XXXXX
OCCI-WG

Augusto Ciuffoletti, Università di Pisa

September 22, 2014
Updated: September 25, 2014

Open Cloud Computing Interface - Monitoring Extension

Status of this Document

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

Copyright Notice

Trademarks

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

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 complimentary documents, which together form the complete specification. The documents are divided into three categories consisting of the OCCI Core, the OCCI Renderings and the OCCI Extensions.

- The OCCI Core specification consists of a single document defining the OCCI Core Model. The OCCI Core Model can be interacted with *renderings* (including associated behaviours) and expanded through *extensions*.
- 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. They do not require changes to the HTTP Rendering specifications as of this version of the specification.

The current specification consists of three documents. This specification describes version 1.1 of OCCI. Future releases of OCCI may include additional rendering and extension specifications. The documents of the current OCCI specification suite are:

OCCI Core describes the formal definition of the the OCCI Core Model [1].

OCCI HTTP Rendering defines how to interact with the OCCI Core Model using the RESTful OCCI API [2]. The document defines how the OCCI Core Model can be communicated and thus serialised using the HTTP protocol.

OCCI Infrastructure contains the definition of the OCCI Infrastructure extension for the IaaS domain [3]. The document extends the OCCI Core Model with additional Entity sub-types and their associated attributes and actions.

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 [4].

3 Motivations

The provider may want to give the user the tools to undertake actions based on the functional parameters of a resource performance: for instance, depending on how many jobs are running, on how much storage is available, or on the current state of a WAN link.

[1] Infrastructure as a Service

These actions, as well as the collected functional parameters, are primarily controlled by the provider, and so is the the way in which they are processed in order to implement the service.

This document defines an OCCI Resource subtype, the **Sensor**, that is in charge of implementing these kinds of activities.

4 OCCI monitoring

Resource monitoring is one component of the service the provider gives to the user: so it is naturally configured as an OCCI-resource. We define the *Sensor* as the Resource subtype for this purpose.

Among the distinguishing capabilities of a monitoring activity is its relationship with time: it may be limited in time, and exhibit real-time properties.

Other capabilities that are specific for a given monitoring service are added as mixins of the **Sensor**.

A **Sensor** instance is the target of *Notification* links from the resources it monitors.

Table 1. The immutable model attributes of the **Sensor** resource. The base URL **http://schemas.ogf.org/occi** has been replaced with <**schema**> in this table for a better reading experience.

Term	Scheme	Title	Attributes	Actions	Parent	
Sensor	<schema>/monitoring#	**Sensor** Resource	see Table 2	{}	{}	<schema>/core#Resource

Table 2. Attributes defined for the **Sensor** type.

Attribute	Type	Multi-plicity	Mutability	Description
occi.sensor.period	number	0..1	true	The time between two following measurements
occi.sensor.accuracy	number	1	false	of period or delay
occi.sensor.timebase	number	0..1	true	The server time when the timestart and timestop are modified
occi.sensor.timestart	number	0..1	true	The delay after which the session is planned to start
occi.sensor.timestop	number	0..1	true	The delay after which the session is planned to stop

5 Application notes and an example

The sensor is characterized mainly by timing attributes. The only atttribute that MUST be defined is the real-time *accuracy* of the monitoring activity. Other non mandatory attributes describe how frequently the monitoring activity is performed, and the time lapse during which the monitoring activity takes place.

The accuracy of the timing represents the maximum distance between an event, and its observation from the **Sensor**.

The **Sensor** gathers the functional parameters for its activity using data that is received from resources: therefore, the **Sensor** SHOULD be the target of *Notification* links, as defined in document [**?**].

The operation of the sensor is defined by mixins: the provider offers the user a choice of mixins, thus allowing the user to inspect the performance of the provision.

The provider uses the capabilities indicated in the mixins associated with a **Sensor** instance to implement the **notification** link and to instrument the Resource at the source of the **Notification** link.

For instance, consider the following simple example, where the mixin that specifies the operation of the **Sensor** is a simple tag:

> A provider offers a *3-out-of-k* mixin that can be associated to a **Sensor** Resource, that keeps in the *active* state 3 of the Compute resources from which it receives notifications.
>
> The user that wants to take advantage of this service instantiates a **Sensor** and associates with it a *3-out-of-k* mixin. Next it associated an **Notify** to each of the k Compute resources $C_{1..k}$. For each of them a **Notification** link N_i is instantiated that originates from C_i and targets the **Sensor**.

The schema is portable across any platform that offers the OCCI-infrastructure, OCCI-notification, and OCCI-monitoring, and that provides a *3-out-of-k* mixin.

Distinct providers may interoperate, for instance if one provides the *3-out-of-k* Resource and another the Compute resources, provided that an agreement exists between the two that allows cross provider information transfer.

6 Security issues

The OCCI Notification specification is an extension to the OCCI Core and Model specification [1]; thus the same security considerations as for the OCCI Core and Model specification apply here.

7 Glossary

Term	Description
Action	An OCCI base type. Represent an invocable operation on a 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 OCCI Attributes and OCCI Actions exposed by an **entity instance**.
Client	An OCCI client.
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 associate 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.
model attribute	An internal attribute of a the Core Model which is *not* client discoverable.
OCCI	Open Cloud Computing Interface.
OCCI base type	One of Entity, Resource, Link or Action.
OCCI Action	see Action.
OCCI Attribute	A client discoverable attribute identified by an instance of the Attribute type. Examples are occi.core.title and occi.core.summary.
OCCI Category	see Category.
OCCI Entity	see Entity.
OCCI Kind	see Kind.
OCCI Link	see Link.
OCCI Mixin	see Mixin.
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.
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.

8 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

9 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.

10 Full Copyright Notice

References

[1] R. Nyrén, A. Edmonds, A. Papaspyrou, and T. Metsch, "Open Cloud Computing Interface – Core," GFD-P-R.183, April 2011. [Online]. Available: http://ogf.org/documents/GFD.183.pdf

[2] T. Metsch and A. Edmonds, "Open Cloud Computing Interface – HTTP Rendering," GFD-P-R.185, April 2011. [Online]. Available: http://ogf.org/documents/GFD.185.pdf

[3] ——, "Open Cloud Computing Interface – Infrastructure," GFD-P-R.184, April 2011. [Online]. Available: http://ogf.org/documents/GFD.184.pdf

[4] 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