1	XXXXX
2	OCCI-WG
3	
4	
5	

## 6 Open Cloud Computing Interface - Notification Extension

- 7 Status of this Document
- 8 This document provides information to the community regarding the specification of the Open Cloud Com-
- <sup>9</sup> puting Interface. Distribution is unlimited.
- 10 Copyright Notice
- <sup>11</sup> Copyright © Open Grid Forum (2009-2011). All Rights Reserved.
- 12 <u>Trademarks</u>
- <sup>13</sup> OCCI is a trademark of the Open Grid Forum.
- 14 Abstract
- <sup>15</sup> This document, part of a document series, produced by the OCCI working group within the Open Grid Forum
- <sup>16</sup> (OGF), provides a high-level definition of a Protocol and API. The document is based upon previously gathered
- 17 requirements and focuses on the scope of important capabilities required to support modern service offerings.

18	18 Contents					
19	1	Introduction	3			
20	2	Notational conventions	3			
21	3	Motivations	3			
22	4	OCCI notification	3			
23		4.1 The <i>notifier</i> mixin	4			
24		4.2 The <i>notification</i> link	4			
25	5	Application notes and an example	4			
26	6	Security issues	6			
27	7	Glossary	7			
28	8	Intellectual Property Statement	7			
29	9	Disclaimer	8			
30	10	Full Copyright Notice	8			

## <sup>31</sup> 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<sup>1</sup> 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 through *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.
- <sup>51</sup> TODO: replace with 1.2, note backwards compatibility. define new set of docs for 1.2 below...

## 52 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].

## 57 **3** Motivations

<sup>58</sup> It is often the case that an entity changes during its lifetime: for instance a *Compute* resource experiences a <sup>59</sup> transient at the beginning of its lifetime during bootup [2].

 $_{60}$  We want to give the provider the tools to allow the user to define entities such that their changes are  $_{61}$  observable.

- 62 We introduce an OCCI Extension that allows the user
- to differentiate an OCCI resource that produces notifications, and
- to describe how such notifications are visible to other OCCI Resources.

## 65 4 OCCI notification

<sup>66</sup> The way to define a property of an OCCI entity is to associate a mixin to it. So the straightforward way to

<sup>67</sup> assert that an entity is one whose changes are observable is by associating a mixin that is related with this

- <sup>68</sup> property. We call this mixin *notifier*.
- <sup>69</sup> To define where a notification is directed the user instantiates a *notification* link.

<sup>1</sup>Infrastructure as a Service

### 70 **4.1** The notifier mixin

Table 1. The immutable model attributes of the *notifier* mixin. 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	Depends	Applies
notifier	<schema $>/$ notification $#$	notifier Mixin	{}	{}	{}	<schema>/core#Resource</schema>

71 The *mixin* instance assigned to the *notifier* type is http://schemas.ogf.org/occi/notification#notifier,

<sup>72</sup> as in table 1. The provider that supports the OCCI Notification extension MUST implement the *notifier* mixin

73 for each provided entity kind.

<sup>74</sup> There is no capability associated with the *notifier* mixin: it is a *tag*.

#### 75 4.2 The notification link

**Table 2.** The immutable model attributes of the *notification* kind. The base URL http://schemas.ogf.org/occi has been replaced with <schema> in this table for a better readability experience.

Term	Scheme	Title	Attributes	Actions	Parent
notification	<schema $>/$ notification $#$	notification Link	{}	{}	<schema>/core#Link</schema>

76 The kind instance assigned to the notification type is http://schemas.ogf.org/occi/notification#notification,

<sup>77</sup> as in table 2. The target of a *notification* is a generic *Resource*, while the source MUST be associated with <sup>78</sup> the *notifier* mixin.

<sup>79</sup> If the user requests the creation of a *notification* whose source is not associated with a *notifier* mixin, the <sup>80</sup> request MUST fail with an error.

<sup>81</sup> If the user removes the *notification* mixin associated with a *Resource*, all outgoing *notification* links MUST <sup>82</sup> be silently removed.

<sup>83</sup> There is no capability associated with the *notification*.

According with the core model [3], the user is able to discover all *notification* links that have their source in a given *Resource*.

According with the core model [3], the removal of the source of an *notification* link determines the removal of the link itself.

# **5** Application notes and an example

From the user perspective, the application of the *notifier* to a *Resource* corresponds to enabling the access to its state. This is significant mainly for *Resources* whose state changes in time.

The *notifier* mixin alone does not specify which changing aspect is in fact notified, and how: notably, not necessarily the attribute with STATE id, if one exists. When needed, additional mixins are used to address specific events.

94 From the provider perspective the association of a notifier mixin is reflected in the implementation of the

<sup>95</sup> functionalities needed to observe and render the change. How this is implemented depends on the *Resources* 

<sup>96</sup> that are targets of the *notification*.

<sup>97</sup> The way in which notifications are used falls ouside the scope of this document: as a general rule, they are for

management purposes. Such aspects can be defined by the user with mixins associated with the *notification* link, or in the *Resource* targeted by the *notification*.



Figure 1. Compute resources that notify their state to a control resource

<sup>100</sup> The notification extension is inappropriate to describe the planned and periodic measurement of operational <sup>101</sup> parameters of a resource for monitoring purposes: the monitoring extension [?] specifically addresses such

aspects, and may be coordinated with notification.

<sup>103</sup> The following example (see figure 1) illustrates a use case where the capabilities of the involved entities are <sup>104</sup> implicit in the *Kind* of the target resource.

<sup>105</sup> A provider offers a Management Resource of *Kind 3-out-of-k*, that keeps in the *active* state three <sup>106</sup> of the Compute resources from which it receives notifications.

The user that wants to take advantage of this service instantes a *3-out-of-k* Resource *3ook*, and associates a *notifier* 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 3ook.

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

From the point of view of cloud management, the existence of a *notifier* entails the activation of a process that is able to detect and notify the occurrence of relevant events. The existence of a *notification* corresponds to a kind of *subscribe* request: the occurrence of relevant events is notified to the target resource.

The cloud management, knowing the type of resource that is the target of the notification, is able to optimize and configure the production of events and the notification protocol. However, such details are kept deliberatly hidden to the user, that has limited capabilities to configure the notifications service, bound to the existence of specialized mixins.

The API is therefore mostly opaque to the user: this is a feature of the notification service that is introduced to improve its useability, performance and robustness. Whenever the user needs better control on the monitoring process, for instance to perform custom management activities, the monitoring extension is preferable.

<sup>122</sup> In figure 2 we give an idea of a possible deployment of a *300k* resource. The definition of the service attached

to the resource is not relevant: all we need to know is that it controls resources based on notifications from the resources themselves, in a closed control loop. On the left side we see the virtual resources: the 3-out-of-k

resource receives notifications in the form of XMPP messages from the *notify processes* running on the Virtual

<sup>126</sup> Machine. The creation of this infrastructure is governed by the presence of *notifier* mixins and *notification*.

<sup>127</sup> On the right there is the Cloud management architecture, that has control over the resources, in particular

to process the input messages of the *300k* implementation (inside the container labelled *User Management*)

<sup>129</sup> and to control the state of the virtual machines (inside the container labelled *Resource Management*).

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



Figure 2. Deploying a *3ook* resource

## 132 6 Security issues

<sup>133</sup> The OCCI Notification specification is an extension to the OCCI Core and Model specification [3]; thus the

<sup>134</sup> same security considerations as for the OCCI Core and Model specification apply here.

## 135 7 Glossary

Term	Description
Action	An OCCI base type. Represents an invocable operation on a Entity sub-type in-
	stance 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 <b>capabilities</b> refer to the OCCI Attributes and
	OCCI Actions exposed by an <b>entity instance</b> .
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
	sub-type. However, the term entity instance is defined to include any instance of a
Kind	A type in the OCCL Core Model. A core component of the OCCL classification
KIIIQ	system
Link	An OCCL base type A Link instance associates one Resource instance with another
Mixin	A type in the OCCL Core Model A core component of the OCCL classification
	system.
mix-in	An instance of the Mixin type associated with an <i>entity instance</i> . 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 <i>not</i> 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.
	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 it associated at instance creation-time pre-populate certain
t	All Thouses.
type	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
URI	Uniform Resource Locator
URN	Uniform Resource Name

137

136

# **138 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

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

<sup>145</sup> 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.

<sup>148</sup> Please address the information to the OGF Executive Director.

## 149 **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

<sup>154</sup> Copyright © Open Grid Forum (2009-2014). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment 155 on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in 156 whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph 157 are included on all such copies and derivative works. However, this document itself may not be modified in 158 any way, such as by removing the copyright notice or references to the OGF or other organizations, except 159 as needed for the purpose of developing Grid Recommendations in which case the procedures for copyrights 160 defined in the OGF Document process must be followed, or as required to translate it into languages other 161 than English. 162

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

### 165 **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

- <sup>169</sup> [2] T. Metsch and A. Edmonds, "Open Cloud Computing Interface Infrastructure," GFD-P-R.184, April <sup>170</sup> 2011. [Online]. Available: http://ogf.org/documents/GFD.184.pdf
- <sup>171</sup> [3] 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