Create)

"The POST method is used to request that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line."RFC2616

POSTing a representation (e.g. OVF) to a collection (e.g. /compute) will result in a new resource being created (e.g. /compute/123) and returned in the Location: header. POST is also used with HTML form data to trigger verbs (e.g. restart)

GET (Retrieve - Metadata and Entity)

"The GET method means retrieve whatever information (in the form of an entity) is identified by the Request-URI."RFC2616

GETting a resource (e.g. /compute/123) 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 (Create or Update)

"The PUT method requests that the enclosed entity be stored under the supplied Request-URI."RFC2616

PUTting a representation (e.g. OVF) to a URL (e.g. /compute/123) 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)

"The DELETE method requests that the origin server delete the resource identified by the Request-URI."RFC2616

DELETE results in the deletion of the resource (and everything "under" it, as appropriate).

6. Additionally the following HTTP methods are used:

| | |
|---------------------------------|---|
| COPY (Duplicate) | “The COPY method creates a duplicate of the source resource identified by the Request-URI, in the destination resource identified by the URI in the Destination header.”RFC4918 |
| HEAD (Retrieve - Metadata Only) | “The HEAD method is identical to GET except that the server MUST NOT return a message-body in the response.”RFC2616 |
| MOVE (Relocate) | “The MOVE operation on a non-collection resource is the logical equivalent of a copy (COPY), followed by consistency maintenance processing, followed by a delete of the source, where all three actions are performed in a single operation.”RFC4918 |
| OPTIONS | “The OPTIONS method represents a request for information about the communication options available on the request/response chain identified by the Request-URI.”RFC2616 |

2.3. Connection

2.3.1. Authentication

7. Servers *may* require that requests be authenticated using standard HTTP-based authentication mechanisms (including OAuth). OAuth They indicate this requirement by returning HTTP 401 with a WWW-Authenticate header and a suitable challenge (e.g. Basic, Digest, OAuth). The client then includes appropriate Authorization headers in its responses. RFC2617

8. Servers *may* set and clients *may* accept cookies in order to maintain authentication state between requests. Such sessions *should not* be used for other purposes in line with RESTful principles. RFC2109

TODO: Add support for SAML 2?

2.3.2. Versioning

9. Every request *should* include an OCCI-Version HTTP header indicating the version of the API requested (e.g. 1.0). If none is provided the latest available version *shall* be used.

2.4. Model

10. The model defines the objects themselves without regard to how they interrelate.

2.4.1. Kinds

11. Each category of resources distinguished by some common characteristic or quality is called a *kind* (e.g. compute, network, storage, queue, application, contact).

12. Kinds defined by this standard live in the `http://purl.org/occi/kind/` namespace but anyone can define a new kind by allocating a URI they control.

Warning

Defining your own kinds can lead to interoperability problems and should be a last resort reserved for unique functionality. A simple peer review process is available for extending the registries which should be used where possible.

13. Each resource *must* specify a kind by way of a *category* within the *scheme* “`http://purl.org/occi/kind/`”.

Tip

The word *type* is not used in this context in order to avoid confusion with Internet media types.

2.4.2. Attributes

14. An *attribute* is a specification that defines a property of an object. It is expressed in the form of key-value pairs.

15. Attributes are divided into namespaces which are separated by the dash character (“-”). They *must* be handled as case-insensitive but *should* be case-preserving by default (depending on the format).

Tip

This scalable approach was derived from the Mozilla Firefox `about:config` page, though the “.” separator was replaced with “-” for maximum compatibility with various formats.

16. Attributes defined by this standard reside under the OCCI namespace (e.g. “OCCI-ABC”) but anyone can define a new attribute by allocating a unique namespace under “X-” (e.g. “X-Acme-ABC”). A number of attributes are common to all *kinds*.

Warning

Defining your own attributes can lead to interoperability problems and should be a last resort reserved for unique functionality. A simple peer review process is available for extending the registries which should be used where possible.

```
OCCI-Compute-Cores: 2
OCCI-Compute-Speed: 3000
OCCI-Memory-Size: 8192
Acme-Network-Identifier: dmz
```

Table 1. Common Attributes

| Attribute | Description | Example |
|------------------|---------------------------------------|---|
| OCCI-Id | Immutable identifier for the resource | urn:uuid:d0e9f0d0-f62d-4f28-bc90-23b0bd871770 |
| OCCI-Kind | Kind of resource | compute |
| OCCI-Title | Display name for the resource | Compute Resource #123 |
| OCCI-Summary | Description of the resource | A virtual compute resource |
| OCCI-Author-Name | Owner of the resource | John Citizen |
| OCCI-Updated | Last updated date/time [RFC3339] | 2020-12-31T23:59:59Z |
| OCCI-Version | Specification version | 1.0 |
| ETag | HTTP Entity Tags [RFC2616] | "dad86c61eea237932f" |

2.4.3. Actions

17. An *action* is some process that can be carried out on one or more *resources*.

18. Each available *action* for a given *resource* is indicated via a *link* with the *action* class.

```
<link rel="http://purl.org/occi/action/restart#cold"
      class="action"
      title="Cold Restart"
      href="http://example.com/123/restart?type=cold" />
```

19. Actions defined by this standard reside under the `http://purl.org/occi/action/` namespace but anyone can define a new action by allocating a URI they control.

Warning

Defining your own actions can lead to interoperability problems and should be a last resort reserved for unique functionality. A simple peer review process is available for extending the registries which should be used where possible.

20. An *action* is triggered via an HTTP POST and depending on the action requested (e.g. *resize*), parameters *may* be provided using HTML forms (e.g. `application/x-www-form-urlencoded`).

21. The specific parameters required and allowable values for them depend on the action and for advanced actions *may* require sending of custom *content types* rather than `application/x-www-form-urlencoded`.

22. Synchronous actions *may* return 200 OK on successful completion or 201 Created with a `Location:` header indicating a new resource for audit purposes.

Tip

Assume that clients are paranoid and want audit trails for all but the most trivial of actions.

23. In the event that the *action* does not complete immediately it *should* return HTTP 202 Accepted and a `Location:` header indicating a new resource where status and other pertinent information can be obtained.

Tip

Don't keep clients waiting - if you're not sure to return immediately then give them a resource they can monitor.

2.5. Meta-model

24. The meta-model defines how objects interrelate.

2.5.1. Categories

25. *Category* information allows for flexible organisation of resources into one or more vocabularies (each of which is referred to as a *scheme*).

26. The meta-model was derived from Atom, consisting of three attributes:

| | |
|-------------------|--|
| term | The term itself (e.g. "compute") |
| scheme (optional) | The vocabulary (e.g. "http://purl.org/occi/kind/") |
| label (optional) | A human-friendly display name for the term (e.g. "Compute Resource") |

```
<category term="compute"
          scheme="http://purl.org/occi/kind/"
          label="Compute Resource" />
```

27. Category schemes and/or terms defined by this standard reside throughout the `http://purl.org/occi/` namespace but anyone can define a new scheme by allocating a URI they control.

Tip

Categories provide a flexible way to manage resources by taxonomy (categories) and/or folksonomy (tags), where both can be shared between [groups of] users or globally. For example, users can create schemes for resource locations (e.g. US-East, US-West, Europe), operating systems (e.g. Windows, Linux) and patch levels (e.g.

TODO: Consider moving to link relations for categories so as to be compatible with existing standards rather than creating new ones. LINK is already standardised within HTML and HTTP and the Web Linking Internet-Draft will proceed to standard status. The Web Category draft is less sure, particularly where a workaround exists.

2.5.2. Collections

28. Where an operation returns multiple resources (e.g. categories, searches) this is referred to as a *collection*.

29. Depending on the format these are returned as:

- A list of pointers to resources (e.g. `text/uri-list` [`http://tools.ietf.org/html/rfc2483#section-5`])
- A list of pointers to resources with metadata (e.g. `application/atom+xml` with link to content)
- A list of embedded resources and metadata (e.g. `application/atom+xml` with content embedded)

Tip

Most collections should be pointers to resources with metadata for performance reasons - $O(1)$ rather than $O(n+1)$ requests for pointers alone. The resources themselves should only be embedded when they are known to be of a reasonable size.

30. Any given URL can be a collection and/or advertise *links* to other *collections* using the *collection* class.

Tip

The root ("`/`") *should* expose collections *in-band* and/or *out-of-band* in order for clients to discover resources.

```
<link rel="http://purl.org/occi/collection/audit"
      class="collection"
      title="Audit Entries"
      href="http://example.com/123/audit" />
```

2.5.2.1. Paging

31. Collections *may* be divided into *pages*, with each linking to the "first", "last", "next" and "previous" *link relations*.

```
<link rel="first" href="http://example.com/xyz;start=0" />
<link rel="previous" href="http://example.com/xyz;start=400" />
<link rel="self" href="http://example.com/xyz;start=500" />
```

```
<link rel="next" href="http://example.com/xyz;start=600" />
<link rel="last" href="http://example.com/xyz;start=900" />
```

2.5.3. Linking

32. Existing linking standards defined for Atom [RFC4287], HTTP [LINK] and HTML [HTML5] are used to indicate associations between resources. All formats *must* support *in-band* linking including:

- Link relations (e.g. `rel="alternate"`)
- Pointers to resources (e.g. `href="http://example.com/"`)
- Internet media types (e.g. `type="text/html"`)
- Extensibility (e.g. `attribute="value"`)

```
<link rel="related"
      title="System Documentation"
      href="http://example.com/user-guide.pdf"
      type="application/pdf" />
```

33. *Link relations* defined by this standard reside under the `http://purl.org/occi/rel` namespace but anyone can define a new *link relation* by allocating a URI they control.

Table 2. Link Relations

| Relation | Description |
|--|--|
| <code>category</code> (<code>http://purl.org/occi/rel#category</code>) | A category mapping whereby: <ul style="list-style-type: none"> • The <i>scheme</i> is required and indicated by the <code>href</code> attribute. • The <i>label</i> is optional and indicated by the <code>title</code> attribute. • The <i>term</i> is required and indicated by the <code>term</code> extended attribute. |
| <code>collection</code> (<code>http://purl.org/occi/rel#collection</code>) | A related collection whereby: <ul style="list-style-type: none"> • The <i>root</i> of the collection is indicated by the <code>href</code> attribute. • The <i>kind</i> of the collection is indicated by the <code>kind</code> extended attribute. |
| <code>first</code> | "An IRI that refers to the furthest preceding resource in a series of resources." [LINK] |
| <code>help</code> | "The referenced document provides further help information for the page as a whole." [HTML5] |
| <code>icon</code> | "The specified resource is an icon representing the page or site, and should be used by the user agent when representing the page in the user interface." [HTML5] |
| <code>last</code> | "An IRI that refers to the furthest following resource in a series of resources." [LINK] |
| <code>next</code> | "A URI that refers to the immediately following document in a series of documents." [LINK] |

| Relation | Description |
|----------|--|
| previous | "A URI that refers to the immediately preceding document in a series of documents." [LINK] |
| search | "The referenced document provides an interface specifically for searching the document and its related resources." [HTML5, OpenSearch] |
| self | "Identifies a resource equivalent to the containing element" [RFC4287] |

2.6. Extensibility

34. The interface is fully extensible, both via a public peer review process (in order to update the specification itself, usually via registries) and via independent allocation of unique namespaces (in order to cater for vendor-specific enhancements).

2.6.1. Foreign markup

35. Implementations *must* accept and forward but otherwise ignore markup they do not understand.

2.7. Security Considerations

36. Encryption is not required by the specification in order to cater for sites that do not or can not use it (e.g. due to export restrictions, performance reasons, etc.), however SSL/TLS *should* be used over public networks including the Internet.

2.8. Registration

2.8.1. IANA Considerations

2.8.1.1. Internet Media Types (MIME Types)

37. The following media types are to be registered:

- text/occi
- application/occi+atom
- application/occi+json

2.8.1.2. Well-Known URI Registry

38. The following well-known URI suffix is to be registered:

| | |
|------------------------|---|
| URI Suffix | <code>/.well-known/occi/</code> |
| Change Controller | OGF |
| Specification Document | Open Cloud Computing Interface (OCCI) [http://purl.org/occi] |
| Related Information | N/A |

2.8.1.3. Link Relation Type Registry

39. The following *link relations* are to be registered:

- Category

| | |
|---------------|---|
| Relation Name | category |
| Description | Assigns the link's context to a category, whereby the <code>scheme</code> is required and indicated by the <code>href</code> attribute, the <code>label</code> is optional and indicated by the <code>title</code> attribute and the <code>term</code> is required and indicated by the <code>term</code> extended attribute. |
| References | <ul style="list-style-type: none">• Atom [RFC4287]• OCCI [this specification] |
| Notes | Category meta-model was derived from Atom for use with OCCI. This relation was defined for compatibility with existing standards including HTTP and HTML. |

- Collection

| | |
|---------------|---|
| Relation Name | collection |
| Description | Identifies a related <i>collection</i> whereby the root of the collection is indicated by the <code>href</code> attribute and the <i>kind</i> of the collection is indicated by the <code>kind</code> extended attribute. |
| References | <ul style="list-style-type: none">• OCCI [this specification] |
| Notes | N/A |

Glossary

| | |
|-------------|--|
| in-band | “Sending of metadata and control information in the same band, on the same channel, as used for data”, for example, by embedding it in HTML. [http://en.wikipedia.org/wiki/In-band] |
| kind | “A category of things distinguished by some common characteristic or quality”, for example events, messages, media. [http://wordnetweb.princeton.edu/perl/webwn?s=kind] |
| out-of-band | “Communications which occur outside of a previously established communications method or channel”, for example, in HTTP headers. [http://en.wikipedia.org/wiki/Out-of-band_signaling] |
| type | Internet media (MIME) type as defined by RFC2045 and RFC2046 |

Bibliography

Normative References

[RFC2045] *RFC 2045 - Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*. <http://tools.ietf.org/html/rfc2045>. Internet Engineering Task Force (IETF) 1996-11.

[RFC2046] *RFC 2046 - Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*. <http://tools.ietf.org/html/rfc2046>. Internet Engineering Task Force (IETF) 1996-11.

- [RFC2109] *RFC 2109 - HTTP State Management Mechanism*. <http://tools.ietf.org/html/rfc2109>. Internet Engineering Task Force (IETF) 1997-02.
- [RFC2616] *RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1*. <http://tools.ietf.org/html/rfc2616>. Internet Engineering Task Force (IETF) 1999-06.
- [RFC2617] *RFC 2617 - HTTP Authentication: Basic and Digest Access Authentication*. <http://tools.ietf.org/html/rfc2617> [<http://tools.ietf.org/html/rfc2616>]. Internet Engineering Task Force (IETF) 1999-06.
- [RFC3339] *RFC 3339 - Date and Time on the Internet: Timestamps*. <http://tools.ietf.org/html/rfc3339>. Internet Engineering Task Force (IETF) 2002-07.
- [RFC4918] *RFC 4918 - HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)*. <http://tools.ietf.org/html/rfc4918> [<http://tools.ietf.org/html/rfc2616>]. Internet Engineering Task Force (IETF) 2007-06.
- [OpenSearch] *OpenSearch 1.1*. <http://www.opensearch.org/Specifications/OpenSearch/1.1> [<http://tools.ietf.org/html/rfc2616>]. A9.com, Inc. (an Amazon company) Clinton DeWitt. Joel Tesler. Michael Fagan. Joe Gregorio. Aaron Sauve. James Snell. 2009.

Informative References

- [RFC4287] *RFC 4287 - The Atom Syndication Format*. <http://tools.ietf.org/html/rfc4287>. Robert Syre. Mark Nottingham. Internet Engineering Task Force (IETF) 2005-12.
- [HTML5] *HTML 5*. <http://www.w3.org/TR/html5/> [<http://tools.ietf.org/html/rfc4287>]. Ian Hickson. David Hyatt. World Wide Web Consortium (W3C) 2009-08-25.
- [OAuth] *OAuth*. <http://oauth.net/core/1.0> [<http://tools.ietf.org/html/rfc2616>]. OAuth Core Workgroup <spec@oauth.net>. 2007-12-04.
- [RWS] *RESTful Web Services*. <http://oreilly.com/catalog/9780596529260/>. 9780596529260. O'Reilly Media Leonard Richardson. Sam Ruby. 2007-05.
- [LINK] *Web Linking*. <http://tools.ietf.org/html/draft-nottingham-http-link-header>. Internet Engineering Task Force (IETF) Mark Nottingham. 2009-07-12.
- [CATEGORY] *Web Categories*. <http://tools.ietf.org/html/draft-johnston-http-category-header>. Internet Engineering Task Force (IETF) Sam Johnston. 2009-07-1.

3. OCCI Infrastructure

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

Table 3. Common Attributes

| Attribute | Type | Description |
|------------------------------|--------|---|
| OCCI-Infrastructure-Hostname | String | Valid DNS hostname for the resource (may be FQDN) |

3.1. Kinds

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

Table 4. Kinds

| Kind | URI | Description |
|---------|---|----------------------------------|
| compute | http://purl.org/occi/kind/compute | Information processing resources |

| Kind | URI | Description |
|---------|-----------------------------------|--------------------------------|
| network | http://purl.org/occi/kind/network | Interconnection resources |
| storage | http://purl.org/occi/kind/storage | Recorded information resources |

3.1.1. Compute

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

Table 5. Compute Attributes

| Attribute | Type | Description |
|---------------------------------|--------------------------------------|--|
| OCCI-Compute-CPU-Arch | Enum (x86, x64) | CPU Architecture (e.g. x64) |
| OCCI-Compute-CPU-Cores | Integer | Number of CPU cores (e.g. 1, 2) |
| OCCI-Compute-CPU-Speed | Float (10 ⁹ Hertz) | Clock speed in gigahertz (e.g. 2.4) |
| OCCI-Compute-Memory-Size | Float (10 ⁶ bytes) | RAM in megabytes (e.g. 8192) |
| OCCI-Compute-Memory-Speed | Float (10 ⁹ bytes/second) | RAM speed in Gbit/s (e.g. 17 for PC-8500 DDR3 per Wikipedia) |
| OCCI-Compute-Memory-Reliability | Enum (standard, checksum) | Qualitative measure of RAM reliability (e.g. ECC) |

3.1.2. Network

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

Table 6. Network Attributes

| Attribute | Type | Description |
|-------------------------|---|---|
| OCCI-Network-VLAN | Integer (0..4095) | 802.1q VLAN ID (e.g. 4095) |
| OCCI-Network-Label | Token | Tag based VLANs (e.g. external-dmz) |
| OCCI-Network-Address | IPv4 or IPv6 Address (in CIDR notation) | IP gateway address or network address where there is none (e.g. 192.168.0.1/24, 2001:db8:a::123/64) |
| OCCI-Network-Allocation | Enum (auto, dhcp, manual) | Address allocation mechanism: <ul style="list-style-type: none"> • auto is handled automatically by infrastructure and/or guest agent • dhcp uses network-based allocation protocol(s) • manual requires preconfiguration or manual allocation |

TODO: Tidy up network interface addressing.

3.1.3. Storage

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

Table 7. Storage Attributes

| Attribute | Type | Description |
|--------------------------|--|---|
| OCCI-Storage-Reliability | Enum (transient, persistent, reliable) | Qualitative device persistence (e.g. transient) |
| OCCI-Storage-Size | Integer (10 ⁹ bytes) | Drive size in gigabytes (e.g. 40, 0.00144) |
| OCCI-Storage-Speed | Integer (10 ⁶ bytes/second) | Drive speed in MB/s (e.g. 600 for SAS/SATA-600 Wikipedia) |

3.2. Extensions

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

Bibliography

Normative References

Informative References

[Wikipedia] *Wikipedia: List of device bandwidths.* http://en.wikipedia.org/wiki/List_of_device_bandwidths.

4. OCCI HTTP Binding

2009-10-07

4.1. Specification

46. The HTTP binding is the default binding for OCCI:

- The HTTP binding is defined by RFC2616 (HTTP).
- Web Categories [CATEGORY] and Web Linking [LINK] specifications are used for the meta-model.
- Server-side cookies ("Attributes") are used for name-value pairs.
- Collections are transferred as the `text/uri-list` content type (defined in RFC 2169, Appendix A).

47. In all cases the process defined in RFC2965 is used to `set/get` message-headers except `[Set-]Attribute:`, `[Set-]Category:` and `[Set-]Link:` are used in place of `Cookie:` and `Set-Cookie:`.

48. `Set-*` headers may be included on PUT or POST requests (including empty POSTs in order to update the metadata independently of the representation).

4.2. Example

4.2.1. Request

```
POST /compute/123 HTTP/1.1
```

```
Host: example.com
Content-Length: 0
Set-Attribute: id="urn:uuid:d0e9f0d0-f62d-4f28-bc90-23b0bd871770"
Set-Category: compute;
  scheme="http://purl.org/occi/kind/";
  label="Compute Resource"
Set-Link: <http://example.com/products/1234>;
  rel="alternate";
  title="Alternate representation"
```

4.2.2. Response

```
Attribute: id="urn:uuid:d0e9f0d0-f62d-4f28-bc90-23b0bd871770"
Attribute: title="Compute Resource #123"
Attribute: summary="A virtual compute resource"
Attribute: updated="2009-12-31T12:59:59Z"
Attribute: compute.cores=2
Attribute: compute.speed=3000
Attribute: compute.memory=2048
ETag: "dad86c61eea237932f"
Category: compute;
  scheme="http://purl.org/occi/kind/";
  label="Compute Resource"
Link: <http://example.com/products/1234>;
  rel="alternate";
  title="Alternate representation"

<?xml version="1.0" encoding="UTF-8"?>
<ovf:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ovf="http://schemas.dmtf.org/ovf/1/envelope"
<!-- snip -->
```

Bibliography

Normative References

[RFC2616] *RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1*. <http://tools.ietf.org/html/rfc2616>. Internet Engineering Task Force (IETF) 1999-06.

[RFC2965] *RFC 2965 - HTTP State Management Mechanism*. <http://tools.ietf.org/html/rfc2965> [<http://tools.ietf.org/html/rfc2822>]. Internet Engineering Task Force (IETF) 2000-10.

Informative References

[CATEGORY] *Web Categories*. <http://tools.ietf.org/html/draft-johnston-http-category-header>. Internet Engineering Task Force (IETF) Sam Johnston. 2009-07-1.

[LINK] *Web Linking*. <http://tools.ietf.org/html/draft-nottingham-http-link-header>. Internet Engineering Task Force (IETF) Mark Nottingham. 2009-07-12.

[HTML5-article] *Designing a great HTTP API - why heavyweight XML is not the answer*. <http://www.elastichosts.com/blog/2009/01/01/designing-a-great-http-api/> [<http://www.smashingmagazine.com/2009/07/29/misunderstanding-markup-xhtml-2-comic-strip/>]. . 2009-01-01.

5. Contributors

49. Next to the members of the OCCI working group the following people actively contributed to this document:

Table 8. List of contributors

| Name | Affiliation | Contact |
|-----------------------|--------------------------------------|---------------------------|
| Andy Edmonds | Intel - SLA@SOI project | andrewx.edmonds@intel.com |
| Sam Johnston (Editor) | Australian Online Solutions | samj@samj.net |
| Gary Mazzaferro | OCCI Counselour - Exxia, Inc. | garymazzaferro@gmail.com |
| Thijs Metsch (Editor) | Sun Microsystems - RESERVOIR project | thijs.metsch@sun.com |
| Andre Merzky | Lousiana State University | andre@merzky.net |

6. OCCI Legal Notices

6.1. Intellectual Property Statement

50. The OGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the OGF Secretariat.

51. The OGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the OGF Executive Director.

6.2. Disclaimer

52. This document and the information contained herein is provided on an "As Is" basis and the OGF disclaims all warranties, express or implied, including but not limited to any warranty that the use of the information herein will not infringe any rights or any implied warranties of merchantability or fitness for a particular purpose.

6.3. Full Copyright Notice

53. Copyright (C) Open Grid Forum (2009). All Rights Reserved.

54. This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the OGF or other organizations, except as needed for the purpose of developing Grid Recommendations in which case the procedures for copyrights defined in the OGF Document process must be followed, or as required to translate it into languages other than English.

55. The limited permissions granted above are perpetual and will not be revoked by the OGF or its successors or assignees.