GFD-R-P.225
OCCI-WG

Ralf Nyrén, Independent
Florian Feldhaus, Independent
Boris Parák, CESNET
Zdeněk Šustr, CESNET

February 25, 2011
Updated: September 5, 2016

# Open Cloud Computing Interface – JSON Rendering

Status of this Document

This document provides information to the community regarding the specification of the Open Cloud Computing Interface. Distribution is unlimited.



Abstract

This document, part of a document series produced by the OCCI working group within the Open Grid Forum (OGF), provides a high-level definition of a Protocol and API. The document is based upon previously gathered requirements and focuses on the scope of important capabilities required to support modern service offerings.

# Contents

**1 Introduction** **3**

**2 Notational Conventions** **3**

**3 OCCI JSON Rendering** **4**

3.1 Entity Instance Rendering . . . 4

3.1.1 Resource Instance Rendering . . . 4

3.1.2 Link Instance Rendering . . . 5

3.2 Category Instance Rendering . . . 5

3.2.1 Kind Instance Rendering . . . 6

3.2.2 Mixin Instance Rendering . . . 6

3.2.3 Action Instance Rendering . . . 7

3.3 Entity Collection Rendering . . . 7

3.3.1 Resource Collection Rendering . . . 7

3.3.2 Link Collection Rendering . . . 7

3.4 Category Collection Rendering . . . 7

3.4.1 Kind Collection Rendering . . . 8

3.4.2 Mixin Collection Rendering . . . 8

3.4.3 Action Collection Rendering . . . 8

3.5 Attributes Rendering . . . 8

3.5.1 Attribute Description Rendering . . . 8

**4 Security Considerations** **9**

**5 Glossary** **10**

**6 Contributors** **10**

**7 Intellectual Property Statement** **11**

**8 Disclaimer** **11**

**9 Full Copyright Notice** **11**

**A JSON Rendering Examples** **13**

A.1 Resouce Instance Example . . . 13

A.2 Action Invocation Example . . . 13

A.3 Link Instance Example . . . 14

A.4 Kind Instance Example . . . 14

A.5 Mixin Instance Example . . . 15

A.6 Action Instance Example . . . 15

**B OCCI JSON Schema** **15**

# 1 Introduction

The Open Cloud Computing Interface (OCCI) is a RESTful Protocol and API for all kinds of management tasks. OCCI was originally initiated to create a remote management API for IaaS[1] model-based services, allowing for the development of interoperable tools for common tasks including deployment, autonomic scaling and monitoring. It has since evolved into a flexible API with a strong focus on interoperability while still offering a high degree of extensibility. The current release of the Open Cloud Computing Interface is suitable to serve many other models in addition to IaaS, including PaaS and SaaS.

In order to be modular and extensible the current OCCI specification is released as a suite of complementary documents, which together form the complete specification. The documents are divided into four categories consisting of the OCCI Core, the OCCI Protocols, the OCCI Renderings and the OCCI Extensions.

- The OCCI Core specification consists of a single document defining the OCCI Core Model. OCCI interaction occurs through *renderings* (including associated behaviors) and is expandable through *extensions*.
- The OCCI Protocol specifications consist of multiple documents, each describing how the model can be interacted with over a particular protocol (e.g. HTTP, AMQP, etc.). Multiple protocols can interact with the same instance of the OCCI Core Model.
- The OCCI Rendering specifications consist of multiple documents, each describing a particular rendering of the OCCI Core Model. Multiple renderings can interact with the same instance of the OCCI Core Model and will automatically support any additions to the model which follow the extension rules defined in OCCI Core.
- The OCCI Extension specifications consist of multiple documents, each describing a particular extension of the OCCI Core Model. The extension documents describe additions to the OCCI Core Model defined within the OCCI specification suite.

The current specification consists of seven documents. This specification describes version 1.2 of OCCI and is backward compatible with 1.1. Future releases of OCCI may include additional protocol, rendering and extension specifications. The specifications to be implemented (MUST, SHOULD, MAY) are detailed in the table below.

**Table 1.** What OCCI specifications must be implemented for the specific version.

| Document | OCCI 1.1 | OCCI 1.2 |
|---|---|---|
| Core Model | MUST | MUST |
| Infrastructure Model | SHOULD | SHOULD |
| Platform Model | MAY | MAY |
| SLA Model | MAY | MAY |
| HTTP Protocol | MUST | MUST |
| Text Rendering | MUST | MUST |
| JSON Rendering | MAY | MUST |

# 2 Notational Conventions

All these parts and the information within are mandatory for implementors (unless otherwise specified). The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [1].

[1] Infrastructure as a Service

# 3 OCCI JSON Rendering

The OCCI JSON Rendering specifies a rendering of OCCI instance types in the JSON data interchange format as defined in [2].

The rendering can be used to render OCCI instances independently of the protocol being used. Thus messages can be delivered by, e.g., the HTTP protocol as specified in [3].

The following media-type MUST be used for the OCCI JSON Rendering:

`application/occi+json`

The OCCI JSON Rendering consists of a JSON object containing information on the OCCI Core instances OCCI Kind, OCCI Mixin, OCCI Action, OCCI Link and OCCI Resource. The rendering also include a JSON object to invoke the operation identified by OCCI Actions. The rendering of each OCCI Core instance will be described in the following sections.

## 3.1 Entity Instance Rendering

Entity instances MUST be rendered as JSON hashmaps.

### 3.1.1 Resource Instance Rendering

The OCCI Resource Instance Rendering consists of a JSON object as shown in the following declaration. Appendix A.1 contains a detailed example. Table 2 defines the object members.

```
{
    "kind": String,
    "mixins": Array,
    "attributes": Object,
    "actions": Array,
    "id": String,
    "links": Array,
    "summary": String,
    "title": String,
}
```

**Table 2.** OCCI Resource instance rendered with the following entries:

| Object member | JSON type | Mutability | Multiplicity | Description |
|---|---|---|---|---|
| kind | String | immutable | 1 | Type identifier. |
| mixins | Array of Strings | mutable | 0..1 | List of type identifiers of associated OCCI Mixins. |
| attributes | Object | mutable | 0..1 | Instance Attributes (see 3.5.1). |
| actions | Array of Strings | mutable | 0..1 | List of type identifiers of OCCI Actions applicable to the OCCI Resource instance. |
| id | String | immutable | 1 | ID of the OCCI Resource. |
| links | Array of Objects | mutable | 0..1 | List of OCCI Links (fully rendered instances, see 3.1.2). |
| summary | String | mutable | 0..1 | Summary text of resource. |
| title | String | mutable | 0..1 | Title of resource. |

**3.1.1.1 Action Invocation Rendering** The OCCI Action Invocation Rendering identifies an invocable operation on a OCCI Resource or OCCI Link instance. To trigger such an operation the OCCI Action Invocation Rendering is required.

The OCCI Action Invocation Rendering consists of a top-level JSON object as shown in the following declaration. Appendix A.2 contains a detailed example. Table 3 defines the object members.

```
{
    "action": String,
    "attributes": Object
}
```

**Table 3.** An OCCI Action invocation is rendered with the following entries:

| Object member | JSON type | Mutability | Multiplicity | Description |
|---|---|---|---|---|
| action | String | immutable | 1 | Type identifier. |
| attributes | Object | mutable | 0..1 | Instance attributes (see 3.5.1). |

### 3.1.2 Link Instance Rendering

The OCCI Link Instance Rendering consists of a JSON object as shown in the following declaration. Appendix A.3 contains a detailed example. Table 4 defines the object members.

```
{
    "kind": String,
    "mixins": Array,
    "attributes": Object,
    "actions": Array,
    "id": String,
    "source": Object,
    "target": Object,
    "title": String
}
```

**Table 4.** OCCI Link instances are rendered with the following entries:

| Object member | JSON type | Mutability | Multiplicity | Description |
|---|---|---|---|---|
| kind | String | immutable | 1 | Type identifier. |
| mixins | Array of Strings | mutable | 0..1 | List of type identifiers of associated OCCI Mixins. |
| attributes | Object | mutable | 0..1 | Instance attributes (see 3.5.1). |
| actions | Array of Strings | mutable | 0..1 | List of type identifiers of OCCI Action Categories applicable to the OCCI Link instance. |
| id | String | immutable | 1 | ID of the OCCI Link. |
| source | Object | immutable | 1 | Hashmap of the link source (see 3.1.2.1). |
| target | Object | immutable | 1 | Hashmap of the link target (see 3.1.2.1). |
| title | String | mutable | 0..1 | Title of the Link |

#### 3.1.2.1 Link Instance Source/Target Rendering

The OCCI Link Instance Source/Target Rendering consists of a JSON object as shown in the following declaration. Appendix A.3 contains a detailed example. Table 5 defines the object members. `location` maps to OCCI Core's `source` and `target` model attributes and `kind` maps to OCCI Core's `target.kind` model attribute. The value of `kind` for `source` is implied by OCCI Core's model attribute value for `source`.

```
{
    "location": String,
    "kind": String
}
```

**Table 5.** OCCI Link sources/targets are rendered with the following entries:

| Object member | JSON type | Mutability | Multiplicity | Description |
|---|---|---|---|---|
| location | String | immutable | 1 | URI of the link target/source. |
| kind | String | immutable | 0..1 | Kind identifier, supplied if location points to an OCCI Resource. |

## 3.2 Category Instance Rendering

Category instances MUST be rendered as JSON hashmaps.

### 3.2.1 Kind Instance Rendering

The OCCI Kind Instance Rendering consists of a JSON object as shown in the following declaration. Appendix A.4 contains a detailed example. Table 6 defines the top-level object members.

**Table 6.** OCCI Kind instances are rendered with the following entries:

| Object member | JSON type | Mutability | Multiplicity | Description |
|---|---|---|---|---|
| term | String | immutable | 1 | Unique identifier within the categorization scheme. |
| scheme | String | immutable | 1 | Categorization scheme. |
| title | String | immutable | 0..1 | Title of the OCCI Kind. |
| attributes | Object | immutable | 0..1 | Attribute description, see 9. |
| parent | String | immutable | 0..1 | OCCI Kind type identifier of the related "parent" Kind instance. |
| actions | Array of Strings | immutable | 0..1 | List of OCCI Action type identifiers. |
| location | String | immutable | 0..1 | Transport protocol specific URI bound to the OCCI Kind instance. MUST be supplied for the OCCI Kinds of all OCCI Entities except OCCI Entity itself. |

```
{
    "term": String,
    "scheme": String,
    "title": String,
    "attributes": Object,
    "actions": Array,
    "parent": String,
    "location": String
}
```

### 3.2.2 Mixin Instance Rendering

The OCCI Mixin Instance Rendering consists of a JSON object as shown in the following declaration. Appendix A.5 contains a detailed example. Table 7 defines the top-level object members.

```
{
    "term": String,
    "scheme": String,
    "title": String,
    "attributes": Object,
    "actions": Array,
    "depends": Array,
    "applies": Array,
    "location": String
}
```

**Table 7.** OCCI Mixin instances are rendered with the following entries:

| Object member | JSON type | Mutability | Multiplicity | Description |
|---|---|---|---|---|
| term | String | immutable | 1 | Unique identifier within the categorization scheme. |
| scheme | String | immutable | 1 | Categorization scheme. |
| title | String | immutable | 0..1 | Title of the OCCI Mixin. |
| attributes | Object | immutable | 0..1 | Attribute description, see 9. |
| depends | Array of Strings | immutable | 0..1 | List of type identifiers of the dependent Mixin instances. |
| applies | Array of Strings | immutable | 0..1 | List of OCCI Kind type identifiers this OCCI Mixin can be applied to. |
| actions | Array of Strings | immutable | 0..1 | List of OCCI Action type identifiers. |
| location | String | immutable | 1 | Transport protocol specific URI bound to the OCCI Mixin instance. |

### 3.2.3 Action Instance Rendering

The OCCI Action Instance Rendering consists of a JSON object as shown in the following declaration. Appendix A.6 contains a detailed example. Table 8 defines the top-level object members.

**Table 8.** OCCI Actions are rendered inside the top-level JSON object with name *actions* as an array of JSON Objects with the following entries:

| Object member | JSON type | Mutability | Multiplicity | Description |
|---|---|---|---|---|
| term | String | immutable | 1 | Unique type identifier within the categorization scheme. |
| scheme | String | immutable | 1 | Categorization scheme. |
| title | String | immutable | 0..1 | Title of the OCCI Action. |
| attributes | Object | immutable | 0..1 | Attribute description, see 9. |

```
{
    "term": String,
    "scheme": String,
    "title": String,
    "attributes": Object,
}
```

## 3.3 Entity Collection Rendering

Collections of Entity instances MUST be rendered as JSON arrays. The content of that array is a set of entity instance renderings.

That array MUST be a member of a JSON hashmap that is associated with the relevant key name specific to the type of Entity collection being rendered.

### 3.3.1 Resource Collection Rendering

The JSON hashmap key-name associated with the array of resource instances MUST be resources.

```
{
    "resources": []
}
```

### 3.3.2 Link Collection Rendering

The JSON hashmap key-name associated with the array of link instances MUST be links.

```
{
    "links": []
}
```

## 3.4 Category Collection Rendering

Collections of Category instances MUST be rendered as JSON arrays. The content of that array is a set of Category instance renderings.

That array MUST be a member of a JSON hashmap that is associated with the relevant key name specific to the type of Category collection being rendered.

### 3.4.1 Kind Collection Rendering

The JSON hashmap key-name associated with the array of kind instances MUST be kinds.

```
{
    "kinds": []
}
```

### 3.4.2 Mixin Collection Rendering

The JSON hashmap key-name associated with the array of mixin instances MUST be mixins.

```
{
    "mixins": []
}
```

### 3.4.3 Action Collection Rendering

The JSON hashmap key-name associated with the array of action instances MUST be actions.

```
{
    "actions": []
}
```

Collections of Category instances are rendered as JSON arrays.

## 3.5 Attributes Rendering

Attribute names consist of alphanumeric characters separated by dots. The dots define a logical namespace-like hierarchy. This hierarchy is NOT reflected in JSON objects. As shown in the following declaration, the attribute name is an opaque identifier rendered as hashmap *key*. The hashmap *value* contains either a Number, a String, a Boolean, an Array or an Object (as an attribute value or an attribute description, following the Attribute Description Rendering, see 3.5.1).

```
{
    "one.two.three": Number|String|Boolean|Array|Object,
    "one.two.four" : Number|String|Boolean|Array|Object
}
```

For examples of rendered Attributes please refer for instance to the Resource instance rendering example in Appendix A.1.

### 3.5.1 Attribute Description Rendering

Attribute Descriptions are rendered as JSON objects as defined in table 9

```
{
    "mutable": Boolean,
    "required": Boolean,
    "type": String,
    "pattern": Object,
    "default": String | Number | Boolean | Array | Object,
    "description": String
}
```

For examples of rendered Attribute Descriptions please refer, e.g., to the Kind rendering example in Appendix A.4.

**Table 9.** All properties of the Attribute definition are optional, but may contain defaults which MUST be used if the Attribute is not present in the instantiated OCCI Entity.

| Object member | JSON type | Default | Description |
|---|---|---|---|
| mutable | Boolean | false | Defines if the Attribute is mutable after initialization. |
| required | Boolean | false | Defines if the Attribute MUST be specified at instantiation of the OCCI Entity. |
| type | String | string | Type of the Attribute. MUST be either "string", "number", "boolean", "array" or "object". |
| pattern | Object | | JSON Schema [4] to validate the value of the attribute. It is recommended to specify the `$schema` property for the schema used. |
| default | String, Number, Boolean, Array, Object | `(none)` | Attribute default. MUST be the same type as defined in the type property and MUST be used if the Attribute is not present in the instantiated OCCI Entity. |
| description | String | `(none)` | Description of the attribute. |

# 4 Security Considerations

OCCI does not require that an authentication mechanism be used nor does it require that client to service communications are secured. It does RECOMMEND that an authentication mechanism be used and that where appropriate, communications are encrypted using HTTP over TLS. The authentication mechanisms that MAY be used with OCCI are those that can be used with HTTP and TLS. For further discussion see the appropiate section in [3].

# 5 Glossary

| Term | Description |
|---|---|
| Action | An OCCI base type. Represents an invocable operation on an Entity sub-type instance or collection thereof. |
| Attribute | A type in the OCCI Core Model. Describes the name and properties of attributes found in Entity types. |
| Category | A type in the OCCI Core Model and the basis of the OCCI type identification mechanism. The parent type of Kind. |
| capabilities | In the context of Entity sub-types **capabilities** refer to the Attributes and Actions exposed by an **entity instance**. |
| Collection | A set of Entity sub-type instances all associated to a particular Kind or Mixin instance. |
| Entity | An OCCI base type. The parent type of Resource and Link. |
| entity instance | An instance of a sub-type of Entity but not an instance of the Entity type itself. The OCCI model defines two sub-types of Entity: the Resource type and the Link type. However, the term *entity instance* is defined to include any instance of a sub-type of Resource or Link as well. |
| Kind | A type in the OCCI Core Model. A core component of the OCCI classification system. |
| Link | An OCCI base type. A Link instance associates one Resource instance with another. |
| Mixin | A type in the OCCI Core Model. A core component of the OCCI classification system. |
| mix-in | An instance of the Mixin type associated with an *entity instance*. The "mix-in" concept as used by OCCI *only* applies to instances, never to Entity types. |
| OCCI | Open Cloud Computing Interface. |
| OGF | Open Grid Forum. |
| Resource | An OCCI base type. The parent type for all domain-specific Resource sub-types. |
| resource instance | See *entity instance*. This term is considered obsolete. |
| tag | A Mixin instance with no attributes or actions defined. Used for taxonomic organisation of entity instances. |
| template | A Mixin instance which if associated at instance creation-time pre-populate certain attributes. |
| type | One of the types defined by the OCCI Core Model. The Core Model types are Category, Attribute, Kind, Mixin, Action, Entity, Resource and Link. |
| concrete type/sub-type | A concrete type/sub-type is a type that can be instantiated. |
| URI | Uniform Resource Identifier. |
| URL | Uniform Resource Locator. |
| URN | Uniform Resource Name. |

# 6 Contributors

We would like to thank the following people who contributed to this document:

| Name | Affiliation | Contact |
|---|---|---|
| Michael Behrens | R2AD | behrens.cloud at r2ad.com |
| Mark Carlson | Toshiba | mark at carlson.net |
| Augusto Ciuffoletti | University of Pisa | augusto.ciuffoletti at gmail.com |
| Andy Edmonds | ICCLab, ZHAW | edmo at zhaw.ch |
| Sam Johnston | Google | samj at samj.net |
| Gary Mazzaferro | Independent | garymazzaferro at gmail.com |
| Thijs Metsch | Intel | thijs.metsch at intel.com |
| Ralf Nyrén | Independent | ralf at nyren.net |
| Alexander Papaspyrou | Adesso | alexander at papaspyrou.name |
| Boris Parák | CESNET | parak at cesnet.cz |
| Alexis Richardson | Weaveworks | alexis.richardson at gmail.com |
| Shlomo Swidler | Orchestratus | shlomo.swidler at orchestratus.com |
| Florian Feldhaus | Independent | florian.feldhaus at gmail.com |
| Zdeněk Šustr | CESNET | zdenek.sustr at cesnet.cz |
| Jean Parpaillon | Inria | jean.parpaillon at inria.fr |
| Philippe Merle | Inria | philippe.merle@inria.fr |


Next to these individual contributions we value the contributions from the OCCI working group.

# 7 Intellectual Property Statement

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.

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.

# 8 Disclaimer

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.

# 9 Full Copyright Notice

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

# References

[1] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119 (Best Current Practice), Internet Engineering Task Force, Mar. 1997. [Online]. Available: http://www.ietf.org/rfc/rfc2119.txt

[2] D. Crockford, "The application/json Media Type for JavaScript Object Notation (JSON)," RFC 4627 (Informational), Internet Engineering Task Force, Jul. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4627.txt

[3] R. Nyren, T. Metsch, and A. Edmonds, "Open Cloud Computing Interface – HTTP Protocol," Open Grid Forum, September 2016. [Online]. Available: https://www.ogf.org/documents/GFD.223.pdf

[4] F. Galiegue, K. Zyp, and G. Court, "Json schema: core definitions and terminology," draft-zyp-json-schema-04, Internet Engineering Task Force, Jan. 2013. [Online]. Available: http://tools.ietf.org/html/draft-zyp-json-schema-04

# A JSON Rendering Examples

## A.1 Resouce Instance Example

The following is an example of a rendered `compute` resource instance as specified in section 3.1.1.

```
{
    "kind": "http://schemas.ogf.org/occi/infrastructure#compute",
    "mixins": [
        "http://example.com/occi/infrastructure/os_tpl#debian9",
        "http://example.com/occi/infrastructure/resource_tpl#medium"
    ],
    "attributes": {
        "occi.compute.speed": 2,
        "occi.compute.memory": 4,
        "occi.compute.cores": 2,
        "com.example.occi.templates.myosmixin": {
            "mykey": "myvalue"
        }
    },
    "actions": [
        "http://schemas.ogf.org/occi/infrastructure/compute/action#start"
    ],
    "id": "urn:uuid:996ad860-2a9a-504f-8861-aeafd0b2ae29",
    "links": [
        {
            "kind":"http://schemas.ogf.org/occi/infrastructure#networkinterface",
            "mixins":[
                "http://schemas.ogf.org/occi/infrastructure/networkinterface#ipnetworkinterface"
            ],
            "attributes":{
                "occi.infrastructure.networkinterface.interface":"eth0",
                "occi.infrastructure.networkinterface.mac":"00:80:41:ae:fd:7e",
                "occi.infrastructure.networkinterface.address":"192.168.0.100",
                "occi.infrastructure.networkinterface.gateway":"192.168.0.1",
                "occi.infrastructure.networkinterface.allocation":"dynamic"
            },
            "actions":[
                "http://schemas.ogf.org/occi/infrastructure/networkinterface/action#up",
                "http://schemas.ogf.org/occi/infrastructure/networkinterface/action#down"
            ],
            "id":"urn:uuid:22fe83ae-a20f-54fc-b436-cec85c94c5e8",
            "target": {
                "location:"/network/b7d55bf4-7057-5113-85c8-141871bf7635",
                "kind":"http://schemas.ogf.org/occi/infrastructure#network"
            },
            "source": {
                "location":"/compute/996ad860-2a9a-504f-8861-aeafd0b2ae29",
                "kind":"http://schemas.ogf.org/occi/infrastructure#compute"
            }
        }
    ]
}
```

## A.2 Action Invocation Example

The following is an example of a rendered `stop` action invocation as specified in section 3.1.1.1.

```
{
    "action":"http://schemas.ogf.org/occi/infrastructure/compute/action#stop",
    "attributes":{
        "method":"graceful"
    }
}
```

## A.3 Link Instance Example

The following is an example of a rendered `networkinterface` link as specified in section 3.1.2.

```
{
    "kind":"http://schemas.ogf.org/occi/infrastructure#networkinterface",
    "mixins":[
        "http://schemas.ogf.org/occi/infrastructure/networkinterface#ipnetworkinterface"
    ],
    "attributes":{
        "occi.infrastructure.networkinterface.interface":"eth0",
        "occi.infrastructure.networkinterface.mac":"00:80:41:ae:fd:7e",
        "occi.infrastructure.networkinterface.address":"192.168.0.100",
        "occi.infrastructure.networkinterface.gateway":"192.168.0.1",
        "occi.infrastructure.networkinterface.allocation":"dynamic"
    },
    "actions":[
        "http://schemas.ogf.org/occi/infrastructure/networkinterface/action#up",
        "http://schemas.ogf.org/occi/infrastructure/networkinterface/action#down"
    ],
    "id":"urn:uuid:22fe83ae-a20f-54fc-b436-cec85c94c5e8",
    "target": {
        "location:"/network/b7d55bf4-7057-5113-85c8-141871bf7635",
        "kind":"http://schemas.ogf.org/occi/infrastructure#network"
    },
    "source": {
        "location":"/compute/996ad860-2a9a-504f-8861-aeafd0b2ae29",
        "kind":"http://schemas.ogf.org/occi/infrastructure#compute"
    }
}
```

## A.4 Kind Instance Example

The following is an example of a rendered Kind instance as specified in section 3.2.1.

```
{
    "term":"compute",
    "scheme":"http://schemas.ogf.org/occi/infrastructure#",
    "title":"ComputeResource",
    "parent":"http://schemas.ogf.org/occi/core#resource",
    "attributes":{
        "occi.compute.hostname":{
            "mutable":true,
            "required":false,
            "type":"string",
            "description":"Hostname of the compute resource"
            "pattern": {
                "$schema": "http://json-schema.org/draft-04/schema#",
                "type": "string",
                "pattern": "\S+"
            }
        },
        "occi.compute.state":{
            "mutable":false,
            "required":false,
            "type":"string",
            "default":"inactive",
            "description":"State the compute resource is in"
        }
    },
    "actions":[
        "http://schemas.ogf.org/occi/infrastructure/compute/action#start",
        "http://schemas.ogf.org/occi/infrastructure/compute/action#stop",
        "http://schemas.ogf.org/occi/infrastructure/compute/action#restart",
        "http://schemas.ogf.org/occi/infrastructure/compute/action#suspend"
    ],
    "location":"/compute/"
}
```

## A.5 Mixin Instance Example

The following is an example of a rendered `medium` Resource Template Mixin as specified in section 3.2.2.

```
{
    "term":"medium",
    "scheme":"http://example.com/template/resource#",
    "depends":[
        "http://schemas.ogf.org/occi/infrastructure#resourcetpl"
    ],
    "applies":[
        "http://schemas.ogf.org/occi/infrastructure#compute"
    ],
    "attributes":{
        "occi.compute.speed":{
            "type":"number",
            "default":2.8
        }
    },
    "title":"MediumVM",
    "location":"/template/resource/medium/"
}
```

## A.6 Action Instance Example

The following is an example of a rendered `stop` Action instance as specified in section 3.2.3.

```
{
    "term":"stop",
    "scheme":"http://schemas.ogf.org/occi/infrastructure/compute/action#",
    "title":"StopComputeinstance",
    "attributes":{
        "method":{
            "mutable":true,
            "required":false,
            "type":"string",
            "default":"poweroff"
        }
    }
}
```

# B OCCI JSON Schema

The JSON schema provided below validates any valid OCCI message courtesy of the `anyOf` construct below. Sub-schemas or fragments need to be used to validate specific OCCI classes.

```
{
    "id": "http://schemas.ogf.org/occi/OCCI-schema.json",
    "$schema": "http://json-schema.org/draft-04/schema#",
    "title": "OCCI v. 1.2 JSON Rendering Schema",
    "definitions": {

        "array_of_strings": {
            "type": "array",
            "items": { "type": "string" }
        },

        "kinded_uri": {
            "id": "#kinded_uri",
            "type": "object",
            "required": ["location"],
            "additionalProperties": false,
            "properties": {
                "location": { "type": "string" },
                "kind": { "type": "string" }
            }
        },

        "resource": {
            "id": "#resource",
            "type": "object",
            "required": ["kind", "id"],
            "additionalProperties": false,
            "properties": {
                "kind": { "type": "string" },
                "mixins": { "$ref": "#/definitions/array_of_strings" },
```

```
480                             "attributes": { "$ref": "#/definitions/attributes" },
481                             "actions": { "$ref": "#/definitions/array_of_strings" },
482                             "id": { "type": "string" },
483                             "links": {
484                                     "type": "array",
485                                     "items": {
486                                             "$ref": "#definitions/link"
487                                     }
488                             },
489                             "summary": { "type": "string" },
490                             "title": { "type": "string" }
491                     }
492             },
493
494             "action_invocation": {
495                     "id": "#action_invocation",
496                     "type": "object",
497                     "required": ["action"],
498                     "additionalProperties": false,
499                     "properties": {
500                             "action": { "type": "string" },
501                             "attributes": { "$ref": "#/definitions/attributes" }
502                     }
503             },
504
505             "link": {
506                     "id": "#link",
507                     "type": "object",
508                     "required": ["kind", "id", "target", "source"],
509                     "additionalProperties": false,
510                     "properties": {
511                             "kind": { "type": "string" },
512                             "mixins": { "$ref": "#/definitions/array_of_strings" },
513                             "attributes": { "$ref": "#/definitions/attributes" },
514                             "actions": { "$ref": "#/definitions/array_of_strings" },
515                             "id": { "type": "string" },
516                             "source": { "$ref": "#/definitions/kinded_uri" },
517                             "target": { "$ref": "#/definitions/kinded_uri" },
518                             "rel": { "type": "string" },
519                             "title": { "type": "string" }
520                     }
521             },
522
523             "kind": {
524                     "id": "#kind",
525                     "type": "object",
526                     "required": ["term", "scheme"],
527                     "additionalProperties": false,
528                     "properties": {
529                             "term": { "type": "string" },
530                             "scheme": { "type": "string" },
531                             "title": { "type": "string" },
532                             "attributes": { "$ref": "#/definitions/attribute_description" },
533                             "actions": { "$ref": "#/definitions/array_of_strings" },
534                             "parent": { "type": "string" },
535                             "location": { "type": "string" }
536                     }
537             },
538
539             "mixin": {
540                     "id": "#mixin",
541                     "type": "object",
542                     "required": ["term", "scheme", "location"],
543                     "additionalProperties": false,
544                     "properties": {
545                             "term": { "type": "string" },
546                             "scheme": { "type": "string" },
547                             "title": { "type": "string" },
548                             "attributes": { "$ref": "#/definitions/attribute_description" },
549                             "actions": { "$ref": "#/definitions/array_of_strings" },
550                             "depends": { "$ref": "#/definitions/array_of_strings" },
551                             "applies": { "$ref": "#/definitions/array_of_strings" },
552                             "location": { "type": "string" }
553                     }
554             },
555
556             "action": {
557                     "id": "#action",
558                     "type": "object",
559                     "required": ["term", "scheme"],
560                     "additionalProperties": false,
561                     "properties": {
562                             "term": { "type": "string" },
563                             "scheme": { "type": "string" },
564                             "title": { "type": "string" },
565                             "attributes": { "$ref": "#/definitions/attribute_description" }
566                     }
567             },
568
569             "resource_collection": {
570                     "id": "#resource_collection",
571                     "type": "object",
572                     "required": ["resources"],
573                     "additionalProperties": false,
574                     "properties": {
575                             "resources": {
576                                     "type": "array",
577                                     "items": {
578                                             "$ref": "#definitions/resource"
579                                     }
580                             }
581                     }
582             },
583
584             "link_collection": {
585                     "id": "#link_collection",
586                     "type": "object",
587                     "required": ["links"],
588                     "additionalProperties": false,
589                     "properties": {
590                             "links": {
```

```
                                        "type": "array",
                                        "items": {
                                                "$ref": "#definitions/link"
                                        }
                                }
                        }
                },

                "kind_collection": {
                        "id": "#kind_collection",
                        "type": "object",
                        "required": ["kinds"],
                        "additionalProperties": false,
                        "properties": {
                                "kinds": {
                                        "type": "array",
                                        "items": {
                                                "$ref": "#definitions/kind"
                                        }
                                }
                        }
                },

                "mixin_collection": {
                        "id": "#mixin_collection",
                        "type": "object",
                        "required": ["mixins"],
                        "additionalProperties": false,
                        "properties": {
                                "mixins": {
                                        "type": "array",
                                        "items": {
                                                "$ref": "#definitions/mixin"
                                        }
                                }
                        }
                },

                "action_collection": {
                        "id": "#action_collection",
                        "type": "object",
                        "required": ["actions"],
                        "additionalProperties": false,
                        "properties": {
                                "actions": {
                                        "type": "array",
                                        "items": {
                                                "$ref": "#definitions/action"
                                        }
                                }
                        }
                },

                "attributes": {
                        "id": "#attributes",
                        "type": "object",
                        "additionalProperties": {
                                "oneOf": [
                                        { "type": "number" },
                                        { "type": "boolean" },
                                        { "type": "string" },
                                        { "type": "object" },
                                        { "type": "array" }
                                ]
                        }
                },

                "attribute_description": {
                        "id": "#attribute_description",
                        "oneOf": [
                                {
                                        "type": "object",
                                        "additionalProperties": false,
                                        "minProperties": 1,
                                        "properties": {
                                                "mutable": { "type": "boolean" },
                                                "required": { "type": "boolean" },
                                                "type": { "type": "string" },
                                                "default": {
                                                        "oneOf": [
                                                                { "type": "number" },
                                                                { "type": "string" },
                                                                { "type": "boolean" }
                                                        ]
                                                },
                                                "description": { "type": "string" },
                                                "pattern": { "type": "object" }
                                        }
                                },
                                {
                                        "type": "object",
                                        "additionalProperties": false,
                                        "patternProperties": {
                                                ".+": {
                                                        "$ref": "#/definitions/attribute_description"
                                                }
                                        }
                                }
                        ]
                },

                "model": {
                        "id": "#model",
                        "type": "object",
                        "additionalProperties": false,
                        "properties": {
                                "resources": {
                                        "type": "array",
                                        "items": {
                                                "$ref": "#definitions/resource"
                                        }
```

```
                                },
                                "links": {
                                        "type": "array",
                                        "items": {
                                                "$ref": "#definitions/link"
                                        }
                                },
                                "mixins": {
                                        "type": "array",
                                        "items": {
                                                "$ref": "#definitions/mixin"
                                        }
                                },
                                "kinds": {
                                        "type": "array",
                                        "items": {
                                                "$ref": "#definitions/kind"
                                        }
                                },
                                "actions": {
                                        "type": "array",
                                        "items": {
                                                "$ref": "#definitions/action"
                                        }
                                }
                        }
                }
        },

        "anyOf": [
                { "$ref": "#/definitions/kind" },
                { "$ref": "#/definitions/mixin" },
                { "$ref": "#/definitions/action" },
                { "$ref": "#/definitions/attributes" },
                { "$ref": "#/definitions/attribute_description" },
                { "$ref": "#/definitions/kind_collection" },
                { "$ref": "#/definitions/mixin_collection" },
                { "$ref": "#/definitions/action_collection" },
                { "$ref": "#/definitions/resource_collection" },
                { "$ref": "#/definitions/link_collection" },
                { "$ref": "#/definitions/model" }
        ]
}
```