

Open Cloud Computing Interface Service Level Agreements extension

Status of This Document

This document provides information to the community regarding the specification of the Open Cloud Computing Interface. The current status of the document is GWD (Group Working Draft). Distribution is unlimited.

Copyright Notice

Copyright © Open Grid Forum (2009-2014). All Rights Reserved.

Trademark

OCCI is a registered trademark and service mark of the Open Grid Forum.

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 in relation with the Service Level Agreements extension of the OCCI core model. The document is based upon previously gathered requirements and focuses on the scope of important capabilities required to support modern service offerings.

Contents

Abstract.....	1
Contents	2
1 Introduction.....	3
2 Notational Conventions	3
3 Service Level Agreements	3
3.1 Agreement	5
3.2 AgreementLink.....	9
3.3 Service Level Agreement example.....	9
4 Security Considerations	12
5 Glossary	12
6 Contributors.....	12
7 Intellectual Property Statement	13
8 Disclaimer.....	13
9 Full Copyright Notice	13
References	14

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 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 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 behaviors) and expanded through extensions
- The OCCI Rendering specifications consist of multiple documents each describing 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.

This OCCI SLA specification is an extension of OCCI Core Model and includes the definition of additional resource types, attributes and related actions.

2 Notational Conventions

The key words ‘MUST,’ ‘MUST NOT,’ ‘REQUIRED,’ ‘SHALL,’ ‘SHALL NOT,’ ‘SHOULD,’ ‘SHOULD NOT,’ ‘RECOMMENDED,’ ‘MAY,’ and ‘OPTIONAL’ are to be interpreted as described in RFC 2119[1].

3 Service Level Agreements

The OCCI Service Level Agreements (OCCI SLAs) document describes how the OCCI Core Model [2] can be extended and used to implement a Service Level Agreement management API. This API allows for the creation and management of resources related with the realization of agreements between an OCCI-enabled cloud service provider and potential consumers of the provider’s resources. The introduced types and Mixins defined in this OCCI SLAs document are the following:

Agreement This resource represents the Service Level Agreement between the provider and the consumer. It includes the basic information for this contract and with the appropriate extensions (Mixins) it can be populated with further information. To this end, we introduce the *AgreementTemplate* and the *AgreementTerms* Mixins which complement the SLAs with template tagging and terms specification respectively.

AgreementLink This is a link entity that associates an *Agreement* instance with any other Resource instance.

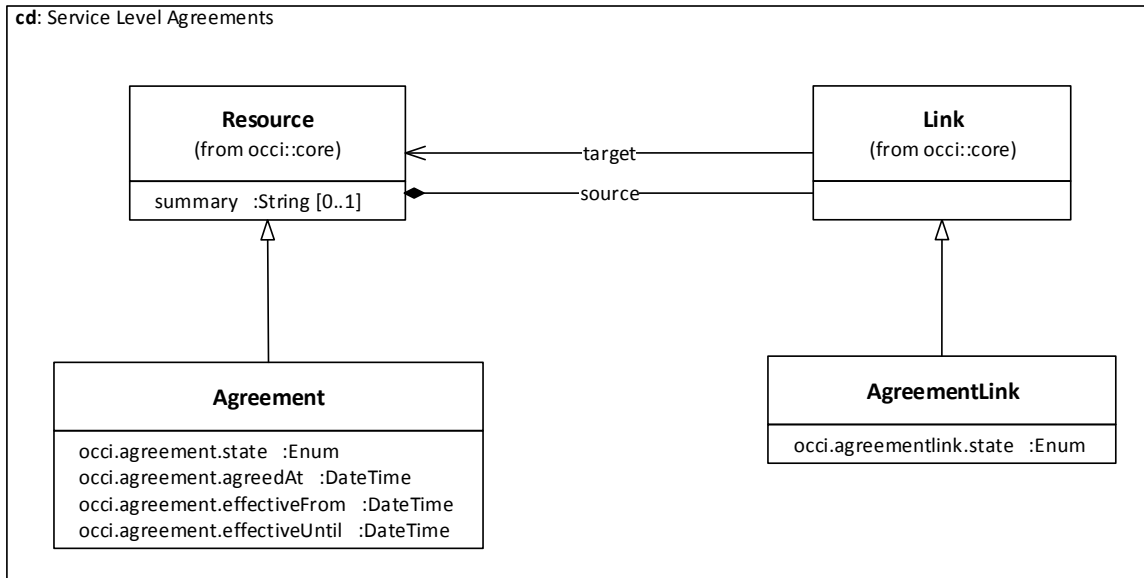


Figure 1. Overview diagram of OCCI Service Level Agreements types

It is REQUIRED by the OCCI Core Model specification that every type instantiated which is a sub-type of a Resource or a Link (i.e. *Agreement* and *AgreementLink*) MUST be assigned a Kind that identifies the instantiated type. To this end, each Kind instance MUST be related to the Resource or Link base type's Kind. That assigned Kind MUST be immutable to any client.

In the following table (Table 1) the Kind instances for the OCCI SLAs Resource, Link sub-types as well as the Mixins are introduced. For information on how to extend these types, please refer to the OCCI Core Model specification [2]. We also present related examples at the end of this document.

Table 1: The Kind instances defined for the SLAs sub-types of Resource, Link and related Mixins.

Term	Scheme ¹	Title	Related Kind
Agreement	<schema>/sla#	A Service Level Agreement	<schema>/core#resource
AgreementLink	<schema>/sla#	Link between a SLA and its associated resources	<schema>/core#link
Agreement_tpl	<schema>/sla#	A SLA template	-
Agreement_terms	<schema>/sla#	Term definition for an agreement	-

The following sections describe the *Agreement* and *AgreementLink* types, with details about their attributes, states and actions. The *AgreementTemplate* and *AgreementTerms* Mixins are also defined and presented. In the end, examples of OCCI SLAs instantiations are shown. These present several phases of the Service Level Agreement lifecycle, as well as specific instances of terms and service qualities.

¹ The base URL <http://schemas.ogf.org/occi> has been replaced with <schema> for better presentation of the table.

3.1 Agreement

The *Agreement* type represents a generic contract resource which holds the information related to a SLA between a cloud service consumer and a provider for the provisioned resources (e.g. compute, storage, network etc.). The Agreement type inherits the Resource base-type defined in the OCCI Core Model [2]. The Kind instance assigned to the *Agreement* type is <http://schemas.ogf.org/occi/sla#agreement>. An *Agreement* instance MUST relate and expose this Kind.

Table 2 describes the attributes defined by the *Agreement* type through its Kind instance. These attributes MUST be exposed by an instance of the *Agreement* type. In Figure 2 the allowed states of an Agreement instance are presented. Those specific states MUST be assigned to an *Agreement* instance by a cloud service provider SHOULD the implements the OCCI SLAs specification.

Table 2: Attributes for the Agreement type.

Attribute	Type	Multiplicity	Mutability	Description
occi.agreement.state	Enum {Rejected, Pending, Completed, Terminated}	1	Immutable	Current state of the instance.
occi.agreement.agreedAt	Timestamp	0..1	Immutable	The point in time when the agreement was made.
occi.agreement.effectiveFrom	Timestamp	0..1	Mutable	The point in time when the agreement's effectiveness begins.
occi.agreement.effectiveUntil	Timestamp	0..1	Mutable	The point in time when the agreement's effectiveness ends.

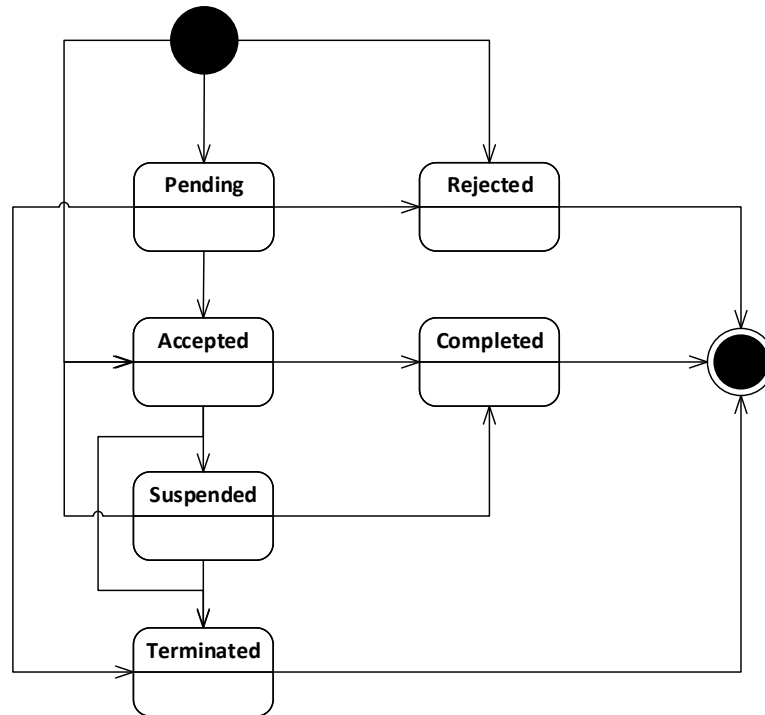


Figure 2. State diagram for Agreement instance.

The Actions that are applicable to *Agreement* instances are presented in Table 3. The Actions are defined by the Kind instance <http://schemas.ogf.org/occi/sla#agreement>. Every Action in the table is identified by a Category instance using the <http://schemas.ogf.org/occi/sla/agreement/action#> categorization scheme. The “Action Term” below refers to the term of the Action's Category identifier.

Table 3: Actions applicable to instances of the Agreement type.

Action Term	Target state	Attributes
Offer	Pending	-
Accept	Accepted	-
Reject	Rejected	-
Suspend	Suspended	-
Complete	Completed	-
Terminate	Terminated	-

These actions MUST be exposed by an instance of *Agreement* type of an OCCI SLAs implementation. The implementation of the Agreement type is REQUIRED if a cloud service provider adopts the OCCI SLAs specification.

3.1.1 AgreementTemplate Mixin

In order to allow the classification of agreements and the provisioning of Service Level Agreement templates, an OCCI Mixin is introduced. The *AgreementTemplate* Mixin is assigned the “scheme” <http://schemas.ogf.org/occi/sla/agreement#> and the term *agreement_tpl*. An *AgreementTemplate* mixin MUST support these values. The use and instantiation of this Mixin is OPTIONAL but

RECOMMENDED for improved classification and management of the agreements. There are no specific attributes defined for the *AgreementTemplate* Mixin, thus every provider that implements the OCCI SLAs specification MAY introduce provider specific attributes using the Attributes Set inherited from the Category type.

As can be seen in the example diagram below, the *AgreementTemplate* mixin can be used either for simple agreement tagging (e.g. gold, silver etc.) but also for introducing specific attributes and features for each tag.

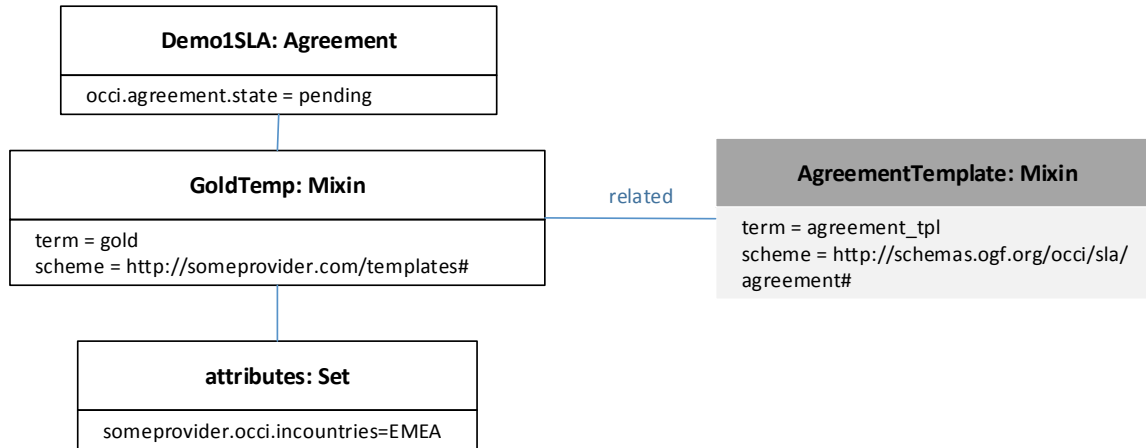


Figure 3: Object diagram of an Agreement instance and its associated AgreementTemplate mixin.

3.1.2 Agreement Terms Mixin

A necessary part of an agreement offer, as well as the consequent agreement, is the section of the agreement terms. To this end, the OCCI SLAs suggests the introduction of the agreement terms through the Mixin mechanism. The *AgreementTerms* Mixin is assigned the “scheme” `http://schemas.ogf.org/occi/sla/agreement#` and the term `agreement_terms`. An *AgreementTerms* mixin MUST support these values. OCCI SLAs implementations SHOULD support this in order to provide a classification and definition mechanism for the various terms and conditions of the agreements. Therefore, the implementation of this functionality is OPTIONAL but RECOMMENDED.

Following the rationale presented in the WS-Agreement specification [5], OCCI SLAs defines two types of agreement terms: service terms and service level objectives (SLOs). The first includes information related with the service description and definition. The second refers to the guarantee terms that specify the service level which the two parties are agreeing to. A cloud service provider MAY introduce domain specific attributes to the *AgreementTerms* mixin instances that he constructs, through the attributes set inherited from the Category type. Mixin relationships MAY be used in order to enforce classification of capabilities but also to allow resource specific instantiation of *AgreementTerms*. For example, an availability Mixin could be defined, which is related with the *AgreementTerms* Mixin type. The provider, then, MAY choose to instantiate different availability mixins for compute or storage resources (or any other offered resource) based on his own definition of availability for those resources.

Table 4: Attributes for the Agreement type.

Attribute	Type	Multiplicity	Mutability	Description
occi.agreement.term.type	Enum {SERVICE-TERM, SLO}	1	Immutable	The type of the term that is being defined.
occi.agreement.term.state	Enum {undefined, fulfilled, violated}	1	Immutable	The state of fulfillment of the specific term.

The *AgreementTerms* state can be either *undefined*, *fulfilled* or *violated* (Figure 4). The *undefined* state is the initial state of the term until an assessment is made. During runtime and while the service and SLA is being monitored the state **MUST** be *fulfilled* or *violated*.

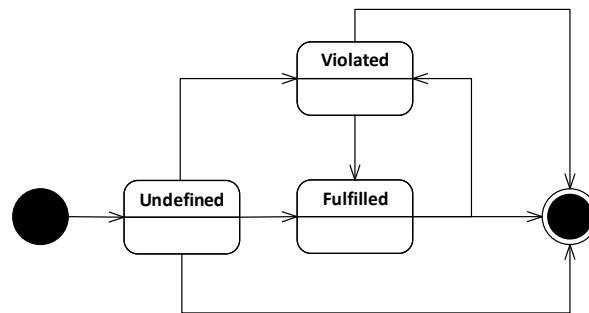


Figure 4: AgreementTerms state diagram

In Figure 5 an example of using the *AgreementTerms* Mixin is shown. In the specific implementation an agreement offer (state: pending) is defined which is about a compute service (termtype: SERVICE-TERM, memory: 16GB, cores: 4). As Service Level Objective (SLO) is the Availability mixin instance defined with attributes specifying the uptime and remedy values. A more detailed example is presented at the end of the document.

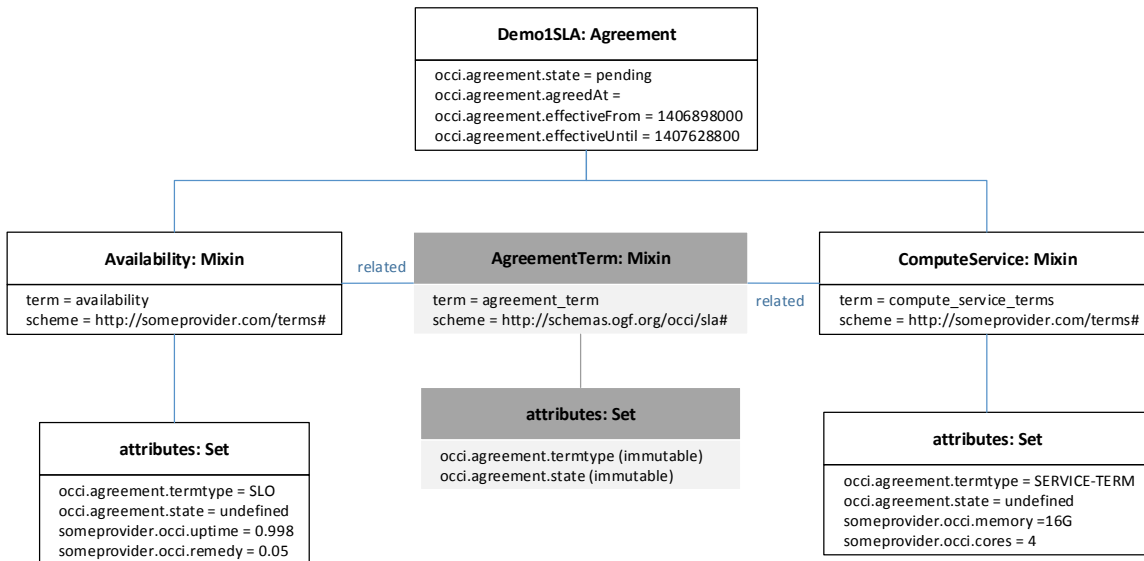


Figure 5: Object diagram of an Agreement instance populated with AgreementTerms mixins.

3.2 AgreementLink

In order to associate signed Service Level Agreements with existing OCCI resource instances, the *AgreementLink* is introduced. This is a sub-type of the OCCI Core Model Link base type. Thus, the instantiation of an *AgreementLink* resource allows the linkage of resources of the previous defined *Agreement* sub-type with any OCCI Core Model Resource sub-type (e.g. Infrastructure sub-types). The implementation of the AgreementLink type is REQUIRED if a cloud service provider adopts the OCCI SLAs specification.

The *AgreementLink* type is assigned the Kind instance `http://schemas.org/occi/sla#agreementlink`. An *AgreementLink* instance MUST use and expose this Kind. The Kind instance assigned to the *AgreementLink* type MUST be related to the `http://schemas.org/occi/core#link` Kind.

The state of the *AgreementLink* is being specified through an attribute (Table 5). A cloud service provider could therefore specify the status of the association between an *Agreement* and its relevant resources e.g. during a re-negotiation phase or suspension phase. Figure 6 shows the state transitions of an AgreementLink instance. The defined attribute and the state transitions of an *AgreementLink* type instance MUST be exposed and implemented by the provider.

Table 5: Attributes for the Agreement Link type.

Attribute	Type	Multiplicity	Mutability	Description
<code>occi.agreementlink.state</code>	Enum {Active, Inactive}	1	Immutable	Current state of the link.

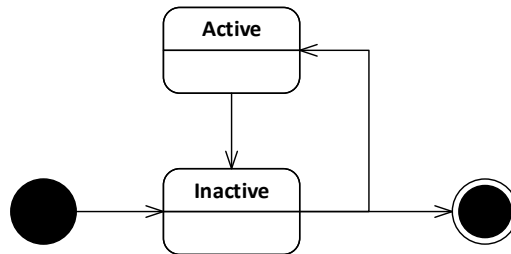


Figure 6: AgreementLink state diagram

Because of the multiple possibilities in terms of design and implementation of an OCCI compatible system, domain specific *AgreementLink* sub-types MAY be defined by cloud service providers. Thus, additional, provider specific attributes in such agreement link sub-types MAY be defined in by its Kinds instances.

3.3 Service Level Agreement example

In this section, an example instantiation of an Agreement type along with provider defined mixins is presented. It is to be noted that the implementation of an OCCI SLA framework is a responsibility of the cloud service provider. Thus, the instantiation of the proposed types and mixins are subject to the requirements and objectives of the provider. The presented instantiation of an OCCI SLA is only an example. Different approaches, mixins and attributes definitions could be followed.

The creation and provisioning of SLAs includes several phases. The process of reaching such agreement could be described by the following steps²:

- Negotiation phase - The cloud service consumer retrieves the SLA templates, completes the REQUIRED values and submits an offer to the cloud service provider. (agreement-state: *pending*)
- Agreement phase - The cloud service provider can decide whether to accept the filled out template (the offer) or not. It is also possible to provide a counter-offer to the customer. (agreement-state: *accepted, rejected, pending*)
- Execution phase - When the agreement has been accepted the Agreement is in place and the (newly) created resource can be linked and associated with the reached agreement. (agreement-state: *accepted*)

The object diagram in Figure 7 represents an Agreement in the execution phase. In the presented example the Demo1SLA agreement is being populated with the *SilverTemp* mixin which is related to the *AgreementTemplate* Mixin type. This is used to tag and classify the agreement as well as to define some generic constraints such as the region in which the resources (under that SLA template) SHOULD be allocated. In addition to the template mixin several *AgreementTerms* mixins are defined either to define and describe the service offered or to introduce Service Level Objectives (SLOs) for the agreement.

To this end, through the *ComputeServiceTerms* mixin, the cloud service provider introduces a set of service terms which characterize the service being offered with this SLA. In this case it is a compute resource with technical specifications defined through provider-specific attributes (e.g. *someprovider.occi.cores*, *someprovider.occi.cpu* etc.). The *Availability*, *ServicePerformance* and *ServiceCapacity* are all Service Level Objective terms that set certain thresholds to metrics which determine the Quality of Service (QoS) of the respective offering. Every SLO term also defines the remedy value which is the compensation to the customer in the event that the cloud service provider fails to meet the specified SLO. The value is usually a percentage of the agreed rate for the offered cloud service. The attributes defined in the mixins can be either mutable or immutable to the customer depending on how the negotiation phase is being realized by the cloud service provider. What is more, every term has a current state value. Depending on the current assessment the terms are fulfilled or violated. Each violation will trigger the respective remedy value.

² This list is not exhaustive. More steps or a different workflow could be followed (e.g. re-negotiation phase etc.).

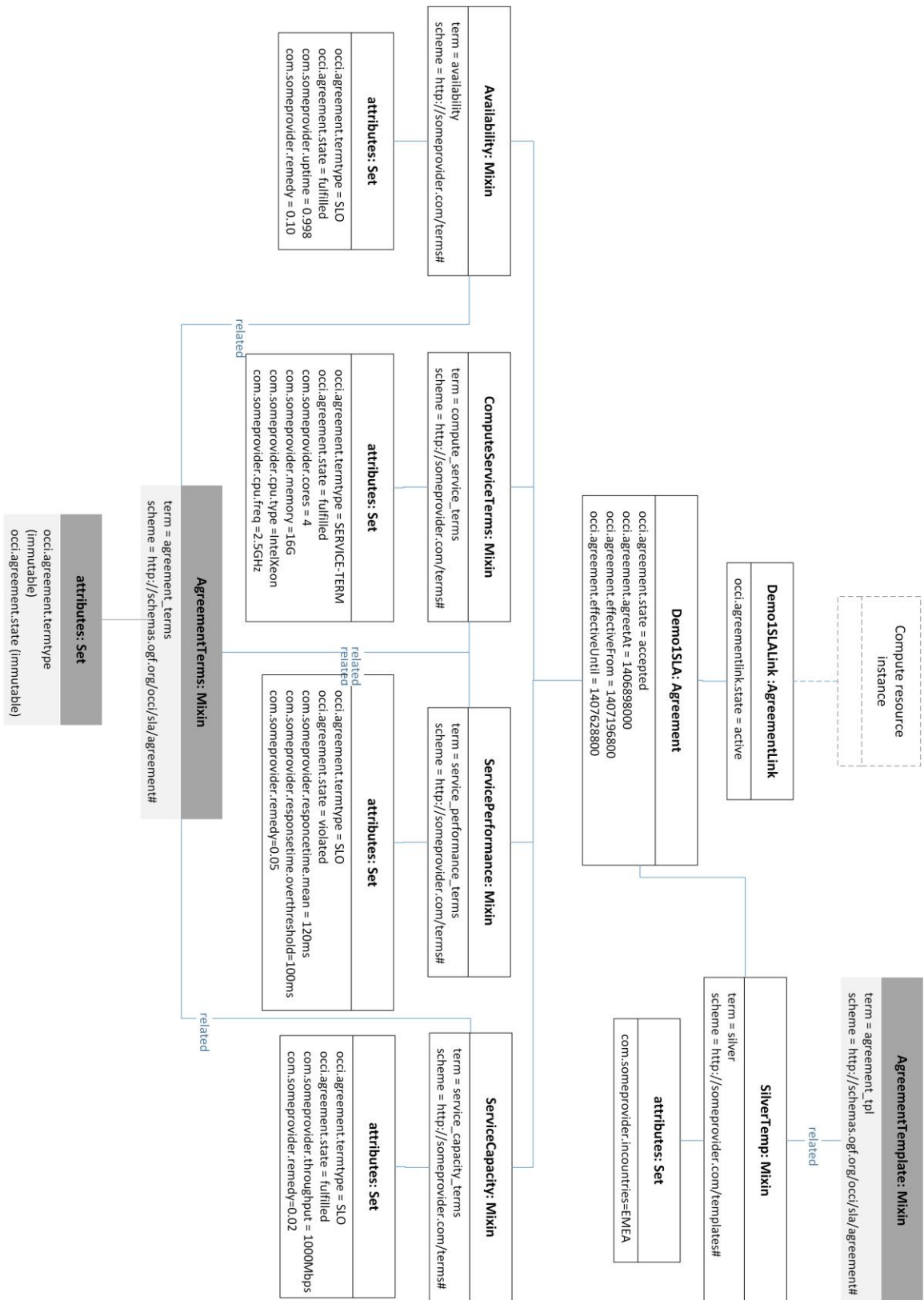


Figure 7: OCCI SLA instantiation example

4 Security Considerations

The OCCI SLAs specification is an extension of the OCCI Core Model specification [2]; thus the same security considerations as for the OCCI Core Model specification apply here.

5 Glossary

Term	Description
Action	An OCCI base type. Represent an invocable operation on an Entity sub-type instance or collection thereof.
Category	A type in the OCCI model. The parent type of Kind.
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.
Kind	A type in the OCCI 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 model. A core component of the OCCI classification system.
mixin	An instance of the Mixin type associated with a resource instance. The "mixin" concept as used by OCCI only applies to instances, never to Entity types.
OCCI	Open Cloud Computing Interface.
OCCI base type	One of Entity, Resource, Link or Action.
Resource	An OCCI base type. The parent type for all domain-specific resource types.
resource instance	An instance of a sub-type of Entity. The OCCI model defines two sub-types of Entity, the Resource type and the Link type. However, the term resource instance is defined to include any instance of a sub-type of Resource or Link as well.
type	One of the types defined by the OCCI model. The OCCI model types are Category, Kind, Mixin, Action, Entity, Resource and Link.
sub-type	A sub-type is a type that can be instantiated.
Cloud service provider	The entity who offers a resource/service.
Cloud service consumer	The party which is in business relationship with the cloud service provider for using a cloud service/resource.
Service Level Agreement (SLA)	The contract or agreement that the two parties (provider, consumer) need to "sign". It includes all the information about the services and the terms they both agree upon.
Service Level Objective (SLO)	The quality of service aspect of the agreement. Specifies a non-functional guarantee in the SLA.
SLA Template	It is a resource that classifies set of terms and qualities for a provisioned service.

6 Contributors

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

Name	Affiliation	Contact
------	-------------	---------

Gregory Katsaros Thijs Metsch John Kennedy	Intel Intel Intel	gregory.katsaros at intel.com thijs.metsch at intel.com john.m.kennedy at intel.com
--	-------------------------	---

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

Copyright (C) Open Grid Forum 2014. All Rights Reserved.

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 as references to the derived portions on all such copies and derivative works. The published OGF document from which such works are derived, however, 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 new or updated OGF documents in conformance with the procedures defined in the OGF Document Process, or as required to translate it into languages other than English. OGF, with the approval of its board, may remove this restriction for inclusion of OGF document content for the purpose of producing standards in cooperation with other international standards bodies.

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

References

- [1] Scott Bradner. Key Words for Use in RFCs to Indicate Requirement Levels, RFC 2119. The Internet Society. March 1997. <http://tools.ietf.org/html/rfc2026>
- [2] R. Nyr_en, 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>
- [3] 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>
- [4] T. Metsch and A. Edmonds, "Open Cloud Computing Interface - Infrastructure," GFD-P-R.184, April 2011. [Online]. Available: <http://ogf.org/documents/GFD.184.pdf>
- [5] Alain Andrieux et.al, "Web Services Agreement Specification (WS-Agreement)" GFD-R-P.107, March 2017. [Online]. Available: <https://www.ogf.org/documents/GFD.107.pdf>